UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Ciências Aplicadas

Luciana Narumi Oshiro Yamada

AS DIFERENTES PERSPECTIVAS DO AGRUPAMENTO DE DADOS ATRAVÉS DA OTIMIZAÇÃO MULTIOBJETIVO

LIMEIRA
2022

UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Ciências Aplicadas

Luciana Narumi Oshiro Yamada

AS DIFERENTES PERSPECTIVAS DO AGRUPAMENTO DE DADOS ATRAVÉS DA OTIMIZAÇÃO MULTIOBJETIVO

Dissertação apresentada à Faculdade de Ciências Aplicadas da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Mestra em Engenharia de Produção e de Manufatura na área de Pesquisa Operacional e Gestão de Processos.

Orientadora: Profª. Drª. Priscila Cristina Berbert Rampazzo.
Coorientadora: Profª. Drª. Flávia Barbosa.

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELA ALUNA LUCIANA NARUMI OSHIRO, E ORIENTADA PELA PROFª. DRª. PRISCILA CRISTINA BERBERT RAMPAZZO.

LIMEIRA
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Faculdade de Ciências Aplicadas
Ana Luiza Clemente de Abreu Valério - CRB 8/10669

Y141d

Yamada, Luciana Narumi Oshiro, 1995-
As diferentes perspectivas do agrupamento de dados através da otimização multiobjetivo / Luciana Narumi Oshiro Yamada. – Limeira, SP : [s.n.], 2022.

Orientador: Priscila Cristina Berbert Rampazzo.
Coorientador: Flávia Barbosa.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Ciências Aplicadas.

1. Otimização multiobjetivo. 2. Análise por agrupamento. 3. Algoritmos evolutivos. I. Rampazzo, Priscila Cristina Berbert, 1984-. II. Barbosa, Flávia, 1989-. III. Universidade Estadual de Campinas. Faculdade de Ciências Aplicadas. IV. Título.

Informações Complementares

Título em outro idioma: The different perspectives of data clustering through multiobjective optimization
Palavras-chave em inglês:
Multiobjective optimization
Cluster analysis
Evolutionary algorithms
Área de concentração: Pesquisa Operacional e Gestão de Processos
Titulação: Mestra em Engenharia de Produção e de Manufatura
Banca examinadora:
Priscila Cristina Berbert Rampazzo [Orientador]
Leonardo Tomazeli Duarte
Luís Filipe Ribeiro dos Santos Guimarães
Data de defesa: 14-12-2022
Programa de Pós-Graduação: Engenharia de Produção e de Manufatura

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-0161-479
- Currículo Lattes do autor: https://lattes.cnpq.br/0245014658066609

FOLHA DE APROVAÇÃO

Autora: Luciana Narumi Oshiro Yamada RA: 147080

Título: As diferentes perspectivas do Agrupamento de Dados através da Otimização Multiobjetivo

Natureza: Dissertação

Área de Concentração: Pesquisa Operacional e Gestão de Processos.

Instituição: Faculdade de Ciências Aplicadas – FCA/Unicamp

Data da defesa: 14 de dezembro de 2022.

BANCA EXAMINADORA:

Profª. Drª. Priscila Cristina Berbert Rampazzo (Orientador)
Faculdade de Ciências Aplicadas - FCA/Unicamp

Prof. Dr. Leonardo Tomazeli Duarte (membro)
Faculdade de Ciências Aplicadas - FCA/Unicamp

Prof. Dr. Luís Filipe Ribeiro dos Santos Guimarães (membro externo)
Faculdade de Engenharia, Universidade do Porto

A ata da defesa, com as respectivas assinaturas dos membros da Banca Examinadora, encontra-se no processo de vida acadêmica da aluna.

Agradecimentos

Aos meus pais, Luís e Marilda, e meu irmão, Guilherme, pelo amor, apoio e incentivo. Sem vocês, nada disso seria possível.

Ao meu marido, Felipe, por todo amor e companheirismo durante todos estes anos. Obrigada por me ajudar e me motivar nos momentos difíceis.

Aos meus sogros, Carlos e Miriam, pelo amor, apoio e pelos bons momentos compartilhados durante essa etapa.

À minha orientadora, Professora Priscila, pela confiança, oportunidade e por me ensinar tanto durante estes anos.

À minha coorientadora, Professora Flávia, pelos conselhos, pela disponibilidade e dedicação em sempre me ajudar.

Aos professores da banca, Professor Leonardo Tomazeli Duarte e Professor Luís Filipe Ribeiro dos Santos Guimarães, por aceitarem o convite, pelos importantes comentários e sugestões ao trabalho.

A todos os professores e colegas do Laboratório Centro de Pesquisa Operacional (CPO) e do Laboratório de Análise de Dados e Apoio à Decisão (LAD2). Obrigada por toda ajuda e aprendizado.

Aos demais familiares e amigos, muito obrigada!

À Faculdade de Ciências Aplicadas (FCA) da Universidade Estadual de Campinas (UNICAMP) por todo suporte.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 - Nº do Processo: 36-P-7733/2022.

Resumo

A associação das temáticas Agrupamento de Dados e Otimização Multiobjetivo tornou-se uma alternativa atraente para alguns dos problemas enfrentados por algoritmos tradicionais de agrupamento. Os pesquisadores têm dedicado esforço considerável para aperfeiçoar os métodos de agrupamento multiobjetivo. Porém, poucos exploram os resultados obtidos através de uma análise a *posteriori*. Neste trabalho, utilizamos um algoritmo com diferentes funções de distância para explorar os conflitos entre diferentes perspectivas do problema. Ao final, obtemos um conjunto de soluções não-dominadas, que carregam informações sobre possíveis estruturas de agrupamento. Depois disso, uma análise a *posteriori* é proposta para explorar o conhecimento dessas soluções e extrair informações complementares sobre o problema. Os resultados sugerem que as informações geradas pela fronteira não-dominada podem auxiliar na análise exploratória do problema e serem utilizadas em tarefas como a rotulação de dados.

Palavras-chave: Otimização multiobjetivo. Análise por agrupamento. Algoritmos evolutivos.

Abstract

The association of the themes of Data Clustering and Multiobjective Optimization has become an attractive alternative to some problems of the traditional clustering algorithm. Researchers have devoted considerable effort to improving multiobjective clustering methods. However, few explore the results obtained in a posteriori analysis. In this work, we use an algorithm with different distance functions to explore conflicts between different perspectives of the problem. By the end, we obtain a set of non-dominated solutions, carrying out information about the possible clustering structure. After that, we pursue a posteriori analysis to exploit the knowledge of these solutions and extract complementary information about the problem. The results suggest that the information generated by the non-dominated frontier can help in the exploratory analysis of the problem and be used in tasks such as data labeling.

Keywords: Multiobjective optimization. Cluster analysis. Evolutionary algorithms.

Lista de ilustrações

Lista de tabelas

Sumário

Introdução

Diariamente, grandes quantidades de dados são geradas para registrar de forma qualitativa e quantitativa um dado evento. Com a atual capacidade de processamento, armazenamento e compartilhamento de dados, podemos ter acesso a informações relevantes para a tomada de decisão. Porém, extrair informações sobre os dados é um processo desafiador, sendo necessário a utilização de métodos para pré-processar, analisar e avaliar os dados, de forma a produzirem conhecimento sobre determinado problema (RUNKLER, 2020). Tarefas que extraem essas informações como, análise de dados, previsão de risco e auxílio em diagnósticos médicos não são consideradas simples, podendo se tornar mais complexas quando os dados são incompletos ou apresentam redundância e inconsistência. O Aprendizado de Máquina é uma área da Inteligência Artificial (IA) que reúne métodos matemáticos, estatísticos e algoritmos computacionais que permitem extrair relações dentro de um conjunto de dados para, a partir das informações extraídas, construir um modelo capaz de realizar previsões ou auxiliar na tomada de decisões frente à novas situações.

Dentro deste campo, os tipos de aprendizados conhecidos são: supervisionado, não supervisionado, semissupervisionado e por reforço. Sendo as duas primeiras categorias as mais conhecidas. Nelas, os métodos podem ser desenvolvidos a partir de dados de entrada com ou sem valores de saídas associados, também denominados de rótulos ou classes. No aprendizado supervisionado, por exemplo, o modelo é criado com o objetivo de encontrar as relações entre dados de entrada e saída conhecidos, de modo que a informação adquirida seja utilizada para predizer a saída de novos dados. Por sua vez, no aprendizado não supervisionado, o objetivo passa a ser a busca por padrões escondidos a partir dos dados de entrada sem rótulos. Nesse caso, os métodos buscam por padrões agrupando dados com características semelhantes. A ideia principal desses dois tipos de aprendizados é que no supervisionado, há um "professor" para guiar o processo de aprendizado, enquanto no não supervisionado, o modelo precisa aprender sozinho, tornando o processo mais difícil pela ausência dessa ajuda (CUNNINGHAM; CORD; DELANY, 2008).

Nos últimos anos, o campo de IA progrediu consideravelmente no desenvolvimento de métodos de aprendizado de máquina supervisionado (LECUN; MISRA, 2021), no qual pressupõe-se a existência de uma grande quantidade de dados rotulados. Porém, tal exigência é um forte limitador para a aplicação em diversos problemas práticos. Assim, um dos grandes desafios que despontam em IA é a busca por estratégias que não necessitam de supervisão.

Além da limitação dada pela falta de dados rotulados, a aplicação dos modelos geralmente envolvem processos de tomada de decisões sob objetivos conflitantes. Um

modelo supervisionado de classificação pode ser utilizado no processo de diagnóstico médico para auxiliar a identificação de pacientes com perfis saudáveis e doentes. Porém, problemas como atribuir um paciente doente como saudável podem ocorrer e gerar um impacto grave no processo de diagnóstico. Dessa forma, um modelo capaz de capturar padrões dessas duas categorias, como os métodos não supervisionados, pode promover melhor generalização na tomada de decisão e permitir explorar o problema a partir de diferentes perspectivas.

Algumas dificuldades dos métodos não supervisionados envolvem a escolha do critério de agrupamento, que deve se adequar às características dos dados, e a definição de parâmetros como a quantidade de grupos. Quando os critérios de agrupamento não conseguem capturar corretamente as características dos dados, os algoritmos tradicionais apresentam desempenho ruim (MUKHOPADHYAY; MAULIK; BANDYOPADHYAY, 2015). Além disso, a maioria dos algoritmos utilizam apenas um critério de agrupamento, podendo limitar o conhecimento e padrão obtidos em conjuntos com diferentes propriedades (FACELI; CARVALHO; SOUTO, 2007). Algoritmos multiobjetivo para agrupamento de dados foram propostos com o objetivo de capturar diferentes padrões nos dados. Combinando os dois grandes temas, esses métodos surgiram como alternativas para alguns problemas enfrentados na aplicação de métodos tradicionais de agrupamento. Com a utilização da otimização multiobjetivo, múltiplos critérios de agrupamentos podem ser utilizados para capturar as diferentes perspectivas do problema, gerando um conjunto de soluções que represente o *trade-off* dessas características.

Trabalhos com a associação dessas duas temáticas têm crescido na literatura, demonstrando ser uma alternativa atrativa para alguns dos problemas enfrentados pelos algoritmos tradicionais. Os pesquisadores têm dedicado esforço considerável para aperfeiçoar os métodos de agrupamento multiobjetivo, porém, poucos exploram os resultados obtidos em uma análise a *posteriori*.

Este trabalho sugere que a Otimização Multiobjetivo, associada às diferentes medidas de distâncias, poderia capturar diferentes perspectivas do problema. Além disso, combinada com uma etapa de análise a *posteriori*, pode-se compreender melhor os resultados obtidos, assim como, utilizar as informações geradas pelas soluções não-dominadas para extrair informações complementares ao agrupamento. A ideia principal de propor essa etapa é que cada solução não-dominada gerada carrega informações sobre o possível padrão presente no conjunto de dados e explorá-las pode ser interessante no contexto do paradigma não supervisionado. As informações geradas pela fronteira não-dominada poderiam auxiliar nos processos de tomada de decisão e na compreensão do problema.

Dessa forma, buscamos responder as seguintes questões:

1. As análises a *posteriori* contribuem para um melhor entendimento dos resultados

obtidos no agrupamento de dados?

2. Como explorar as soluções não-dominadas da otimização multiobjetivo para obter informações complementares ao agrupamento de dados?

Organização da Dissertação

A primeira parte da dissertação abrange todo o referencial teórico, apresentando o Aprendizado de Máquina, a Otimização Multiobjetivo e o Agrupamento de Dados Multiobjetivo.

A segunda parte da dissertação discute as principais contribuições da pesquisa, apresenta o desenvolvimento do algoritmo de agrupamento de dados multiobjetivo, os experimentos realizados e as conclusões finais do trabalho.

Sendo assim, esta dissertação está organizada da seguinte maneira:

- **Parte I: Referencial Teórico**
  - **Capítulo 1 - Aprendizado de Máquina** introduz o referencial teórico dos diferentes tipos de aprendizados existentes, enfatizamos principalmente o aprendizado não supervisionado;
  - **Capítulo 2 - Otimização Multiobjetivo** apresenta a formalização de um problema multiobjetivo e os métodos para resolução. Explicamos com mais detalhes o algoritmo utilizado nesta pesquisa, o NSGA-II;
  - **Capítulo 3 - Agrupamento de Dados Multiobjetivo** revisa os principais conceitos envolvendo o desenvolvimento de algoritmos de Agrupamento de Dados Multiobjetivo. Neste capítulo, apresentaremos os trabalhos que serviram de referência para a proposta desta pesquisa;
- **Parte II: Desenvolvimento**
  - **Capítulo 4 - Contribuições** discute as principais contribuições da pesquisa do ponto de vista teórico e prático;
  - **Capítulo 5 - Metodologia** apresenta de forma detalhada a proposta desta pesquisa;
  - **Capítulo 6 - Experimentos e Resultados** apresenta os testes e os resultados obtidos em conjuntos de dados sintéticos e reais;
  - **Capítulo 7 - Conclusão e Perspectivas Futuras** finaliza com as principais conclusões e as perspectivas de estudos complementares para continuação do estudo.

Parte I

Referencial Teórico

1 Aprendizado de Máquina

A década de 50 foi marcada por um desenvolvimento intenso na área de aprendizado de máquina. Sua origem é geralmente associada ao trabalho do psicólogo Frank Rosenblatt, que baseado em ideias sobre o sistema nervoso humano desenvolveu o *Perceptron*, protótipo das conhecidas Redes Neurais Artificiais (FRADKOV, 2020). Mas foi a partir da primeira década do século XXI que o interesse da comunidade científica cresceu, consequência do *Big data*, da redução do custo computacional e do desenvolvimento de novos algoritmos no campo de aprendizagem profunda (TSYGANOV, 2020).

Atualmente, o campo de estudo de aprendizado de máquina têm mostrado grandes avanços e contribuições para diversas áreas científicas. Algumas das aplicações mais conhecidas abrangem áreas de análises preditivas para tomada de decisão; cuidados de saúde; comércio digital e recomendação de produtos; processamento de linguagem natural e análise de sentimentos; e reconhecimento de padrão, imagem e voz (SARKER, 2021).

Os aprendizados são divididos em quatro tipos: supervisionado, não supervisionado, semissupervisionado e aprendizado por reforço. No aprendizado supervisionado, dados de entrada e saída são utilizados para construir um modelo de aprendizado automático (THEODORIDIS; KOUTROUMBAS, 2008). O objetivo desses métodos é prever a saída de novas variáveis a partir do conhecimento adquirido *a priori*. As tarefas mais conhecidas desse tipo de aprendizado são a classificação e a regressão.

Entretanto, informações sobre a classe final dos dados (rótulos) nem sempre estão disponíveis. Nesse sentido, métodos de aprendizado não supervisionado auxiliam na busca de padrões baseados nos dados de entrada. Na tarefa de agrupamento de dados, os padrões são representados pelos grupos (*clusters*) e dados que compartilham o mesmo grupo são mais similares, segundo alguma métrica, do que dados contidos em grupos diferentes (SANCHES, 2003). Outra tarefa conhecida é a redução de dimensionalidade e o objetivo é representar conjuntos de dados de alta dimensão em dimensões menores preservando suas estruturas e informações mais significativas, diminuindo assim a redundância ou irrelevância dos dados, o custo computacional e melhorando a interpretação dos modelos (SARKER, 2021).

Na literatura, outra tarefa encontrada dos métodos não supervisionados é a aprendizagem autossupervisionada. Para LeCun e Misra (2021) o autossupervisionado é uma das maneiras mais promissoras de construir um conhecimento parecido com o dos humanos em modelos de aprendizado de máquina. Essa abordagem têm ganhado destaque nos últimos anos, principalmente pelo uso de aprendizagem profunda e em aplicações

de processamento de linguagem natural e em imagens. O objetivo principal dessa tarefa é criar pseudo rótulos de forma automática apenas explorando alguma propriedade dos dados, sem precisar envolver dados rotulados (SCHMARJE et al., 2021).

No aprendizado semissupervisionado o objetivo é entender como a combinação de dados rotulados e não rotulados pode auxiliar no processo de aprendizagem (ZHU; GOLDBERG, 2009). O aprendizado semissupervisionado é de grande interesse em aprendizado de máquina porque pode usar dados não rotulados, que estão mais amplamente disponíveis, para melhorar as tarefas de aprendizado supervisionado quando os dados rotulados são escassos ou caros. Para além do aprendizado não supervisionado, o semissupervisionado, considera que os dados devem ser agrupados não apenas porque são semelhantes, mas também porque possuem algum significado conceitual (SANCHES, 2003).

O aprendizado por reforço é uma técnica que permite tomadores de decisões artificiais aprenderem em um ambiente interativo usando como entrada as suas ações e experiências (MAIMON; COHEN, 2009). Esses tomadores de decisões, chamados de agentes, desempenham ações que ocasionam um sinal de recompensa ou penalização. Dessa forma, seu comportamento e suas próximas ações são moldadas com base no aprendizado adquirido por essas consequências, ou seja, ações com boas recompensas são reforçadas no aprendizado. Um algoritmo que segue essa abordagem é o AlphaGo (SILVER et al., 2016), conhecido por ser o primeiro programa de computador a derrotar um campeão mundial do jogo de tabuleiro Go. O aprendizado por reforço pode resolver inúmeros problemas reais e em diversos campos, como otimização baseada em simulação, sistemas multiagentes, logística da cadeia de suprimentos, teoria dos jogos, entre outros (SARKER, 2021). Para mais detalhes, uma revisão da literatura pode ser encontrada em Dayan e Niv (2008), Maimon e Cohen (2009) e Nian, Liu e Huang (2020).

A Figura 1 resume os tipos de aprendizado e suas principais categorias. Nas próximas seções, descreveremos com mais detalhes cada um dos tópicos: os aprendizados supervisionado, não supervisionado, com ênfase nas tarefas de agrupamento de dados, redução de dimensionalidade e na aprendizagem autossupervisionada, e semissupervisionado.

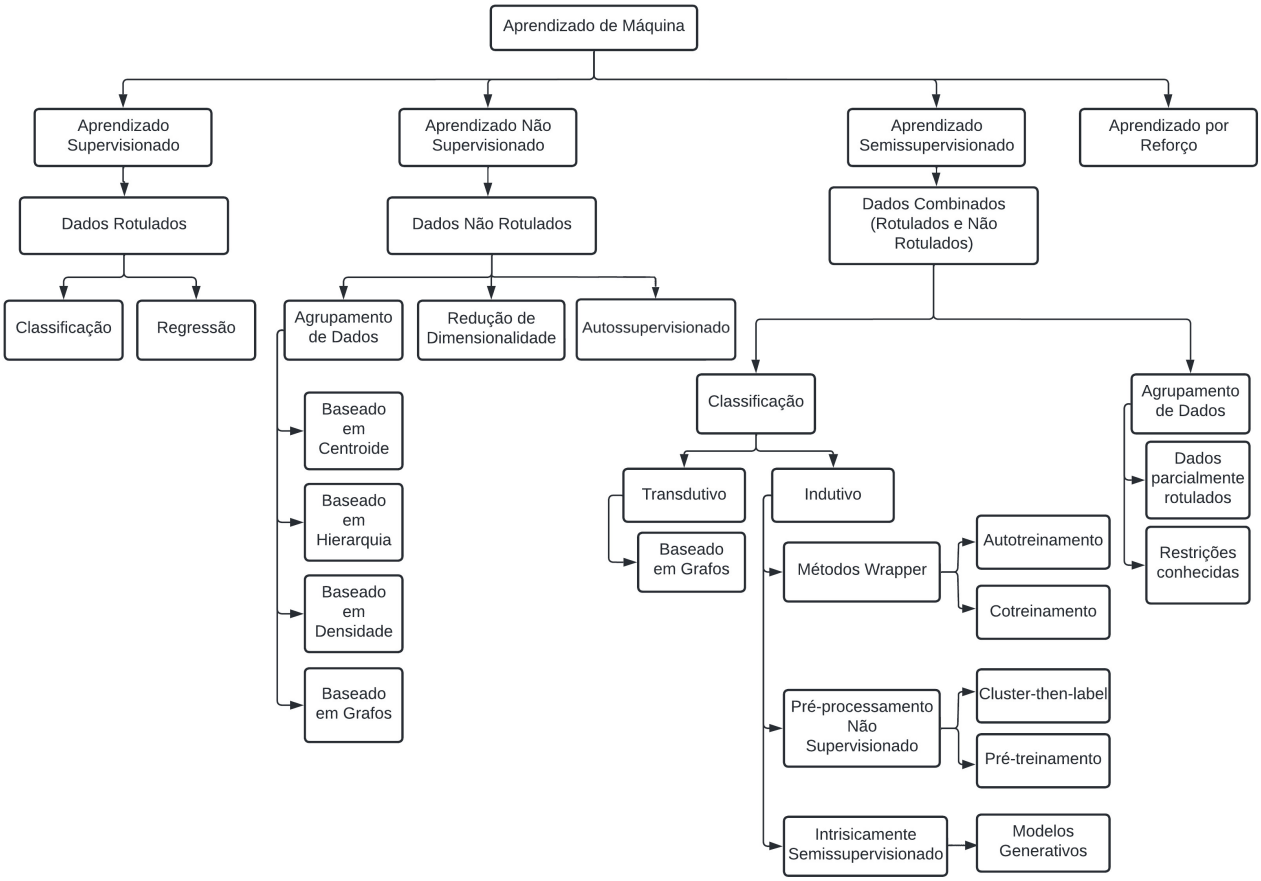
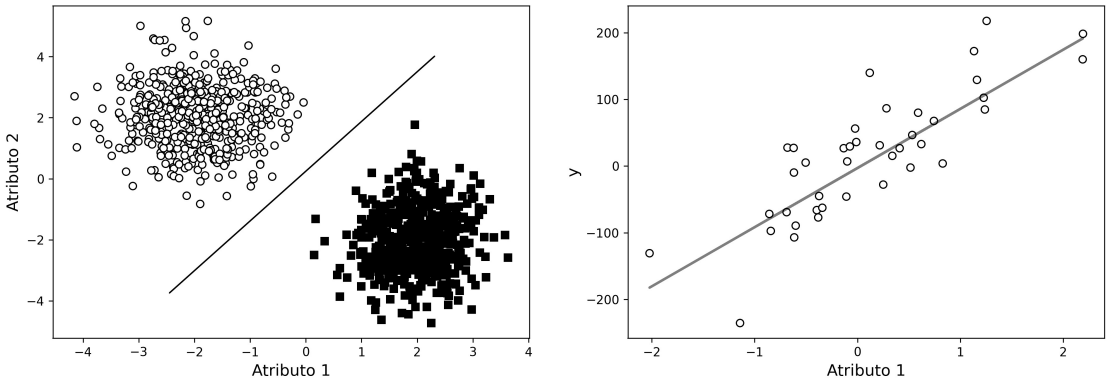


Figura 1 – Adaptado de Sarker (2021) e Engelen e Hoos (2020).

1.1 Métodos supervisionados

Os métodos supervisionados podem ser divididos pelas tarefas de Classificação e Regressão. Na primeira tarefa (Figura 2a), o objetivo do modelo de aprendizado é atribuir uma classe a um dado desconhecido a partir do conhecimento adquirido de eventos anteriores. De forma matemática, o modelo produz uma função de mapeamento $g : \mathbb{R}^n \rightarrow \{1, ..., m\}$, tal que m são as classes das amostras, que permitirá encontrar uma saída y de um vetor de entrada $\boldsymbol{x}$, quando $y = g(\boldsymbol{x})$ (GOODFELLOW; BENGIO; COURVILLE, 2016). A saída do modelo de classificação pode ser tanto um valor discreto, associado a classe da amostra, como uma distribuição de probabilidade sobre a classe. Alguns exemplos de aplicações são em problemas de detecção de e-mail spam, reconhecimento de objetos em imagens e obtenção de diagnósticos.

Na tarefa de regressão (Figura 2b), dado um conjunto de amostras de treinamento $(y_i, \boldsymbol{x}_i), y_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathbb{R}^d, i = 1, 2, ..., N$, a tarefa é estimar uma função g, que se adeque aos dados e consiga predizer novos dados de entrada (THEODORIDIS, 2015). Diferente da classificação, a saída do modelo de regressão é contínua. Alguns exemplos de aplicação da regressão abrangem a área financeira, com predição de dados, análise de tendências e estimativa de séries temporais (SARKER, 2021).



(a) Classificação (b) Regressão

Figura 2 – Exemplo de um problema de classificação e regressão. Conjunto de dados gerados pela biblioteca Scikit-learn.

Dentre os algoritmos de aprendizagem supervisionada, podemos citar as Máquina de Vetores de Suporte (VAPNIK; CHERVONENKIS, 1964); os algoritmos baseados em árvores de decisão como o CART (LEO et al., 1984); algoritmos baseados em estatística como o Classificador *Naïve Bayes* e a Regressão Logística (CESSIE; HOUWELINGEN, 1992), que exploram a tarefa de predição através da distribuição de probabilidade sobre as possíveis classes; algoritmos com a abordagem *ensemble* como as Florestas Aleatórias (HO, 1995) e AdaBoost (FREUND; SCHAPIRE et al., 1996);

Outro método conhecido é o algoritmo K-vizinhos mais próximos (KNN, do inglês *k-nearest neighbors*). Diferente dos outros algoritmos de classificação, o KNN é considerado um "algoritmo preguiçoso", pois não constrói uma função de mapeamento, apenas utiliza a distância entre dados para chegar em uma conclusão sobre a classe final de uma amostra desconhecida (SEN; HAJRA; GHOSH, 2020). Com essa estrutura fácil de ser compreendida, o KNN pode produzir resultados altamente competitivos, inclusive em grandes conjuntos de dados e dados ruidosos (SARKER, 2021).

A ideia principal do KNN é predizer a saída de um novo dado de entrada a partir dos seus k-vizinhos mais próximos. Formalmente, dado uma amostra c_0, encontramos as k amostras de treino $c_{(j)}$, $j = 1, ..., k$ mais próximas de c_0 segundo uma medida de distância como a distância euclidiana, e classificamos a amostra usando o rótulo predominante entre os k vizinhos (HASTIE et al., 2009).

1.2 Métodos não supervisionados

1.2.1 Agrupamento de dados

O agrupamento é uma das formas mais simples para compreender dados desconhecidos. Com sua natureza exploratória, seus principais objetivos são encontrar estruturas escondidas, obter um agrupamento natural dos dados e utilizar essas informações para compreender melhor os dados (JAIN, 2010).

Ao particionar o conjunto de dados em subconjuntos $S = S_1, ..., S_k$, as amostras contidas em cada grupo (*cluster*) S_i, são mais similares entre si do que amostras contidas em outros grupos (THEODORIDIS; KOUTROUMBAS, 2008).

Essa divisão dos dados é realizada a partir de critérios de agrupamento, que verificam quão similar ou diferente um dado é quando comparado a outro. Dependendo da escolha desses critérios, encontramos diferentes formas de agrupar o conjunto de dados, podendo ser úteis em dados com diferentes características como exemplificado na Figura 3.

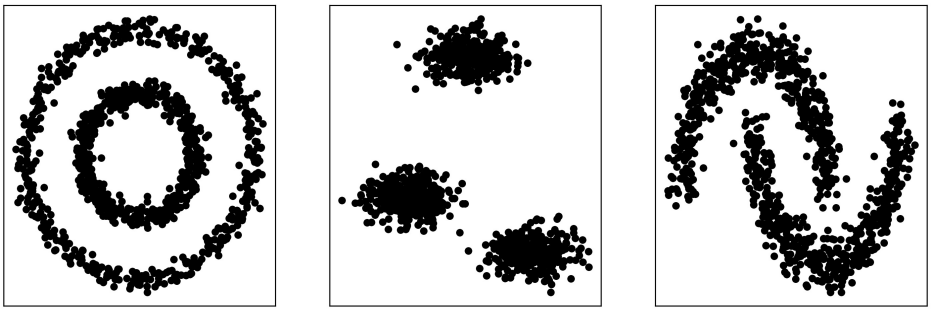


Figura 3 – Conjuntos de dados com diferentes características gerados a partir da biblioteca Scikit-learn. (1) *Circles*; (2) *Blobs*; (3) *Moons*.

Em relação à similaridade, quanto maior o valor observado, mais parecido são os dados (exemplo: coeficiente de correlação). Em relação à dissimilaridade, quanto maior o valor observado, menos parecidos os dados são (exemplo: distâncias). Medidas de similaridades são mais importantes para conjunto de dados com variáveis qualitativas, enquanto as dissimilaridades são utilizadas para medir atributos contínuos (XU; WUNSCH, 2005). Os critérios de agrupamento mais utilizados são: distância de Minkowski, distância Euclidiana, distância do Cosseno, distância de correlação de Pearson, distância de Mahalanobis, similaridade de Jaccard e de Hamming.

Na literatura, diversas formas de categorizar os algoritmos de agrupamento podem ser encontradas, nesse projeto dividiremos os métodos em quatro principais grupos: baseado em centroide, baseado em hierarquia, baseado em densidade e baseado em grafo. Outros tipos de categorias de algoritmos podem ser encontrada em Xu e Wunsch (2005) e Xu e Tian (2015).

Segundo Oskolkov (2019), essas diferentes abordagens de algoritmos seguem estratégias que consideram características do conjunto de dados. O agrupamento baseado em centroide, por exemplo, tem bom desempenho apenas com grupos com simetria esférica ou elipsoidal. O agrupamento hierárquico pode ser sensível a dados com valores atípicos (*outliers*). Agrupamentos baseados em grafos levam em consideração o número de vizinhos compartilhados e têm melhor desempenho em espaços de dimensões altas quando comparados com métodos que utilizam a distância euclidiana; porém, para construir o grafo, a própria distância euclidiana também é utilizada. Alguns destes métodos necessitam que o número de grupos seja especificado implicitamente a *priori*, através da definição (às vezes arbitrária) de hiperparâmetros. Os métodos de agrupamento baseados em densidade, por meio de janelas deslizantes que se movem em direção à alta densidade de pontos, permitem realizar a tarefa de agrupamento sem especificar o número de grupos.

Na Tabela 1, algumas vantagens e desvantagens de cada tipo de agrupamento são resumidas. Nas próximas seções, abordaremos cada tipo de agrupamento e seus respectivos métodos.

Tipo de Agrupamento	Vantagem	Desvantagem
Baseado em Centroide	Baixa complexidade computacional e alta eficiência.	Não adequados para dados não convexos; sensíveis a *outliers* e a quantidade de *clusters*; suscetíveis a ótimos locais; necessidade de definir a quantidade de grupos.
Baseado em Hierarquia	Adequados para conjunto de dados com formatos e tipos de atributos arbitrários; relações de hierarquia são facilmente detectáveis.	Alto tempo computacional; Não aplicável a altas dimensões; sensíveis a *outliers*; necessidade de definir a quantidade de grupos.
Baseado em Densidade	Alta eficiência; adequados para conjunto de dados com formatos arbitrários; sem necessidade de definir o número de *clusters*.	Sensível a configuração de parâmetros; Não aplicável a altas dimensões.
Baseado em Grafos	Alta eficiência; agrupamento com alta acurácia.	Alta complexidade computacional com o aumento da complexidade dos grafos.

Tabela 1 – Vantagens e desvantagens de cada tipo de agrupamento apresentado por Xu e Tian (2015).

1.2.1.1 Baseado em Centroide

Algoritmos baseados em centroide tem como ideia principal utilizar o centro do conjunto de dados como o centro do grupo correspondente. Métodos como o *K-means* (MACQUEEN et al., 1967) e o *K-medoids* (KAUFMAN; ROUSSEEUW, 1990) são os mais conhecidos, porém, podemos citar também os métodos PAM (*Partitioning Around*

Medoids) e CLARA (*Clustering Large Applications*), propostos por Kaufman e Rousseeuw (1990), e CLARANS (*Clustering Large Applications based on Randomized Search*) (NG; HAN, 2002).

O *K-means* é considerado um dos algoritmos mais populares de agrupamento de dados e tem como objetivo minimizar a soma dos erros quadrátricos de cada amostra ao centroide mais próximo. Os centroides são representados pela média das amostras contidas em cada grupo e, por isso, o algoritmo realça estruturas de grupos compactos. Seu funcionamento pode ser descrito como no Algoritmo 1 (THEODORIDIS, 2015):

Algoritmo 1 – Algoritmo *K-means*

1 Selecione um número K de *clusters*
2 Inicialize aleatoriamente os centroides $\boldsymbol{\mu_k}$ de cada *cluster*, com $k = 1, 2, ..., K$
3 **enquanto** *o critério de parada não é satisfeito* **faça**
4 **para** *cada amostra* $\boldsymbol{x_n}$, $n = 1, 2, ..., N$, *do conjunto de dados* **faça**
5 Determine o centroide $\boldsymbol{\mu_k}$ mais perto de $\boldsymbol{x_n}$ minimizando
6 $C_k := \sum_{\boldsymbol{x_n} \in C_k} ||\boldsymbol{x_n} - \boldsymbol{\mu_k}||^2$
7 **fim**
8 **para** *cada cluster* $\boldsymbol{\mu_k}$, $k = 1, 2, ..., K$ **faça**
9 Atualize $\boldsymbol{\mu_k}$ como a média de todos os pontos do *cluster*
10 $\boldsymbol{\mu_k} := \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x_n}$
11 **fim**
12 **fim**
13 Conjunto de *clusters*

O *K-means* se tornou atrativo para diversas aplicações devido sua simplicidade computacional, fácil implementação, eficiência e sua adequação a grandes conjuntos de dados. Apesar disso, o algoritmo possui algumas desvantagens como: a não garantia da convergência para o mínimo global devido suas inicializações aleatórias; sensibilidade a *outliers* e dados ruidosos; não aplicável a conjuntos de dados não compactos, como é possível observar os resultados do algoritmo nos conjuntos de dados *Circles* e *Moons* apresentados na Figura 4.

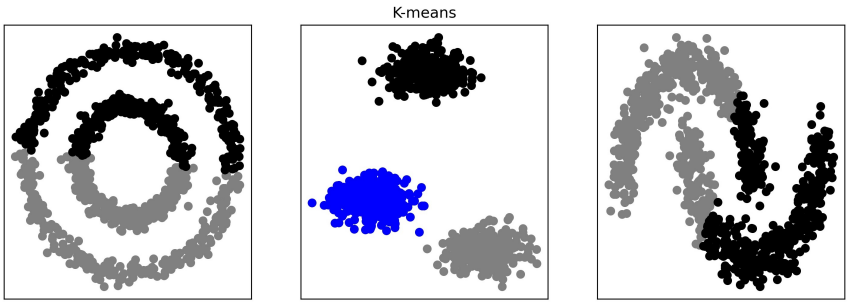


Figura 4 – Resultado do algoritmo *K-means* gerado respectivamente para as bases *Circles*, *Blobs* e *Moons*.

Para solucionar a influência de *outliers* e dados ruidosos enfrentada pelo *K-means*, o algoritmo *K-medoids* foi criado utilizando uma nova representação para os centroides. Essa nova representação é denominada *medoids* e é atribuída por um ponto do conjunto de dados, geralmente as amostras localizadas mais ao centro do conjunto. Uma outra vantagem de representar os grupos por *medoids* é que o algoritmo pode ser utilizado em conjuntos de dados com domínios contínuos e discretos, diferente do *K-means* que é mais adequado para domínios contínuos (THEODORIDIS; KOUTROUMBAS, 2008). Porém, uma desvantagem do algoritmo é que essa representação de *medoids* aumenta sua complexidade computacional.

1.2.1.2 Baseado em Hierarquia

Algoritmos hierárquicos representam o conjunto de dados de acordo com estruturas de hierarquias, que podem ser divididas entre aglomerativos e divisivos. Os aglomerativos são baseados na abordagem *bottom-up*, no qual inicialmente cada amostra pertence a um grupo e iterativamente os grupos vão se formado a partir da similaridade entre eles. Os divisivos seguem a abordagem *top-down*, no qual inicialmente todas as amostras pertencem a um mesmo grupos e iterativamente grupos menores vão sendo formados a partir da divisão dos dados. A junção ou divisão dos grupos é feita de acordo com alguma medida de similaridade, escolhida para otimizar algum critério como, por exemplo, a soma dos quadrados (ROKACH, 2009). Os métodos também podem ser divididos conforme ligações *Single-link*, *Complete-link* e *Average-link* (SAXENA et al., 2017).

Na ligação *Single-link*, também chamada de conectividade ou método do vizinho mais próximo, a ligação entre dois grupos é feita por dois elementos, um em cada grupo, que estão mais próximos um do outro. De forma oposta, a *Complete-link*, também chamada de diâmetro ou método do vizinho mais distante, considera a ligação entre amostras mais distantes. Por fim, a *Average-link*, conhecida como método da mínima variância, considera a distância entre dois grupos como sendo a média de qualquer membro de um grupo para qualquer membro do outro. Por essas características, as ligações podem trazer diferentes

formas de agrupamento para um mesmo conjunto de dados. Na Figura 5, o resultado de um algoritmo hierárquico aglomerativo é apresentado. Diferente dos resultados obtidos pelo *K-means* na seção anterior, podemos observar que os agrupamentos gerados representam melhor os conjuntos que destacam mais a conectividade dos dados.

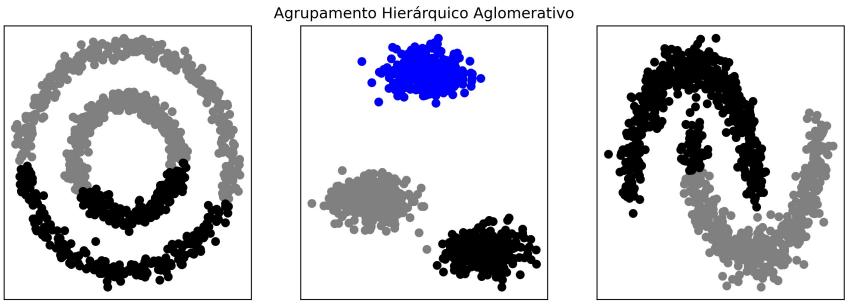


Figura 5 – Resultado do algoritmo de agrupamento hierárquico aglomerativo com ligação *Average-link* gerado respectivamente para as bases *Circles*, *Blobs* e *Moons*.

Uma forma de visualizar estruturas hierárquicas é por meio dos Dendrogramas. Nesses gráficos podemos observar vários níveis de agrupamento e a semelhança dos subconjuntos formados. A altura das linha verticais mostram o grau de diferença entre os ramos. Na Figura 6 apresentamos um dendrograma do conjunto de dados *Moons*, onde é possível observar em um nível mais alto uma separação entre dois grandes grupos.

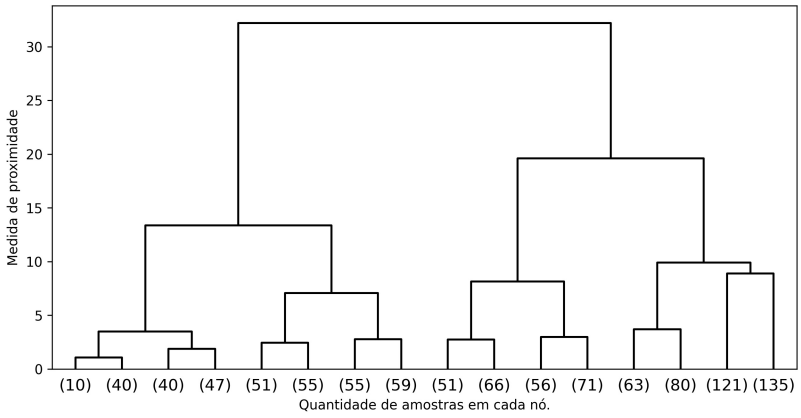


Figura 6 – Dendrograma do conjunto de dados *Moons*, gerado a partir da biblioteca *Scipy* e *Scikit-learn*.

Métodos de agrupamento hierárquico conhecidos são: BIRCH (*Balanced Iterative Reducing and Clustering using Hierarchies*) (ZHANG; RAMAKRISHNAN; LIVNY, 1996), CURE (*Clustering using Representatives*) (GUHA; RASTOGI; SHIM, 1998), ROCK (Robust Clustering using Links) (RAJEEV; RASTOGI; SHIM, 1999) e Chameleon (KARYPIS; HAN; KUMAR, 1999).

1.2.1.3 Baseado em Densidade

Algoritmos baseados nessa abordagem assumem que regiões de alta densidade são separados em grupos. Os grupos então são criados a partir de distribuições de probabilidades específicas e uma mistura delas podem representar a distribuição geral do conjunto de dados (XU; WUNSCH, 2005). Métodos baseado em densidade sofrem com conjuntos em que os grupos possuem densidades similares e em aplicações de alta dimensão (SARKER, 2021).

Métodos conhecidos dessa abordagem são: DBSCAN (*Density-based Spatial Clustering of Applications with Noise*) (ESTER et al., 1996), OPTICS (*Ordering points to Identify the Clustering Structure*) (ANKERST et al., 1999) e *Mean Shift* (COMANICIU; MEER, 2002).

1.2.1.4 Baseado em Grafos

Nessa abordagem de agrupamento, os grupos são representados por grafos, os nós ou vértices correspondem as amostras do conjunto de dados e as arestas refletem as proximidades entre duas amostras. A árvore geradora mínima (MST, do inglês *Minimal Spanning Tree*) (ZAHN, 1971) é muito utilizada na construção desses algoritmos, busca representar um grafo de modo que as ligações entre as amostras tenham o menor custo geral associado. Uma das formas de se encontrar uma MST é através do algoritmo de *Kruskal* (KRUSKAL, 1956).

Algoritmos hierárquicos também podem ser vistos como algoritmos baseados em grafos, um exemplo são os agrupamentos gerados por ligação *single-link* que podem ser considerados subgrafos de uma árvore geradora mínima (ROKACH, 2009). Outros algoritmos que seguem essa abordagem são: *Spectral clustering* (DONATH; HOFFMAN, 2003) e CLICK (Clustering Identification via Connectivity Kernels) (SHARAN; SHAMIR, 2000).

1.2.2 Redução de dimensionalidade

Muitos problemas reais possuem conjuntos de dados que estão em um espaço de alta dimensão, ou seja, que possuem diversos atributos ou características. Em tarefas de agrupamentos de dados, por exemplo, o ser humano é capaz de analisar os resultados de forma muito eficiente em até três dimensões. Porém, a partir do momento que a dimensão aumenta, essa compreensão se torna mais difícil. O mesmo acontece com os algoritmos, a análise dos dados fica mais complexa e perde-se a intuição do comportamento dos modelos.

A "maldição da dimensionalidade" (*Curse of Dimensionality*), cunhado por Bellman (1961), é uma expressão muito utilizada no campo científico para representar os eventos que ocorrem com dados de alta dimensão e que podem impactar o desempenho dos

modelos de aprendizado. Verleysen e François (2005) apresentam esses fenômenos; alguns estão relacionados às propriedades geométricas e às concentrações de normas. Em relação às propriedades geométricas, os autores demonstram que elas se tornam contra intuitivas e longe de propriedades que observamos em dimensões menores. Nas concentrações de normas é apresentado como as funções de distância em altas dimensões podem apresentar outros tipos de comportamentos. Beyer et al. (1999), por exemplo, demonstraram que métricas como a abordagem de vizinhos mais próximos quando utilizadas em alta dimensão não são capazes de continuar representando da maneira correta as noções de distâncias; isso significa que, se considerarmos uma amostra como referência, os pontos mais distantes e os mais próximos dela não vão possuir diferenças significativas. Em Aggarwal, Hinneburg e Keim (2001), os pesquisadores demonstram que em alta dimensão, métricas como a de *Manhattan* produzem melhor desempenho do que a distância Euclidiana.

Uma forma de enfrentar problemas em alta dimensão é utilizar métodos não supervisionados capazes de reduzir a dimensão do conjunto de dados, de uma maneira que preservem as informações importantes, facilitando a compreensão e diminuindo a complexidade computacional.

Um método muito utilizado para esse objetivo é a Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis*). Formalmente, suponha que, inicialmente, temos um conjunto de dados de H amostras e D dimensões. O objetivo do PCA é projetar os dados originais $\{\boldsymbol{a_1}, ..., \boldsymbol{a_h}\}$, $h = 1, ..., H$, para um espaço de dimensão menor G, de modo que o novo conjunto $\{\boldsymbol{b_1}, ..., \boldsymbol{b_j}\}$, $j = 1, ..., J$, preservem a máxima variância dos dados originais (WATT; BORHANI; KATSAGGELOS, 2016).

1.2.3 Autossupervisionado

Na literatura, o aprendizado autossupervisionado é considerado uma subcategoria dos métodos não supervisionados (SCHMARJE et al., 2021). Esse novo aprendizado surge com o questionamento de como os humanos e animais aprendem mesmo sem receber informações explícitas, apenas com as observações e interações no mundo real. Um exemplo disso é a comparação do aprendizado humano com o de carros autônomos. Carros autônomos enfrentam dificuldades mesmo com muitas horas de treinamento, enquanto para os humanos com poucas horas de prática é possível aprender a dirigir (LECUN; MISRA, 2021). O objetivo principal da aprendizagem autossupervisionada é tentar construir e reproduzir esse tipo de conhecimento nos modelos de aprendizado de máquina. Os modelos são treinados com amostras não rotuladas de forma a obter representações e reconstruir os dados, criando pseudo rótulos de forma automática e que serão utilizadas posteriormente para outras tarefas (QI; LUO, 2020).

Aplicações em processamento de linguagem natural têm se destacado por obter bons desempenhos. Um exemplo de aplicação é em problemas em que os modelos precisam

preencher sentenças a partir do aprendizado adquirido no treinamento (DEVLIN et al., 2018). Uma revisão completa sobre o tema pode ser encontrada em Jing e Tian (2020) e Qi e Luo (2020).

1.3 Métodos semissupervisionados

Segundo Chapelle, Schölkopf e Zien (2006), o tema de aprendizado semissupervisionado se iniciou com as ideias da autoaprendizagem, conhecida também como autotreinamento e autorrotulagem. Trabalhos como o de Scudder (1965) e Fralick (1967), por exemplo, são citados por apresentarem há muito tempo ideias sobre o uso de dados não rotulados na classificação. Porém, foi na década de 90 que o interesse pelo aprendizado semissupervisionado aumentou, Merz, Clair e Bond (1992) foram os primeiros a utilizarem o termo semissupervisionado no contexto da utilização de dados rotulados e não rotulados na tarefa de classificação (CHAPELLE; SCHÖLKOPF; ZIEN, 2006).

O principal objetivo dos algoritmos de aprendizagem semissupervisionada é combinar muitas amostras rotuladas com poucas não rotuladas de forma a desenvolver modelos mais eficientes (ZHU, 2005). Esses algoritmos tentam melhorar a performance do aprendizado supervisionado ou do não supervisionado utilizando a combinação dos métodos e informações geradas por cada um (ENGELEN; HOOS, 2020).

No aprendizado supervisionado, por exemplo, a obtenção de dados rotulados pode ser um desafio, pois pode ser custoso em termos de tempo e dinheiro, e é o ponto principal para o desenvolvimento dos modelos. Se os dados de treinamento são escassos, o modelo não conseguirá representar da maneira correta todo o conjunto de dados do problema. Além disso, no mundo real existe uma grande quantidade de dados não rotulados e saber utilizá-los pode trazer vantagens nas construções de modelos de aprendizado. Por sua vez, um dos desafios do aprendizado não supervisionado é contornar os ótimos locais das tarefas de agrupamento, consequência das inicializações aleatórias dos algoritmos. Utilizar conhecimentos sobre as amostras ou classes para direcionar o aprendizado, pode ser uma alternativa para resolver esse problema.

No aprendizado semissupervisionado, as tarefas podem ser tanto de classificação quanto de agrupamento de dados. Os algoritmos de classificação semissupervisionada são divididas em Transdutivo e Indutivo, e em um segundo nível, pela forma como os métodos incorporam os dados não rotulados. Já os algoritmos de agrupamento podem estar relacionados com a existência de dados parcialmente rotulados ou o conhecimento sobre restrições das amostras. Nos tópicos abaixo abordaremos os principais métodos de classificação e agrupamento de dados semissupervisionados, porém, uma revisão completa dessas duas tarefas podem ser encontrados em Engelen e Hoos (2020) e Bair (2013).

1.3.1 Classificação Semissupervisionada

Na classificação semissupervisionada, os métodos são categorizados pela aprendizagem transdutiva e indutiva. No primeiro caso, o objetivo do modelo é rotular amostras desconhecidas a partir do conhecimento adquirido sobre os rótulos de algumas amostras. Nesses métodos, os modelos não conseguem generalizar e lidar com dados não vistos, por esse motivo o modelo pode não ser capaz de encontrar rótulos para todas as amostras desconhecidas. No segundo caso, a aprendizagem considera tanto amostras rotuladas como não rotuladas; as não rotuladas ajudam o modelo a compreender melhor a distribuição do conjunto inteiro de dados. Diferente do primeiro caso, o modelo criado a partir desse aprendizado consegue predizer amostras não vistas, pois é capaz de utilizar o aprendizado adquirido para generalizar casos futuros.

Para a revisão dos métodos de classificação semissupervisionada, seguiremos a categorização apresentada por Engelen e Hoos (2020). Para os autores a aprendizagem transdutiva está relacionada a métodos baseados em grafos, enquanto os indutivos são divididos em três categorias: métodos *wrapper*, pré-processamento não supervisionado e métodos intrinsecamente semissupervisionado. Nos próximos tópicos apresentaremos os principais métodos dessas duas abordagens.

1.3.1.1 Transdutivo

Os métodos transdutivos não são capazes de generalizar sua predição para dados não vistos, pois não trabalham com o espaço inteiro das variáveis de decisão, por esse motivo são muito associados a métodos com estruturas de grafos. Diferente dos modelos de aprendizagem que possuem fase de treinamento e teste, esses métodos se dividem apenas pelos dados rotulados e não rotulados. A partir da estrutura de grafos, os rótulos são propagados para amostras não rotuladas de acordo com medidas de similaridade.

1.3.1.2 Indutivo

Na primeira categoria do aprendizado indutivo estão os algoritmos baseados na abordagem *Wrapper*, modelos supervisionados são utilizados de forma iterativa de modo que a cada novo treinamento o modelo encontre classes para amostras não rotuladas. Alguns exemplos são os algoritmos de Autotreinamento ou Autoaprendizagem, e Cotreinamento.

No Autotreinamento, um modelo supervisionado é treinado com amostras rotuladas e utilizado para predizer amostras não rotuladas na etapa de teste. Após a predição, as amostras que foram rotuladas e selecionadas conforme um limiar de decisão são incluídas no conjunto de treinamento. O classificador é treinado novamente considerando o novo conjunto de dados e essa etapa é repetida a cada iteração até que todas as amostras não rotuladas encontrem seus respectivos rótulos.

Uma extensão da técnica anterior é o Cotreinamento, na qual classificadores são treinados em diferentes conjuntos de atributos das amostras rotuladas. Posteriormente, as amostras não rotuladas são classificadas, resultando em pseudo rótulos. Essas novas amostras são adicionadas no conjunto de dados do outro classificador, e cada classificador é treinado novamente, o processo se repete até que todas as amostras sejam rotuladas. A ideia de utilizar diferentes visualizações dos dados surge com o intuito de gerar maior diversidade no treinamento do modelo de aprendizado (ENGELEN; HOOS, 2020). Um exemplo de aplicação de cotreinamento são em classificações de imagem e vídeo. Um vídeo legendado, por exemplo, pode ser representado pelas suas características visuais e textuais. Utilizar as informações provenientes de cada uma delas e combinar com a inclusão de dados não rotulados pode complementar o treinamento de modelos de aprendizado e melhorar o desempenho dos algoritmos (GUPTA et al., 2008).

Na segunda categoria, estão os métodos de pré-processamento não supervisionado. Nessa abordagem, as amostras não rotuladas são incorporadas em uma etapa anterior à classificação. Os resultados obtidos a partir desse pré-processamento de dados são utilizados para complementar o treinamento de modelos de classificação. Diferente dos métodos *wrapper*, o classificador é treinado apenas com as amostras rotuladas originais. Técnicas como a extração de atributos, pré-agrupamento de dados ou pré-treinamento de modelos são tipos de pré-processamento não supervisionados utilizados (ENGELEN; HOOS, 2020). A extração de atributos, também considerada como uma etapa da redução da dimensionalidade, tem como intuito selecionar as características que melhor representam a estrutura do conjunto de dados. Além do PCA, citado na seção 1.2.2, os *Autoencoder* também são algoritmos utilizados para essa tarefa. O pré-treinamento de modelos são conhecidos pelas aplicações em aprendizagem profunda, principalmente por auxiliar na configuração de parâmetros das redes neurais profundas.

Na terceira categoria estão os métodos intrinsecamente semissupervisionados. Esses métodos são chamados dessa forma pois são algoritmos supervisionados que são adaptados para incorporar amostras não rotuladas nas funções objetivo (ENGELEN; HOOS, 2020). Um exemplo, são os modelos generativos (*Generative models*) como a Mistura Gaussiana. Esses métodos utilizam as informações de dados rotulados e não rotulados para identificar a distribuição do conjunto de dados.

1.3.2 Agrupamento de Dados Semissupervisionado

Ao depararmos com dados rotulados, pensamos logo em aplicações como os algoritmos supervisionados, porém, quando esses dados existem em conjunto com dados não rotulados, podemos pensar em aproveitar essas informações para guiar processos não supervisionados (GRIRA; CRUCIANU; BOUJEMAA, 2004). Essa é a ideia principal dos algoritmos de agrupamento de dados semissupervisionados, utilizar informações prévia dos

dados de modo a direcionar o aprendizado dos modelos. Para Bair (2013), os métodos podem ser categorizados por incorporarem dados parcialmente rotulados ou pelo conhecimento de restrições dessas amostras. Nesta seção apresentaremos algoritmos que utilizaram o método clássico *K-means* como base para criar algoritmos semissupervisionados, mas na literatura outros algoritmos como os hierárquicos podem ser encontrados. Em Basu, Bilenko e Mooney (2003) é possível encontrar outros tipos de categorização dos agrupamento de dados semissupervisionados.

Nos métodos que consideram amostras parcialmente rotuladas, algoritmos conhecidos são o *Constrained K-means* e *Seeded K-means*. No *Constrained K-means* proposto por Basu, Banerjee e Mooney (2002), as amostras rotuladas são utilizadas para inicializar os centroides de cada grupo. Além disso, na etapa de atualização e cálculo do novos centroide, as amostras rotuladas sempre farão parte do grupo inicial, mesmo que elas fiquem mais próximas de outros grupos. Uma extensão desse método é o *Seeded K-means*, proposto também por Basu, Banerjee e Mooney (2002). Diferente do anterior, os pesquisadores alteraram a etapa de atualização considerando a atribuição das amostras conhecidas sempre para o centroide do grupo mais próximo. Essa alteração foi realizada para evitar o agrupamento incorreto das amostras rotuladas quando elas eram fixadas ao grupo inicial.

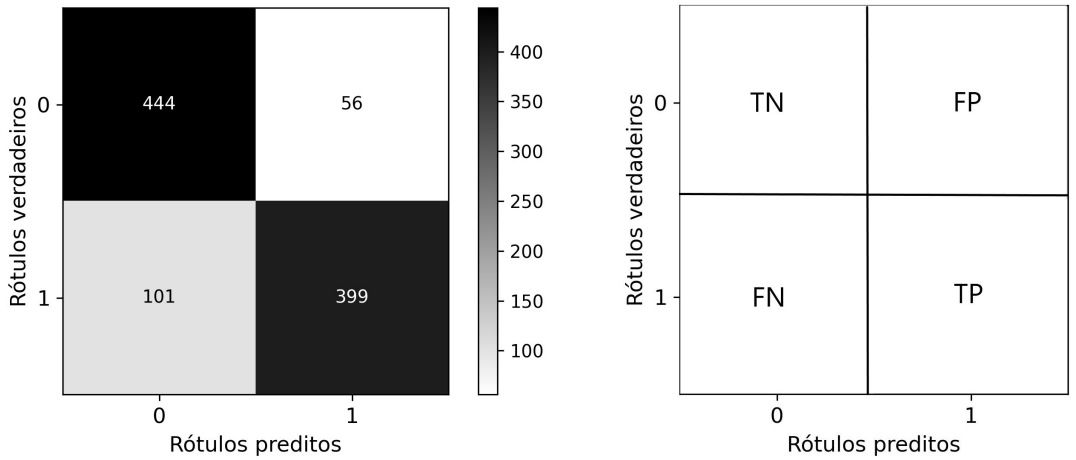
Nos métodos que utilizam o conhecimento de restrições, os algoritmos conhecidos são o *COP-Kmeans* e o *PCKmeans* (*Pairwise Constrained K-Means*). Esses algoritmos podem ser vantajosos em situações onde é mais fácil adquirir informações sobre as relações entre as amostras do que informações sobre as classes (BASU; BANERJEE; MOONEY, 2004). No *COP-Kmeans* desenvolvido por Wagstaff et al. (2001), o algoritmo incorpora o conhecimento prévio da relação entre as amostras rotuladas através de restrições do tipo *must-link* e *cannot-link*. Restrições *must-link* indicam que duas amostras devem ser agrupadas juntamente, por outro lado, para a restrição *cannot-link* as amostras não devem pertencem ao mesmo grupo. Essas restrições são consideradas na etapa de atribuição das amostras a seus respectivos grupos, evitando a violação das restrições definidas no problema. Uma extensão do algoritmo anterior é o *PCKmeans*, proposto por Basu, Banerjee e Mooney (2004), os autores modificaram a etapa de atualização permitindo a violação de restrições através da minimização de uma função que combina a distância de cada amostra ao centroide mais próximo e o custo de violação das restrições.

1.4 Métricas

Nesta seção apresentaremos as métricas que são utilizadas para analisar o desempenho dos métodos de aprendizado de máquina.

Matriz de Confusão

A matriz de confusão (*Confusion Matrix*) é uma ferramenta utilizada para avaliar o desempenho de modelos de aprendizado de máquina; por meio dela diversas métricas são construídas. A matriz apresenta a quantidade de amostras através das classes preditas no modelo e seus verdadeiros valores. A Figura 7a foi gerada para avaliar uma classificação de um conjunto de dados de 1000 amostras e duas classes, cada uma das classes possui 500 mostras. Os quadrantes no qual o modelo previu corretamente os rótulos verdadeiros, identificados pela cor preta, estão associados aos valores de Verdadeiro Negativo (TN, *True Negative*) como 444 e Verdadeiro Positivo (TP, *True Positive*) como 399. De maneira oposta, os quadrantes com rótulos diferentes representam os erros de classificação, associados aos valores de Falso Positivo (FP, *False Positive*) como 56 e Falso Negativo (FN, *False Negative*) como 101.



(a) Matriz de confusão gerada para um problema de classificação com TP=399, TN=444, FP=56 e FN=101.

(b) Quadrantes correspondentes aos valores de TP, TN, FP e FN.

Figura 7 – Matriz de confusão.

Acurácia

A métrica de acurácia, Equação 1.1, nos traz a percepção do desempenho geral do modelo, pois contabiliza a proporção entre a quantidade de acertos da classificação e o total das amostras. A acurácia assume valores entre 0 a 1, sendo 1 o acerto total da predição das amostras. Para o modelo de classificação da Figura 7a, a acurácia é de 0,84.

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.1}$$

Taxa de Verdadeiro Positivo (Sensibilidade ou *Recall*)

A Taxa de Verdadeiro Positivo (TPR), Equação 1.2, avalia o desempenho do algoritmo em classificar corretamente as amostras positivas. A métrica é calculada a partir da proporção entre quantidade de amostras com rótulos positivos classificadas corretamente e o total de amostras positivas. Para o modelo de classificação da Figura 7a, a taxa de verdadeiro positivo é de 0,80.

$$\text{TPR} = \frac{TP}{TP + FN} \tag{1.2}$$

Taxa de Verdadeiro Negativo (Especificidade)

Semelhante à métrica anterior, a Taxa de Verdadeiro Negativo (TNR), Equação 1.3, calcula a proporção da quantidade de amostras com rótulos negativos classificadas corretamente em relação a todas as amostras negativas. Para o modelo de classificação da Figura 7a, a taxa de verdadeiro positivo é de 0,88.

$$\text{TNR} = \frac{TN}{TN + FP} \tag{1.3}$$

Precisão

A Precisão calcula de acordo com a Equação 1.4 a proporção entre a quantidade de amostras positivas rotuladas corretamente pelo total de amostras positivas prevista pelo modelo, ou seja, avalia a precisão do modelo em predizer rótulos positivos. Para o modelo de classificação da Figura 7a, a precisão é de 0,88.

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{1.4}$$

F1-score

O F1-score, calculada como na Equação 1.5, representa um balanço da matriz de confusão quando se tem um conjunto de dados com classes desbalanceadas. A métrica assume valores entre 0 e 1, sendo 1 seu valor ótimo. Para o modelo de classificação da Figura 7a, o F1-score é de 0,84.

$$\text{F1-score} = \frac{2TP}{2TP + FP + FN} \tag{1.5}$$

A escolha das métricas deve se adequar ao tipo de problema e é tão importante quanto a escolha do modelo. Ao avaliar os resultados com uma determinada métrica podemos entender que o modelo está satisfatório, porém quando utilizamos outra, o modelo pode apresentar uma performance ruim. Por esse motivo, para cada tipo de problema deve-se analisar como os erros impactam nas tomadas de decisões, de modo a refletir o desempenho verdadeiro do modelo. Em alguns problemas, por exemplo, obter um valor menor de falso negativo é preferível e pode gerar menos impacto do que um valor menor de falso positivo. Dessa forma, utilizar mais de uma métrica para avaliar os resultados obtidos complementa a análise final e torna-a mais robusta.

2 Otimização Multiobjetivo

Os primeiros conceitos da Otimização Multiobjetivo são associados aos trabalhos de Edgeworth (1881) e Pareto (1906). Com trabalhos iniciais na área de ciências econômicas, foi a partir da década de 70, com a tradução do trabalho de Pareto, que o interesse da comunidade científica e o desenvolvimento de métodos cresceram nas áreas de engenharia e matemática aplicada (WECK, 2004). Atualmente, a Otimização Multiobjetivo é conhecida por ser um campo de estudo da área de Pesquisa Operacional e por sua grande importância prática. Várias aplicações de problemas reais utilizam a otimização multiobjetivo, alguns exemplos abrangem a gestão da cadeia de suprimentos, tomada de decisão médica, otimização de design aerodinâmico, bioinformática, entre outros.

A otimização multiobjetivo busca encontrar formas de tratar problemas com objetivos conflitantes. Esses objetivos assumem uma relação de compromisso, onde a melhora em um implica na piora do outro. Além disso, os objetivos geralmente podem não ser comparáveis em relação às suas grandezas e por isso, podem não ser agregados em uma única função-objetivo. Um exemplo prático da aplicação é no problema de seleção de portfólios. Nesse problema, um investidor procura obter uma carteira de ativos que melhor represente seu perfil de investimento, minimizando o risco e maximizando o retorno. Os objetivos (risco e retorno) são considerados conflitantes, pois sabe-se que investimentos em ativos com maiores riscos podem significar retornos maiores.

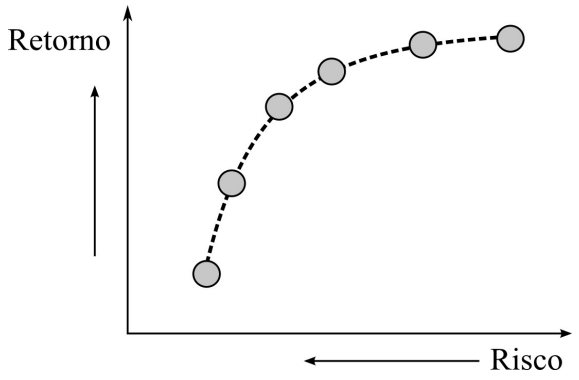


Figura 8 – Exemplo de um problema de seleção de portfólios.

Diferente da otimização mono-objetivo, que busca encontrar uma solução ótima única, a otimização multiobjetivo procura obter um conjunto de soluções que represente o equilíbrio entre os conflitos dos objetivos. Nos próximos tópicos abordaremos as principais definições de um problema multiobjetivo e formas de resolução através das Metaheurísticas.

2.1 Formalização de um problema de Otimização Multiobjetivo

Formalmente, um problema multiobjetivo pode ser visto como a minimização de $\boldsymbol{G}(\boldsymbol{x}) = [g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), ..., g_n(\boldsymbol{x})]^T$, sujeito a $\boldsymbol{x} \in \Omega$. Sendo n a quantidade de funções a serem minimizadas, Ω o espaço de decisão e $\boldsymbol{x} = (x_1, x_2, ..., x_n)^T$ o vetor das variáveis de decisão. Na otimização multiobjetivo lidamos com dois espaços de busca, o das variáveis e o dos objetivos. Para cada solução encontrada no espaço de busca, existe um ponto associado no espaço dos objetivos.

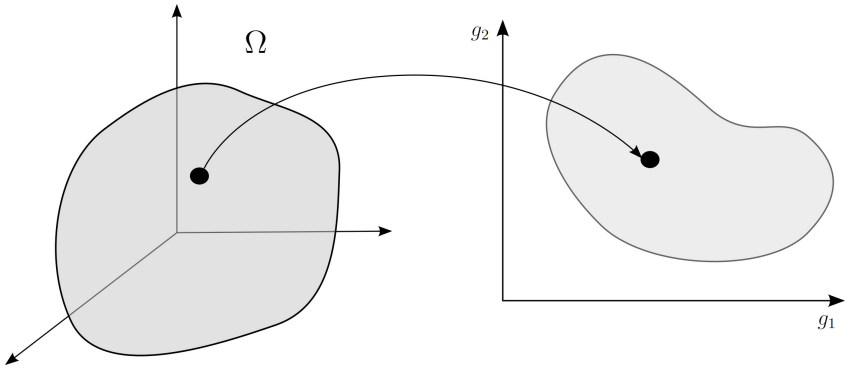


Figura 9 – Espaço das variáveis de decisão e espaço dos objetivos.

Uma solução x^* é dita eficiente, Pareto-ótima ou não-dominadas, quando não existe nenhuma outra solução factível do problema que seja melhor que ela. Formalmente, sejam $\boldsymbol{x_1}$ e $\boldsymbol{x_2} \in \Omega$, $\boldsymbol{x_1}$ domina $\boldsymbol{x_2}$, com $\boldsymbol{G}(\boldsymbol{x_1}) \preceq \boldsymbol{G}(\boldsymbol{x_2})$ para um problema de minimização, se e somente se $\forall\ i \in \{1, 2, ..., n\},\ g_i(\boldsymbol{x_1}) \leqslant g_i(\boldsymbol{x_2})$ e $\exists\ j \in \{1, 2, ..., n\},\ g_j(\boldsymbol{x_1}) < g_j(\boldsymbol{x_2})$ (FERREIRA, 1999). Dessa forma, uma solução $\boldsymbol{x}^*$ é eficiente, se e somente se, $\boldsymbol{x}^*$ é não-dominada em relação à Ω, ou seja, nenhum vetor do espaço de busca domina $\boldsymbol{x}^*$.

O conjunto das soluções eficientes é denominado Conjunto Pareto-ótimo e pode ser definido por $S^* = \{\boldsymbol{x} \in \Omega \mid \nexists\ \boldsymbol{z} \in \Omega, \boldsymbol{G}(\boldsymbol{z}) \preceq \boldsymbol{G}(\boldsymbol{x})\}$. Dado o conjunto Pareto-ótimo, podemos definir a Fronteira de Pareto como $FP = \{\boldsymbol{G}(\boldsymbol{x}) = [g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), ..., g_n(\boldsymbol{x})]^T \mid \boldsymbol{x} \in S^*\}$.

No exemplo da Figura 10, considerando o espaço dos objetivos de um problema de minimização, as soluções 1, 2 e 3 correspondem às soluções eficientes e suas imagens constituem a Fronteira de Pareto. As soluções 4, 5 e 6 são dominadas por uma ou mais soluções, por isso, não fazem parte do conjunto Pareto-ótimo.

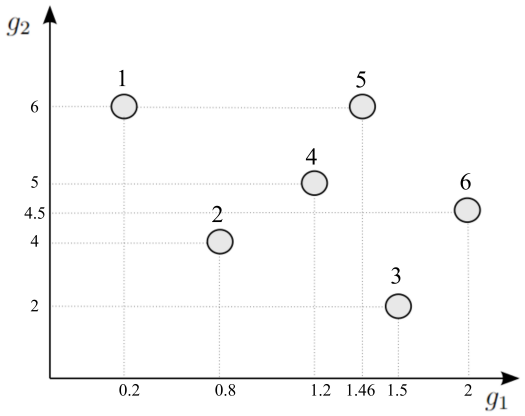


Figura 10 – Conceito de dominância em um problema multiobjetivo.

Na otimização multiobjetivo, além das soluções eficientes, existem as soluções especiais, denominadas como ideal, utópica e nadir. Essas soluções podem ser utilizadas para a construção de algoritmos, auxiliando na limitação do espaço de busca. Na prática, a solução ideal e a solução utópica são soluções que podem não existir, pois não representam o conflito entre os objetivos. Em um problema de minimização, a solução ideal é definida como o vetor $\boldsymbol{w^{ideal}} = (g_1^*, g_2^*, ..., g_n^*)^T$, tal que g_n^* é o menor valor assumido para a função g_n. A solução utópica $\boldsymbol{w^{utópica}}$ é definida como $\boldsymbol{w_i^{utópica}} = \boldsymbol{w_i^{ideal}} - \beta_i$ com $\beta_i > 0$, $\forall i = 1, 2, ..., n$ (DEB, 2014). Nesse caso, a solução utópica assume um valor menor que a ideal. Por fim, a solução nadir $\boldsymbol{w^{nadir}}$ é definida como a combinação entre os valores máximos de cada uma das funções assumidas pelas soluções do conjunto Pareto-ótimo.

Na Figura 11 é possível observar as soluções ideal, utópica e nadir, juntamente com as soluções $\boldsymbol{w_1}$ e $\boldsymbol{w_2}$ que minimizam, respectivamente, as funções g_2 e g_1.

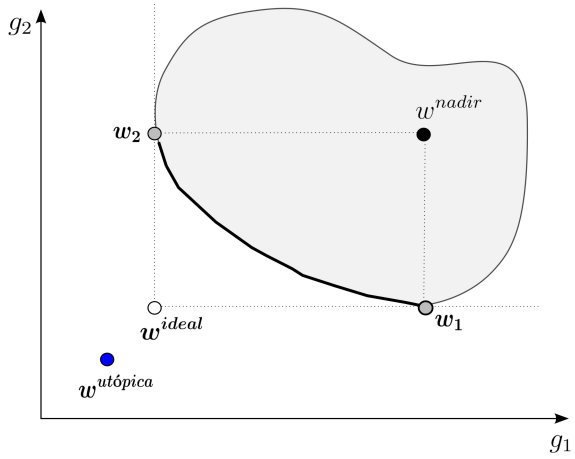


Figura 11 – Soluções especiais de um problema multiobjetivo.

2.2 Métodos de Resolução para Otimização Multiobjetivo

Diversos métodos foram criados com o intuito de encontrar respostas para problemas multiobjetivo. Dentre eles estão os clássicos, como os Métodos de Programação Matemática, e os chamados de alternativos, como as Metaheurísticas. Os clássicos são utilizados em problemas onde as funções-objetivo e restrições podem ser definidas analiticamente. Esses métodos são divididos em métodos a *priori*, *posteriori* e interativos, e diferem principalmente pela forma como incorporam as preferências do decisor no processo de otimização.

Métodos a *priori* mais conhecidos são os métodos Lexicográficos e Programação por Metas (CHARNES; COOPER, 1961). Nos métodos a *posteriori*, podemos citar o Método da Soma Ponderada (GASS; SAATY, 1955) e do ε-restrito (HAIMES, 1971). Métodos interativos conhecidos são o Método de Geoffrion, Dyer e Feinberg (GDF) (GEOFFRION; DYER; FEINBERG, 1972) e STEM (BENAYOUN et al., 1971).

Apesar dos métodos clássicos serem amplamente utilizados, em alguns problemas reais, funções-objetivo e restrições podem não ser definidas analiticamente, limitando a aplicação desses métodos. Nesses casos, a otimização pode seguir estratégias mais flexíveis, sem a definição direta das funções ou ser direcionada através dos dados, tornando a aplicação das metaheurísticas a abordagem mais adequada. Além disso, as técnicas de programação matemática possuem limitações quando aplicadas a problemas com fronteiras não convexas e descontínuas. As metaheurísticas são menos impactadas por esses problemas. Por esse motivo, nesse trabalho, focaremos principalmente nas metaheurísticas.

Uma metaheurística é uma estratégia de busca que aplica iterativamente técnicas heurísticas para resolução de problemas de otimização, fornecendo boas soluções a um custo computacional aceitável. Durante as iterações, a geração de soluções diferentes ou melhores podem ser obtidas com o uso de procedimentos de busca local, recombinação de soluções e reinserção de novas soluções (SÖRENSEN; GLOVER, 2013). Essa técnica dispõe-se a reduzir um pouco a qualidade das soluções em troca de uma flexibilidade na formulação e implementação de problemas que não podem ser obtidos com outras técnicas (CHOPARD; TOMASSINI, 2018). Apesar de ter a desvantagem da não garantia do ótimo global, algumas vantagens de sua utilização é a capacidade de se adequar facilmente aos problemas e de fugir de ótimos locais, consequência da exploração do espaço de busca. As metaheurísticas podem ser classificadas pelas estratégias de buscas utilizadas ou pela quantidade de soluções finais dos algoritmos (HUSSAIN et al., 2019). Na primeira categoria, as estratégias são divididas entre local e global, enquanto que na segunda, existem os algoritmos com uma única solução ou populacionais. Outras formas de categorização podem ser encontradas em Abdel-Basset, Abdel-Fatah e Sangaiah (2018) e Stegherr, Heider e Hähner (2020).

Algoritmos de buscas locais são considerados métodos de explotação, pois priorizam buscas em regiões com soluções de melhor qualidade. Alguns exemplos são a Busca Tabu (*Tabu Search*) (GLOVER, 1989), *Simulated Annealing* (KIRKPATRICK; JR; VECCHI, 1983), *Variable Neighborhood Search* (MLADENOVIĆ; HANSEN, 1997) e *Greedy Randomized Adaptive Search Procedure* (GRASP) (FEO; RESENDE, 1989). Por sua vez, algoritmos de buscas globais, priorizam a exploração, ou seja, geram soluções mais diversificadas, que expandem na fronteira e percorrem regiões desconhecidas. Exemplos conhecidos são os Algoritmos Genéticos (AG) (HOLLAND, 1992), *Scatter Seach* (GLOVER, 1977), Otimização por Colônias de Formigas (ACO, do inglês *Ant Colony Optimization*) (DORIGO; BIRATTARI; STUTZLE, 2006) e por Enxame de Partículas (PSO, do inglês *Particle Swarm Optimization*) (EBERHART; KENNEDY, 1995). Alguns desses métodos ainda são conhecidos por usarem uma combinação entre estratégias locais e globais. Estratégias locais geralmente apresentam uma única solução, enquanto as globais, que fornecem um conjunto de soluções a cada execução, são chamadas de populacionais. A Busca Tabu é um exemplo de algoritmo local com saída de uma única solução, enquanto, os AG são algoritmos globais e populacionais.

Os AG também são conhecidos por serem Metaheurísticas Evolutivas. Os métodos são chamados dessa forma por utilizarem conceitos de evolução para a construção de algoritmos. A literatura sugere que as metaheurísticas evolutivas aplicadas a problemas de otimização multiobjetivo apresentam melhor desempenho quando comparados a outras estratégias de busca (TORO et al., 2006). Os AG são amplamente utilizados, com sucesso, em inúmeras aplicações: configuração de sistemas complexos, alocação de tarefas, seleção de rotas, e outros problemas de otimização e aprendizado de máquina. No tópico a seguir, abordaremos as metaheurísticas evolutivas. Na Figura 12, resumimos os principais métodos de resolução de problemas multiobjetivo.

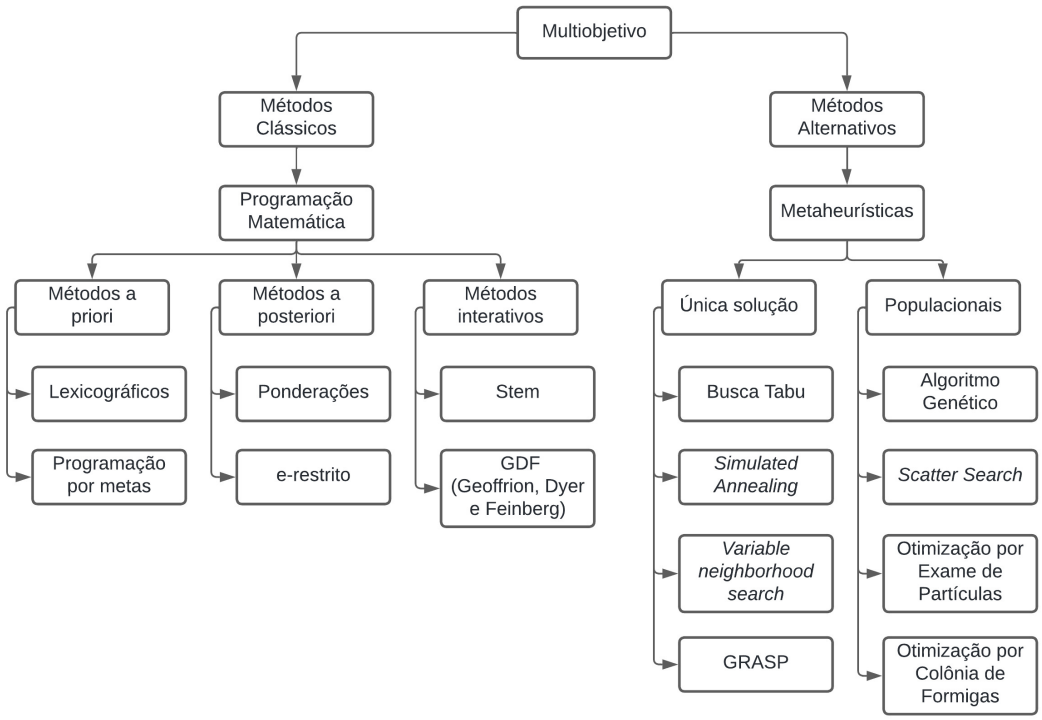


Figura 12 – Categorização dos métodos de resolução para multiobjetivo.

2.2.1 Metaheurísticas Evolutivas

Inspirados pelos princípios de seleção natural de Charles Darwin, Fogel, Owens e Walsh (1965), Holland (1973), Rechenberg (1973) e Schwefel (1995) são conhecidos por serem os pioneiros a trabalharem ideias sobre programação e estratégias evolutivas. Na abordagem evolutiva, uma população de indivíduos corresponde as possíveis soluções de um problema. Esses indivíduos são avaliados através de uma medida de desempenho, chamada de *fitness*, que irá avaliar a qualidade do indivíduo gerado e indicar sua chance de sobreviver e se multiplicar no ambiente. Em um problema de otimização, o *fitness* corresponde às funções-objetivo do problema. Através dos operadores de cruzamento e mutação novos indivíduos são criados para trazer variabilidade na população. Indivíduos considerados mais aptos são então selecionados e passam para as próximas gerações; a cada geração indivíduos melhores são criados. O resultado desse processo iterativo é a resposta final do problema a ser resolvido; no multiobjetivo, corresponde ao conjunto de soluções Pareto-ótimas.

O primeiro algoritmo evolutivo multiobjetivo criado foi proposto por Schaffer (1984), denominado de *Vector Evaluated Genetic Algorithm* (VEGA). O VEGA utilizava uma estrutura simples de algoritmo genético com um mecanismo de seleção baseada na proporção de cada função-objetivo. Esse algoritmo inicial apresentava alguns problemas como, por exemplo, a perda de boas soluções durante as gerações, provocada pelo mecanismo de seleção (HUSSAIN et al., 2019). Após a criação do VEGA, surgiram os algoritmos

chamados de primeira geração, marcados por estratégias de buscas com a incorporação dos conceitos de soluções Pareto-ótimas. Os mais conhecidos são o *Nondominated Sorting Genetic Algorithm* (NSGA) (SRINIVAS; DEB, 1994), *Niched-Pareto Genetic Algorithm* (NPGA) (HORN; NAFPLIOTIS; GOLDBERG, 1994) e o *Multi-Objective Genetic Algorithm* (MOGA) (FONSECA; FLEMING et al., 1993). Posteriormente, os algoritmos da segunda geração surgiram com a aplicação de estratégias mais eficientes e focadas no elitismo, dentre os mais conhecidos estão o *Strength Pareto Evolutionary Algorithm* (SPEA) (ZITZLER; THIELE, 1998), SPEA-II (ZITZLER; LAUMANNS; THIELE, 2001), *Pareto Archived Evolution Strategy* (PAES) (KNOWLES; CORNE, 2000) e NSGA-II (DEB et al., 2002).

Desde a criação desses métodos, as metaheurísticas evolutivas foram aplicadas em diversos problemas por apresentarem bons resultados em relação à convergência e garantia de diversificação de soluções não-dominadas (DEB; JAIN, 2013). Porém, quando aplicadas em problemas com mais de três objetivos, denominado problemas com muitos objetivos (*Many-objective*), algoritmos como o NSGA-II e SPEA-II podem se tornar menos eficientes. Pensando nessas aplicações, outros métodos foram propostos como o *Multiobjective Evolutionary Algorithm Based Upon Decomposition* (MOEA/D) (ZHANG; LI, 2007) e o NSGA-III (DEB; JAIN, 2013). Ishibuchi et al. (2016) apresentaram resultados comparando os algoritmos NSGA-II e NSGA-III, demonstrando que o desempenho dos métodos pode depender do problema a ser resolvido; um exemplo, são em problemas da mochila com muitos objetivos, no qual o NSGA-II apresenta resultados melhores quando comparados ao NSGA-III.

Além dos problemas com muitos objetivos, trabalhos mais recentes têm indicado uma linha promissora de pesquisa com aplicações das metaheurísticas evolutivas em Otimização Orientada por Dados (*Data-Driven Evolutionary Optimization*) (JIN et al., 2018). Nessa abordagem, problemas que não possuem funções-objetivo definidas analiticamente ou que possuem custo computacional alto para avaliar as funções, utilizam métodos de aprendizado de máquina em estruturas de algoritmos evolutivos com o objetivo de tornar o processo menos custoso. Um exemplo é o uso de métodos supervisionados e não supervisionados como substitutos das funções de avaliação de um algoritmo genético. Alguns exemplos de algoritmos propostos são *Reference Vector Based Evolutionary Algorithm* (RVEA) (CHENG et al., 2016), *Kriging assisted RVEA* (K-RVEA) (CHUGH et al., 2016), *Classification based Surrogate-assisted Evolutionary Algorithm* (CSEA) (PAN et al., 2018), *Hybrid Surrogate-assisted Many-objective Evolutionary Algorihm* (HSMEA) (HABIB et al., 2019).

Para o desenvolvimento desse projeto, o NSGA-II foi escolhido por ser tratar de um método clássico considerado eficiente para resolver problemas complexos. Além disso, é conhecido como um dos algoritmos estado-da-arte para tratamento de problemas

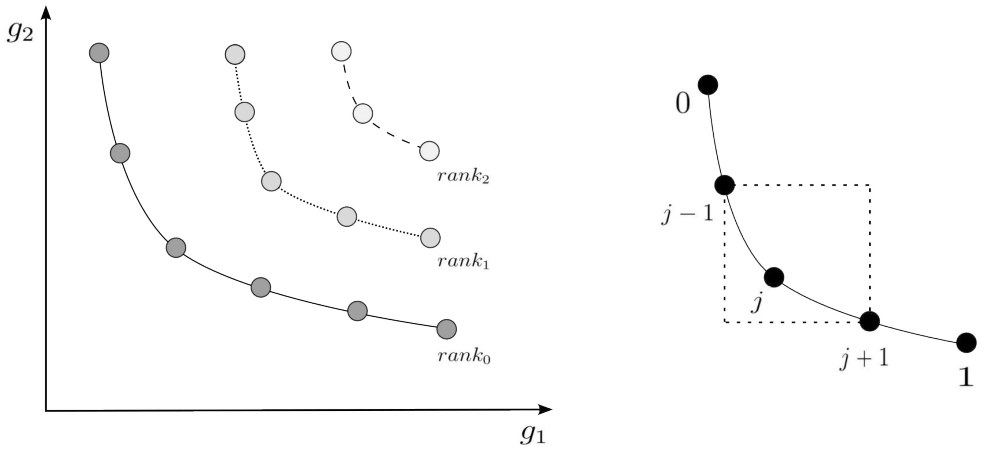
de otimização multiobjetivo. A seguir, apresentaremos o método com mais detalhes.

2.2.1.1 NSGA-II

Proposto por Deb et al. (2002), o NSGA-II buscou solucionar algumas críticas que permearam o trabalho anterior (NSGA). As principais melhorias do novo algoritmo foram: a utilização do elitismo com o uso de uma população mista; a proposta de uma classificação não-dominada mais eficiente; e o uso da distância de aglomeração como mecanismo de diversidade da população, sem necessidade de configuração de parâmetros.

No algoritmo, operadores de cruzamento e mutação são utilizados na população de pais Q_i para gerar uma população de filhos P_i, ambas com tamanho TP. As populações se juntam para a criação de uma população mista $T_i = Q_i \cup P_i$, de tamanho $2TP$, permitindo que os melhores indivíduos da população de pais e filhos passem para a próxima geração. As abordagens de classificação não-dominada e a distância de aglomeração são então aplicadas para selecionar a população da próxima geração. Dessa forma, cada indivíduo da população é avaliado com base nesses dois critérios, *rank* e *crowding distance*, respectivamente.

Na classificação não-dominada (Figura 13a), considerando as funções-objetivo de minimização, as soluções são organizadas em fronteiras considerando a relação de dominância entre elas. A primeira fronteira ($rank_0$) corresponde aos melhores indivíduos e será aquela em que os indivíduos são dominantes, ou seja, não existe nenhuma outra solução no conjunto que seja melhor do que ela. Após a classificação das fronteiras, calcula-se a distância de aglomeração (Figura 13b).



(a) Ordenação da população em fronteiras.

(b) Distância de aglomeração (*Crowding distance*).

Figura 13 – Classificação não-dominada e distância de aglomeração utilizada no NSGA-II.

A distância de aglomeração é utilizada para manter a diversidade da população e possui a vantagem da não necessidade de configuração de parâmetros por parte do usuário. Soluções mais diversificadas se encontram em regiões menos povoada e, por consequência, possuem maior distância de aglomeração. O cálculo dessa métrica representa

a soma das distâncias normalizadas dos vizinhos mais próximos da solução ao longo de cada objetivo. Para as soluções extremas, atribui-se grandes valores de distância e para o restante, calcula-se a distância de aglomeração. Formalmente, dada uma solução j, a distância d_j é definida como:

$$d_j = \sum_{n=1}^{N} \frac{g_n(j+1) - g_n(j-1)}{g_n^{max} - g_n^{min}} \tag{2.1}$$

Sendo N a quantidade de funções-objetivo, $j+1$ e $j-1$ os vizinhos mais próximos de uma solução j, g_n^{max} e g_n^{min} os valores máximos e mínimos de cada função g_n, considerando indivíduos de uma mesma fronteira (mesmo *rank*).

A população da próxima geração Q_{i+1}, de tamanho TP, será formada pelos melhores indivíduos e mais diversos. Inicialmente, indivíduos em fronteiras menores (menor *rank*) são escolhidos para compor a nova população, o processo continua até que o tamanho TP seja atingido. Quando dois indivíduos pertencerem a uma mesma fronteira, o critério de seleção será o que possuir maior distância de aglomeração. Todos os outros indivíduos que não couberem na população serão rejeitados. Na Figura 14, esse processo é ilustrado. Todos os indivíduos das fronteiras R_0, R_1 *e* R_2 são incluídos na nova população Q_{i+1}. Porém, como a fronteira R_3 possui mais indivíduos do que o necessário; os indivíduos são comparados entre si pela distância de aglomeração e os mais diversos são escolhidos.

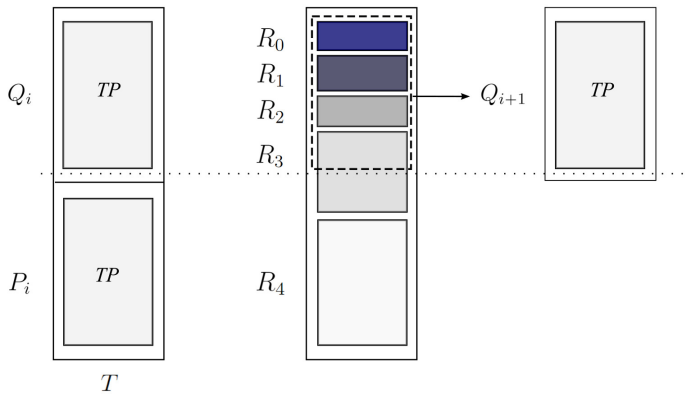


Figura 14 – Processo de seleção da próxima geração no NSGA-II.

Dessa forma, utilizando os critérios de classificação não-dominada (*rank*) e distância de aglomeração (*crowd*), uma solução a é considerada melhor que uma solução b, se:

$$(rank_a < rank_b) \vee \big((rank_a = rank_b) \ \wedge \ (crowd_a > crowd_b)\big) \tag{2.2}$$

Então, o NSGA-II se propõe a cuidar do avanço das soluções em direção à Fronteira de Pareto (através do operador de classificação não-dominada) e a garantir uma boa representatividade de diferentes regiões da fronteira não-dominada (através do operador de *crowding distance*), duas características importantes para o que se propõe fazer através deste trabalho: a análise a *posteriori* das soluções não-dominadas.

3 Agrupamento de Dados Multiobjetivo

No capítulo 1, foram apresentados os principais conceitos dos paradigmas supervisionados, não supervisionados e semissupervisionados. Dentre os métodos não supervisionados, estão as técnicas de agrupamento, que são utilizadas para a exploração de dados, auxiliando na identificação e na compreensão de padrões relevantes para o problema. Quando não há nenhuma informação sobre os dados, o desempenho do método depende do quanto o critério de agrupamento do algoritmo consegue capturar as características dos dados. Se métodos que enfatizam *clusters* esféricos são aplicados em conjuntos de dados com formatos arbitrários, o agrupamento não terá um bom desempenho visto que o algoritmo não conseguirá representar os diferentes formatos de *clusters*. Nesse sentido, o desafio da aplicação desses métodos é definir qual o melhor critério de agrupamento utilizar dentre os diversos algoritmos desenvolvidos.

Uma forma de lidar com esse desafio é tratar o problema de agrupamento como um problema multiobjetivo. Nessa abordagem, otimiza-se, simultaneamente, diferentes critérios de agrupamento de forma a encontrar um conjunto de soluções de alta qualidade e que represente o conflito entre essas funções-objetivo. No conjunto de dados com *clusters* arbitrários, uma forma de gerar boas soluções de agrupamento é considerar a formulação multiobjetivo com a otimização de funções de compactação e conectividade, pois o compromisso entre essas funções irá gerar soluções que apresentam essas duas características, facilitando a captura dos diferentes formatos de *clusters*. Dessa forma, a abordagem multiobjetivo pode ser vantajosa para capturar mais de uma característica em conjuntos de dados desconhecidos.

Considerar o problema de agrupamento de dados como um problema multiobjetivo, torna a aplicação das metaheurísticas evolutivas adequada. Como apresentado no capítulo 2, as metaheurísticas evolutivas são métodos eficientes de resolução para problemas com dois ou mais objetivos, sendo os algoritmos genéticos amplamente utilizados. Nesse trabalho, propomos a utilização de um algoritmo genético para resolver o problema de agrupamento multiobjetivo.

Agrupamento multiobjetivo evolutivo, tem sido alvo de pesquisas recentes. Segundo José-García e Handl (2021), a utilização dessa abordagem ganhou tração devido à habilidade do multiobjetivo de capturar diversas propriedades de agrupamento. Na literatura, a maioria dos algoritmos de agrupamento multiobjetivo são baseados nos algoritmos genéticos (MUKHOPADHYAY; MAULIK; BANDYOPADHYAY, 2015). Segundo Nanda e Panda (2014), após o trabalho de Handl e Knowles (2007), que propôs o *Multiobjective Clustering with Automatic K-determination* (MOCK), houve um crescimento em pesqui-

sas nessa área. Além disso, os algoritmos desenvolvidos demonstraram vantagens sobre métodos clássicos não supervisionados, evidenciando o benefício do uso da abordagem multiobjetivo, e foram aplicados em diversos problemas reais, ressaltando sua importância prática.

Em algumas pesquisas, o agrupamento multiobjetivo é ainda combinado com uma abordagem semissupervisionada, onde pressupõe a existência de muitos dados sem rótulos e poucos com rótulos. Nesses trabalhos, o aprendizado semissupervisionado é geralmente integrado nas funções-objetivo, introduzindo o conhecimento sobre as amostras rotuladas no critério de agrupamento.

3.1 Agrupamento Multiobjetivo

Os trabalhos apresentados nesta seção enfatizam o uso da otimização multiobjetivo em tarefas de agrupamento de dados. Os algoritmos desenvolvidos seguem estruturas de métodos conhecidos como o NSGA-II, PESA-II e MOEA/D, e adaptam as funções-objetivo para considerar o critério de agrupamento a ser otimizado. Toda a estrutura do algoritmo genético é modificada para se adequar ao problema de agrupamento de dados. Parte das informações foram baseadas na extensa revisão bibliográfica de Mukhopadhyay, Maulik e Bandyopadhyay (2015), onde os principais algoritmos de agrupamento multiobjetivo são apresentados.

Para a representação da solução (Figura 15), existem dois principais tipos: os baseados em protótipos de *clusters* e os baseados em ponto. Na primeira categoria, as soluções são representadas por centroide, *medoid* ou moda dos *clusters*. Essa representação é conhecida por ser mais simplista, de rápida convergência e adequada para identificar e incorporar sobreposições de *clusters*. Porém, possui desvantagem de não conseguir capturar *clusters* de formatos arbitrários, visto que enfatizam *clusters* compactos. Na segunda categoria, cada solução representa um possível agrupamento de dados. Dessa forma, o tamanho do indivíduo está associado a quantidade de amostras presentes no conjunto de dados. Cada posição do indivíduo pode assumir um valor do rótulo do *cluster*, as representações baseadas em *Cluster Label*, ou indicar a ligação com outras amostras em uma estrutura baseada em grafo, chamada de *Locus-based Adjacency Graph*. Esses dois tipos de representações possuem a vantagem de capturar formatos arbitrários de *clusters*, visto que não estão fixados nos protótipos e são independentes do número de *clusters*. Porém, a desvantagem dessa abordagem é a complexidade computacional, a convergência mais lenta e a não adequação a sobreposições de *clusters*.

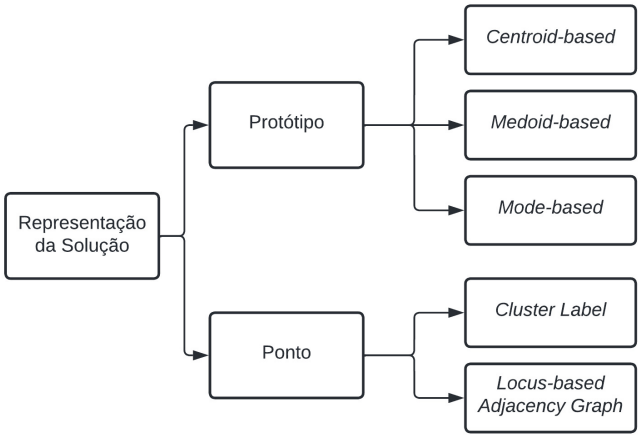


Figura 15 – Tipos de representações de indivíduos em algoritmos de agrupamento multiobjetivo, adaptado de Mukhopadhyay, Maulik e Bandyopadhyay (2015).

Para a escolha das funções-objetivo, diferentes características de agrupamento são considerados, os principais tipos são divididos em: compactação, separação dos *clusters* e densidade. Na Tabela 2 estão apresentados as principais funções-objetivo utilizadas nos algoritmos presentes na literatura. As funções com características de compactação e separação dos *clusters* tendem a capturar formatos mais esféricos e podem não desempenhar bem em conjuntos de dados com formatos arbitrários. Já funções como a densidade, ajudam a capturar formatos arbitrários, pois priorizam a conectividade dos dados.

Outra característica comum dos algoritmos de agrupamento de dados multiobjetivo é a etapa de seleção da solução final (Figura 16), nessa etapa, procura-se encontrar a solução não-dominada que melhor represente o agrupamento para determinado conjunto de dados. A seleção pode ser feita considerando uma métrica de validação de agrupamento de dados, utilizando o método *Knee-based* ou extraindo informações da fronteira final.

Na primeira categoria, uma métrica diferente da utilizada nas funções-objetivo é escolhida para selecionar uma solução. Na segunda categoria, métodos *Knee-based* são utilizados para identificar soluções que se encontram em uma região promissora da Fronteira Pareto-ótima, denominada de região *knee*. A região, localizada no meio da fronteira, tende a ter soluções com o *trade-off* mais interessante entre as funções-objetivo. De acordo com Garza-Fabre et al. (2022), em problemas onde a preferência do decisor é desconhecida, geralmente as soluções localizadas na região *knee* são preferíveis pelo decisor. Em Deb e Gupta (2011) são apresentados diversos métodos para encontrar essas regiões.

Na terceira categoria, considera-se que a exploração de todas as soluções da fronteira final pode fornecer informações sobre a estrutura de agrupamento procurada. Duas formas são utilizadas para encontrar a estrutura final: através de uma estrutura de grafos ou a partir do consenso entre as soluções. Na primeira forma, a informação da frequência em que as amostras são agrupadas juntamente se tornam as arestas (pesos)

de um grafo e, posteriormente, o grafo é particionado para gerar a solução final. Na segunda forma, amostras que são agrupadas juntamente podem ser consideradas do mesmo *cluster*. Assim, todas as soluções fornecem a informação de agrupamento e o rótulo final da amostra será o que for predominante, nesses casos parâmetros são criados para selecionar as informações mais confiáveis.

Tipo	Funções-objetivo	Compactação	Separação	Densidade
Protótipo	J_m *index*	x		
	Desvio Geral	x		
	Davies-Bouldin index (DB)	x	x	
	Xie-Beni index (XB)	x	x	
	Entropia intracluster (H)	x		
	I index	x	x	
	Separação de *Cluster*		x	
	Average Between Group Sum of Squares (ABGSS)		x	
Cluster Label	*Dunn index*	x	x	
	Conectividade			x
	Edge index		x	
	Silhouette index (SI)	x	x	
	Min-Max Cut	x	x	
	Total Within-Cluster Variance (TWCV)	x		

Tabela 2 – Principais tipos de funções-objetivo utilizadas em algoritmos de agrupamento multiobjetivo apresentados por Mukhopadhyay, Maulik e Bandyopadhyay (2015).

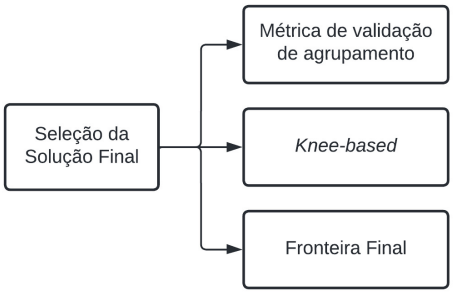


Figura 16 – Métodos de seleção da solução final em algoritmos de agrupamento multiobjetivo, adaptado de Mukhopadhyay, Maulik e Bandyopadhyay (2015).

Na Tabela 3 estão expostos os trabalhos que envolvem o tema de agrupamento multiobjetivo. Uma breve descrição de cada um deles será apresentado a seguir.

Referência	Algoritmo	Algoritmo Genético	Codificação	Função-objetivo
Handl e Knowles (2004) e Handl e Knowles (2007)	MOCK	PESA-II	*Locus-based*	Compactação e Conectividade.
Handl e Knowles (2006)	MOCK + Semi	PESA-II	*Locus-based*	Compactação, Conectividade e ARI
	Semi	PESA-II	*Locus-based*	ARI e Silhouette Index
Maulik, Mukhopadhyay e Bandyopadhyay (2009)	MOGA-SVM	NSGA-II	*Centroid-based*	*Xie-Beni index* e J_m *index*
Mukhopadhyay e Maulik (2011)	MOVGA	NSGA-II	*Centroid-based*	Compactação *Fuzzy* e Separação *Fuzzy*
Garza-Fabre, Handl e Knowles (2017)	Δ-MOCK	NSGA-II	*Locus-based*	Variância intracluster e Conectividade.
José-García et al. (2021)	MVMC	MOEA/D	*Medoid-based*	Matrizes de dissimilaridade: distância euclidiana, cosseno e *maximum edge distance* (MED).

Tabela 3 – Trabalhos relacionados a agrupamento de dados multiobjetivo.

Handl e Knowles (2007) apresentam o MOCK, um algoritmo multiobjetivo de agrupamento de dados automático. Pesquisas iniciais com esse algoritmo haviam sido apresentadas anteriormente em Handl e Knowles (2004), mas foi na publicação seguinte que o algoritmo ganhou popularidade. O trabalho proposto utiliza a codificação *Locus-based Adjacency Graph*, que permite tornar o algoritmo independente do números de *clusters*. Além disso, a proposta utiliza o algoritmo evolutivo PESA-II e otimizam duas funções-objetivo, compactação e conectividade. Na etapa de seleção, os autores propõem um método *Knee-based* para selecionar a melhor solução da fronteira não dominada.

Questões relacionadas à abordagem multiobjetivo para agrupamento de dados foram discutidas nesse trabalho inicial. A primeira delas envolve o benefício de se utilizar a abordagem multiobjetivo para o agrupamento de dados. A partir dos resultados obtidos, o MOCK apresentou melhor desempenho geral quando comparado com algoritmos clássicos de agrupamento de dados, que otimizam um único critério. Isso se deve porque esses algoritmos apresentam bons desempenhos apenas quando o critério de agrupamento é adequado às características dos dados. Em conjunto de dados onde a combinação entre os critérios permitiria melhor agrupamento, apenas o MOCK seria eficiente (HANDL; KNOWLES, 2007). Outra questão abordada é o conflito existente entre os objetivos. Quando os dados possuem *clusters* com formatos arbitrários e estruturas mais complexas foi observado o aumento do conflito entre os objetivos, diferente de estruturas mais compactadas e bem separadas, onde a melhor solução priorizou uma das funções-objetivo.

Em Handl e Knowles (2006), incorpora-se o aprendizado semissupervisionado

no problema de agrupamento através da extensão do MOCK (HANDL; KNOWLES, 2004), denominado de "MOCK + semi". Esse algoritmo inclui uma nova função-objetivo baseada na métrica *Adjusted Rand Index* (ARI), que calcula o grau de preservação do conhecimento inserido em uma possível solução. O ARI considera apenas as amostras rotuladas, enquanto, os outros dois objetivos, compactação e conectividade consideram todo o conjunto de dados. Além disso, outro algoritmo multiobjetivo, denominado Semi, é proposto utilizando duas funções-objetivo: ARI e a *Silhouette Width* (SI). A primeira função é calculada com as amostras rotuladas e a segunda com todos os dados. Os testes consideraram diferentes porcentagens de dados rotulados e ruídos, utilizando conjuntos de dados sintéticos e reais. Quando comparado com sua versão original, o MOCK + semi apresentou pouca diferença, obtendo uma pequena vantagem em alguns testes. Já o algoritmo multiobjetivo Semi, demonstrou superioridade em relação à abordagem semissupervisionada, supervisionada e não supervisionada.

Em Matake et al. (2007), propõe-se uma versão do MOCK para dados de grande escala. Considerando que a etapa final de seleção é mais custosa computacionalmente, os pesquisadores propõem a diminuição da fronteira não dominada com a remoção das soluções não promissoras e redundantes antes da utilização do método *Knee-based*. Apesar de demonstrar que o algoritmo desenvolvido é menos custoso, o desempenho foi inferior ao MOCK.

Maulik, Mukhopadhyay e Bandyopadhyay (2009) propõe o MOGA-SVM, um algoritmo genético multiobjetivo baseado no agrupamento *Fuzzy* com uma etapa posterior de classificação. O trabalho proposto utiliza uma codificação baseada em centroide, o NSGA-II e duas funções-objetivo, *Xie-Beni index* e J_m *index*. Diferente dos algoritmos anteriores, a etapa de seleção da solução utiliza todas as soluções da fronteira não dominada a fim de explorar as informações geradas e definir um agrupamento final. A ideia principal é que amostras agrupadas com frequência possuem semelhança e podem ser consideradas do mesmo *cluster*. Dessa forma, após a otimização, constrói-se a matriz de pertencimento *fuzzy* e a partir dela, amostras que atingiram um grau de pertencimento maior que um limiar α ($0 \leqslant \alpha \leqslant 1$) são consideradas com o mesmo rótulo e selecionadas para compor um conjunto de dados de treinamento, que será utilizado posteriormente no SVM. A quantidade de soluções selecionadas é definida pelo parâmetro βN, *sendo* $0 \leqslant \beta \leqslant 1$ *e* N, o número de soluções não-dominadas. O classificador SVM é treinado utilizando as amostras selecionadas e rótulos são encontrados para o restante do conjunto de dados. A solução final de agrupamento é obtida combinando os rótulos das amostras de treino e teste. O método proposto foi aplicado em um problema de análise de expressão genética e demonstrou desempenho superior a métodos simples de agrupamento, agrupamento fuzzy e multiobjetivo. O uso do método supervisionado em uma etapa posterior demonstrou melhorar a qualidade da solução final do agrupamento de dados.

Em Mukhopadhyay e Maulik (2011), o *Multiobjective Variable String Length Genetic Fuzzy Clustering* (MOVGA) é proposto com o objetivo de segmentar imagens. O algoritmo utiliza uma codificação baseada em centroide, o NSGA-II e otimiza duas funções-objetivo, compactação *fuzzy* e separação *fuzzy*. Assim como no MOCK, a codificação do MOVGA permite tornar o parâmetro de números de *clusters* flexível. Após a obtenção da fronteira, uma métrica externa de validação do agrupamento é utilizado para selecionar uma solução final. O método demonstrou resultados superiores a algoritmos como *K-means*, *Fuzzy C-means*, hierárquico, mono-objetivo e outras técnicas de agrupamento *fuzzy*. Além disso, foi aplicado com sucesso em um conjunto de dados de imagens de ressonância magnética do cérebro humano.

Dando continuidade ao MOCK, no segundo trabalho, Garza-Fabre, Handl e Knowles (2017) apresentam o Δ-MOCK. Diferente do seu antecessor, o algoritmo considera uma nova função objetivo em substituição à compactação, denominada de variância intra-cluster. Além disso, mudanças na inicialização e no esquema de representação de indivíduos são realizadas e substitui-se a estrutura de otimização do PESA-II para o NSGA-II. Esse trabalho concentrou-se em melhorar o desempenho do algoritmo em relação à produção de soluções com melhor qualidade de agrupamento e à redução do tempo computacional, as mudanças implementadas demonstraram que o algoritmo tornou-se mais eficiente.

Em trabalhos recentes, José-García et al. (2021) propõem o *Evolutionary Many-objective Approach to Multiview Clustering* (MVMC). Na abordagem *Multiview*, utiliza-se diferentes origens de visualizações de dados no processo de agrupamento. Essas visualizações podem estar associadas a diferentes conjuntos de atributos ou funções que realçam possíveis relações entre os dados, como as de dissimilaridades e similaridades. Considera-se que ao utilizar essas múltiplas visualizações, as informações se complementem e gerem soluções mais robustas e com melhores acurácias. No MVMC, utiliza-se a abordagem *many-objective* para atender a limitação dos algoritmos existentes de agrupamento *multiview*, que utilizam apenas duas visualizações de dados. O algoritmo proposto utiliza uma decodificação baseada em *medoids*, sendo necessária a definição prévia da quantidade de *clusters*, uma metaheurística evolutiva baseada em decomposição, MOEA/D e otimizam o critério de agrupamento *within-cluster scatter (WCS)*, combinado com diferentes matrizes de dissimilaridades. As matrizes representam as visualizações e foram criadas com diferentes métricas de distâncias como a do cosseno, *Maximum Edge Distance* (MED) e distância Euclidiana. Na etapa de seleção da solução, utiliza-se o *Silhouette index* para obter a solução mais adequada ao conjunto de dados. O trabalho foi aplicado em conjuntos de dados sintéticos e reais, sendo uma delas do Instituto Nacional do Câncer do Rio de Janeiro. O MVMC produziu melhores resultados quando comparado com o *K-means* e outros algoritmos de *Multiview*. Assim como em Handl e Knowles (2007), foi observado que em alguns conjuntos de dados a fronteira final representava a predominância de uma das métricas de distância, enquanto em outras era perceptível o equilíbrio entre

as visualizações, demonstrando a utilidade da abordagem multiobjetivo. Além disso, o algoritmo mostrou ser um algoritmo escalável, apresentando bons desempenhos com o aumento de funções-objetivo.

Em José-García e Handl (2021), os algoritmos Δ-MOCK e MVMC são comparados com o objetivo de investigar a influência do critério de agrupamento empregado nos algoritmos e as diferentes métricas de distâncias utilizadas. A principal diferença entre os algoritmos são as funções-objetivo utilizadas, no Δ-MOCK, utiliza-se a variância e conectividade, enquanto no MVMC, utiliza apenas o critério WCS, com diferentes matrizes de distância. Na comparação, ambos os algoritmos foram capazes de gerar soluções de qualidade para cada conjunto de dados. Para os pesquisadores, o trabalho demonstra que a escolha adequada das funções de distâncias em um algoritmo de agrupamento evolutivo é capaz de gerar boa soluções, mesmo utilizando apenas um critério de agrupamento. Além disso, demonstram que o uso das diferentes funções superam as limitações do uso de codificações baseadas em protótipos de *clusters*, que sofrem para capturar formatos arbitrários de *clusters*.

Na Figura 17, apresentamos uma linha do tempo com um breve resumo dos trabalhos citados anteriormente nessa seção.

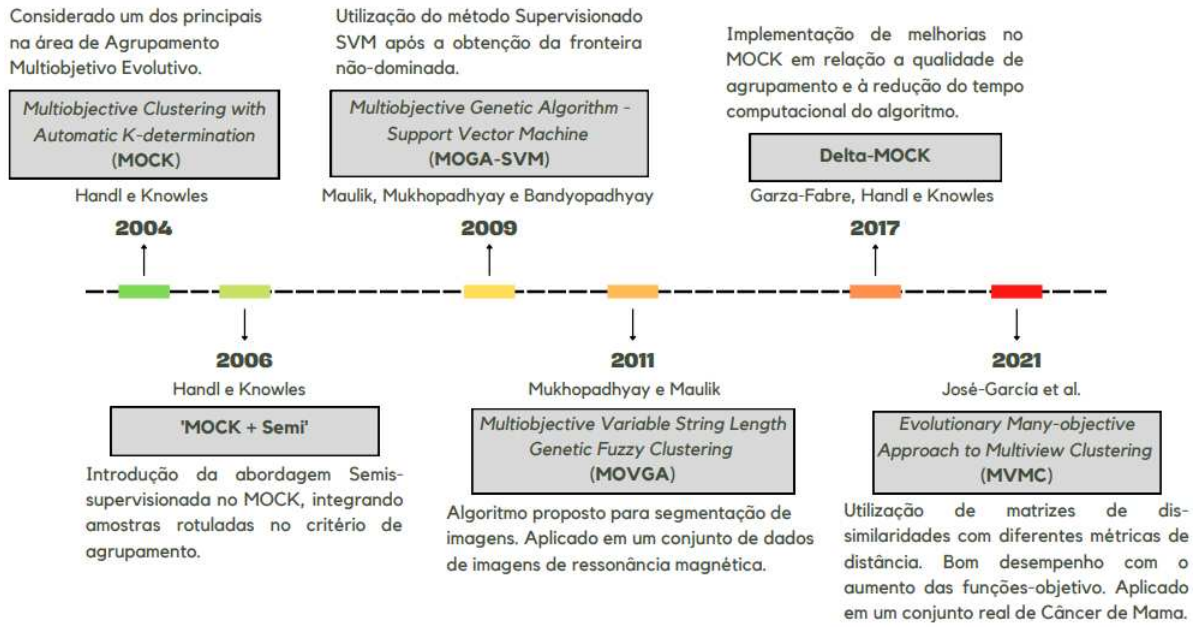


Figura 17 – Linha do tempo com o resumo dos trabalhos de agrupamento de dados multiobjetivo.

3.2 Agrupamento Semissupervisionado

Nesta seção, os trabalhos apresentados utilizam o agrupamento de dados semissupervisionado com métodos supervisionados em uma abordagem de autotreinamento.

Os métodos não supervisionados são modificados incluindo informações conhecidas sobre as amostras rotuladas, direcionando o agrupamento de dados; a partir das informações geradas, procura-se rotular as amostras desconhecidas. Após o agrupamento, o classificador é treinado com as amostras rotuladas e testado com as amostras não rotuladas e em seguida, combina-se os resultados das duas etapas de forma a selecionar as amostras com rótulos mais confiáveis. Essas amostras são incluídas no conjunto de treinamento e o modelo de classificação é treinado novamente, o processo se repete até que todas as amostras estejam rotuladas. Os limiares de seleção das amostras são utilizados para evitar erros de atribuições no processo de rotulagem e, consequentemente, a piora no desempenho de classificação.

Gan et al. (2013) propõem um método que utiliza o agrupamento de dados semissupervisionado *Fuzzy C-means* com o classificador SVM, denominado de SSFCM + SVM. Na primeira etapa do algoritmo proposto, amostras rotuladas e não rotuladas são utilizadas no SSFCM para revelar estruturas do espaço de dados e geram informações de pertencimento de cada amostra aos possíveis rótulos de dados. Na segunda etapa, o classificador SVM é treinado com o conjunto de dados rotulados e testado com as amostras da etapa anterior que obtiveram alto grau de pertencimento. Dessa forma, encontra-se os rótulos para amostras não rotuladas e o ciclo do autotreinamento continua até que todas as amostras estejam rotuladas. O algoritmo proposto demonstrou melhor desempenho em comparação a outros algoritmos de autotreinamento como *Self-training* e *Help-training*. Em comparação a outros algoritmos semissupervisionados que incluem o SVM, o algoritmo proposto apresentou resultados similares e, em alguns casos, melhores.

Semelhante ao trabalho anterior, Arora, Tushir e Kashyap (2020) propõem a combinação do SSFCM com o método supervisionado de classificação, *Naive Bayes*. O algoritmo apresentou resultados superiores em termos de acurácia e eficácia em comparação com algoritmos supervisionados e semissupervisionados, incluindo o proposto anteriormente por Gan et al. (2013). A Tabela 4 inclui os trabalhos citados anteriormente.

Referência	Algoritmo	Método	
		Semissupervisionado	Supervisionado
Gan et al. (2013)	SSFCM + SVM	SSFCM	SVM
Arora, Tushir e Kashyap (2020)	SSFBC	SSFCM	*Naive Bayes*

Tabela 4 – Trabalhos relacionados a agrupamento de dados semissupervisionado

Parte II

Desenvolvimento

4 Contribuições

Embora os pesquisadores tenham dedicado um esforço considerável para melhorar os métodos de agrupamento multiobjetivo em relação à otimização multiobjetivo, analisando convergência e diversidade na fronteira não-dominada, a maioria dos trabalhos da literatura não enfatizam análises a *posteriori*, para melhor compreensão dos resultados obtidos sob a ótica do aprendizado não supervisionado.

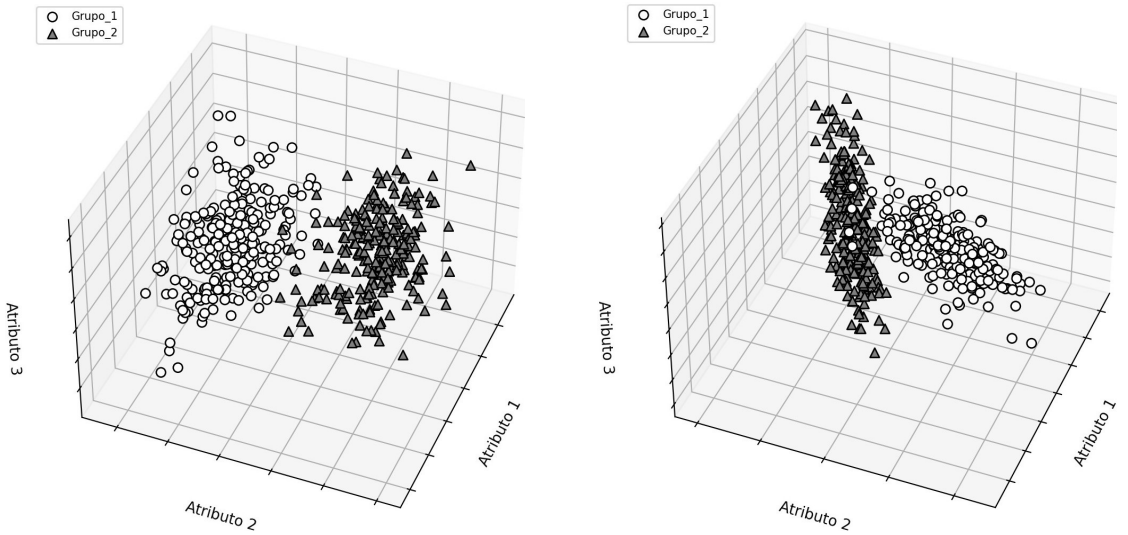
Sendo assim, a principal contribuição deste trabalho é analisar os resultados de um algoritmo de agrupamento multiobjetivo sob esses dois pontos de vista. Na visão da otimização multiobjetivo, além das análises tradicionais, investigaremos as soluções não-dominadas, a capacidade da fronteira de representar diferentes agrupamentos de dados e de gerar informações complementares sobre o problema, que podem ser importantes na análise exploratória de dados. Sob a perspectiva do aprendizado de máquina, utilizaremos métricas e ferramentas, como os *heatmaps*, para analisar o desempenho do agrupamento e da tarefa de rotulação de dados.

Dividiremos esse capítulo em três partes, na seção 4.1, apresentaremos uma breve motivação da pesquisa. Na seção 4.2, serão apresentados as contribuições teóricas, enfatizando a análise do agrupamento sob os pontos de vistas mencionados anteriormente. Na subseção 4.3, apresentaremos possíveis aplicações da abordagem desenvolvida nesse trabalho para a área de saúde e o problema do aprendizado semissupervisionado.

4.1 Motivação

Para ilustrar a motivação do nosso trabalho, trazemos um exemplo de uma possível aplicação na área de saúde. A área de saúde possui grande quantidade de dados disponível, porém, nem todos os dados possuem rótulos devido à complexidade do processo de atribuição ou à necessidade de conhecimentos específicos para realizar esta tarefa. A disponibilidade de muitos dados não rotulados torna adequada à aplicação de métodos não supervisionados.

Considere, então, um conjunto de dados com dados clínicos de diversos pacientes com suspeita de Covid-19. A fim de verificar os padrões dos dados para perfis de pacientes, agrupamos os dados considerando dois padrões de grupos, doentes e saudáveis. Como desconhecemos as distribuições dos dados, podemos ter grupos com diferentes características como, por exemplo, dados mais compactados (Figura 18a) ou mais conectados (Figura 18b).



(a) Dados com característica de compactação. (b) Dados com característica de conectividade.

Figura 18 – Diferentes distribuições de dados.

Agora, suponha que foram identificado alguns pacientes com o diagnóstico da doença. A Tabela 5, exemplifica esse novo conjunto de dados, com $S = S_1, S_2, ...$, sendo os dados clínicos coletados de cada paciente. Como rótulo do diagnóstico, o resultado 1 representa que o paciente está positivo para a doença e 0 como negativo. Pacientes que ainda não possuem o diagnóstico final são identificados com o rótulo -1.

Paciente	S_1	S_2	...	Resultado
1	1	1	...	**-1**
2	1	0	...	**0**
3	1	1	...	**1**
4	0	0	...	**0**
5	0	0	...	**-1**
6	1	0	...	**-1**
...	...	...	...	...
99	0	1	...	**0**
100	1	1	...	**1**

Tabela 5 – Conjunto de dados de pacientes com suspeita de Covid-19.

Com o objetivo de obter informações sobre os pacientes, um método de agrupamento multiobjetivo é aplicado considerando o novo conjunto de dados. O objetivo é aproveitar as informações dos diagnósticos existentes para encontrar informações sobre os dados não rotulados. Após aplicar o agrupamento, em uma das soluções não-dominadas, obtemos dois subconjuntos identificados pelas Tabelas 6 e 7.

Paciente	S_1	S_2	...	Resultado
4	0	0	...	**0**
5	0	0	...	**-1**
6	1	0	...	**-1**
...	...	...	...	...
99	0	1	...	**0**

Tabela 6 – Subconjunto 1.

Paciente	S_1	S_2	...	Resultado
1	1	1	...	**-1**
2	1	0	...	**0**
3	1	1	...	**1**
...	...	...	...	...
100	1	1	...	**1**

Tabela 7 – Subconjunto 2 .

A partir das diferentes medidas de similaridades podemos extrair informações complementares a respeito do diagnóstico, por exemplo, informações sobre probabilidade de atribuição de um determinado rótulo.

Na área de saúde, obter informações probabilísticas podem tornar a análise dos casos clínicos mais robusta e auxiliar os profissionais a tomarem melhores decisões acerca do acompanhamento e tratamentos dos pacientes. Dessa forma, neste projeto buscamos explorar as informações de agrupamento carregadas por cada solução não-dominada em conjunto com uma pequena quantidade de dados rotulados para gerar as informações complementares.

4.2 Teórico

O algoritmo de agrupamento multiobjetivo utilizado nessa pesquisa se baseia no trabalho de Handl e Knowles (2007), o MOCK. Neste projeto, propomos pequenas modificações na inicialização, nos operadores genéticos e utilizamos o NSGA-II, proposto posteriormente no Δ-Mock (GARZA-FABRE; HANDL; KNOWLES, 2017).

Os algoritmos encontrados na literatura propõem uma etapa de seleção de uma solução final após encontrar a fronteira não-dominada. Nesta dissertação, propomos após a obtenção da fronteira não-dominada, a criação de uma análise a *posteriori*, com o objetivo de compreender os resultados do agrupamento e explorar a extração de informações complementares. Assim como no trabalho de Maulik, Mukhopadhyay e Bandyopadhyay (2009), consideramos que explorar todas as soluções do multiobjetivo pode ser interessante, pois cada solução fornece uma informação de agrupamento. Dessa forma, assumimos que amostras agrupadas juntas com frequência nas soluções, possuem uma semelhança e podem ser definidas como sendo do mesmo *cluster*.

Na análise a *posteriori* dividimos a fronteira não-dominada em três partes, a ideia principal é identificar soluções com características diferentes e a que possui a melhor distribuição para determinado conjunto de dados. Duas das soluções possuem características predominantes em cada uma das funções e a outra pode apresentar características mistas. Além disso, para a extração de informações complementares introduzimos o conhecimento das amostras rotuladas para direcionar a rotulação das amostras desconhecidas. Assim,

amostras que são agrupadas frequentemente com as conhecidas podem ser consideradas do mesmo *cluster* e obter o mesmo rótulo. Para essa aplicação, calculamos a probabilidade de determinada amostra pertencer ao mesmo grupo que uma amostra conhecida.

Além da divisão da fronteira em porções e a inclusão do conhecimento sobre amostras rotuladas, nosso trabalho se difere do proposto por Maulik, Mukhopadhyay e Bandyopadhyay (2009) pelo o uso de outra ferramenta visual. Enquanto os pesquisadores propõem o uso dos *Eisen plot* para o problema de análise de expressão genética, nesse trabalho propomos o uso dos *heatmaps* para analisar as características dos agrupamentos e da exploração do problema. Os *heatmaps* são utilizados para visualizar a frequência em que as amostras são agrupadas juntamente em um mesmo *cluster*. Essa informação, vinda das soluções não-dominadas, gera um padrão que representa a característica de determinada parte da fronteira. Através dele, podemos identificar os possíveis *clusters* formados e extrair informações complementares do agrupamento como, amostras que podem pertencer a mais de uma classe, sendo útil para aplicação na área de saúde que será explicada mais adiante. O comportamento de amostras que pertencem a mais de uma classe pode ser observada através da frequência em que são agrupadas com outras de diferentes rótulos. A ferramenta proposta permite acessar de forma visual as características das soluções de agrupamento e informações de sobreposições mesmo em uma codificação baseada em grafos, evitando o uso de uma codificação baseada em protótipo ou o uso de métricas *fuzzy* no algoritmo.

A seguir descreveremos potenciais aplicações da nossa metodologia em problemas da área da saúde e do aprendizado semissupervisionado.

4.3 Aplicações

4.3.1 Saúde

A área de saúde é um dos campos potenciais para a aplicação da metodologia proposta nesse trabalho.

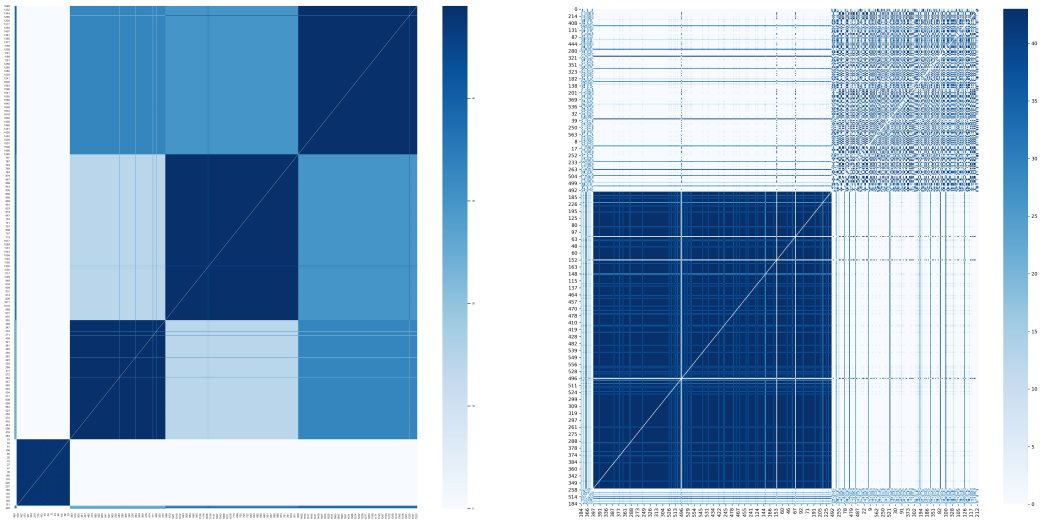
Com os algoritmos de agrupamento de dados, as informações dos pacientes podem ser exploradas de forma a separar os dados com características semelhantes. A partir das informações obtidas, os resultados podem ser analisados para auxiliar profissionais no diagnóstico, tratamento e acompanhamento dos pacientes. Já a rotulação de dados se torna adequada para essa área por possuir grande quantidade de dados não rotulados e poucos rotulados. Em uma aplicação real, diagnósticos realizados no passado servem como referência para novos pacientes com diagnósticos desconhecidos. Dessa forma, a presença de informações conhecidas na etapa de rotulação de dados pode auxiliar a geração de soluções de agrupamento com melhor qualidade. Além disso, rotular dados pode ser custoso por demandar tempo de um especialista, tornando a criação de modelos de rotulação cada vez

mais necessária.

Outra característica desse campo que tornam as aplicações dos métodos adequados é a necessidade de se obter informações probabilísticas ou que indiquem possíveis sobreposições de classes. Por exemplo, saber que um paciente tem 70% de chance de ter a doença x e 30% a doença y é preferível pelos profissionais de saúde do que um método que tenha como resposta um diagnóstico fixo; isso porque é importante a avaliação de um especialista nesta área. Outro exemplo foi o enfrentado pela pandemia com os casos de COVID-19, gripe e pneumonia, as doenças foram frequentemente confundidas devido seus sintomas similares. Ao utilizar métodos como o proposto nesse trabalho, os pacientes com sintomas que se enquadram em mais de uma doença poderiam ser identificados pelas sobreposições de classes, possibilitando que o profissional acompanhe melhor a evolução dos sintomas do paciente e tome uma decisão mais assertiva quanto ao diagnóstico e tratamento.

O *heatmap*, proposto nesse trabalho, é a ferramenta que auxiliará na obtenção das informações complementares explicadas no parágrafo anterior. Além de visuais são ferramentas de fácil compreensão e que complementam os resultados de agrupamento de dados. A intensidade das cores, representam a frequência em que amostras foram agrupadas em um mesmo *cluster*, e as linhas e colunas são as amostras do conjunto de dados. Na Figura 19a, apresentamos um exemplo de *heatmap* gerado para uma base sintética. No exemplo, podemos visualizar quatro principais grupos com alto compartilhamento de *clusters*, representado pela cor azul escuro. Dentre os quatro, três possuem diferentes graus de compartilhamento com outros *clusters*, representado pelas tonalidades de azuis mais claros. Essas diferenças de cores nos trazem a percepção dos graus de similaridades com outros *cluster* e a partir delas, podemos extrair as informações de probabilidade.

Na Figura 19b, apresentamos um segundo exemplo de *heatmap*, gerado para o problema de câncer de mama. Nela, podemos visualizar duas separações principais, que estão associados ao diagnóstico final do tumor, maligno ou benigno. Como as colunas e linhas dos *heatmaps* são ordenadas pelo rótulos finais de cada amostra, obtidas na etapa de atribuição final, podemos observar algumas características das amostras diferentes do *cluster* que ela pertence. Esse comportamento pode ser identificado pelas linhas brancas nos *clusters* formados e estão associadas a amostras com nenhum compartilhamento, podendo representar possíveis *outliers*.



(a) Exemplo de *heatmap* gerado para o problema 2d-4c-no6.

(b) Exemplo de *heatmap* gerado para o problema *Breast Cancer*.

Figura 19 – Exemplo de *heatmaps*.

4.3.2 Semissupervisionado

Outra potencial aplicação seria no campo de aprendizagem semissupervisionada. Poderíamos contribuir para essa área pelo desenvolvimento de um algoritmo de rotulação de dados multiobjetivo, auxiliando no enfrentamento do desafio da falta de dados rotulados. Muitos dos trabalhos dessa área utilizam métodos não supervisionados juntamente com supervisionados e a informação de amostras conhecidas pode ser inserida tanto no agrupamento de dados ou como amostras de treinamento em um método classificação, como os trabalhos apresentados na seção 3.2.

A ideia principal seria aperfeiçoar a etapa final de rotulação de dados do método proposto nesse trabalho, combinando a rotulação baseada na Fronteira de Pareto com um método de classificação em uma abordagem de autotreinamento. De forma semelhante com os trabalhos de Arora, Tushir e Kashyap (2020) e Gan et al. (2013), utilizaríamos a saída de um algoritmo de agrupamento para rotular previamente as amostras desconhecidas, selecionando aquelas com maior confiança para compor os dados de treinamento da classificação. Por sua vez, o método de classificação seguiria uma abordagem de autotreinamento, onde o modelo seria treinado repetidas vezes a partir dos novos dados obtidos.

Em Handl e Knowles (2006), observou-se que o uso da abordagem semissupervisionada no agrupamento de dados multiobjetivo pode ser vantajosa em comparação com métodos não supervisionados, supervisionados e semissupervisionados. Diferente desse trabalho, que utiliza a informação semissupervisionada nas funções-objetivo, estamos interessado em investigar a capacidade de rotulação de dados a partir das informações geradas pelas solução não-dominadas e com a ajuda do conhecimento prévio das amostras. Além

disso, investigaremos também a capacidade de aprimorar a classificação final dos dados, visto que aumentar os rótulos de treinamento poderia contribuir para uma classificação com melhores resultados.

Nesse trabalho, priorizamos inicialmente o uso das amostras rotuladas na etapa a *posteriori*. Porém, como perspectivas futuras desejamos incluir a etapa de classificação posterior a rotulação de dados para verificar se a abordagem melhora a classificação final dos dados.

Além disso, diversas aplicações enfrentam a falta de dados rotulados e a criação de métodos de rotulação se torna cada vez mais necessária. Dessa forma, destaca-se que o método proposto possui potencial aplicação para diferentes problemas reais.

5 Metodologia

Neste capítulo será apresentada a metodologia proposta da pesquisa. O método está dividido em três principais etapas (Figura 20): pré-processamento, algoritmo de agrupamento multiobjetivo e análise pós-otimização. Cada etapa será discutida com mais detalhes nas próximas seções.

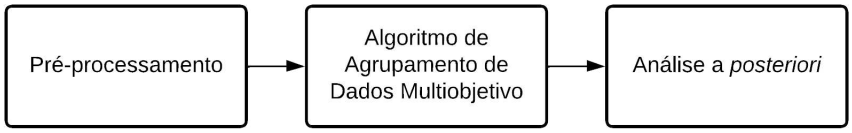


Figura 20 – Fluxo geral do método proposto.

5.1 Pré-processamento

5.1.1 Normalização

Inicialmente, na etapa de pré-processamento, o conjunto de dados é normalizado através do método Min-Max. Cada atributo x do conjunto de dados passará pela Equação 5.1 e será transformado em um valor entre 0 e 1, evitando que atributos com escalas maiores tenham mais influência no cálculo das distâncias utilizadas no algoritmo.

$$x_{normalizado} = \frac{x - min(x)}{max(x) - min(x)}. \tag{5.1}$$

5.2 Agrupamento de dados multiobjetivo

Como apresentado no capítulo anterior, nosso algoritmo de agrupamento de dados multiobjetivo se baseia no algoritmo MOCK, com pequenas diferenças na inicialização, nos operadores genéticos e no algoritmo de otimização. A inicialização da população é realizada através de três algoritmos, *K-means*, *K-medoids* e MST. Para os operadores genéticos, utilizamos o cruzamento uniforme e dois tipos de mutação, centroide e vizinhos. Além disso, utilizamos a estrutura do NSGA-II, incluída posteriormente no Δ-MOCK, para classificar e selecionar os indivíduos da população na etapa de torneio binário e seleção da próxima geração. O critério de parada do algoritmo é definido empiricamente em 100 gerações. No Algoritmo 2, apresenta-se a estrutura geral do algoritmo proposto e cada etapa é descrita com mais detalhes nas subseções 5.2.1 a 5.2.6.

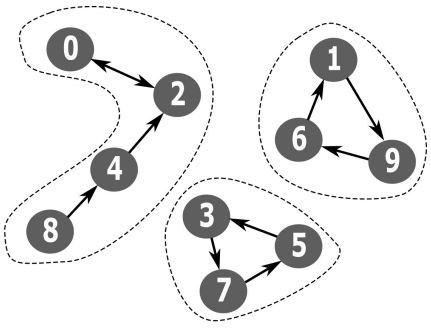
Algoritmo 2 – Algoritmo Genético

```
Inicializa população P_inicial com tamanho TP
enquanto o critério de parada não é satisfeito faça
    Selecione pais p1 e p2 pelo torneio binário
    Gere F_c filhos a partir de p1 e p2 (cruzamento uniforme)
    Gere F_m filhos a partir de p1 e p2 (mutação centroide ou vizinhos)
    Crie P_mista = P_inicial + F_c + F_m
    Avalie P_mista nas funções-objetivo
    Selecione os melhores TP indivíduos de P_mista
fim
Retorne as soluções não-dominadas
```

5.2.1 Representação da solução

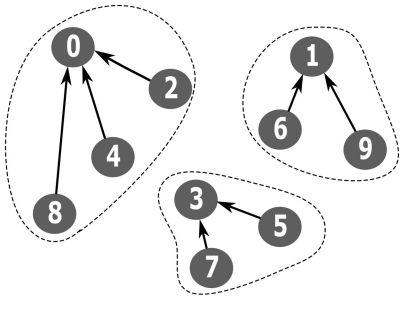
Para codificação da solução, cada indivíduo da população corresponde a uma possível forma de agrupar o conjunto de N dados. Para representar o indivíduo foi utilizada a codificação *locus-based adjacency* (HANDL; KNOWLES, 2007). Nessa codificação, um indivíduo é um vetor $[a_0, a_1, ..., a_N]$ de tamanho N, onde cada elemento se relaciona com uma das amostras do conjunto de dados e seus valores são suas conexões. No final, as conexões das amostras geram os agrupamentos. No exemplo (a) da Figura 21, a posição 0, está associada à amostra 0, que por sua vez, está conectada com a amostra 2, demonstrando que as amostras 0 e 2 estão contidas no mesmo *cluster*.

A decodificação do indivíduo é realizada utilizando a raiz do grupo. Por raiz, denominamos como a amostra de menor valor pertencente ao grupo. Ao identificar a raiz, todos os elementos terão o mesmo valor associado no indivíduo. No exemplo (b) da Figura 21, destacamos a raiz de cada *cluster*. Além disso, mostramos o indivíduo do exemplo (a) decodificado com as raízes de cada grupo (0, 1 e 3).



Indivíduo	2	9	0	7	2	3	1	5	4	6
Índices	0	1	2	3	4	5	6	7	8	9

(a) Codificação *locus-based adjacency*.



Indivíduo	0	1	0	3	0	3	1	3	0	1
Índices	0	1	2	3	4	5	6	7	8	9

(b) Indivíduo (a) decodificado pelas raízes (0, 1, 3).

Figura 21 – Exemplo da codificação e decodificação utilizada no algoritmo.

5.2.2 Funções-objetivo

Nesse trabalho, utilizamos três funções-objetivo: Compactação, Conectividade e *Within-cluster scatter*, sendo a última utilizada com duas funções de distâncias diferentes, Mahalanobis e Cosseno. Os operadores do NSGA-II (*rank* e *crowd*) selecionam os indivíduos com melhores valores de funções-objetivo e considerados mais diversos. Cada uma das funções evidencia uma forma de agrupar os dados. A distribuição de pontos pode variar de um conjunto para outro, podendo surgir *clusters* compactos, alongados ou de formatos arbitrários. As funções de Compactação e Mahalanobis possuem, respectivamente, características esféricas e elipsoidais, enquanto Conectividade e Cosseno, priorizam mais a conectividade dos dados. Com o intuito de explorar as relações conflitantes entre as funções e a aplicação da otimização multiobjetivo no agrupamento de dados, testamos diferentes combinações entre as funções-objetivo.

Para compactação e conectividade, o aumento do número de *clusters* reflete em maiores valores de conectividade e menores em compactação. Em contrapartida, a diminuição de *clusters* ocasiona maiores valores de compactação e menores em conectividade. Esse conflito pode ser visualizado na Figura 22. Os pontos a, b, c, d representam soluções não-dominadas ou eficientes (melhores indivíduos da população final). Cada um destes pontos tem associado: (i) f_1 e f_2, no espaço dos objetivos representando respectivamente, compactação dos grupos e conectividade dos grupos; (ii) um agrupamento para os dados, contendo o número total de *clusters* e a atribuição de pontos em cada *cluster* (que podem variar de indivíduo para indivíduo).

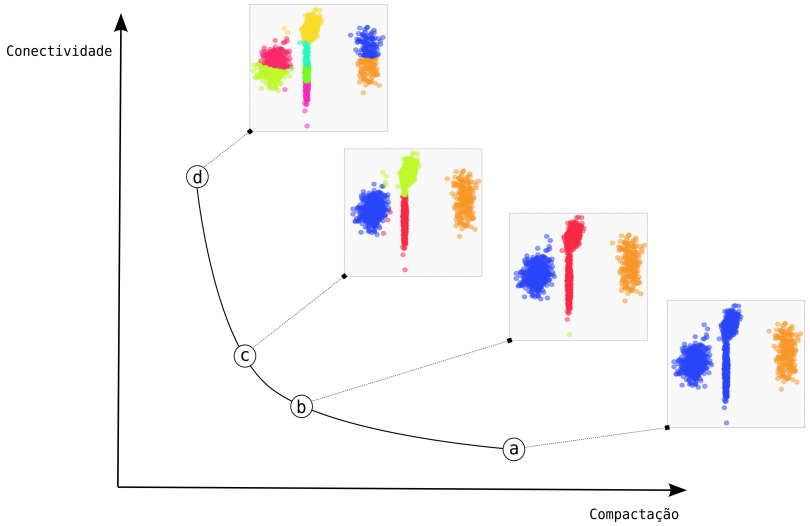


Figura 22 – Representação da fronteira com as soluções não-dominadas, destacando diferentes formas de agrupar os dados. Solução a (K=2), b (K=4), c (K=4) e d (K=8).

As definições das funções-objetivo de compactação e conectividade foram retiradas de Handl e Knowles (2007) e a *Within-cluster scatter* de José-García et al. (2021). Neste projeto, deseja-se minimizar todas as Equações 5.2, 5.3 e 5.4.

5.2.2.1 Compactação

$$\text{Compactação} = \sum_{C_k \in C} \sum_{i \in C_k} \delta(i, \mu_k). \tag{5.2}$$

$\delta(i, \mu_k)$ = distância euclidiana entre as amostras do *cluster* C_k e seu centróide μ_k.
C: conjunto de todos os *clusters* C_k do indivíduo.
μ_k: k-ésimo centroide do *cluster* C_k.
i: i-ésima amostra do *cluster* C_k .

5.2.2.2 Conectividade

$$\text{Conectividade} = \sum_{i=1}^{N} (\sum_{j=1}^{L} p_{i,nn_{i,j}}). \tag{5.3}$$

$nn_{i,j}$: j-ésimo vizinho mais próximo da amostra i.
$p_{i,nn_{i,j}}$: penalidade aplicada para cada par $(i, nn_{i,j})$.

$$p_{i,nn_{i,j}} = \begin{cases} \frac{1}{j} & \text{se } nn_{i,j} \text{ não pertencer ao mesmo } \textit{cluster} \text{ da amostra } i \\ 0 & \text{caso contrário} \end{cases}$$

5.2.2.3 *Within-Cluster Scatter* (WCS)

$$\text{WCS} = \sum_{C_k \in C} \sum_{\mathbf{a}, \mathbf{b} \in C_k} d_j(\mathbf{a}, \mathbf{b}). \tag{5.4}$$

$d_j(\mathbf{a}, \mathbf{b})$ = dissimilaridade entre as amostras $\mathbf{a}$ *e* $\mathbf{b}$.
C : conjunto de todos os *clusters* C_k do indivíduo.

A seguir são apresentadas as funções de distâncias utilizadas para o cálculo do WCS.

5.2.2.3.1 Mahalanobis

Sendo $\mathbf{p}$ e $\mathbf{q}$ vetores de objetos do conjunto de dados e IV a matriz de covariância das amostras, a distância de Mahalanobis é calculada como:

$$\text{Mahalanobis} = \sqrt{(\mathbf{p} - \mathbf{q}) IV^{-1} (\mathbf{p} - \mathbf{q})^T}. \tag{5.5}$$

5.2.2.3.2 Cosseno

Sendo $\mathbf{p}$ e $\mathbf{q}$ vetores de objetos do conjunto de dados e $\|\mathbf{p}\|$ e $\|\mathbf{q}\|$ a norma Euclidiana, a distância de cosseno é calculada como:

$$\text{Cosseno} = 1 - \frac{\mathbf{pq}}{\|\mathbf{p}\|\|\mathbf{q}\|}. \tag{5.6}$$

5.2.3 População Inicial

Para a geração da população inicial do algoritmo[1], escolhemos métodos que gerassem boas soluções iniciais para o problema de agrupamento de dados e que estivessem relacionadas às funções-objetivo escolhidas. Três métodos foram utilizados: algoritmo de *Kruskal*, *K-means* e *K-medoids*. No algoritmo original do MOCK, duas inicializações são utilizadas: MST, pelo algoritmo de Prim e o *K-means*. Nesse trabalho, pela facilidade de implementação escolhemos o algoritmos de *Kruskal* e incluímos o *K-medoids* para gerar soluções menos impactadas por *outliers*.

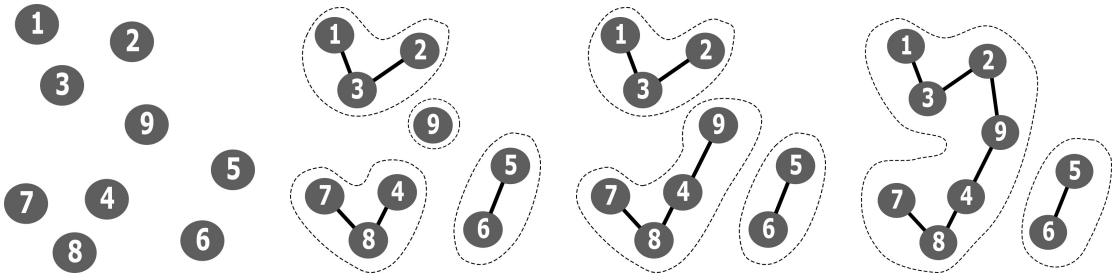
A população inicial de tamanho TP é composta pelos indivíduos gerados pelos três métodos. A quantidade de indivíduos de cada inicialização é definida pelo usuário e precisa estar associada ao intervalo de K_{min} e K_{max}.

5.2.3.1 Algoritmo de *Kruskal*

No Algoritmo de *Kruskal*, busca-se crescer uma floresta de modo que se torne conexa e com custo mínimo (FEOFILOFF, 2019). Utilizando esse algoritmo na inicialização, introduzimos na população inicial indivíduos com menores distâncias de conexão entre as amostras, destacando a conectividade dos dados.

Para inicializar os indivíduos através deste algoritmo, inicialmente utilizamos uma matriz de distância entre as amostras para encontrar os pares de arestas de menor valor. A distância utilizada foi a distância Euclidiana. Iterativamente, incluímos as arestas com menores distâncias, uma a uma, até que a quantidade de *clusters* no indivíduo alcance o intervalo de $[K_{min}, K_{max}]$. Dessa forma, quando a quantidade de *clusters* presentes no indivíduo alcançar o intervalo definido pelo usuário, a população inicial começará a ser gerada. Na Figura 23, um exemplo da inicialização é apresentada.

[1] Nos testes iniciais, também foi implementada a inicialização aleatória da população, mas a convergência do algoritmo foi muito lenta e a qualidade das soluções geradas foi bastante inferior. Por isso, esta inicialização não é apresentada aqui.



(a) Conjunto de dados. (b) Indivíduo com K = 4. (c) Indivíduo com K = 3. (d) Indivíduo com K = 2.

Figura 23 – Exemplo de inicialização de indivíduos pelo Algoritmo de *Kruskal*, com $[K_{min}, K_{max}] = [2, 4]$.

5.2.4 Seleção

Para a etapa de seleção da população, utilizaremos a estratégia de não dominância do NSGA-II (capítulo 2 seção 2.2.1.1). Dessa forma, cada solução será comparada com base no *rank* da sua fronteira e da distância de aglomeração, os melhores serão escolhidos para compor a população de pais ou da próxima geração.

5.2.4.1 Torneio Binário

No torneio binário, dois indivíduos da população são escolhidos aleatoriamente e o melhor é selecionado para compor a população de pais, de tamanho TP. Para comparar as soluções foi utilizado a ordenação de soluções não-dominadas e a distância de aglomeração do NSGA-II. Os indivíduos escolhidos serão utilizados nos operadores de cruzamento e mutação.

5.2.4.2 Próxima Geração

Para compor a população da próxima geração, utiliza-se a ordenação não-dominada e a distância de aglomeração para selecionar TP indivíduos da população mista (pais + filhos gerados no *crossover* + filhos gerados na mutação).

5.2.5 Cruzamento

Após a seleção da população de pais pelo torneio binário, F_c pais são escolhidos aleatoriamente para gerar filhos pelo *crossover* uniforme. Nesse operador, dois pais p_1 e p_2 geram uma solução filho. O funcionamento do *crossover* pode ser descrito a seguir:

1. Escolher aleatoriamente dois pais p_1 e p_2, sendo $p_1 \neq p_2$.
2. Gerar máscara $M = [i, ..., n]$ com $i \in \{0, 1\}$. O filho terá 50% de chance de copiar o material genético de p_1 e 50% de p_2.

3. Gerar filho e decodificá-lo.

   a) Se após a decodificação, $K = 1$ (indivíduo infactível), gerar novamente a máscara até que $K > 1$ ou máximo de tentativas = 3. No caso do filho permanecer com $K = 1$ após as tentativas, gerar indivíduo aleatoriamente.

   b) Se $K > 1$, incluir na população de filhos e repetir passo 1.

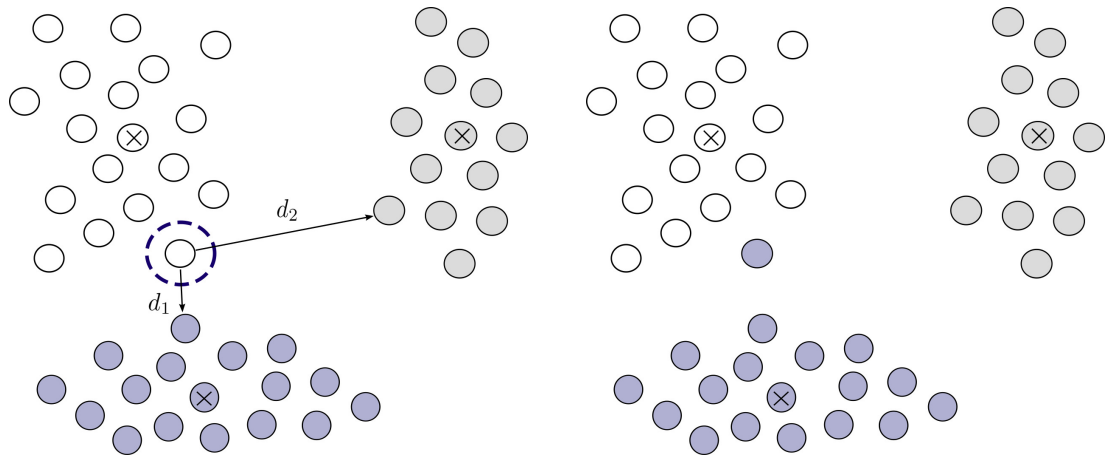
5.2.6 Mutação

Assim como no operador anterior, F_m pais são escolhidos aleatoriamente para gerar filhos por uma das mutações. As mutações escolhidas foram Mutação Centroide e Mutação Vizinhos e cada uma delas tende a privilegiar as características das funções-objetivo.

5.2.6.1 Mutação Centroide

A mutação centroide tem como objetivo atribuir a amostra selecionada para o *cluster* com o centroide mais próximo. Um exemplo pode ser observado na Figura 24 e o operador funciona da seguinte forma:

1. Escolher aleatoriamente uma amostra i do indivíduo.
2. Calcular a distância euclidiana da amostra i com todos os outros centroides do indivíduo.
3. Alocar a amostra para um novo *cluster*, considerando o centroide mais próximo.



(a) Seleção da amostra a ser realocada, destacada em azul. d_1 e d_2 representam as distâncias entre a amostra e os outros centroides do indivíduo.

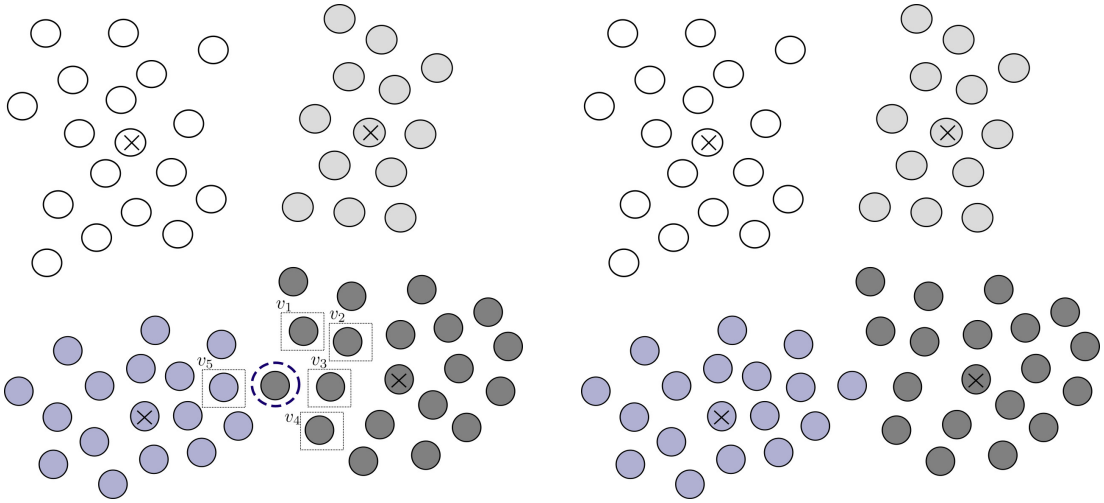
(b) A amostra selecionada será realocada para o *cluster* com o centroide mais próximo, considerando que $d_1 < d_2$.

Figura 24 – Exemplo de mutação centroide.

5.2.6.2 Mutação Vizinhos

A mutação vizinhos tem como objetivo atribuir a amostra selecionada para o *cluster* do vizinho mais próximo. Um exemplo pode ser observado na Figura 25 e o operador funciona da seguinte forma:

1. Escolher aleatoriamente uma amostra i do indivíduo.
2. Identificar os L vizinhos mais próximos da amostra i através do algoritmo *k-nearest neighbors*.
3. Alocar a amostra para um novo *cluster*, considerando o vizinho mais próximo.



(a) Seleção da amostra a ser realocada, destacada em azul. v_1, v_2, v_3, v_4 e v_5 representam os vizinhos mais próximos da amostra.

(b) A amostra selecionada será realocada para o *cluster* do vizinho v_5, pois é o vizinho mais próximo pertencente a outro *cluster*.

Figura 25 – Exemplo de mutação vizinhos.

5.3 Análise a *posteriori*

Após a execução do Algoritmo Genético (Algoritmo 2), a análise a *posteriori* tem como objetivo analisar as soluções não-dominadas de forma a obter informações sobre o agrupamento de dados e verificar como elas podem auxiliar na compreensão do problema.

5.3.1 Divisão da Fronteira não-dominada

Inicialmente, dividimos a fronteira não-dominada como ilustrado na Figura 26. A partir das soluções extremas da fronteira (P_1 e P_2), encontramos a solução utópica ($P_{\text{utópica}}$) correspondente e determinamos a equação da reta que passa pelos dois pontos

extremos. Conhecendo a reta, conseguimos dividir a fronteira em três partes, traçando novas retas que interceptam a solução utópica e os pontos P_3 e P_4.

O objetivo desta divisão é identificar qual porção da fronteira contém a melhor distribuição dos dados em *clusters*. Por exemplo, em compactação e conectividade, *clusters* com simetria esférica apresentarão melhores agrupamentos quando associados a soluções da Parte 1 (que minimiza g_1, compactação). Por outro lado, *clusters* alongados apresentarão melhores agrupamentos quando associados a soluções da Parte 3 (que minimiza g_2, conectividade). A Parte 2 pode apresentar bons agrupamentos para bases mistas ou mais variadas, pois pondera compactação e conectividade.

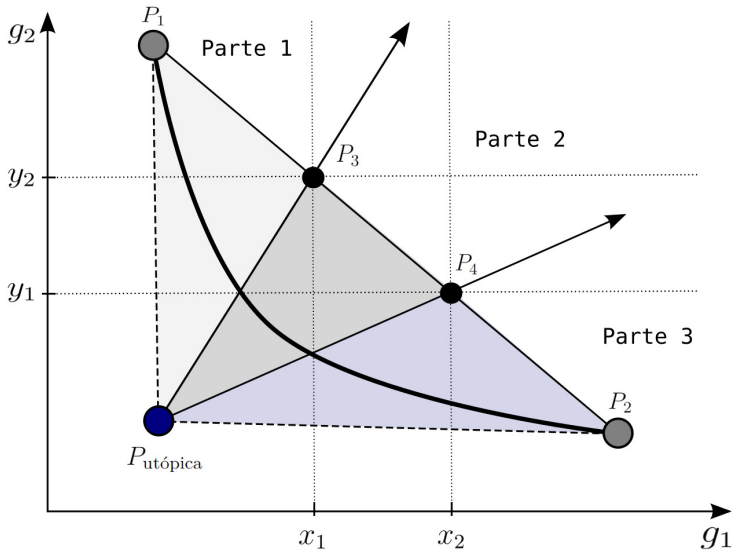


Figura 26 – Divisão da fronteira não-dominada em três partes. As soluções em cinza representam os extremos da fronteira, $P_1 = (min(g_1), max(g_2))$ e $P_2 = (max(g_1), min(g_2))$. A solução em azul representa a solução utópica, $P_{\text{utópica}} = (min(g_1), min(g_2))$.

5.3.2 Matriz de compartilhamento de grupos

Após a divisão, em cada parte da fronteira não-dominada, gera-se uma matriz M de compartilhamento de grupos, com dimensão nxn, sendo n o número total de amostras do conjunto de dados. Os elementos $c_{i,j}$ da matriz representam a quantidade de vezes que determinada amostra i foi agrupada juntamente com a amostra j. Considerando que em cada parte da fronteira existem diferentes números de soluções, o número máximo de grupos compartilhados será igual ao número de soluções não-dominadas presentes em cada

parte (S_p).

$$M_{n,n} = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{pmatrix}.$$

5.3.3 *Heatmaps* de compartilhamento de grupos

Para a criação dos *heatmaps* de compartilhamento de grupos, ordenamos a matriz de compartilhamento de grupos (Subseção 5.3.2) pelas classes finais encontradas em cada solução de agrupamento.

A ideia de visualizar os resultados pelos *heatmaps* de compartilhamento de grupos é avaliar se o algoritmo está se aproximando dos agrupamentos reais e identificar possíveis padrões nos dados.

5.3.4 Extração das informações complementares

Com o intuito de analisar a extração de informações a partir das soluções não-dominadas, consideramos 20% das amostras do conjunto de dados como rotuladas e o restante como não rotuladas. As amostras com rótulos serão utilizadas na etapa de análise a *posteriori* para verificar se esses dados em conjunto com as informações geradas pela fronteira não-dominada será capaz de obter informações de probabilidade e gerar possíveis rótulos para amostras não rotuladas.

A partir da matriz de compartilhamento de grupos, criamos uma matriz MP com dimensão nxq, sendo n o número total de amostras do conjunto de dados e q as amostras com rótulos conhecidos. Para encontrar a probabilidade de uma amostra n pertencer ao mesmo grupo de uma amostra q, dividimos cada elemento da matriz MP pela soma total de cada linha.

$$MP_{n,q} = \begin{pmatrix} \frac{c_{1,1}}{\sum c_{1,q}} & \frac{c_{1,2}}{\sum c_{1,q}} & \cdots & \frac{c_{1,1}}{\sum c_{1,q}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{c_{n,1}}{\sum c_{n,q}} & \frac{c_{n,2}}{\sum c_{n,q}} & \cdots & \frac{c_{n,q}}{\sum c_{n,q}} \end{pmatrix}.$$

Para determinar a probabilidade final, para cada amostra não rotulada, soma-se as probabilidades de amostras de mesmo rótulo da matriz MP, obtendo assim as probabilidades acumuladas, P_{ac}. A Tabela 8 apresenta um exemplo das probabilidade encontradas na matriz MP. Considere as amostras 20, 30 e 52 como sem rótulos e as amostras 1, 2, 5, 10, 11, 15, como rotuladas. P_1 representa a probabilidade de uma amostra

desconhecida pertencer ao mesmo grupo da amostra 1 e, consequentemente, obter o mesmo rótulo.

Amostra \ P	Rótulo 0		Rótulo 1		Rótulo 2	
	P_1	P_2	P_5	P_{10}	P_{11}	P_{15}
20	0.35	0.35	0	0	0.1	0.2
30	0	0.4	0.6	0	0	0
52	0	0	0	0	0	1

Tabela 8 – Probabilidade obtida na matriz MP.

A partir das informações encontrada anteriormente, calcula-se as probabilidades acumuladas por rótulos, apresentada na Tabela 9. Nesse exemplo, a amostra 20 possui 70% de chance de ter o rótulo 0 e 30% de ter o rótulo 2.

Amostra \ P_{ac}	Probabilidade acumulada		
	Rótulo 0	Rótulo 1	Rótulo 2
20	0.7	0	0.3
30	0.4	0.6	0
52	0	0	1

Tabela 9 – Probabilidade acumulada por rótulo.

Para cada parte da fronteira, encontraremos as informações de probabilidade acumulada por rótulo, associadas a soluções não-dominadas com diferentes distribuições. Para validação do resultado final, consideramos a maior probabilidade obtida entre as três partes e comparamos com os rótulos verdadeiros de cada amostra.

5.3.5 Métricas

Para a primeira métrica, acurácia geral (Equação 5.7), medimos o desempenho do modelo quanto a atribuição correta das classes. Nos experimentos realizados, todos os conjuntos de dados possuíam rótulos associados, dessa forma, a métrica é utilizada para validar principalmente a etapa da rotulação de dados da análise a *posteriori*. A etapa de rotulação de dados pode ser vista no contexto do aprendizado supervisionado, como uma classificação final das amostras. Para o cálculo, consideramos quantas amostras com as classes atribuídas pelo algoritmo foram classificadas corretamente em relação ao total de amostra.

$$Acc = \frac{amostras\ classificadas\ corretamente}{total\ de\ amostras}. \tag{5.7}$$

Por ser uma métrica para métodos supervisionados,a acurácia não é ideal para validar a qualidade do agrupamento de dados. No contexto da aplicação de métodos não

supervisionados, optamos em utilizar o *Adjusted Rand Index* (ARI) (HUBERT; ARABIE, 1985) como a segunda métrica. Diferente da acurácia, que valida os resultados com os rótulos conhecidos, o ARI analisa a qualidade dos grupos formados, sem olhar necessariamente para os rótulos.

Assim como no trabalho de MOCK, o ARI foi escolhido por considerar uma normalização em que partições aleatórias assumem valores perto de 0, sendo importante para avaliar as soluções produzidas pelo multiobjetivo com diferentes números de grupos (HANDL; KNOWLES, 2007). A métrica pode assumir valores negativos, sendo o valor ideal igual a 1.

Sejam $C = \{c_1, c_2, ..., c_i\}$, o agrupamento alvo conhecido de um conjunto de dados; $G = \{g_1, g_2, ..., g_j\}$, um agrupamento produzido pelo multiobjetivo; P_{ij} o número de amostras da classe c que foram agrupadas no grupo g; e n, a quantidade de dados do conjunto de dados, constrói-se a tabela de contingência abaixo:

Classe \ Grupo	g_1	g_2	...	g_j	Soma
c_1	p_{11}	p_{12}	...	p_{1j}	$p_{1.}$
c_2	p_{21}	p_{22}	...	p_{2j}	$p_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
c_i	p_{i1}	p_{i2}	...	p_{ij}	$p_{i.}$
Soma	$p_{.1}$	$p_{.2}$	...	$p_{.j}$	n

Tabela 10 – Tabela de contingência de uma solução C e G.

Para o cálculo do ARI, são utilizadas comparações de pares entre uma solução alvo e uma produzida pelo algoritmo de agrupamento. As comparações consideradas são: a = pares de mesma classe em C e mesmo grupo em G; b = pares de mesma classe em C e grupo distintos em G; c = pares de classes distintas em C e mesmo grupo em G; d = pares de classes distintas em C e grupos distintos em G. Os valores de a e d refletem os acertos de uma solução produzida pelo algoritmo em comparação a solução alvo e os valores de b e c, os erros.

$$a = \sum_{i,j} \binom{p_{ij}}{2} \tag{5.8}$$

$$b = \sum_{i} \binom{p_{i.}}{2} - \sum_{i,j} \binom{p_{ij}}{2} \tag{5.9}$$

$$c = \sum_{j} \binom{p_{.j}}{2} - \sum_{i,j} \binom{p_{ij}}{2} \tag{5.10}$$

$$a + b + c + d = \binom{n}{2} \tag{5.11}$$

A equação do ARI, pode ser definida como:

$$ARI = \frac{\sum_{i,j}\binom{p_{ij}}{2} - [\sum_i \binom{p_{i.}}{2}\sum_j \binom{p_{.j}}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{p_{i.}}{2} + \sum_j \binom{p_{.j}}{2}] - [\sum_i \binom{p_{i.}}{2}\sum_j \binom{p_{.j}}{2}]/\binom{n}{2}} \tag{5.12}$$

A fim de exemplificar o cálculo das métricas, considere uma solução alvo $C = \{1, 1, 1, 2, 2, 2\}$, com duas classes: classe 1 representada pelo círculos e classe 2 pelos triângulos. Considere a solução produzida pelo multiobjetivo como sendo $G = \{1, 1, 2, 2, 2, 2\}$, com dois grupos formados: grupo 1 representada pela cor branca e grupo 2 pela cor cinza. A Figura 27 apresenta as duas soluções C e G em uma mesma representação e a Tabela 11, a comparação de pares entre as soluções.

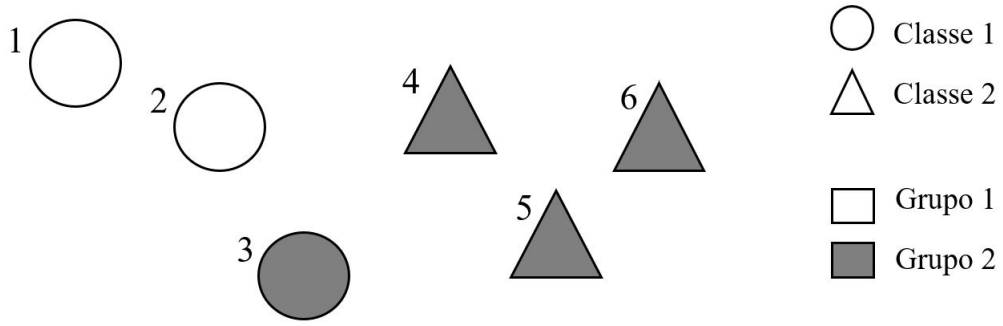


Figura 27 – Representação da solução alvo e da solução multiobjetivo.

Classe \ Grupo	g_1	g_2	Soma
c_1	2	1	3
c_2	0	3	3
Soma	2	4	$n = 6$

Tabela 11 – Tabela de contingência da Figura 27.

Calculando as métricas do exemplo, obtemos $ARI = 0.324$ e $Acc = 0.833$.

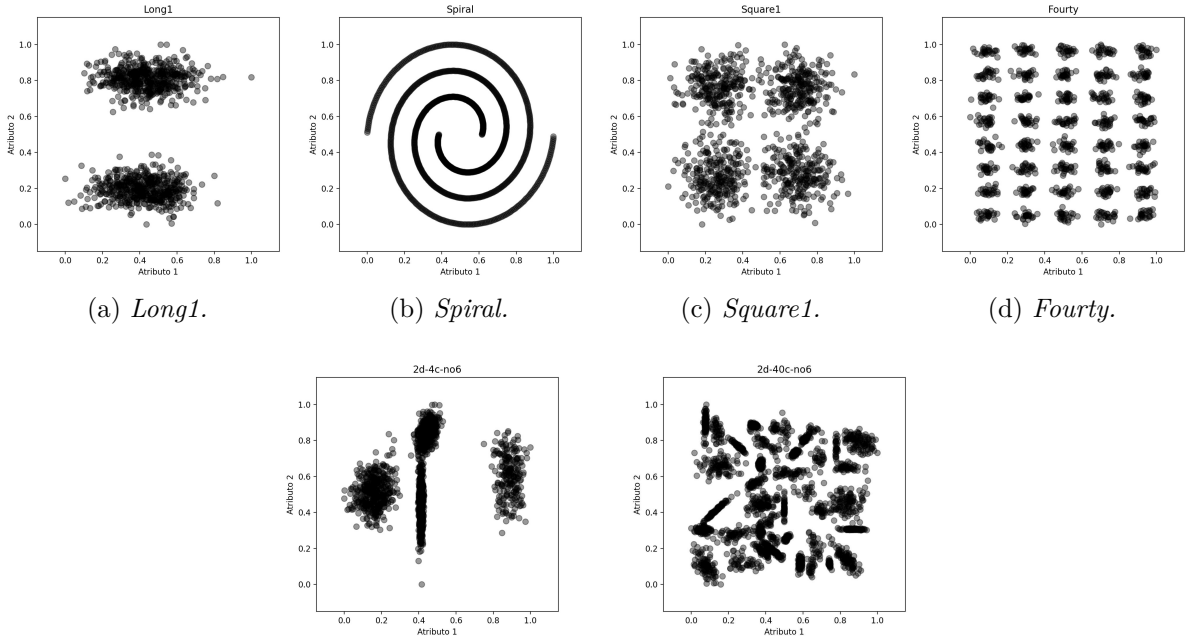
6 Experimentos e Resultados

6.1 Conjuntos de dados

Os experimentos com a metodologia proposta foram realizados com dois conjuntos de dados: dados sintéticos (Seção 6.1.1) e dados reais (Seção 6.1.2). Os conjuntos são apresentados nas seções a seguir.

6.1.1 Dados sintéticos

Selecionamos alguns conjuntos de dados sintéticos do trabalho de Handl e Knowles (2007) para serem utilizados nos experimentos desse trabalho. Na Figura 28 é possível observar as características dos conjuntos de dados. *Long1* e *Spiral* representam conjuntos com *clusters* alongados, que preservam a conectividade dos dados. O *Square1* representa *clusters* com algumas sobreposições de dados e o conjunto *Fourty* possui dados compactos e bem separados. Por fim, 2d-4c-no6 e 2d-40c-no6 são dados gerados aleatoriamente por uma distribuição normal multivariada. Na Tabela 12 são apresentadas as informações conhecidas dos conjuntos de dados. Dentre eles, apenas 2d-4c-no6 e 2d-40c-no6 são conjuntos desbalanceados, ou seja, cada classe possui diferentes quantidades de amostras.



(a) *Long1.* (b) *Spiral.* (c) *Square1.* (d) *Fourty.*

(e) 2d-4c-no6. (f) 2d-40c-no6.

Figura 28 – Conjuntos de dados sintéticos.

Conjunto de dados	Parâmetros		
	Nº de *clusters*	Amostras	Atributos
Long1	2	1000	2
Spiral	2	1000	2
Square1	4	1000	2
2d-4c-no6	4	1670	2
Fourty	40	1000	2
2d-40c-no6	40	2355	2

Tabela 12 – Informações conhecidas dos conjuntos sintéticos.

6.1.2 Dados reais

Para os experimentos com dados reais, utilizamos os conjuntos *Iris* e *Breast Cancer* da *UCI Machine Learning Repository*. Na Tabela 13 são apresentadas as informações conhecidas sobre os dados.

Conjunto de dados	Parâmetros			
	Nº de *clusters*	Amostras	Nº de amostras por classe	Atributos

Iris	3	150	classe 0 = 50;
			classe 1 = 50;
			classe 2 = 50.
				4

Breast Cancer	2	569	classe 0 = 357; classe 1 = 212.	30

Tabela 13 – Informações conhecidas dos conjuntos reais.

6.2 Análise sob a perspectiva da Otimização Multiobjetivo

Para analisar os algoritmos sob a ótica da otimização multiobjetivo, definimos inicialmente os parâmetros que serão utilizados nas bases sintéticas e reais. Os parâmetros foram estabelecidos experimentalmente e são apresentados na Tabela 14.

Para a definição do intervalo de *clusters* $[K_{min}, K_{max}]$, deve-se considerar um conhecimento prévio do conjunto de dados e da quantidade de *clusters* esperados pelo tomador de decisão. A atribuição equivocada deste intervalo pode afetar a qualidade das soluções geradas pelo modelo, pois a diminuição ou aumento dos *clusters* pode não gerar boas soluções de agrupamento dependendo do conjunto de dados. Além disso, a definição do intervalo influencia na quantidade de soluções geradas por cada uma das inicializações, podendo haver repetição de indivíduos. Nos testes, considerando que o intervalo definido foi de $[2, 25]$, para os conjuntos com até 4 *clusters*, e de $[2, 50]$, com até 40 *clusters*, a população inicial de filhos criados pelo *K-means* ($Q_{Kmeans} = 38$) e *K-medoids* ($Q_{Kmedoids} = 38$) gera indivíduos repetidos.

Parâmetro		Valor
$[K_{min}, K_{max}]$	Intervalo de *clusters*	$[2, 25]$ ou $[2, 50]$
TP	Tamanho da população	100
G	Gerações	100
L	Vizinhos mais próximos	10 ou 20
Q_{MST}	Soluções *Kruskal*	24
Q_{Kmeans}	Soluções *K-means*	38
$Q_{Kmedoids}$	Soluções *K-medoids*	38
F_c	Filhos gerados por *crossover*	20
F_m	Filhos gerados pela mutação	20

Tabela 14 – Parâmetros utilizados nas bases sintéticas e reais.

Nos experimentos a seguir analisaremos a fronteira não dominada obtida a partir do uso de diferentes combinações de funções-objetivo.

6.2.1 Análise variações funções-objetivo

Para verificar o uso das diferentes combinações de função-objetivo, analisamos as fronteiras finais obtidas nas 5 rodadas do algoritmo quanto: a exploração do espaço de busca, o conflito presente entre as combinações e a capacidade de gerar boas soluções de agrupamento em termos de acurácia e da métrica *Adjusted Rand Index* (ARI).

Neste conjunto de experimentos, buscamos entender a relação de compromisso entre as funções-objetivo utilizadas e verificamos se as combinações de funções de distâncias conseguem explorar diferentes características dos conjuntos de dados.

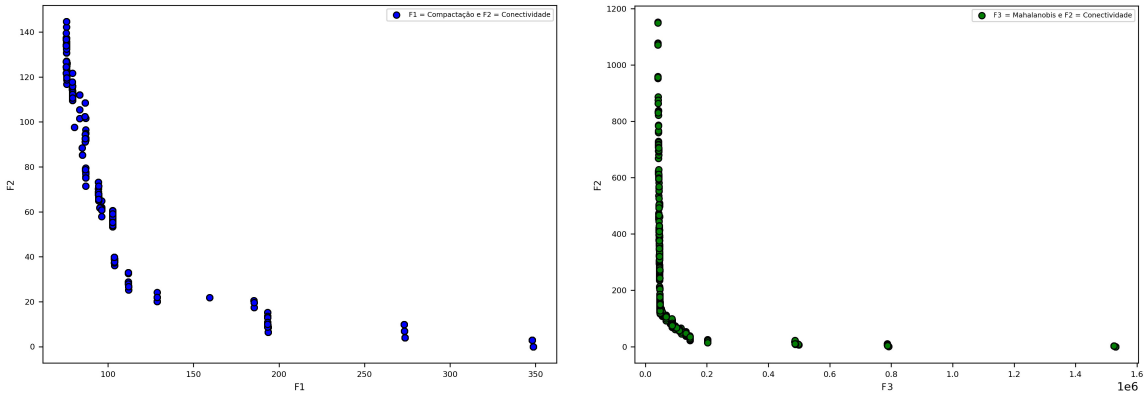
Para os experimentos, quatro combinações foram testadas: f_1 = Compactação e f_2 = Conectividade, f_1 = Compactação e f_4 = Cosseno , f_3 = Mahalanobis e f_2 = Conectividade, f_3 = Mahalanobis e f_4 = Cosseno. As funções Compactação e Mahalanobis não foram utilizadas juntas por possuírem pouca diferença de característica a ser capturada, dado que uma enfatiza *clusters* esféricos e outra, *clusters* elipsoidais. Todos os conjuntos de dados apresentados anteriormente na Subseção 6.1.1 e 6.1.2 foram utilizados neste experimento.

Para verificar o conflito das funções-objetivo, analisamos as correlações geradas entre todas as combinações de funções. Para todos os conjuntos de dados, com exceção da *Fourty*, foi possível observar que as combinações de Compactação/Conectividade e Mahalanobis/Conectividade apresentaram uma correlação negativa. Esse resultado demonstra que a maximização e minimização das funções utilizadas seguem direções opostas, evidenciando o conflito entre as funções. Dentre as combinações, em Compactação/Conectividade obteve-se maior correlação negativa em comparação a Mahalanobis/Conectividade. Avaliando as combinações que não geraram fronteira, observou-se correlação positiva entre os valores de funções-objetivo, demonstrando que Compactação/Cosseno e Mahalanobis/Cosseno não se mostraram conflitantes.

Para o conjunto *Fourty*, observou-se correlação baixa para todas as combinações, demonstrando que o uso da otimização multiobjetivo não é necessária para conjuntos com características semelhantes. Esse resultado nos indica que a utilização de um algoritmo mono-objetivo ou um algoritmo clássico de agrupamento que utilize um critério baseado em compactação seria o suficiente para encontrar uma solução de qualidade para o problema.

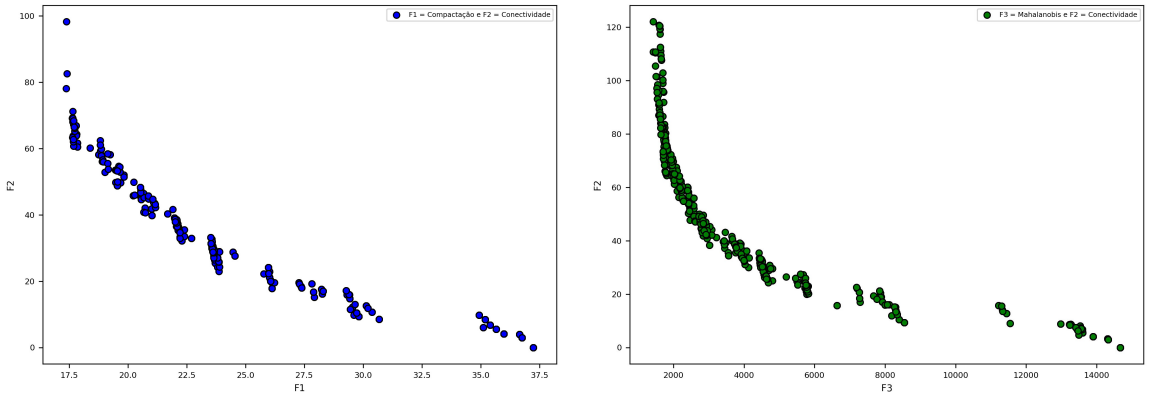
Dessa forma, analisando os resultados sob a ótica do multiobjetivo, para todos os conjuntos de dados, observamos a formação de fronteiras não-dominadas somente para as combinações de Compactação/Conectividade e Mahalanobis/Conectividade. Para as outras duas combinações, poucas soluções finais foram geradas e não foi possível observar a formação de uma fronteira não dominada. Por esse motivo, não apresentaremos as soluções finais obtidas em Compactação/Cosseno e Mahalanobis/Cosseno.

Nas Figuras 30 e 29, apresentamos as fronteiras finais obtidas, respectivamente, nos conjuntos 2d-4c-no6 e *Iris*, para Compactação/Conectividade e Mahalanobis/Conectividade. Analisando as fronteiras, observamos que em Mahalanobis/Conectividade são geradas soluções com maiores valores de conectividade em comparação a Compactação/Conectividade, tornando a fronteira mais preenchida na região que minimiza f_3 e maximiza f_2. Esse mesmo comportamento foi encontrado em todos os outros conjuntos de dados.



(a) f_1 = Compactação e f_2 = Conectividade. (b) f_3 = Mahalanobis e f_2 = Conectividade.

Figura 29 – Variações das funções-objetivo conjunto 2d-4c-no6.



(a) f_1 = Compactação e f_2 = Conectividade. (b) f_3 = Mahalanobis e f_2 = Conectividade.

Figura 30 – Variações das funções-objetivo conjunto *Iris*.

Para analisar as soluções geradas pelas fronteiras finais em relação ao agrupamento de dados, calculamos a acurácia (ACC) e o *Adjusted Rand Index* (ARI) das soluções não-dominadas com base na solução alvo do conjunto. A solução alvo dos conjuntos sintéticos e reais são conhecidas, dessa forma, calculamos o quão próxima a solução produzida pelo multiobjetivo está do ideal. Para o cálculo das métricas, para cada solução obtida no multiobjetivo atribui-se um vetor de rótulos de acordo com o agrupamento gerado. Dessa forma, cada grupo C_k gerado pela solução está associado a um rótulo r_k.

Na Tabela 15, apresentamos as médias dos melhores valores de acurácias e ARI encontrados nas combinações de Compactação/Conectividade e Mahalanobis/Conectividade. Para os conjuntos *Long1*, *Spiral*, *Square1*, foram consideradas 50 rodadas, enquanto que para os conjuntos *Fourty*, 2d-4c-no6 e 2d-40c-no6, 31 rodadas. Os resultados de ARI obtidos pelo MOCK foram retirados de Handl e Knowles (2007). Para os conjuntos *Fourty*, 2d-4c-no6 e 2d-40c-no6, os resultados não foram incluídos pois os autores não apresentaram no artigo. As colunas identificadas com "Agrupamento Multiobjetivo"referem-se aos resultados da abordagem proposta nesta dissertação.

Conjunto de dados	MOCK	Agrupamento Multiobjetivo			
	■/△	■/△		□/△	
	ARI	ARI	ACC	ARI	ACC
Long1	0.9998	1	1	1	1
Spiral	1	1	1	1	1
Square1	0.9622	0.9740	0.499	0.9739	0.499
Fourty	-	0.9733	0.8637	0.9733	0.8703
2d-4c-no6	-	0.9565	0.406	0.9569	0.3917
2d-40c-no6	-	0.8364	0.1821	0.83714	0.2019

Tabela 15 – Média dos melhores valores de acurácia e ARI obtidas pelas soluções não-dominadas dos conjuntos sintéticos. As funções-objetivo utilizadas pelos algoritmos multiobjetivos são: ■ = Compactação ; △ = Conectividade; □ = Mahalanobis.

Os valores de ARI obtidos nos conjuntos sintéticos são similares ao apresentado no MOCK, evidenciando que o algoritmo implementado nesse trabalho não possui diferenças significativas que afetam o desempenho e os resultados obtidos.

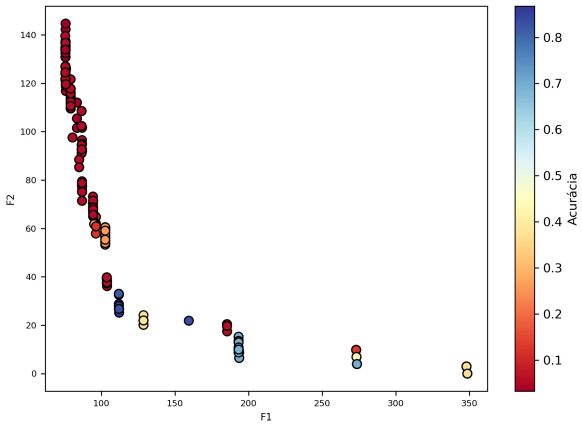
Na Tabela 16, apresentamos os resultados dos conjuntos reais em comparação ao MVMC, visto que no MOCK esses conjuntos não foram utilizados. O MVMC utiliza a função-objetivo *Within-Cluster Scatter* com diferentes funções de distâncias, para esses conjuntos o resultado apresentado foi o obtido com a distância euclidiana e a *Maximum Edge Distance* (MED), baseado na distância euclidiana. A partir dos resultados, podemos observar que o algoritmo de agrupamento multiobjetivo obteve resultados inferiores ao MVMC.

Conjunto de dados	MVMC	Agrupamento Multiobjetivo			
	$\nabla/\bigcirc$	■/△		□/△	
	ARI	ARI	ACC	ARI	ACC
Iris	0.922	0.7199	0.7568	0.7518	0.7748
Breast Cancer	0.837	0.7312	0.6309	0.7291	0.5763

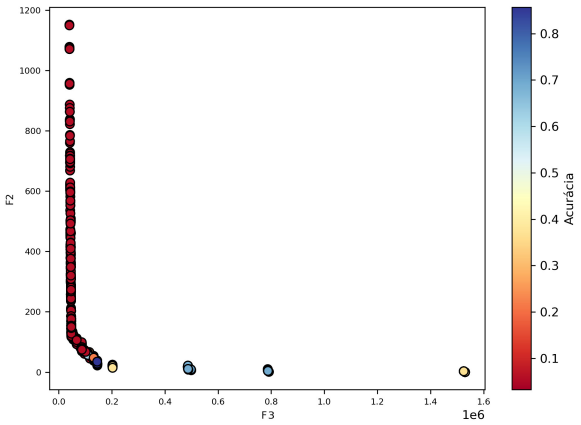
Tabela 16 – Média dos melhores valores de acurácia e ARI obtidas pelas soluções não-dominadas dos conjuntos reais. As funções-objetivo utilizadas pelos algoritmos multiobjetivos são: ∇ = Distância Euclidiana; $\bigcirc$ = MED_{euc}; ■ = Compactação ; △ = Conectividade; □ = Mahalanobis.

Nas combinações não conflitantes de Compactação/Cosseno e Mahalanobis/-Cosseno, observou-se a geração de soluções com baixa acurácia e ARI. Além disso, as soluções encontradas apresentaram maiores quantidades de grupos formados, o que explica baixa qualidade de agrupamento visto que não foram geradas soluções próximas da solução alvo. No conjunto *Iris*, por exemplo, para Compactação/Cosseno as soluções apresentavam de 16 a 23 grupos, enquanto em Compactação/Conectividade de 2 a 11 *clusters*.

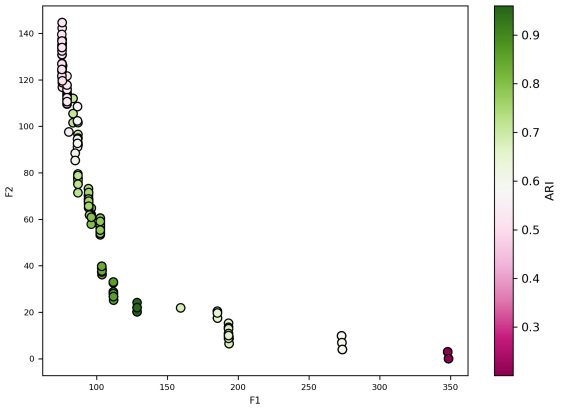
Na Figura 31, apresentamos as fronteiras não-dominadas em relação a acurácia e ARI do conjunto 2d-4c-no6. Para esse conjunto de dados, as soluções de maiores acurácia e ARI estão localizadas no meio da fronteira. O que nos mostra que, apesar da expansão da fronteira em termos de f_2 ser importante para otimização multiobjetivo, não implica necessariamente em melhor agrupamento, pois, a solução alvo do conjunto de dados pode estar localizada em diferentes regiões da fronteira. Nas Figuras 31c e 31d, podemos observar que soluções com baixa acurácia podem possuir alto ARI, resultado da diferença do cálculo de cada métrica.



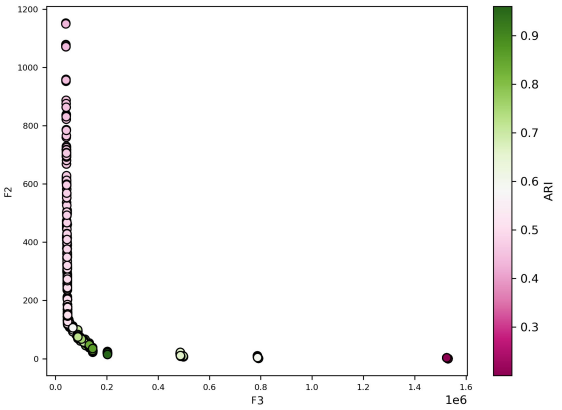
(a) Acurácia: f_1 = Compactação e f_2 = Conectividade.



(b) Acurácia: f_3 = Mahalanobis e f_2 = Conectividade.



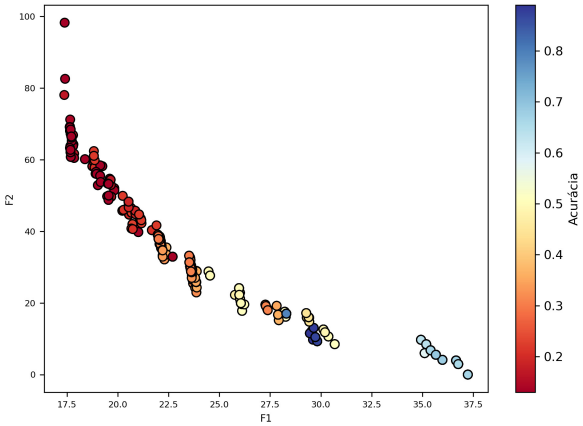
(c) ARI: f_1 = Compactação e f_2 = Conectividade.



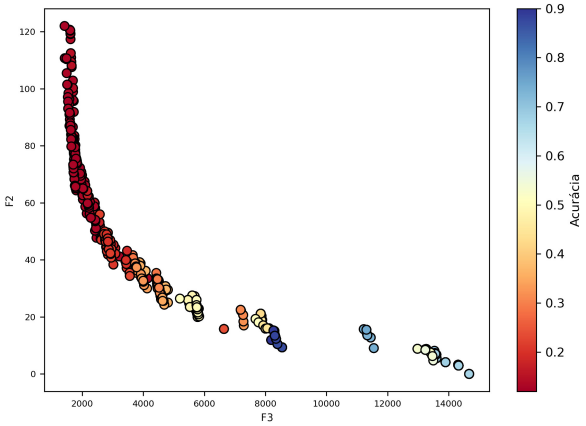
(d) ARI: f_3 = Mahalanobis e f_2 = Conectividade.

Figura 31 – Acurácia e ARI das soluções não-dominadas de 5 rodadas do conjunto 2d-4c-no6.

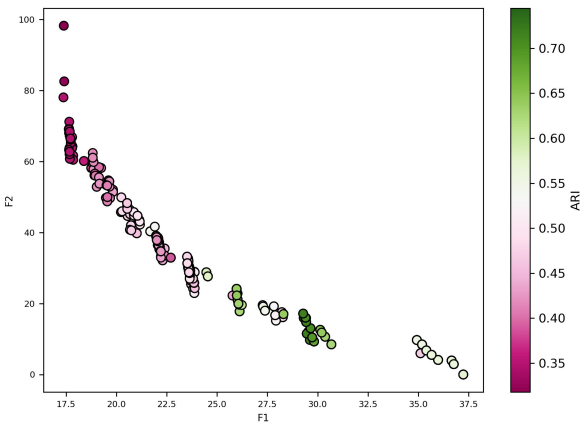
Na Figura 32, podemos observar as acurácias e ARI das soluções das fronteiras geradas pelas combinações Compactação/Conectividade (32a) e Mahalanobis/Conectividade (32b) para o conjunto *Iris*. Para esse conjunto as soluções de maiores acurácia e ARI estão localizadas na região de minimização de f_2.



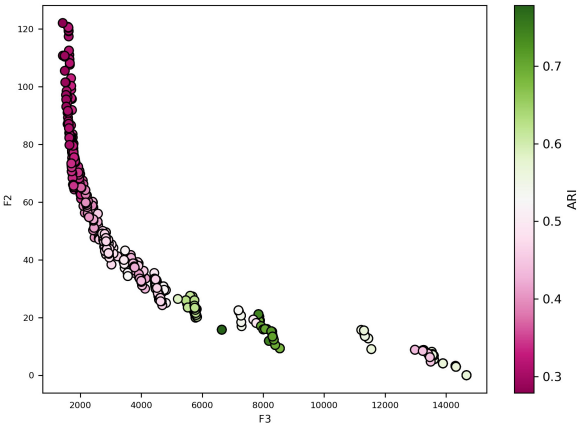
(a) Acurácia: f_1 = Compactação e f_2 = Conectividade.



(b) Acurácia: f_3 = Mahalanobis e f_2 = Conectividade.



(c) ARI: f_1 = Compactação e f_2 = Conectividade.



(d) ARI: f_3 = Mahalanobis e f_2 = Conectividade.

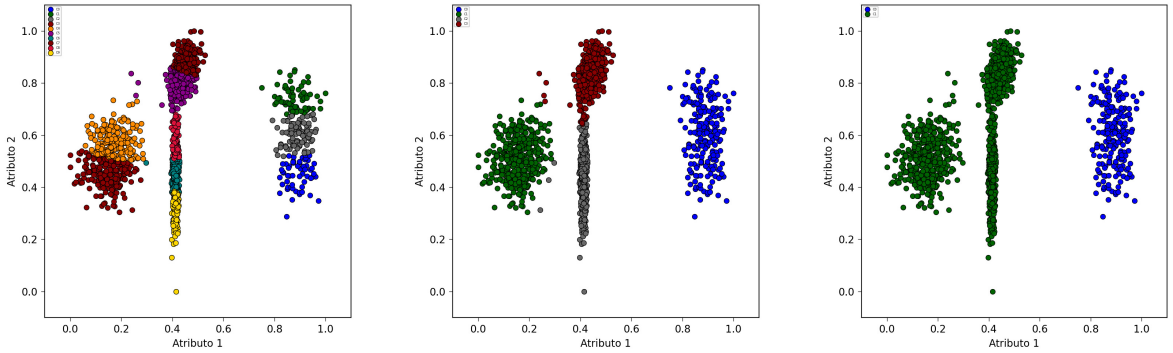
Figura 32 – Acurácia e ARI das soluções não-dominadas de 5 rodadas do conjunto *Iris*.

Em relação aos agrupamentos gerados pelas fronteiras de Compactação/Conectividade e Mahalanobis/Conectividade, como se utiliza funções de distâncias com características semelhantes, não foi possível observar diferenças nas soluções geradas pelas duas combinações.

Para melhor explicar as características de agrupamento, selecionaremos três soluções da fronteira de Compactação/Conectividade para o conjunto 2d-4c-no6 e *Iris* com diferentes quantidades de *clusters* e acurácia. Considerando que o conjunto *Iris* possui 4 atributos, para a visualização das soluções foi aplicado o PCA, reduzindo a dimensão dos dados. As soluções selecionadas são de regiões distintas da fronteira final: a solução 1 corresponde a um dos extremos da fronteira, associada a menores valores de compactação e maiores de conectividade; a solução 2 está localizada no meio da fronteira, associada com uma distribuição mista das duas funções; a solução 3 corresponde ao outro extremo da fronteira, associada a menores valores de conectividade e maiores de compactação.

Na Figura 33, apresentamos as três soluções do conjunto de dados 2d-4c-no6. Podemos observar as diferentes características das soluções: na solução 1, predomina-se os

grupos compactos, o que resulta em um número maior de grupos na solução não-dominadas; na solução 2, percebemos uma mistura das características de compactação e conectividade; na solução 3, a função f_2 assume valor igual a 0, visto que os grupos consideram os dados mais próximos.



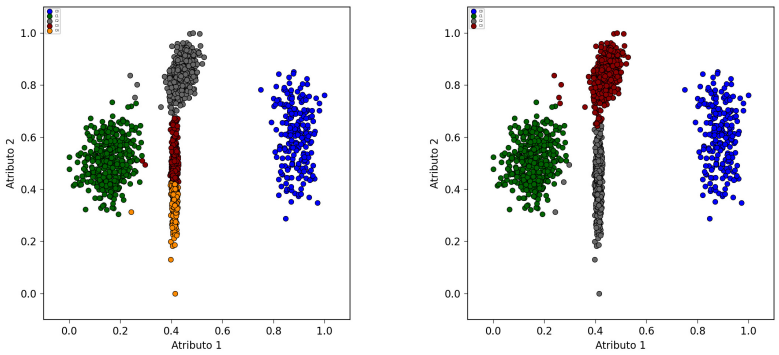
(a) Solução 1: Acc= 0.0347, ARI= 0.5286, $K = 10$, $f_1 = 75.7675$, $f_2 = 144.611$.

(b) Solução 2: Acc= 0.376, ARI= 0.9584, $K = 4$, $f_1 = 128.671$, $f_2 = 20.157$.

(c) Solução 3: Acc= 0.3707, ARI= 0.20, $K = 2$, $f_1 = 348.664$, $f_2 = 0$.

Figura 33 – Indivíduos não dominados de f_1 = Compactação e f_2 = Conectividade do conjunto 2d-4c-no6.

Na Figura 34, apresentamos as soluções com maiores acurácia e ARI do conjunto 2d-4c-no6. Observamos, que a solução de maior ARI é o que mais se aproxima da solução alvo do conjunto. A métrica de acurácia, por considerar os rótulos, não consegue capturar de forma efetiva a qualidade do agrupamento gerado.

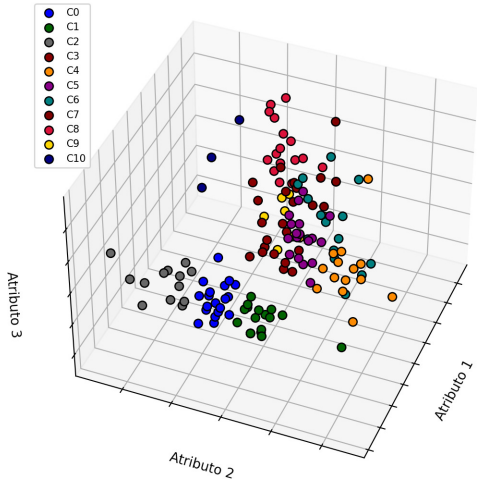


(a) Solução de maior acurácia: Acc= 0.8677, ARI= 0.8626, $K = 5$, $f_1 = 111.928$, $f_2 = 27.983$.

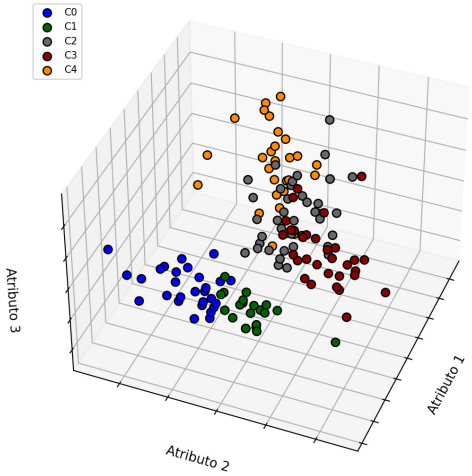
(b) Solução de maior ARI: Acc= 0.3754, ARI= 0.9602, $K = 4$, $f_1 = 128.669$, $f_2 = 24.137$.

Figura 34 – Soluções de maior acurácia e ARI de f_1 = Compactação e f_2 = Conectividade.

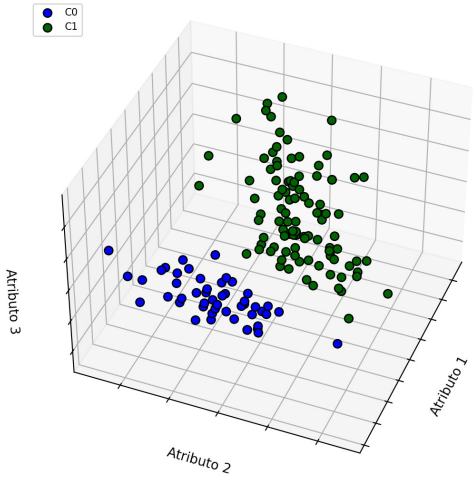
Na Figura 35, apresentamos as soluções do conjunto *Iris*. Nas soluções observamos o mesmo padrão encontrado no conjunto anterior.



(a) Solução 1: Acc= 0.1267, ARI= 0.318, $K = 11$, $f_1 = 17.3868$, $f_2 = 98.2468$.



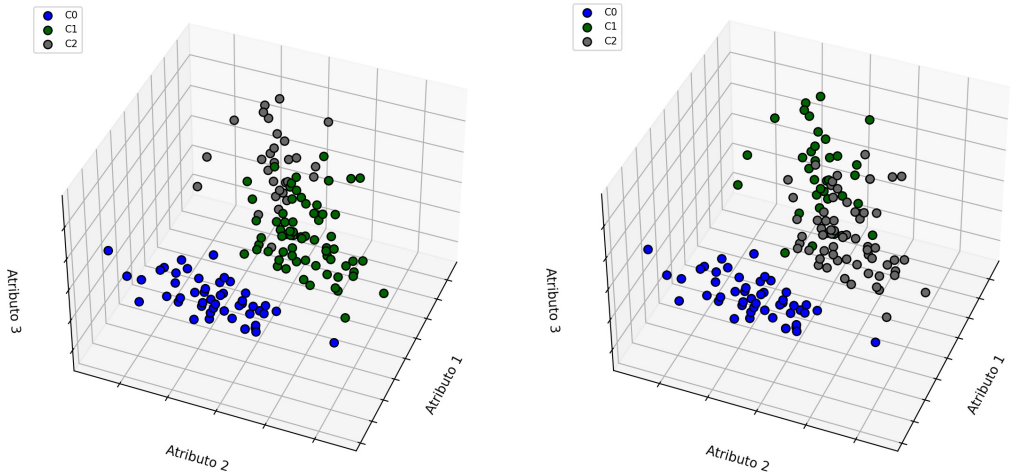
(b) Solução 2: Acc= 0.3067, ARI= 0.4833, $K = 5$, $f_1 = 23.5605$, $f_2 = 32.70$.



(c) Solução 3: Acc= 0.6667, ARI= 0.5681, $K = 2$, $f_1 = 37.2348$, $f_2 = 0$.

Figura 35 – Indivíduos não dominados de f_1 = Compactação e f_2 = Conectividade do conjunto *Iris*.

Na Figura 36, apresentamos as soluções com maior acurácia e ARI para o conjunto *Iris*.



(a) Solução de maior acurácia: Acc= 0.8867, ARI= 0.7196, $K = 3$, $f_1 = 29.55$, $f_2 = 12.142$.

(b) Solução de maior ARI: Acc= 0.4333, ARI= 0.7445, $K = 3$, $f_1 = 29.3801$, $f_2 = 15.529$.

Figura 36 – Soluções de maior acurácia e ARI de f_1 = Compactação e f_2 = Conectividade.

Nesse primeiro experimento, verificamos que para todos os conjuntos de dados, as combinações Compactação/Conectividade e Mahalanobis/Conectividade foram capazes de gerar soluções não-dominadas e com a mesma qualidade de agrupamento de dados. Não foi possível observar características diferentes entre as combinações de funções. Além disso, para Compactação/Cosseno e Mahalanobis/Cosseno, não foi possível observar a formação da Fronteira de Pareto e gerou-se soluções de baixa qualidade de agrupamento.

6.3 Análises a *posteriori*

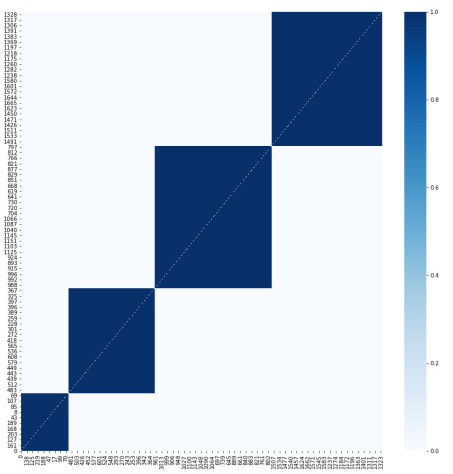
Após a análise inicial da otimização multiobjetivo, exploraremos como as análises a *posteriori* poderiam contribuir para um melhor entendimento dos resultados obtidos no agrupamento de dados.

Na seção anterior, observamos que soluções localizadas em partes distintas da fronteira não-dominada possuem características diferentes. Com o intuito de explorar essas características, propomos a divisão da fronteira não dominada em três partes, agrupando soluções com distribuições semelhantes. A ideia é analisar a característica predominante em cada parte, sem necessidade de olhar individualmente para cada solução.

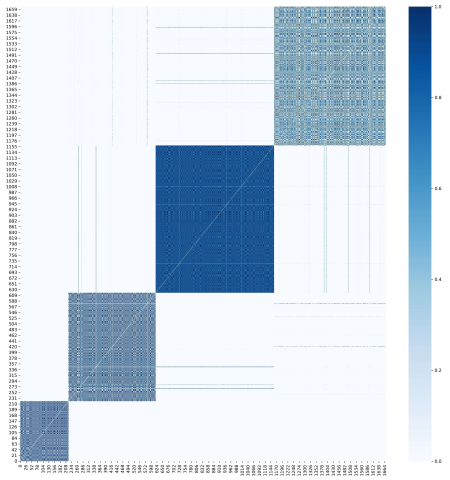
Para essa análise, utilizaremos os *heatmaps* como ferramentas visuais para e exploração dos resultados obtidos. Os *heatmaps* finais de cada parte da fronteira serão analisados em comparação com o *heatmap* da solução alvo. O *heatmap* alvo corresponde ao padrão gerado pela solução alvo de determinado conjunto de dados. Para a criação desse

heatmap associa-se uma matriz de compartilhamento de *clusters* ao vetor de rótulos da solução alvo, portanto, amostras que pertencem ao mesmo grupo assumem o valor igual a 1. Nos *heatmaps* gerados por cada parte da fronteira da otimização multiobjetivo, a matriz de compartilhamento de *clusters* é calculada com base na amostras pertencentes a cada parte. As linhas e colunas dos *heatmaps* são ordenadas de acordo com os rótulos verdadeiros do conjunto de dados. Para fins de comparação com a solução alvo, normalizamos a matriz obtida. A seguir, apresentaremos os resultados obtidos para Compactação/Conectividade.

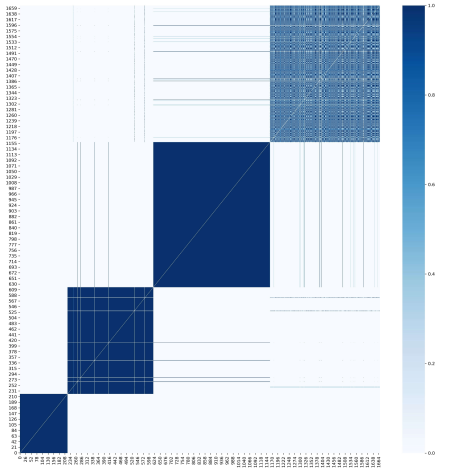
Na Figura 37, apresentamos os *heatmaps* do conjunto 2d-4c-no6. Nas três figuras produzidas pelo algoritmo multiobjetivo é possível observar semelhança com o padrão alvo, presente na diagonal dos *heatmaps*.



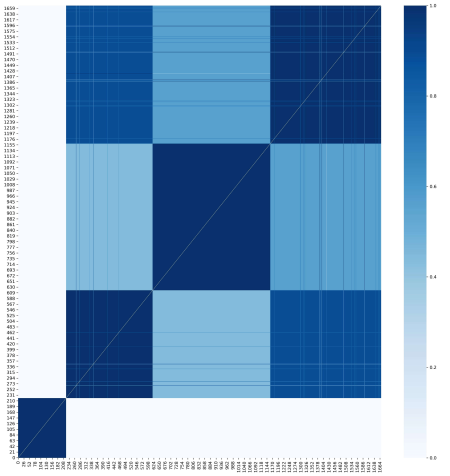
(a) *Heatmap* alvo.



(b) *Heatmap* 1: parte 1 da fronteira.



(c) *Heatmap* 2: parte 2 da fronteira.



(d) *Heatmap* 3: parte 3 da fronteira.

Figura 37 – *Heatmaps* 2d-4c-no6.

O *heatmap* 1 está associado a parte 1 da fronteira, correspondente as soluções com maiores números de grupos, menores valores de compactação e maiores de conectividade. Podemos observar que três dos quatro grupos formados na imagem possuem uma

cor de azul mais clara, nesses grupos as amostras não apresentaram compartilhamento de *clusters* de forma homogênea.

Por sua vez, o *heatmap* 2 está associado ao meio da fronteira com soluções de distribuições mistas das duas funções utilizadas, contendo soluções mais próximas da solução alvo. Nessa figura, podemos observar a formação dos grupos de forma mais homogênea, representada pelos três grupos de cor azul escuro. Esse padrão indica que nesses grupos as amostras foram agrupadas juntamente em quase todas as soluções não-dominadas, podendo estar associada as soluções com melhores valores de acurácia e ARI.

Por fim, o *heatmap* 3 está associado as soluções com menores números de grupos, menores valores de conectividade e maiores em compactação. Na figura, podemos observar a formação homogênea dos grupos na diagonal e a formação de outros grupos, representados pelas diferentes cores de azuis. Esse padrão nos indica que em algumas soluções não-dominadas, amostras de grupos diferentes foram incluídas no mesmo grupo.

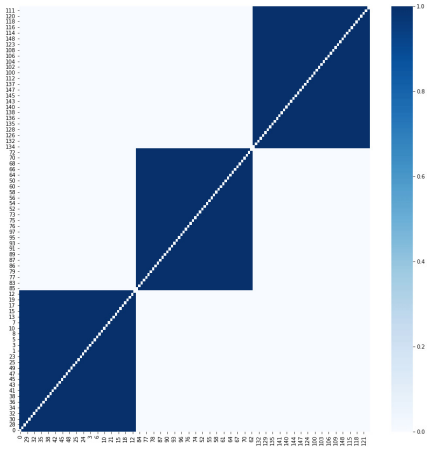
A partir dos *heatmaps* gerados, podemos observar padrões que podem ajudar na compreensão do problema. Nos *heatmaps* 1 e 2, observamos linhas e colunas brancas formadas dentro dos grupos e linhas azuis entre amostras de diferentes grupos. As linhas brancas indicam que a amostra obteve baixo compartilhamento com amostras de um mesmo grupo e as linhas azuis indicam que amostras de grupos diferentes apresentaram alto compartilhamento de *clusters*. Analisando essas amostras, observamos que elas geralmente estão localizadas distantes das demais amostras do grupo e por isso, são frequentemente agrupadas com amostras de outros grupos. No contexto de uma aplicação real, esse padrão poderia auxiliar na detecção de amostras com características fora do comum ou amostras presentes em regiões de sobreposições de grupos, podendo possuir características de mais de um grupo.

Outro padrão é o observado no *heatmap* 3, representado pelos grupos formados pelas diferentes intensidade de azuis. Os grupos formados indicam que as amostras, além de possuir alto grau de compartilhamento com as amostras do mesmo grupo, possuem também compartilhamento com outros grupos. A partir desse padrão poderíamos extrair possíveis informações de probabilidade, como por exemplo, "amostra x possui 70% chance de pertencer ao grupo 2, 10% ao grupo 3 e 20% ao grupo 4". A obtenção dessa saída poderia gerar informações complementares a um agrupamento.

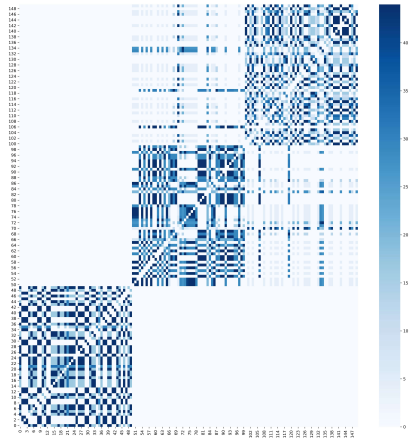
Na Figura 38 estão apresentados os *heatmaps* do conjunto *Iris*. Diferente do conjunto anterior, podemos observar maior dificuldade do algoritmo em se aproximar do *heatmap* alvo do problema. Os grupos formados na diagonal correspondem, respectivamente as classes *Setosa*, *Versicolor* e *Virginica*. Nas figuras podemos observar que os grupos formados não são homogêneos e amostras das classes *Versicolor* e *Virginica* são frequentemente agrupadas no mesmo grupo. Além disso, podemos observar os mesmos

padrões encontrados no conjunto anterior, como as linhas brancas e azuis, dentro e fora dos grupos principais.

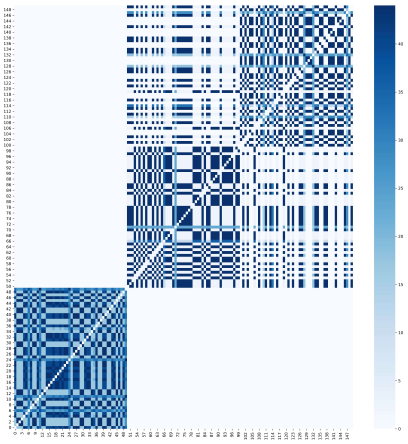
Para todos os conjuntos, os *heatmaps* de Mahalanobis/Conectividade seguiram o mesmo padrão encontrado em Compactação/Conectividade.



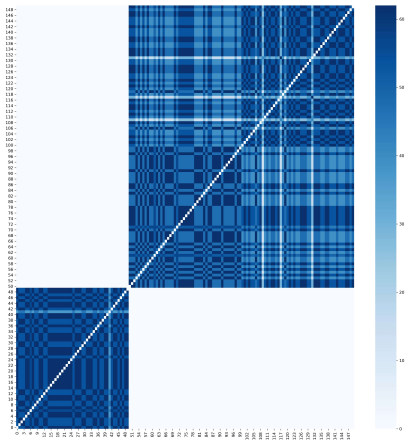
(a) *Heatmap* alvo.



(b) *Heatmap* 1: parte 1 da fronteira.



(c) *Heatmap* 2: parte 2 da fronteira.



(d) *Heatmap* 3: parte 3 da fronteira.

Figura 38 – *Heatmaps Iris*.

Na análise a *posteriori*, exploramos as informações geradas pelas soluções não-dominadas a partir da divisão da fronteira e o uso dos *heatmaps*. Na divisão das fronteiras, foi possível separar soluções finais com distribuições semelhantes e analisar os padrões gerados em uma ferramenta visual. Com o uso dos *heatmaps*, compreendemos melhor as distribuições presentes em cada parte, analisando a homogeneidade gerada pelos grupos, e identificamos padrões que poderiam ser utilizados para extrair informações complementares sobre, probabilidades, sobreposições e *outliers*.

Em um problema onde se conhece pouco sobre os dados, a análise a *posteriori* poderia contribuir para melhor entendimento dos resultados obtidos no agrupamento e na exploração dos dados. Em aplicações reais, fornecer os resultados a partir dos 3 *heatmaps* gerados pelo multiobjetivo pode auxiliar o tomador de decisão a compreender melhor o

problema a ser tratado, podendo ser preferível do que um resultado como os *heatmaps* alvo.

6.3.1 Extração de informações complementares

Este experimento tem como objetivo verificar se as informações geradas pela fronteira não-dominada serão capazes de obter informações complementares ao agrupamento de dados. Para isso, utilizamos dois conjuntos de dados, *Iris* e *Breast Cancer*, para validar a aplicação prática no contexto da área de saúde.

A partir das matrizes de compartilhamento de grupos obtidas ao final da otimização multiobjetivo, calculamos as probabilidades das amostras desconhecidas em relação as amostras rotuladas. Como descrito no Capítulo 5.3.4, consideraremos 20% das amostras do conjunto de dados como rotuladas. Assim, utilizamos uma pequena quantidade de amostras rotuladas em conjunto com as informações geradas pela fronteira não-dominada para a extração das informações.

A Tabela 17 apresenta os resultados obtidos em uma das rodadas para algumas das amostras desconhecidas do conjunto *Iris*. Podemos observar que cada parte da fronteira sugere diferentes informações de probabilidade, baseado nas diferentes distribuições das soluções não-dominadas. A amostra 1, por exemplo, obteve probabilidade igual a 1 para a classe 0 nas três partes da Fronteira de Pareto. Para a amostra 70, a maior probabilidade obtida entre as três partes corresponde a segunda parte da fronteira, com probabilidade de 0.833 para a classe 1 e 0.167 para a classe 2.

Amostra \ Classes	P_{ac} por classe									Rótulo verdadeiro
	Parte 1			Parte 2			Parte 3			
	0	1	2	0	1	2	0	1	2	
1	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	0
49	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	0
70	0	0.478	0.522	**0**	**0.833**	**0.167**	0	0.704	0.296	1
84	**0**	**1**	**0**	0	0.978	0.022	0	0.704	0.296	1
114	**0**	**0.135**	**0.865**	0	0.384	0.616	0	0.5	0.453	2
116	**0**	**0.08**	**0.92**	0	0.833	0.167	0	0.547	0.453	2

Tabela 17 – Probabilidade acumulada por rótulos obtidas nas três partes da fronteira não-dominada para o conjunto *Iris*. Classe 0 = Classe *Setosa*; Classe 1 = Classe *Versicolor*; Classe 2 = Classe *Virginica*. Destaca-se os maiores valores alcançados para cada amostra.

Para esses casos, podemos observar que o maior valor obtido de P_{ac} corresponde a classe verdadeira de cada amostra. Para as amostras 1 e 49, pertencentes a classe 0 (Setosa), observamos que a análise *posteriori* de todas as partes da fronteira indicou probabilidade igual a 1, representando alta chance de pertencer a classe verdadeira. Esse resultado está relacionado a característica da *Setosa*, que é separável linearmente das outras duas classes e, por isso, o algoritmo possui facilidade em agrupar amostras dessa classe.

Para as classes 1 (*Virginica*) e 2 (*Versicolor*), observamos probabilidades distribuídas entre as classes 1 e 2. Considerando que as classes não são separáveis linearmente, o agrupamento de dados se torna mais difícil. Apesar de em algumas amostras dessas classes, o maior valor de probabilidade não corresponder ao rótulo verdadeiro, as informações obtidas podem ser um indicativo de possíveis sobreposições entre as classes.

A Tabela 18, apresenta os resultados para o conjunto *Breast Cancer*. No contexto de uma aplicação na área de saúde, um resultado como o da amostra 36, poderia ser preferível para auxiliar o tratamento da doença, do que apenas o rótulo final da amostra. Assim como no conjunto anterior, os resultados de P_{ac} se aproximaram dos rótulos verdadeiros das amostras.

Amostra \ Classes	P_{ac} por classe						Rótulo verdadeiro
	Parte 1		Parte 2		Parte 3		
	0	1	0	1	0	1	
21	**1**	**0**	0.966	0.034	0.67	0.330	0
340	0.698	0.302	**0.967**	**0.033**	0.67	0.33	0
2	**0**	**1**	0.003	0.997	0.585	0.415	1
36	0.403	0.597	**0.23**	**0.77**	0.585	0.415	1

Tabela 18 – Probabilidade acumulada por rótulos obtidas nas três partes da fronteira não-dominada para o conjunto *Breast Cancer*. Classe 0 = Benigno; Classe 1 = Maligno. Destaca-se os maiores valores alcançados para cada amostra.

Para a validação do método quanto ao uso para uma possível atribuição de rótulos, calculamos a acurácia das amostras desconhecidas (80% do conjunto de dados). Para cada amostra desconhecida, atribuímos um rótulo único, considerado o rótulo que obteve maior valor de probabilidade entre as três partes da fronteira. Dessa forma, para o cálculo da métrica, comparamos o rótulo atribuído na análise a *posteriori* com o rótulo verdadeiro da amostra. No teste do conjunto *Iris*, obtemos uma acurácia de 0.86, e no *Breast Cancer*, 0.91.

Os experimentos demonstraram que o método é capaz de gerar informações complementares ao agrupamento de dados utilizando em conjunto as soluções não-dominadas. Os resultados obtidos complementam a análise *posteriori*, apresentando de forma quantitativa os padrões obtidos visualmente com os *heatmaps*.

7 Conclusão e Perspectivas Futuras

Nesta dissertação, no Capítulo 5, propomos um algoritmo de otimização multiobjetivo para agrupamento de dados combinado com uma etapa de análise a *posteriori*. No Capítulo 6, através dos experimentos, buscamos entender como a análise a *posteriori* poderia contribuir para o entendimento dos resultados obtidos no agrupamento de dados e como poderíamos explorar as soluções não-dominadas para extrair informações complementares ao agrupamento. Os algoritmos presentes na literatura enfatizam a eficiência do algoritmo sob a perspectiva da otimização multiobjetivo, sendo poucos os trabalhos que buscam explorar as informações geradas pela fronteira não-dominada.

Na primeira parte dos experimentos, analisamos o algoritmo proposto sob a perspectiva da Otimização Multiobjetivo. Nessa análise, observamos a relação de compromisso entre diferentes funções-objetivo e a capacidade de gerar diferentes características dos conjuntos de dados. Observamos que o algoritmo proposto apresentou resultados similares ao MOCK para as funções de Compactação/Conectividade e Mahalanobis/Conectividade, evidenciando que as diferenças apresentadas no algoritmo proposto não afetaram o desempenho do algoritmo original. Além disso, analisamos as fronteiras não-dominadas obtidas através das métricas de acurácia e ARI; e observamos como soluções localizadas em diferentes partes da fronteira correspondem a diferentes distribuições de dados.

Na segunda parte dos experimentos, propomos a análise a *posteriori* dividindo a fronteira em três partes para agrupar soluções com características semelhantes. Utilizamos os *heatmaps* para compreender de forma visual os agrupamento obtidos. Através dos *heatmaps*, identificamos que as informações carregadas por cada solução não-dominada, pode nos auxiliar na compreensão dos padrões presentes nos dados, especialmente quando desconhecemos o conjunto de dados. Além disso, identificamos que ao utilizar a informação de agrupamento de cada parte da fronteira não-dominada em conjunto com algumas amostras rotuladas, podemos obter informações complementares ao agrupamento, como informações de probabilidade. A etapa de extração de informações, permitiu reproduzir de forma quantitativa os padrões encontrados nos *heatmaps*, demonstrando que essas informações podem ser utilizadas para a rotulação de dados ou como complemento ao agrupamento.

Dessa forma, as questões colocadas inicialmente nesta dissertação foram respondidas, demonstrando que a proposta de uma análise a *posteriori* e a extração de informações complementares das soluções não-dominadas podem auxiliar os tomadores de decisão a entender melhor as características do problema.

Como perspectivas futuras, inicialmente, pretendemos aperfeiçoar a metodologia

proposta incluindo outras métricas que avaliam a qualidade do agrupamento, sendo importante para aplicações onde se desconhece os rótulos das amostras.

Além disso, aplicar a metodologia proposta em problemas semissupervisionados. Neste trabalho, a etapa de extração de informação, mostrou ter potencial para aplicação em tarefas de rotulação de dados. Além disso, a metodologia poderia ser aplicada em conjunto com métodos supervisionados, com o objetivo de melhorar a tarefa de classificação. Para isso, experimentos devem ser realizados com o objetivo de validar a aplicação, assim como, comparar os resultado obtidos com métodos já existentes na literatura.

Outra potencial aplicação seria em problemas de classificação *multi-label*, onde amostras do conjunto de dados podem ter mais de um rótulo associado. Nesses casos, acreditamos que as informações de probabilidade extraídas pela fronteira não-dominada, poderia contribuir para identificar amostras com mais de um rótulo.

Referências

ABDEL-BASSET, M.; ABDEL-FATAH, L.; SANGAIAH, A. K. Metaheuristic algorithms: A comprehensive review. *Computational intelligence for multimedia big data on the cloud with engineering applications*, Elsevier, p. 185–231, 2018. Citado na página 38.

AGGARWAL, C. C.; HINNEBURG, A.; KEIM, D. A. On the surprising behavior of distance metrics in high dimensional space. In: SPRINGER. *International conference on database theory*. [S.l.], 2001. p. 420–434. Citado na página 27.

ANKERST, M.; BREUNIG, M. M.; KRIEGEL, H.-P.; SANDER, J. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, ACM New York, NY, USA, v. 28, n. 2, p. 49–60, 1999. Citado na página 26.

ARORA, J.; TUSHIR, M.; KASHYAP, R. Improving semi-supervised classification using clustering. *EAI Endorsed Transactions on Scalable Information Systems*, European Alliance for Innovation, v. 7, n. 25, 2020. Citado 2 vezes nas páginas 53 e 60.

BAIR, E. Semi-supervised clustering methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, Wiley Online Library, v. 5, n. 5, p. 349–361, 2013. Citado 2 vezes nas páginas 28 e 31.

BASU, S.; BANERJEE, A.; MOONEY, R. Semi-supervised clustering by seeding. In: CITESEER. *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*. [S.l.], 2002. Citado na página 31.

BASU, S.; BANERJEE, A.; MOONEY, R. J. Active semi-supervision for pairwise constrained clustering. In: SIAM. *Proceedings of the 2004 SIAM international conference on data mining*. [S.l.], 2004. p. 333–344. Citado na página 31.

BASU, S.; BILENKO, M.; MOONEY, R. J. Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering. In: CITESEER. *Proceedings of the ICML-2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*. [S.l.], 2003. p. 42–49. Citado na página 31.

BELLMAN, R. E. Adaptive control processes. In: *Adaptive Control Processes*. [S.l.]: Princeton university press, 1961. Citado na página 26.

BENAYOUN, R.; MONTGOLFIER, J. D.; TERGNY, J.; LARITCHEV, O. Linear programming with multiple objective functions: Step method (stem). *Mathematical programming*, Springer, v. 1, n. 1, p. 366–375, 1971. Citado na página 38.

BEYER, K.; GOLDSTEIN, J.; RAMAKRISHNAN, R.; SHAFT, U. When is "nearest neighbor" meaningful? In: SPRINGER. *International conference on database theory*. [S.l.], 1999. p. 217–235. Citado na página 27.

CESSIE, S. L.; HOUWELINGEN, J. C. V. Ridge estimators in logistic regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, Wiley Online Library, v. 41, n. 1, p. 191–201, 1992. Citado na página 20.

CHAPELLE, O.; SCHÖLKOPF, B.; ZIEN, A. *Semi-Supervised Learning. Adaptive Computation and Machine Learning series*. [S.l.]: The MIT Press, 2006. Citado na página 28.

CHARNES, A.; COOPER, W. W. Management models and industrial applications of linear programming. *Management science*, INFORMS, v. 4, n. 1, p. 38–91, 1961. Citado na página 38.

CHENG, R.; JIN, Y.; OLHOFER, M.; SENDHOFF, B. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 20, n. 5, p. 773–791, 2016. Citado na página 41.

CHOPARD, B.; TOMASSINI, M. Problems, algorithms, and computational complexity. In: *An Introduction to Metaheuristics for Optimization*. [S.l.]: Springer, 2018. p. 1–14. Citado na página 38.

CHUGH, T.; JIN, Y.; MIETTINEN, K.; HAKANEN, J.; SINDHYA, K. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 22, n. 1, p. 129–142, 2016. Citado na página 41.

COMANICIU, D.; MEER, P. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 24, n. 5, p. 603–619, 2002. Citado na página 26.

CUNNINGHAM, P.; CORD, M.; DELANY, S. J. Supervised learning. In: *Machine learning techniques for multimedia*. [S.l.]: Springer, 2008. p. 21–49. Citado na página 13.

DAYAN, P.; NIV, Y. Reinforcement learning: the good, the bad and the ugly. *Current opinion in neurobiology*, Elsevier, v. 18, n. 2, p. 185–196, 2008. Citado na página 18.

DEB, K. Multi-objective optimization. In: *Search methodologies*. [S.l.]: Springer, 2014. p. 403–449. Citado na página 37.

DEB, K.; GUPTA, S. Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering optimization*, Taylor & Francis, v. 43, n. 11, p. 1175–1204, 2011. Citado na página 47.

DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, IEEE, v. 18, n. 4, p. 577–601, 2013. Citado na página 41.

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, IEEE, v. 6, n. 2, p. 182–197, 2002. Citado 2 vezes nas páginas 41 e 42.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. Citado na página 28.

DONATH, W. E.; HOFFMAN, A. J. Lower bounds for the partitioning of graphs. In: *Selected Papers Of Alan J Hoffman: With Commentary*. [S.l.]: World Scientific, 2003. p. 437–442. Citado na página 26.

DORIGO, M.; BIRATTARI, M.; STUTZLE, T. Ant colony optimization. *IEEE computational intelligence magazine*, IEEE, v. 1, n. 4, p. 28–39, 2006. Citado na página 39.

EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. In: IEEE. *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*. [S.l.], 1995. p. 39–43. Citado na página 39.

EDGEWORTH, F. Y. *Mathematical psychics: An essay on the application of mathematics to the moral sciences*. [S.l.]: CK Paul, 1881. Citado na página 35.

ENGELEN, J. E. V.; HOOS, H. H. A survey on semi-supervised learning. *Machine Learning*, Springer, v. 109, n. 2, p. 373–440, 2020. Citado 5 vezes nas páginas 8, 19, 28, 29 e 30.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *kdd*. [S.l.: s.n.], 1996. v. 96, n. 34, p. 226–231. Citado na página 26.

FACELI, K.; CARVALHO, A. C. de; SOUTO, M. C. de. Multi-objective clustering ensemble with prior knowledge. In: SPRINGER. *Brazilian Symposium on Bioinformatics*. [S.l.], 2007. p. 34–45. Citado na página 14.

FEO, T. A.; RESENDE, M. G. A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, Elsevier, v. 8, n. 2, p. 67–71, 1989. Citado na página 39.

FEOFILOFF, P. *Algoritmo de Kruskal*. São Paulo, 2019. Disponível em: <https://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/kruskal.html>. Acesso em: 20 nov. 2021. Citado na página 66.

FERREIRA, P. A. V. Otimização multiobjetivo: Teoria e aplicações. *Tese de Livre Docência*, 1999. Citado na página 36.

FOGEL, L. J.; OWENS, A. J.; WALSH, M. J. Artificial intelligence through a simulation of evolution. *Biophysics and Cybernetic Systems*, Spartan, Washington DC, p. 131–156, 1965. Citado na página 40.

FONSECA, C. M.; FLEMING, P. J. et al. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In: CITESEER. *Icga*. [S.l.], 1993. v. 93, n. July, p. 416–423. Citado na página 41.

FRADKOV, A. L. Early history of machine learning. *IFAC-PapersOnLine*, Elsevier, v. 53, n. 2, p. 1385–1390, 2020. Citado na página 17.

FRALICK, S. Learning to recognize patterns without a teacher. *IEEE Transactions on Information Theory*, IEEE, v. 13, n. 1, p. 57–64, 1967. Citado na página 28.

FREUND, Y.; SCHAPIRE, R. E. et al. Experiments with a new boosting algorithm. In: CITESEER. *icml*. [S.l.], 1996. v. 96, p. 148–156. Citado na página 20.

GAN, H.; SANG, N.; HUANG, R.; TONG, X.; DAN, Z. Using clustering analysis to improve semi-supervised classification. *Neurocomputing*, Elsevier, v. 101, p. 290–298, 2013. Citado 2 vezes nas páginas 53 e 60.

GARZA-FABRE, M.; HANDL, J.; KNOWLES, J. An improved and more scalable evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 22, n. 4, p. 515–535, 2017. Citado 3 vezes nas páginas 49, 51 e 57.

GARZA-FABRE, M.; SÁNCHEZ-MARTÍNEZ, A. L.; ALDANA-BOBADILLA, E.; LANDA, R. Decision making in evolutionary multiobjective clustering: A machine learning challenge. *IEEE Access*, IEEE, 2022. Citado na página 47.

GASS, S.; SAATY, T. The computational algorithm for the parametric objective function. *Naval research logistics quarterly*, Wiley Online Library, v. 2, n. 1-2, p. 39–45, 1955. Citado na página 38.

GEOFFRION, A. M.; DYER, J. S.; FEINBERG, A. An interactive approach for multi-criterion optimization, with an application to the operation of an academic department. *Management science*, INFORMS, v. 19, n. 4-part-1, p. 357–368, 1972. Citado na página 38.

GLOVER, F. Heuristics for integer programming using surrogate constraints. *Decision sciences*, Wiley Online Library, v. 8, n. 1, p. 156–166, 1977. Citado na página 39.

_____. Tabu search—part i. *ORSA Journal on computing*, Informs, v. 1, n. 3, p. 190–206, 1989. Citado na página 39.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 19.

GRIRA, N.; CRUCIANU, M.; BOUJEMAA, N. Unsupervised and semi-supervised clustering: a brief survey. *A review of machine learning techniques for processing multimedia content*, Citeseer, v. 1, p. 9–16, 2004. Citado na página 30.

GUHA, S.; RASTOGI, R.; SHIM, K. Cure: An efficient clustering algorithm for large databases. *ACM Sigmod record*, ACM New York, NY, USA, v. 27, n. 2, p. 73–84, 1998. Citado na página 25.

GUPTA, S.; KIM, J.; GRAUMAN, K.; MOONEY, R. Watch, listen & learn: Co-training on captioned images and videos. In: SPRINGER. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. [S.l.], 2008. p. 457–472. Citado na página 30.

HABIB, A.; SINGH, H. K.; CHUGH, T.; RAY, T.; MIETTINEN, K. A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 23, n. 6, p. 1000–1014, 2019. Citado na página 41.

HAIMES, Y. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE transactions on systems, man, and cybernetics*, v. 1, n. 3, p. 296–297, 1971. Citado na página 38.

HANDL, J.; KNOWLES, J. Multiobjective clustering with automatic determination of the number of clusters. In: INST. SCI. TECHNOL., UNIV. MANCHESTER, MANCHESTER, U.K., TECH. REP, TR-COMPSYSBIO-2004-02. [S.l.], 2004. Citado 2 vezes nas páginas 49 e 50.

_____. On semi-supervised clustering via multiobjective optimization. In: *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. [S.l.: s.n.], 2006. p. 1465–1472. Citado 2 vezes nas páginas 49 e 60.

_____. An evolutionary approach to multiobjective clustering. *IEEE transactions on Evolutionary Computation*, IEEE, v. 11, n. 1, p. 56–76, 2007. Citado 9 vezes nas páginas 45, 49, 51, 57, 63, 64, 73, 75 e 79.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H.; FRIEDMAN, J. H. *The elements of statistical learning: data mining, inference, and prediction*. [S.l.]: Springer, 2009. v. 2. Citado na página 20.

HO, T. K. Random decision forests. In: IEEE. *Proceedings of 3rd international conference on document analysis and recognition*. [S.l.], 1995. v. 1, p. 278–282. Citado na página 20.

HOLLAND, J. H. Genetic algorithms and the optimal allocation of trials. *SIAM journal on computing*, v. 2, p. 88–105, 1973. Citado na página 40.

_____. Genetic algorithms. *Scientific american*, JSTOR, v. 267, n. 1, p. 66–73, 1992. Citado na página 39.

HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. E. A niched pareto genetic algorithm for multiobjective optimization. In: IEEE. *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence*. [S.l.], 1994. p. 82–87. Citado na página 41.

HUBERT, L.; ARABIE, P. Comparing partitions. *Journal of classification*, Springer, v. 2, n. 1, p. 193–218, 1985. Citado na página 73.

HUSSAIN, K.; SALLEH, M. N. M.; CHENG, S.; SHI, Y. Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review*, Springer, v. 52, n. 4, p. 2191–2233, 2019. Citado 2 vezes nas páginas 38 e 40.

ISHIBUCHI, H.; IMADA, R.; SETOGUCHI, Y.; NOJIMA, Y. Performance comparison of nsga-ii and nsga-iii on various many-objective test problems. In: IEEE. *2016 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.], 2016. p. 3045–3052. Citado na página 41.

JAIN, A. K. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, Elsevier, v. 31, n. 8, p. 651–666, 2010. Citado na página 21.

JIN, Y.; WANG, H.; CHUGH, T.; GUO, D.; MIETTINEN, K. Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 23, n. 3, p. 442–458, 2018. Citado na página 41.

JING, L.; TIAN, Y. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 43, n. 11, p. 4037–4058, 2020. Citado na página 28.

JOSÉ-GARCÍA, A.; HANDL, J. On the interaction between distance functions and clustering criteria in multi-objective clustering. In: SPRINGER. *International Conference on Evolutionary Multi-Criterion Optimization*. [S.l.], 2021. p. 504–515. Citado 2 vezes nas páginas 45 e 52.

JOSé-GARCíA, A.; HANDL, J.; GóMEZ-FLORES, W.; GARZA-FABRE, M. An evolutionary many-objective approach to multiview clustering using feature and relational data. *Applied Soft Computing*, Elsevier, v. 108, p. 107425, 2021. Citado 3 vezes nas páginas 49, 51 e 64.

KARYPIS, G.; HAN, E.-H.; KUMAR, V. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, IEEE, v. 32, n. 8, p. 68–75, 1999. Citado na página 25.

KAUFMAN, L.; ROUSSEEUW, P. J. Finding groups in data: an introduction to cluster analysis. *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics*, John Wiley & Sons, v. 344, p. 68–125, 1990. Citado 2 vezes nas páginas 22 e 23.

KIRKPATRICK, S.; JR, C. D. G.; VECCHI, M. P. Optimization by simulated annealing. *science*, American association for the advancement of science, v. 220, n. 4598, p. 671–680, 1983. Citado na página 39.

KNOWLES, J. D.; CORNE, D. W. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, MIT Press, v. 8, n. 2, p. 149–172, 2000. Citado na página 41.

KRUSKAL, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, JSTOR, v. 7, n. 1, p. 48–50, 1956. Citado na página 26.

LECUN, Y.; MISRA, I. *Self-supervised learning: The dark matter of intelligence*. USA, 2021. Disponível em: <https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/>. Acesso em: 20 nov. 2021. Citado 3 vezes nas páginas 13, 17 e 27.

LEO, B.; FRIEDMAN, J. H.; OLSHEN, R. A.; STONE, C. J. Classification and regression trees. *Wadsworth International Group*, v. 8, p. 452–456, 1984. Citado na página 20.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], 1967. v. 1, n. 14, p. 281–297. Citado na página 22.

MAIMON, O.; COHEN, S. A review of reinforcement learning methods. *Data Mining and Knowledge Discovery Handbook*, Springer, p. 401–417, 2009. Citado na página 18.

MATAKE, N.; HIROYASU, T.; MIKI, M.; SENDA, T. Multiobjective clustering with automatic k-determination for large-scale data. In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. [S.l.: s.n.], 2007. p. 861–868. Citado na página 50.

MAULIK, U.; MUKHOPADHYAY, A.; BANDYOPADHYAY, S. Combining pareto-optimal clusters using supervised learning for identifying co-expressed genes. *BMC bioinformatics*, BioMed Central, v. 10, n. 1, p. 1–16, 2009. Citado 4 vezes nas páginas 49, 50, 57 e 58.

MERZ, C. J.; CLAIR, D. S.; BOND, W. E. Semi-supervised adaptive resonance theory (smart2). In: IEEE. *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*. [S.l.], 1992. v. 3, p. 851–856. Citado na página 28.

MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. *Computers & operations research*, Elsevier, v. 24, n. 11, p. 1097–1100, 1997. Citado na página 39.

MUKHOPADHYAY, A.; MAULIK, U. A multiobjective approach to mr brain image segmentation. *Applied Soft Computing*, Elsevier, v. 11, n. 1, p. 872–880, 2011. Citado 2 vezes nas páginas 49 e 51.

MUKHOPADHYAY, A.; MAULIK, U.; BANDYOPADHYAY, S. A survey of multiobjective evolutionary clustering. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 47, n. 4, p. 1–46, 2015. Citado 7 vezes nas páginas 8, 10, 14, 45, 46, 47 e 48.

NANDA, S. J.; PANDA, G. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary computation*, Elsevier, v. 16, p. 1–18, 2014. Citado na página 45.

NG, R. T.; HAN, J. Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, IEEE, v. 14, n. 5, p. 1003–1016, 2002. Citado na página 23.

NIAN, R.; LIU, J.; HUANG, B. A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, Elsevier, v. 139, p. 106886, 2020. Citado na página 18.

OSKOLKOV, N. *How to cluster in High Dimensions*. [S.l.], 2019. Disponível em: <https://towardsdatascience.com/how-to-cluster-in-high-dimensions-4ef693bacc6>. Acesso em: 20 nov. 2021. Citado na página 22.

PAN, L.; HE, C.; TIAN, Y.; WANG, H.; ZHANG, X.; JIN, Y. A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 23, n. 1, p. 74–88, 2018. Citado na página 41.

PARETO, V. Manuale di economia politica. *Societa Editrice Libraria, Milano, Italy*, 1906. Citado na página 35.

QI, G.-J.; LUO, J. Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, 2020. Citado 2 vezes nas páginas 27 e 28.

RAJEEV, S. G.; RASTOGI, R.; SHIM, K. Rock: A robust clustering algorithm for categorical attributes. In: CITESEER. *Information systems*. [S.l.], 1999. Citado na página 25.

RECHENBERG, I. Evolutionstrategie: Optimierung technisher systeme nach prinzipien des biologischen evolution. *Frommann-Hollboog Verlag, Stuttgart*, 1973. Citado na página 40.

ROKACH, L. A survey of clustering algorithms. In: *Data mining and knowledge discovery handbook*. [S.l.]: Springer, 2009. p. 269–298. Citado 2 vezes nas páginas 24 e 26.

RUNKLER, T. A. *Data analytics*. [S.l.]: Springer, 2020. Citado na página 13.

SANCHES, M. K. *Aprendizado de Máquina semi-supervisionado: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados*. Dissertação (Dissertação de Mestrado) — Universidade de São Paulo - USP - São Carlos, 2003. Citado 2 vezes nas páginas 17 e 18.

SARKER, I. H. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, Springer, v. 2, n. 3, p. 1–21, 2021. Citado 6 vezes nas páginas 8, 17, 18, 19, 20 e 26.

SAXENA, A.; PRASAD, M.; GUPTA, A.; BHARILL, N.; PATEL, O. P.; TIWARI, A.; ER, M. J.; DING, W.; LIN, C.-T. A review of clustering techniques and developments. *Neurocomputing*, Elsevier, v. 267, p. 664–681, 2017. Citado na página 24.

SCHAFFER, J. D. *Multiple objective optimization with vector evaluated genetic algorithms*. Tese (Doutorado) — Vanderbilt University, 1984. Citado na página 40.

SCHMARJE, L.; SANTAROSSA, M.; SCHRÖDER, S.-M.; KOCH, R. A survey on semi-, self-and unsupervised learning for image classification. *IEEE Access*, IEEE, v. 9, p. 82146–82168, 2021. Citado 2 vezes nas páginas 18 e 27.

SCHWEFEL, H.-P. Evolution and optimum seeking. *John Wiley and Sons, New York*, 1995. Citado na página 40.

SCUDDER, H. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, IEEE, v. 11, n. 3, p. 363–371, 1965. Citado na página 28.

SEN, P. C.; HAJRA, M.; GHOSH, M. Supervised classification algorithms in machine learning: A survey and review. In: *Emerging technology in modelling and graphics*. [S.l.]: Springer, 2020. p. 99–111. Citado na página 20.

SHARAN, R.; SHAMIR, R. Click: a clustering algorithm with applications to gene expression analysis. In: MARYLAND, MD. *Proc Int Conf Intell Syst Mol Biol*. [S.l.], 2000. v. 8, n. 307, p. 16. Citado na página 26.

SILVER, D.; HUANG, A.; MADDISON, C. J.; GUEZ, A.; SIFRE, L.; DRIESSCHE, G. V. D.; SCHRITTWIESER, J.; ANTONOGLOU, I.; PANNEERSHELVAM, V.; LANCTOT, M. et al. Mastering the game of go with deep neural networks and tree search. *nature*, Nature Publishing Group, v. 529, n. 7587, p. 484–489, 2016. Citado na página 18.

SÖRENSEN, K.; GLOVER, F. Metaheuristics. *Encyclopedia of operations research and management science*, Springer New York, v. 62, p. 960–970, 2013. Citado na página 38.

SRINIVAS, N.; DEB, K. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, MIT Press, v. 2, n. 3, p. 221–248, 1994. Citado na página 41.

STEGHERR, H.; HEIDER, M.; HÄHNER, J. Classifying metaheuristics: Towards a unified multi-level classification system. *Natural Computing*, Springer, p. 1–17, 2020. Citado na página 38.

THEODORIDIS, S. *Machine learning: a Bayesian and optimization perspective*. USA: Academic Press, 2015. Citado 2 vezes nas páginas 19 e 23.

THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition*. USA: Academic Press, 2008. Citado 3 vezes nas páginas 17, 21 e 24.

TORO, F.; ROS, E.; MOTA, S.; ORTEGA, J. Evolutionary algorithms for multiobjective and multimodal optimization of diagnostic schemes. *IEEE Transactions on Biomedical Engineering*, v. 53, n. 2, p. 178–189, 2006. Citado na página 39.

TSYGANOV, V. Learning of quartering in digital control of refit. In: IEEE. *2020 Global Smart Industry Conference (GloSIC)*. [S.l.], 2020. p. 163–170. Citado na página 17.

VAPNIK, V.; CHERVONENKIS, A. On a class of algorithms of learning pattern recognition. *Automation and Remote Control*, v. 25, p. 937–945, 1964. Citado na página 20.

VERLEYSEN, M.; FRANÇOIS, D. The curse of dimensionality in data mining and time series prediction. In: SPRINGER. *International work-conference on artificial neural networks*. [S.l.], 2005. p. 758–770. Citado na página 27.

WAGSTAFF, K.; CARDIE, C.; ROGERS, S.; SCHRÖDL, S. et al. Constrained k-means clustering with background knowledge. In: *Icml*. [S.l.: s.n.], 2001. v. 1, p. 577–584. Citado na página 31.

WATT, J.; BORHANI, R.; KATSAGGELOS, A. K. *Machine learning refined: Foundations, algorithms, and applications*. [S.l.]: Cambridge University Press, 2016. Citado na página 27.

WECK, O. L. D. Multiobjective optimization: History and promise. In: *Invited Keynote Paper, GL2-2, The Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems, Kanazawa, Japan*. [S.l.: s.n.], 2004. v. 2, p. 34. Citado na página 35.

XU, D.; TIAN, Y. A comprehensive survey of clustering algorithms. *Annals of Data Science*, Springer, v. 2, n. 2, p. 165–193, 2015. Citado 3 vezes nas páginas 10, 21 e 22.

XU, R.; WUNSCH, D. Survey of clustering algorithms. *IEEE Transactions on neural networks*, Ieee, v. 16, n. 3, p. 645–678, 2005. Citado 2 vezes nas páginas 21 e 26.

ZAHN, C. T. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on computers*, IEEE, v. 100, n. 1, p. 68–86, 1971. Citado na página 26.

ZHANG, Q.; LI, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, IEEE, v. 11, n. 6, p. 712–731, 2007. Citado na página 41.

ZHANG, T.; RAMAKRISHNAN, R.; LIVNY, M. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, ACM New York, NY, USA, v. 25, n. 2, p. 103–114, 1996. Citado na página 25.

ZHU, X.; GOLDBERG, A. B. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, Morgan & Claypool Publishers, v. 3, n. 1, p. 1–130, 2009. Citado na página 18.

ZHU, X. J. Semi-supervised learning literature survey. University of Wisconsin-Madison Department of Computer Sciences, 2005. Citado na página 28.

ZITZLER, E.; LAUMANNS, M.; THIELE, L. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische ..., v. 103, 2001. Citado na página 41.

ZITZLER, E.; THIELE, L. Multiobjective optimization using evolutionary algorithms—a comparative case study. In: SPRINGER. *International conference on parallel problem solving from nature*. [S.l.], 1998. p. 292–301. Citado na página 41.