



**INSTITUTO
TECNOLÓGICO
VALE**

**Programa de Pós-Graduação em Instrumentação, Controle e
Automação de Processos de Mineração (PROFICAM)
Escola de Minas, Universidade Federal de Ouro Preto (UFOP)
Associação Instituto Tecnológico Vale (ITV)**

Dissertação

**SELEÇÃO DE ATRIBUTOS BASEADO NO ALGORITMO DE OTIMIZAÇÃO POR
COLÔNIA DE FORMIGAS PARA PROCESSOS MINERADORES**

Pedro Fontes Ayres

**Ouro Preto
Minas Gerais, Brasil
2021**

Pedro Fontes Ayres

**SELEÇÃO DE ATRIBUTOS BASEADO NO ALGORITMO DE OTIMIZAÇÃO POR
COLÔNIA DE FORMIGAS PARA PROCESSOS MINERADORES**

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração da Universidade Federal de Ouro Preto e do Instituto Tecnológico Vale, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Controle e Automação.

Orientador: Prof. Bruno Nazário Coelho,
D.Sc.

Coorientador: Prof. Jodelson Aguilar Sabino,
D.Sc.

Ouro Preto
2021

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

A985s Ayres, Pedro Fontes.

Seleção de atributos baseado no algoritmo de otimização por colônia de formigas para processos mineradores. [manuscrito] / Pedro Fontes Ayres. - 2021.
64 f.

Orientador: Prof. Dr. Bruno Nazário Coelho.

Coorientador: Prof. Dr. Jodelson Aguilar Sabino.

Dissertação (Mestrado Profissional). Universidade Federal de Ouro Preto. Programa de Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração. Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração.

Área de Concentração: Engenharia de Controle e Automação de Processos Minerais.

1. Mineração de dados(Computação). 2. Processamento de dados. 3. Programação heurística. I. Coelho, Bruno Nazário. II. Sabino, Jodelson Aguilar. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 681.5:622.2

Bibliotecário(a) Responsável: Angela Maria Raimundo - SIAPE: 1.644.803



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
PROGR. POS GRAD. PROF. INST. CONT. E AUT.
PROCESSOS DE MIN.



FOLHA DE APROVAÇÃO

Pedro Fontes Ayres

Seleção de atributos baseado no algoritmo de otimização por colônia de formigas para processos mineradores

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração (PROFICAM), Convênio Universidade Federal de Ouro Preto/Associação Instituto Tecnológico Vale - UFOP/ITV, como requisito parcial para obtenção do título de Mestre em Engenharia de Controle e Automação na área de concentração em Instrumentação, Controle e Automação de Processos de Mineração.

Aprovada em 04 de novembro de 2021

Membros da banca

Doutor - Bruno Nazário Coelho - Orientador(a) - Universidade Federal de Ouro Preto
Doutor - Jodelson Aguilar Sabino - Coorientador - Vale
Doutor - Agnaldo José da Rocha Reis - Universidade Federal de Ouro Preto
Doutor - Igor Machado Coelho - Universidade Federal Fluminense

Bruno Nazário Coelho, orientador do trabalho, aprovou a versão final e autorizou seu depósito no Repositório Institucional da UFOP em 26/01/2022



Documento assinado eletronicamente por **Bruno Nazário Coelho, COORDENADOR(A) DE CURSO DE PÓS-GRAD EM INSTRUMENTAÇÃO CONTROLE E AUTOMAÇÃO DE PROCESSOS DE MINERAÇÃO**, em 27/01/2022, às 11:34, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0272528** e o código CRC **B8AD2B30**.

*... a todos pesquisadores
brasileiros, por acreditarem neste
país.*

*“Não sabem ler é verdade
Aqueles pra quem eu escrevo.
Mas há sempre uma candeia
Dentro da própria desgraça
Há sempre alguém que semeia
Canções no vento que passa.
Mesmo na noite mais triste
Em tempo de servidão
Há sempre alguém que resiste
Há sempre alguém que diz não.”
(Manuel Alegre)*

Resumo

Resumo da Dissertação apresentada ao Programa de Pós Graduação em Instrumentação, Controle e Automação de Processos de Mineração como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SELEÇÃO DE ATRIBUTOS BASEADO NO ALGORITMO DE OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS PARA PROCESSOS MINERADORES

Pedro Fontes Ayres

Novembro/2021

Orientadores: Bruno Nazário Coelho
Jodelson Aguilar Sabino

Análise avançada de dados referentes aos processos relacionados à atividade mineradora apresentam um cenário desafiador que é extrair conhecimento útil a partir de diferentes bancos de dados. Diante disto, inserido na ciência de Mineração de Dados, uma tecnologia utilizada na fase de pré-processamento dos dados conhecida como Seleção de Atributos visa reduzir a dimensionalidade dos dados e identificar suas variáveis relevantes, aumentando assim o poder preditivo do classificador. Abordagens computacionais baseadas em Inteligência de Enxame vem se destacando para esta etapa. A seguinte Dissertação aborda a evolução dos algoritmos baseado na Otimização por Colônia de Formigas, conhecido algoritmo meta-heurístico de exploração, aplicados à Seleção de Atributos, assim como a proposta de um novo método com características do tipo *Filter* e *Wrapper*. Seu desempenho é avaliado a partir de diferentes bancos de dados e também comparando-se seus resultados com outros algoritmos referenciais acerca do tema. Uma aplicação prática da redução da dimensionalidade de um banco de dados relacionados à gestão de pátios, adotando-se um algoritmo referencial é descrito através de artigo. O trabalho foi orientado através da metodologia de dados CRISP-DM e disponibilizado um projeto no *Github* a partir do produzido.

Palavras-chave: Seleção de Atributos, Otimização por Colônia de Formiga, Redução da Dimensionalidade, Meta-Heurística, .

Macrotema: Usina; **Linha de Pesquisa:** Tecnologias da Informação, Comunicação e Automação Industrial; **Tema:** Manutenção Preventiva, Preditiva e Corretiva.

Abstract

Abstract of Dissertation presented to the Graduate Program on Instrumentation, Control and Automation of Mining Process as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

FEATURE SELECTION BASED ON ANT COLONY OPTIMIZATION FOR MINING PROCESS

Pedro Fontes Ayres

November/2021

Advisors: Bruno Nazário Coelho
Jodelson Aguilar Sabino

Advanced Data Mining and Analysis related to Mining activity present a challenging scenario that is to extract useful knowledge from databases. Given this, a technology used in the data pre-processing step known as Feature Selection aims to reduce Data Dimensionality and to identify the relevant feature subset from the original features, thus increasing the performance of classification or clustering algorithms. Computational approaches based on Swarm Intelligence have been highlighted for this field and the following Dissertation presents an overview of algorithms based on Ant Colony Optimization applied to Feature Selection, as well as a new proposed method classified as a Filter-Wrapper based. Our proposed approach has been evaluated on various real-life datasets. The experimental outcomes have been compared to reference algorithms on the subject. A case of study is described in which a reference algorithm has run for the analysis. The project was guided through the CRISP-DM methodology and all the algorithms and references for this project are available on *Github* from the author.

Keywords: Feature Selection, Ant Colony Optmization, Dimensionality Reduction, Meta-Heuristics, .

Macrotheme: Plant; **Research Line:** Information Technology, Communication and Automation Industrial; **Theme:** Preventive, Predictive and Corrective Maintenance.

Lista de Figuras

2.1	Ciclo de vida do modelo CRISP-DM. Fonte: Otari (2018).	18
2.2	Processo de Seleção de Atributos e suas etapas. Fonte: Sutha e Tamilselvi (2015).	19
2.3	Método <i>Embedded</i> : A Seleção de Atributos requer dados de treinamento para fins de aprendizado. Fonte: Kabir <i>et al.</i> (2011).	22
2.4	Taxonomia dos métodos de Seleção de Atributos não-supervisionados - UFS. Fonte: Solorio-Fernández <i>et al.</i> (2020).	23
2.5	Formigas convergem para o caminho mais curto. Fonte: Gutjahr e Rauner (2007).	26
2.6	Grafo de construção assimétrico para problema do caixeiro viajante. Fonte: Gutjahr (2007).	28
3.1	Representação gráfica de um problema de Seleção de Atributos onde $S_{i,j}$ representa a similaridade associada a cada nó entre as variáveis i e j . Em outras palavras, $S_{i,j} = sim(F_i, F_j)$. Fonte: Tabakhi <i>et al.</i> (2014).	34
3.2	Representação gráfica de múltiplo caminho através de grafo. Dois caminhos, conforme ilustrado em negrito e tracejado, possuem o mesmo sub-conjunto $\{A, B, C, D, E\}$. Fonte: Ghosh <i>et al.</i> (2019).	36
3.3	Fluxograma WFACOFs. Fonte: Ghosh <i>et al.</i> (2019).	37
3.4	Fluxograma FS_ACO. Fonte: O autor.	38
3.5	Matriz de similaridade para banco de dados <i>Wine</i> . Fonte: O autor.	40

Lista de Tabelas

2.1	Resumo das vantagens e desvantagens dos métodos para Seleção de Atributos Não-Supervisionados.	24
3.1	Exemplos mais destacados de algoritmos baseados em ACO para Seleção de Atributos.	32
3.2	Bibliotecas e funções Python utilizadas no algoritmo FS_ACO.	39
3.3	Descrição dos Bancos de Dados utilizados nas análises	45
4.1	Cenários obtidos após execuções	49
4.2	Descrição dos parâmetros e valores ACO.	51
4.3	Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados Wine	51
4.4	Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados Vowel	51
4.5	Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados Breast Cancer	52
4.6	Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados Zoo	52
4.7	Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados Lymphography	53
4.8	Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados Horse	53
4.9	Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados Soybean-small	53
4.10	Descrição dos parâmetros e valores ACO do algoritmo WFACOFS	54

Lista de Siglas e Abreviaturas

ACC	<i>Clustering Accuracy</i>
ACO	<i>Ant Colony Optimization</i>
ACS	<i>Ant Colony System</i>
AS	<i>Ant System</i>
CBA	<i>Congresso Brasileiro de Automática</i>
CRISP-DM	<i>Cross-Industry Standard Process of Data Mining</i>
EAS	<i>Elitist Ant System</i>
FER	<i>Facial Emotion Recognition</i>
FS	<i>Feature Selection</i>
GA	<i>Genetic Algorithm</i>
KNN	<i>K-Nearest Neighbors</i>
MA	<i>Memetic algorithm</i>
MAE	<i>Mean absolute error</i>
MBEGS	<i>Markov blanket-embedded Genetic Algorithm</i>
MLP	<i>Multi-layer Perceptron</i>
MMAS	<i>Max-Min Ant System</i>
NB	<i>Nayve Bayes</i>
NMI	<i>Normalized Mutual Information</i>
PSO	<i>Particle Swarm Optimization</i>

SA *Simulated Annealing*

SVD *Decomposição em Valores Singulares*

SVM *Support Vector Machine*

TML *Transportable Moisture Limit*

UFS *Unsupervised Feature Selection*

UFSACO *Unsupervised Feature Selection Ant Colony Optimization*

WFACOFS *Wrapper-Filter ACO Feature Selection*

Sumário

1	INTRODUÇÃO	13
1.1	Motivação	14
1.2	Objetivos	15
1.3	Estrutura do trabalho	16
2	REFERENCIAL TEÓRICO	17
2.1	CRISP-DM	17
2.2	Seleção de Atributos	18
2.3	Meta-Heurística	19
2.4	Seleção de Atributos Não-Supervisionados	21
2.4.1	Comparação entre métodos para Seleção de Atributos Não-Supervisionados	23
2.5	Otimização por Algoritmos de Computação Evolutiva	24
2.5.1	Otimização por Colônia de Formigas	25
2.5.2	Algoritmo geral da ACO	26
2.5.3	Primeiras variantes da ACO	29
3	MATERIAIS E MÉTODOS	31
3.1	Algoritmos ACO utilizados para Seleção de Atributos	31
3.1.1	UFSACO	33
3.1.2	WFACOFs	34
3.2	Método <i>Filter-Wrapper</i> proposto: FS_ACO	37
3.2.1	Preparação dos dados	39
3.2.2	Adaptação do <i>Ant System</i> para método de Seleção de Atributos do tipo <i>Filter - Wrapper</i>	40
3.2.3	Componente <i>Filter</i>	42
3.2.4	Componente <i>Wrapper</i>	42
3.3	Bancos de Dados	44
3.4	Metodologia	45
4	RESULTADOS E DISCUSSÕES	47
4.1	Estudo de Caso: TML	47

4.2	Análise do FS_ACO através dos classificadores MLP e KNN para diferentes Bancos de Dados	50
4.3	Análise entre FS_ACO e WFACOFS	54
5	CONCLUSÃO	58
6	SUGESTÕES PARA DESENVOLVIMENTO E DISCUSSÕES	59
	Referências Bibliográficas	60

1 INTRODUÇÃO

Os avanços na área de Tecnologia da Informação relacionados às atividades mineradoras vêm nos proporcionando oportunidades para aplicação de conceitos de análise avançada de dados no desenvolvimento de modelos preditivos como ferramenta de suporte à decisão e aumento da produtividade das operações das plantas industriais. Inseridos dentro do contexto da Indústria 4.0, grandes volumes de diferentes formatos vem sendo capturados ou gerados e armazenados, o que representa uma ótima oportunidade para transformação dos mesmos em informações que agregue valor ao negócio da mineração.

A forma mais simples de representação dos dados é realizada por meio de atributos. O tamanho do conjunto de dados pode ser medido em duas dimensões: número de atributos e número de registros, sendo que seu volume pode interferir na aplicação direta de métodos para construção de modelos que representam o conhecimento embutido nesse conjunto de dados. Torna-se necessário que eles sejam representados de maneira apropriada, processados e o modelo construído, avaliado e validado (GOLDSCHMIDT *et al.*, 2015).

Na visão geral do ciclo de vida de um projeto de Mineração de dados, contendo suas fases e tarefas relacionadas ao projeto para a aquisição do conhecimento, a etapa de Preparação dos Dados é um conjunto de atividades que envolvem organização, estruturação e transformação dos dados, precedentes fundamentais para a realização de análises e predições. Frequentemente é uma das etapas que mais consomem tempo em um projeto de análise avançada. De fato, estima-se que esta etapa consuma entre 50-70% do tempo e esforço do projeto e, ao mesmo tempo, torna-se crucial para o sucesso da etapa posterior de modelagem (CHAPMAN *et al.*, 1999).

Atributos redundantes prejudicam o desempenho do algoritmo de aprendizagem de máquina tanto na velocidade devido à dimensionalidade dos dados, quanto na taxa de acerto pois a presença de informações redundantes podem confundir o algoritmo ao invés de auxiliá-lo na busca de um modelo correto para o conhecimento (WITTEN *et al.*, 2016). Inseridos nesse contexto, uma técnica denominada Seleção de Atributos (FS - *Feature Selection*), visa selecionar, de maneira apropriada e fidedigna, as variáveis de entrada, reduzindo custo computacional e melhorando a acurácia do processo de classificação.

Os métodos de Seleção de Atributos podem ser classificados em relação à maneira de como as informações são apresentadas em um conjunto de dados. Devido ao aumento tecnológico considerável nos últimos anos, assim como à vasta quantidade de dados não-rotulados gerados em diferentes aplicações como na mineração de textos e recuperação de imagens, os métodos não-supervisionados ganharam um interesse significativo na comunidade científica (SOLORIO-FERNÁNDEZ *et al.*, 2020).

Da literatura, observa-se que algoritmos bioinspirados tem sido aplicados para o domínio de Seleção de Atributos e o seguinte projeto aborda métodos baseados na ACO - *Ant Colony Optimization* inseridos neste contexto. É apresentado um breve descritivo da evolução dos métodos

baseados na ACO, desde seu primeiro algoritmo proposto, destacando-se as tecnologias inseridas em cada abordagem. Algoritmos referenciais do estudo, aqueles dos quais se destacam nos artigos e que suas propostas elevaram significativamente os índices registrados, foram executados e seus resultados avaliados.

Em uma dessas experimentações, um artigo foi produzido a partir de um estudo de caso aplicado na indústria mineradora. Neste experimento, um algoritmo disponibilizado em *.python* foi utilizado para a redução da dimensionalidade de um banco de dados. O estudo de caso envolve a predição de uma medida de segurança crítica para o transporte de cargas a granel, incluindo finos de minério de ferro, que possuem tendências à liquefação.

Tomando como direcionamento observado na literatura, um método para Seleção de Atributos Não-Supervisionado do tipo *Filter-Wrapper* é proposto e seu desempenho avaliado através de Bancos de Dados com propriedades diferentes. Para as análises, foi considerado como função de avaliação a medida acurácia obtida através da classificação de sub-conjuntos com dimensão variada de atributos selecionados. Um breve estudo da arte do método é apresentado, destacando-se os conceitos dos quais o atribuem a classificação *Filter-Wrapper*.

Introduzido uma metodologia para se avaliar o desempenho do método proposto, é realizado uma comparação com outro algoritmo referencial e atual acerca do tema. Visando proporcionar o máximo possível de compatibilidade entre cenários, foram configurados os mesmos parâmetros possíveis ACO e seus resultados comentados.

O algoritmo desenvolvido através da linguagem *python* assim como os principais artigos e referências que contribuíram para este projeto se encontram disponíveis no GitHub do autor (AYRES). O trabalho é orientado através da metodologia de mineração de dados CRISP-DM - *Cross-Industry Standard Process of Data Mining*, Chapman *et al.* (1999), muito utilizada no meio profissional como referencial de boas práticas em projetos de Ciência de Dados.

1.1. Motivação

A motivação consiste na potencialidade da otimização de processos em operação a partir de uma seleção mais fidedigna e coesa de seus atributos de entrada. Para isto, torna-se necessário o fácil acesso, a implementação e execução de algoritmos para Seleção de Atributos em bancos com natureza distinta de dados.

Métodos de otimização podem ser utilizados no processo de seleção dos melhores atributos e nesse sentido os algoritmos bioinspirados baseados em populações e meta-heurísticas são encontrados com frequência na literatura com aplicações em diversos domínios destacando-se os algoritmos evolutivos e algoritmos de enxames. Novas abordagens considerando o algoritmo ACO para Seleção de Atributos vem apresentando números relevantes em comparação com outras abordagens na literatura levando-nos a considerar o desenvolvimento de uma proposta de algoritmo ACO do tipo *Filter-Wrapper*.

A contribuição à literatura a respeito do tema é outro fator motivacional visto de poucos artigos relacionados produzidos nacionalmente. São constantes as propostas de novos algoritmos com índices superando seus antecessores a partir de tecnologias propostas ao longo da ACO. Atualmente está evidente que o método proposto por (GHOSH *et al.*, 2019) é o que atinge os melhores índices e superá-los a partir de uma nova abordagem seria de grande estímulo e contribuição ao tema.

Cada vez mais se torna necessário a pesquisa e utilização de técnicas que reduzam a Dimensionalidade dos Dados em virtude do rápido avanço na computação e nas tecnologias da informação, mantendo-se a integridade da informação. A escalabilidade dos dados é outro desafio importante na área de estudos sobre seleção de atributos, uma vez que cada vez mais aplicações envolvem bancos de dados com grande número de amostras e atributos.

Há evidências de que estes números irão aumentar, como apresentado no artigo de Junior *et al.* (2016) do qual utiliza a otimização por colônia de formigas para detecção de anomalias em redes de telecomunicações, o autor destaca o constante crescimento do volume de dados neste setor.

A otimização por colônia de formigas segue sendo aplicada em diferentes áreas de negócios, com bons resultados e seus desdobramentos superando os índices de seus anteriores e outras otimizações convencionais na literatura.

1.2. Objetivos

O objetivo geral deste trabalho é desenvolver um novo algoritmo para Seleção de Atributos Não-Supervisionado utilizando-se o método do tipo *Filter-Wrapper*.

No alcance desse objetivo contemplam-se os seguintes objetivos específicos:

- Entendimento do funcionamento da ACO, *Ant Colony Optimization*, com suas etapas moduladas, comentadas e de fácil acesso
- Revisão bibliográfica de ACO aplicados à Seleção de Atributos, ressaltando as tecnologias introduzidas em cada.
- Execução dos algoritmos considerados referenciais no estudo a partir de diferentes Bancos de Dados e aplicações práticas.
- Construir um modelo de análise preditiva para avaliação dos subconjuntos selecionados a partir da ACO, adotando-se a medida acurácia como Função de Avaliação.
- Desenvolvimento de um algoritmo ACO de características do tipo *Filter-Wrapper* para a Seleção de Atributos Não-Supervisionados.
- Criar um repositório para o método proposto, disponibilizando além do algoritmo desenvolvido, artigos e outros algoritmos considerados referenciais do projeto.

1.3. Estrutura do trabalho

A estrutura deste trabalho apresenta a seguinte distribuição: no capítulo 1, introduz os temas relacionados à pesquisa em foco. Uma abordagem das fundamentações teóricas das tecnologias e conceitos utilizadas ao longo do estudo, como Seleção de Atributos, Otimização por Colônia de Formiga, além da descrição da metodologia de Mineração de Dados CRISP-DM é apresentado no capítulo 2. Já no capítulo 3, são apresentados os Materiais e Métodos considerados neste projeto, como os algoritmos referenciais utilizados, a descrição dos bancos de dados avaliados, o estado da arte do método *Filter-Wrapper* proposto e a descrição da metodologia adotada para avaliação. O capítulo 4 apresenta os resultados entregues nesse projeto, através de uma avaliação do desempenho do método proposto a partir de diferentes bancos de dados, do seu desempenho em relação a outro algoritmo referencial acerca do tema e um Estudo de Caso a partir da execução prática de outro algoritmo referencial. A conclusão do projeto é apresentado no capítulo 5.

2 REFERENCIAL TEÓRICO

O presente capítulo aborda brevemente as fundamentações teóricas das tecnologias e conceitos utilizadas ao longo do estudo, como Meta-Heurística, Seleção de Atributos, Otimização por Colônia de Formigas, além da descrição da metodologia de Mineração de Dados CRISP-DM de modo a respaldar a pesquisa, bem como servir de apoio para os procedimentos metodológicos inerentes ao tema proposto.

2.1. CRISP-DM

O objetivo da Mineração de Dados é descobrir o conhecimento por meio da realização de fases e tarefas dentro de um contexto que requer tomada de decisão diante de um problema (CAMILO e SILVA, 2009). Dentre as diversas metodologias de mineração de dados a CRISP-DM, um modelo padrão aberto para práticas de Ciência de Dados, tem se destacado pela ampla utilização especialmente no meio profissional.

Sua estrutura propõe auxiliar os pesquisadores desde o planejamento até a execução da mineração de dados, passando pela especificação do processo da descoberta do conhecimento até a apresentação dos resultados alcançados. De acordo com Chapman *et al.* (1999), a metodologia CRISP-DM é composta por 6 fases organizadas de maneira cíclica, cujo fluxo não é unidirecional, possibilitando ir e voltar entre as suas fases e tarefas. As fases da metodologia CRISP-DM e alguns tópicos compreendidos em cada são:

- *Business Understanding* (Entendimento do Negócio): Objetivo, análise de viabilidade, premissas, restrições, objetivo e plano do projeto.
- *Data Understanding* (Entendimento dos Dados): Aquisição inicial dos dados, análise descritiva, verificação da qualidade dos dados.
- *Data Preparation* (Preparação dos Dados): Seleção de campos, atributos e registros, limpeza e tratamento, balanceamento, integração.
- *Modeling* (Modelagem): Entendimento mais profundo da técnica, construção inicial de modelos, avaliação de modelos.
- *Evaluation* (Avaliação): Avaliação em relação aos dados iniciais, conhecer se o projeto atende aos critérios, revisão geral para implementação.
- *Deployment* (Implementação do Modelo): Prontidão operacional, operação assistida, planejamento de monitoração e manutenção, apresentação final.

O ciclo das fases e suas relações multilaterais podem ser vistos na Figura 2.1 abaixo:

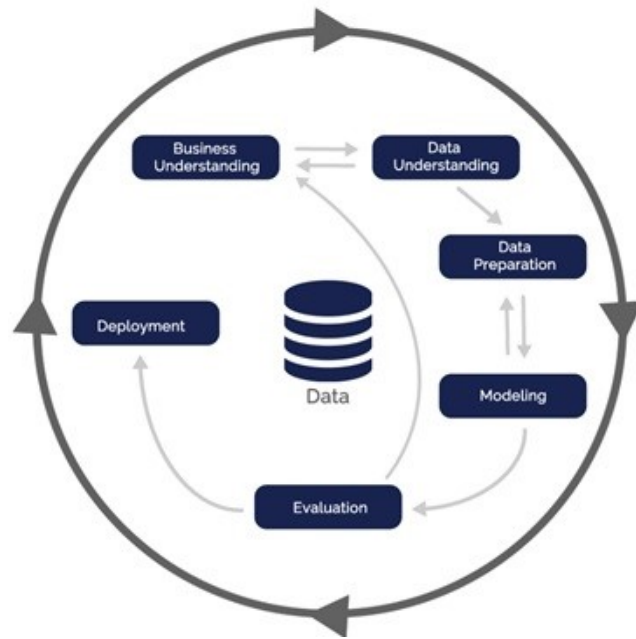


Figura 2.1: Ciclo de vida do modelo CRISP-DM. Fonte: Otaris (2018).

2.2. Seleção de Atributos

O termo Seleção de Atributos, do inglês *Feature Selection* (FS), também encontrado como “Seleção de Características”, é um dos principais e mais utilizados métodos na etapa de *Data Preparation*, seguindo a metodologia CRISP-DM. De acordo com Liu e Motoda (2002), o objetivo da FS é selecionar o subconjunto de características mais apropriado do conjunto de dados original, eliminando as características irrelevantes.

A partir do momento em que bancos de dados tornam-se complexos e volumosos, o método FS se comporta de forma a refinar as informações restringindo apenas variáveis relevantes e úteis para o processo, e consequentemente, diminuir o esforço e tempo computacional devido a redução da dimensionalidade dos dados. Dado um conjunto de variáveis de dimensão n , a FS visa encontrar um subconjunto mínimo de variáveis de dimensão m ($m < n$), adequados à representação das variáveis originais. É uma técnica amplamente utilizada e destaca-se nas áreas de *Pattern recognition* Theodoridis e Koutroumbas (2008), *Machine learning* Kotsiantis (2011), e *Data mining* García *et al.* (2015).

As quatro principais etapas de um processo de FS são ilustradas na Figura 2.2 e suas propriedades descritas como:

1. *Geração de subconjunto de variáveis*: é um processo de pesquisa heurística que resulta na seleção de um subconjunto candidato para avaliação. Ele usa estratégias de pesquisa como pesquisa completa, sequencial e aleatória para gerar subconjuntos de variáveis.
2. *Avaliação do subconjunto*: a qualidade do subconjunto gerado é aferida usando um

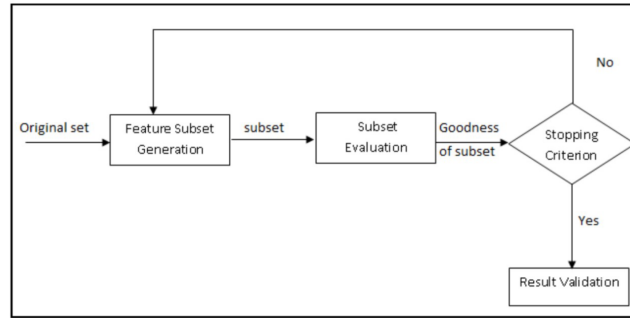


Figura 2.2: Processo de Seleção de Atributos e suas etapas. Fonte: Sutha e Tamilselvi (2015).

critério de avaliação. Se o subconjunto recém-gerado for melhor que o subconjunto anterior, ele substituirá o subconjunto anterior pelo melhor.

3. *Critério de parada*: os dois processos acima são repetidos até que o critério de parada estipulado seja alcançado.
4. *Validação de resultados*: o subconjunto final de melhores variáveis é validado por conhecimento prévio ou por testes diferentes.

2.3. Meta-Heurística

Heurística remete ao verbo da língua grega *eurisko* que significa “encontrar” ou “descobrir” do qual deu origem à palavra da língua inglesa *eureka* que é a interjeição utilizada para expressar a satisfação de se ter encontrada a solução para um problema. Algoritmos heurísticos são métodos que usam regras gerais ou abordagens de senso comum para resolver um problema. Geralmente, não se espera que os algoritmos heurísticos encontrem a melhor resposta para um problema, mas espera-se que eles encontrem soluções “próximas o suficiente” para o melhor. O termo meta-heurística é usado para descrever uma família de algoritmos heurísticos (SIMON, 2013). Por exemplo, a família de todos os algoritmos de otimização de colônia de formigas pode ser chamada de meta-heurística de colônia de formigas.

Uma meta-heurística é uma estrutura algorítmica de alto nível, independente de problemas, que fornece um conjunto de diretrizes ou estratégias para desenvolver algoritmos de otimização heurística. Exemplos notáveis de meta-heurísticas incluem os Algoritmos Genéticos e Evolutivos, Pesquisa por Tabu, Simulated Annealing, Pesquisa de Vizinhança Variável e Otimização por Colônia de Formiga (GLOVER e SÖRENSEN, 2015).

As meta-heurísticas são métodos flexíveis e por isso propiciam uma adaptação mais simples a problemas reais utilizando modelos dinâmicos (ALOISE *et al.*, 2002). Complementando, Kalra e Singh (2015) propõem que técnicas baseadas em metaheurísticas atingem boas soluções em um tempo aceitável. Para problemas que não necessariamente precisam de um ótimo global, mas de um bom resultado em pouco tempo, as metodologias meta-heurísticas são de grande

aplicabilidade.

A resolução de um problema utilizando um modelo determinístico torna-se mais complexa, ou mesmo inviável, quando a relação entre um conjunto candidato a solução e a função objetivo a ser perseguida não é matematicamente direta ou a dimensionalidade do espaço de busca é muito alta. A resolução determinística envolveria uma enumeração exaustiva dentro do espaço de busca, o que não é viável mesmo para problemas relativamente pequenos em virtude do alto esforço computacional e tempo empregado. Esse tipo de limitação dos algoritmos determinísticos abriu espaço para o crescimento dos algoritmos probabilísticos que vêm sendo desenvolvidos e estudados há cerca de 55 anos. Muitos deles são baseadas na técnica de Monte Carlo e trocam a garantia de atingimento do ótimo global por um tempo de execução mais curto (WEISE, 2009).

No campo da computação meta-heurística, os processos bio-inspirados vêm ganhando destaque por seus algoritmos serem inteligentes e adaptativos tal qual os organismos biológicos. A inteligência coletiva (*swarm intelligence*) tem se mostrado com técnicas interessantes entre as heurísticas, elas partem do conceito de que a inteligência coletiva de um grupo de indivíduos é mais do que a soma da inteligência individual. Com base nesse princípio, foram identificados uma grande variedade de algoritmos como: otimização de colônias de formigas, bando de aves, bactérias e peixes; muitos desses algoritmos são também eficientes na resolução de otimização multiobjetivo (KAR, 2016).

Há basicamente duas maneiras de abordar um problema combinatorial de maneira computacionalmente viável: uma é a construção passo a passo de uma solução usando algum tipo de informação heurística ou obtida por experiência para produzir uma solução ótima. Outra abordagem é a partir de uma solução pronta e modificar alguns dos seus elementos para se obter outras soluções melhores. Algoritmos baseados na primeira abordagem são chamados de algoritmos de construção de solução, enquanto aqueles baseados na segunda abordagem são chamados de algoritmos de busca local. Uma meta-heurística, enquanto estratégia mais geral para obter soluções ótimas a um custo razoável, pode se basear em uma das duas abordagens ou em ambas conjuntamente (CARVALHO e RAMOS). A ACO, detalhada a seguir neste projeto, é primordialmente uma meta-heurística baseada na construção de solução, embora possa ser combinada também com estratégias de busca local.

Dois conceitos importantes para o estudo de metaheurísticas são o de *intensificação* e *diversificação*. A intensificação é a exploração mais exaustiva de uma região do espaço de busca onde se espera encontrar boas soluções, ao passo que a diversificação é a mudança do foco da busca para regiões ainda não exploradas (BLUM e ROLI, 2003). Uma boa metaheurística deve obter um equilíbrio entre intensificação e diversificação.

2.4. Seleção de Atributos Não-Supervisionados

Os métodos de Seleção de Atributos podem ser classificados em relação à maneira de como as informações são apresentadas em um conjunto de dados. Os métodos supervisionados (*supervised*), necessitam que o conjunto de dados sejam rotulados para identificar e selecionar as variáveis relevantes; rótulo atribuído a cada objeto podendo ser uma categoria, um valor ordenado ou um valor real. Os métodos semi-supervisionados (*semi-supervised*) necessitam apenas que alguns objetos sejam rotulados e os métodos não-supervisionados (UFS - *Unsupervised Feature Selection*) não necessitam que o conjunto de dados seja rotulado (KOTSIANTIS, 2011).

Nas últimas décadas, muitos métodos de Seleção de Atributos foram propostos, sendo a maioria desenvolvidos para tarefas de classificação supervisionada (TANG *et al.*, 2014). No entanto, devido ao aumento tecnológico considerável nos últimos anos, assim como à vasta quantidade de dados não-rotulados gerados em diferentes aplicações como na mineração de textos, bioinformática, recuperação de imagem, mídia social e detecção de intrusos, os métodos não-supervisionados ganharam um interesse significativo na comunidade científica, (SOLORIO-FERNÁNDEZ *et al.*, 2020).

De acordo com Guyon e Elisseeff (2003); Nijima e Okuno (2008), os métodos UFS tem duas vantagens importantes:

- são imparciais e apresentam bom desempenho quando o conhecimento anterior não está disponível; não há uma rotulação prévia disponível.
- podem reduzir o risco de *overfitting* dos dados em relação aos métodos supervisionados que ocasionalmente não são capazes de lidar com uma nova classe de dados.

Da mesma forma que na Seleção de Atributos supervisionados e semi-supervisionados, os métodos de Seleção de Atributos não-supervisionados são categorizados de acordo com a estratégia adotada para selecionar as variáveis. De acordo com Dong e Liu (2018), são abordados em três categorias, *Filter*, *Wrapper* e *Embedded*:

- Método *Filter*: seleciona o subconjunto de variáveis com base em critérios intrínsecos, características dos dados, independente de um algoritmo de aprendizado. Pode ser aplicado a dados com alta dimensionalidade e suas vantagens são sua generalidade e alta eficiência computacional. Geralmente a acurácia de FS utilizando métodos *Filter* é menor do que métodos *Wrapper* devido a não presença do algoritmo de aprendizado de máquina. Alguns exemplos nesta categoria incluem o ReliefF e Medidas de Similaridade.
- Método *Wrapper*: utilizam um algoritmo de aprendizado de máquina predeterminado para avaliar o melhor subconjunto de variáveis. A precisão do algoritmo preditivo é usada como critério de avaliação. Esse método garante melhores resultados mas é computacionalmente caro para grandes conjuntos de dados. Alguns métodos populares nessa

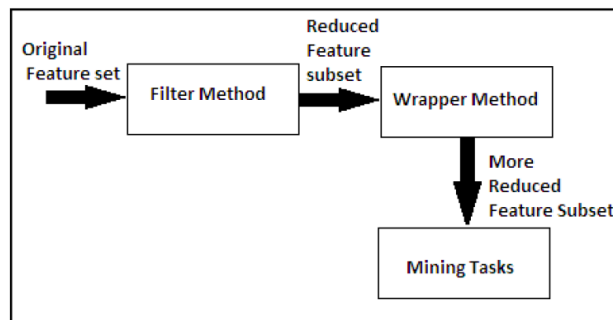


Figura 2.3: Método *Embedded*: A Seleção de Atributos requer dados de treinamento para fins de aprendizado. Fonte: Kabir *et al.* (2011).

categoria são os Algoritmos Genéticos (GA - *Genetic Algorithm*) e Otimização por Enxame (PSO - *Particle Swarm Optimization*).

- Método *Embedded*: incorpora técnicas dos métodos *Filter* e *Wrapper* para obter as vantagens de ambos. Usa uma medida independente e um algoritmo de aprendizado de máquina para medir a acurácia do subconjunto recém-gerado. Nessa abordagem apresentada na Figura 2.3, o método *Filter* é aplicado primeiro para reduzir o espaço das soluções e, em seguida, um método *Wrapper* é aplicado para obter o melhor subconjunto de variáveis. Exemplos de abordagens *Embedded* incluem Otimização por Colônia de Formigas (ACO - *Ant Colony Optimization*), Algoritmos Meméticos (MA - *Memetic algorithm*) e *Markov blanket-embedded Genetic Algorithm* - MBEGS.

Em “*A review of Unsupervised Feature Selection Methods*”, trabalho de doutorado de Solorio-Fernández *et al.* (2020), o autor descreve uma compreensível e estruturada revisão dos métodos para seleção de atributos não-supervisionados. Neste trabalho, uma avaliação dos mais recentes e destacados algoritmos não-supervisionados encontrados na literatura é apresentada, descrevendo suas vantagens e comentários. O autor propõe a taxonomia dos métodos conforme apresentado pela Figura 2.4, categorizadas de acordo com a estratégia utilizada para a seleção dos atributos e destaca os autores que contribuíram para cada grupo.

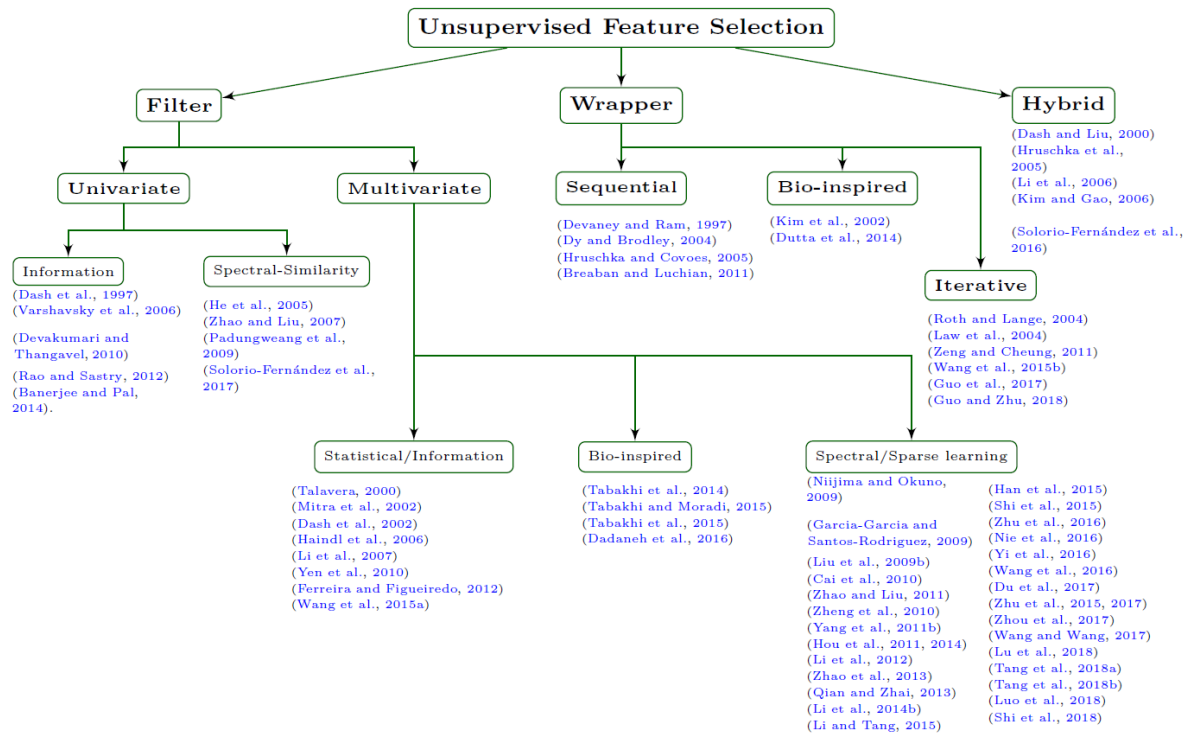


Figura 2.4: Taxonomia dos métodos de Seleção de Atributos não-supervisionados - UFS. Fonte: Solorio-Fernández *et al.* (2020).

2.4.1. Comparação entre métodos para Seleção de Atributos Não-Supervisionados

Os métodos do tipo *Filter* selecionam os atributos sem a ajuda de classificadores; portanto, é mais rápido e oferece uma precisão considerável. Os métodos do tipo *Wrapper* selecionam o atributo com a interação de um classificador definido e fornece maior precisão em predições do que métodos do tipo *Filter*, devido a presença do algoritmo de aprendizado e ajuste dos parâmetros com os classificadores. Os métodos *Embedded* também selecionam o recurso com o auxílio de um classificador e fornecem melhor precisão em comparação ao método *Wrapper*, porém não considera o problema de *overfitting* resultando pouco poder preditivo em caso de ocorrência, (MANIKANDAN e ABIRAMI, 2020). Complementando, Ghosh *et al.* (2019) cita que a tendência geral está direcionada para o desenvolvimento de métodos de características *Embedded*.

Como podemos ver na Tabela 2.1, em geral, não existe uma abordagem ou método UFS melhor para todos os tipos de dados e domínios, sendo que cada abordagem tem seus prós e contras. No entanto, podemos destacar algumas importantes características dos diferentes métodos pertencentes às diferentes abordagens e tipos (SOLORIO-FERNÁNDEZ *et al.*, 2020).

Tabela 2.1: Resumo das vantagens e desvantagens dos métodos para Seleção de Atributos Não-Supervisionados.

Abordagem	Vantagem	Nº Desvantagem
<i>Filter</i>	Rápido Escalonável Independente de algoritmos para agrupamento	Ignora interações com Algoritmos de agrupamento
<i>Wrapper</i>	Interage com algoritmos de agrupamento	Risco de <i>overfitting</i> , Alto custo computacional A seleção é específica para o algoritmo de agrupamento usado
<i>Embedded</i>	Interage com algoritmos de agrupamento Consome menos tempo que <i>Wrappers</i> Dependências dos Atributos podem ser modeladas	A seleção é específica para o algoritmo de agrupamento usado

Fonte: Solorio-Fernández *et al.* (2020).

2.5. Otimização por Algoritmos de Computação Evolutiva

A natureza vem inspirando pesquisadores a desenvolverem novos algoritmos utilizados para resolver complexos problemas de otimização. Vários elementos naturais e formas únicas de sobrevivência tem atraído pesquisadores ao longo dos anos. Fenômenos naturais como o *metal annealing* também seguem inspirando o algoritmo *Simulated Annealing* - SA, (FOGEL, 1994). Até o momento, muitos algoritmos complexos para seleção de atributos derivam-se de técnicas exclusivas de enxames, no sentido de coletar alimentos para a sobrevivência ou escapar de predadores. Esses algoritmos são comumente chamados de Algoritmos de Otimização por Inteligência de Enxame e alguns de seus exemplos são o ACO e PSO, conforme cita Ghosh *et al.* (2019).

Recentemente, o comportamento dos animais tem inspirado alguns algoritmos meta-heurísticos e Ghosh *et al.* (2019) conclui que uma variedade dessas abordagens tem sido utilizadas para realizar a seleção de atributos.

O princípio fundamental desses algoritmos se baseia na utilização de um método construtivo para a obtenção da população inicial (soluções factíveis iniciais) e uma técnica de busca local para melhorar a solução da população, considerando que os indivíduos (soluções) dessa população evoluem de acordo com regras especificadas que consideram o intercâmbio de informações entre os indivíduos. Este processo conduz a população em direção à obtenção da solução ótima (OLARIU e ZOMAYA, 2005).

Os algoritmos de inspiração biológica são bem conhecidos por sua aplicabilidade a pro-

blemas de otimização em vários domínios. Cada indivíduo representa uma solução candidata ao problema, e o algoritmo converge para a solução ideal por meio das interações evolutivas dos indivíduos no espaço das soluções (SOLORIO-FERNÁNDEZ *et al.*, 2020).

2.5.1. Otimização por Colônia de Formigas

A meta-heurística Otimização por Colônia de Formigas (ACO – Ant Colony Optimization) é baseada em um processo de construção de solução inspirado no comportamento coletivo de formigas reais para solucionar inúmeros problemas de otimização (DORIGO *et al.*, 1996). A ideia é imitar o comportamento das formigas ao procurarem o caminho mais curto entre o formigueiro e uma fonte de alimentos. O ACO foi proposto por Marco Dorigo e colegas, em 1992, em sua tese de doutorado.

Foi observado que, na vida real, as formigas depositam no solo uma substância, chamada feromônio, ao longo do caminho percorrido entre o formigueiro e uma fonte de alimento. Quando encontra o alimento, ela faz o caminho de volta para o ninho reforçando trilha de feromônios, garantindo que outras formigas também sigam a trilha. Desta forma, as formigas seguintes tendem a ser atraídas pelo feromônio depositado pela formiga anterior. Também se observa, na vida real, que o feromônio sofre os processos de acumulação, quando uma nova formiga percorre o mesmo caminho, e evaporação, ao longo do tempo. Ao acabar o alimento a trilha deixa de ser reforçada até que ela não seja mais perceptível (DORIGO e STÜTZLE, 2003). Essa comunicação indireta entre as formigas, e também em outros animais, é conhecida como stigmergia e refere-se a um sistema emergente auto organizado a partir da modificação do ambiente local pelos indivíduos (MULLEN *et al.*, 2009).

Na meta-heurística ACO, formigas virtuais, implementadas sob forma de agentes em um programa de computador, simulam o comportamento das formigas no mundo real e a quantidade de feromônio acumulada em cada trilha é decisiva para a escolha do caminho a ser seguido por cada formiga da colônia: quanto mais feromônio a trilha contiver maior será a probabilidade da mesma ser seguida por uma nova formiga que vá fazer o caminho do formigueiro até a fonte de alimento (DORIGO *et al.*, 2006). Como as formigas, ao seguir as trilhas, continuam depositando feromônio nas mesmas, com o passar do tempo as formigas tendem a seguir caminhos mais curtos (o sistema converge para a solução ótima) devido a um maior trânsito de formigas e consequentemente um maior acúmulo de feromônio, conforme exemplificado na Figura 2.5.

Na ACO, as formigas são consideradas procedimentos estocásticos e constroem os subconjuntos de variáveis, iterativamente, usando tanto as informações heurísticas quanto o quantidade de feromônio acumulada nas trilhas. O componente estocástico traz uma exploração mais completa do espaço das soluções e cria uma grande variedade de subconjuntos em comparação com uma heurística gulosa. A estratégia de busca de formigas é remanescente do aprendizado por reforço (DORIGO e STÜTZLE, 2019).

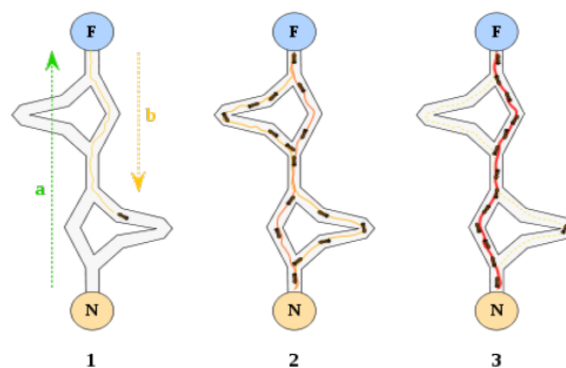


Figura 2.5: Formigas convergem para o caminho mais curto. Fonte: Gutjahr e Rauner (2007).

A ACO se destaca como um dos mais bem sucedidos exemplos de Inteligência Coletiva, proposto inicialmente para a resolução do conhecido problema do Caixeiro Viajante e com aplicações para Seleção de Atributos em diversos domínios. O algoritmo já foi aprovado para outros problemas, como o conhecido “problema da mochila” e também em diversas aplicações na área de telecomunicações (BEDI e SINGH, 2013).

Resumidamente, a ACO apresenta como principais vantagens:

- Robustez: mesmo quando um ou mais indivíduos falham, a colônia continua a executar suas tarefas.
- Flexibilidade: a colônia tem a capacidade de se adaptar rapidamente a mudanças externas e internas.
- Auto-organização: colônia requer relativamente pouca supervisão ou controle

2.5.2. Algoritmo geral da ACO

O termo “Otimização por Colônia de Formigas - ACO”, é um termo genérico para os algoritmos baseados no comportamento de formigas reais utilizado para resolver problemas de otimização discreta de natureza estática ou dinâmica (DORIGO e STÜTZLE, 2003). Como caracterizada por Dorigo, seu pseudo-código pode ser formulado conforme abaixo:

O ACO é um algoritmo estocástico de busca que possui dois tipos possíveis de convergência: a convergência em valor e a convergência em solução. Quando se trata de problemas que possuem mais de uma solução ótima, o objetivo do algoritmo será convergir para qualquer um deles. Portanto, a convergência em valor avalia a probabilidade de que o algoritmo gere uma solução ótima pelo menos uma vez. Todavia, a convergência em solução analisa a probabilidade de que o algoritmo atinja um estado que continua gerando a mesma solução ideal.

Algorithm 1 Meta-heurística ACO

: Iniciar parâmetros

: Iniciar rastros do feromônio

1: **procedure** : **AGENDAR ATIVIDADES**

2: Construção de Soluções

3: Ações Globais [Opcional]

4: Atualização dos Feromônios

Fim do agendamento

Embora a convergência em solução aparente ser mais desejada que a convergência de valor, na otimização só é necessário encontrar a solução ideal uma vez, já que após encontrada o problema é resolvido e o algoritmo pode ser interrompido; isso faz com que a convergência de valor seja suficiente para o sucesso da otimização (DORIGO e STÜTZLE, 2003).

Conforme mencionado anteriormente, a ACO é uma meta-heurística baseada em construção de soluções e também uma meta-heurística baseada em populações, no caso, na cooperação entre formigas. A partir do pseudo-código acima, nota-se que a ACO é dividido basicamente em três etapas.

Na primeira fase, as formigas da colônia constroem passo a passo uma solução, aplicando uma decisão local estocástica com base na informação de feromônio e na informação heurística. É importante notar que não é necessário que as formigas caminhem em sincronia na construção da solução, o que facilita implementações concorrentes e assíncronas da ACO (DORIGO e STÜTZLE, 2003). Enquanto caminha, construindo uma solução, a formiga avalia a solução parcial e, dependendo do algoritmo ACO, pode vir a depositar feromônio do último componente visitado, cooperando, assim, com as formigas que vierem em seguida por aquele caminho.

Depois que uma formiga completa uma solução, algumas ações globais podem ser realizadas sobre essa solução. Essa fase é opcional. A idéia é um agente com conhecimento global realize sobre uma solução ações que formigas individuais não teriam condições de fazer. Por exemplo, um agente global pode observar o caminho de cada formiga e resolver depositar feromônio extra nos componentes usados pela formiga que construiu a melhor solução (DORIGO e STÜTZLE, 2003). Outro procedimento muito comum usado nesta fase é a aplicação de uma busca local sobre as soluções construídas na primeira etapa, obtendo assim um efeito maior de intensificação. De acordo com Stützle e Hoos (2000), usar a busca local nesta etapa é uma forma de hibridizar o ACO e as melhores implementações da ACO geralmente se valem desse recurso.

Na última etapa, temos a etapa de atualização dos feromônios. Essa etapa envolve tanto o depósito quanto a evaporação de feromônio. As formigas podem depositar feromônio após cada passo da construção, o que acontece na fase de construção da solução. No entanto, a formiga também pode depositar feromônio após construir uma solução completa, o que pode ser feito nesta terceira fase do algoritmo. Além disso, para evitar uma convergência muito precoce da

ACO, ou que ele fique preso em um ótimo local, o feromônio depositado deve evaporar ao longo do tempo. Essa atividade também é realizada neste etapa e através dela obtemos um efeito de diversificação.

Problemas de otimização combinatória tratados através da ACO são normalmente codificados baseado no conceito de *grafo de construção* completo.

Adotando-se um problema de otimização combinatória representado através da Eq. 2.1:

$$F(x) \mapsto \max, x \in S. \quad (2.1)$$

Onde x é uma solução para o problema de otimização combinatória, S é um conjunto finito de soluções possíveis e F é a Função de Avaliação, também chamada de *fitness function*. Alguns problemas podem apresentar restrições, e para estes casos, o conjunto S de soluções deve satisfazer todas as fornecidas pelo problema (GUTJAHR, 2007). O *grafo de construção* a partir da Eq. 2.1 pode ser representado através de um *grafo dirigido* completo dado por:

$$C = (V, A) \quad (2.2)$$

onde V representa o conjunto de nós componentes das soluções e A as conexões, conjunto de arestas. Construir uma solução significa encontrar um caminho viável em C . Por exemplo, no problema do caixeiro viajante, os nós correspondem aos clientes, as arestas correspondem às ruas que conectam os clientes e uma solução viável é um caminho *Hamiltoniano* no grafo, ou seja, de menor distância/custo total (GUTJAHR, 2007). O grafo na Figura 2.6 representa uma construção para um problema assimétrico do caixeiro viajante para $n = 5$ cidades.

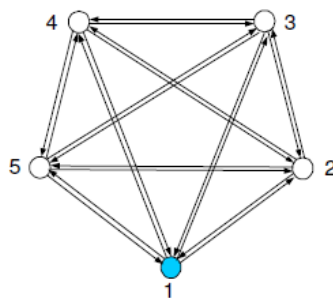


Figura 2.6: Grafo de construção assimétrico para problema do caixeiro viajante. Fonte: Gutjahr (2007).

O próximo capítulo apresenta as primeiras variações da ACO. A partir dos comentários, evidencia que a principal diferença entre eles geralmente recai sobre quando fazem o incremento do feromônio e como o fazem. A compreensão dessas primeiras versões torna-se crucial para a formulação de novos métodos. Atualmente outras propostas são encontradas e constantemente apresentado novas abordagens tentando superar os índices de seus antecessores.

2.5.3. Primeiras variantes da ACO

A partir da primeira proposta baseado em colônias de formigas, de Dorigo (1992), foram sugeridos tecnologias a partir da metodologia e aplicada com sucesso para a resolução do problema do caixeiro viajante. A princípio, foram propostas três diferentes versões do AS (*Ant System*): *ant-density* (Colorni, Dorigo e Maniezzo, 1992); *ant-quantity* (Colorni, Dorigo e Maniezzo, 1992); e *ant-cycle*. (Dorigo, 1992). As primeiras duas versões atualizavam o feromônio logo após cada passo dentro da iteração, enquanto, na última versão (*ant-cycle*), a quantidade de feromônio só era atualizada depois que todas as formigas construíam a resposta, ou seja, ao final de cada iteração. Atualmente quando se refere ao AS, remete-se a versão *ant-cycle* que se demonstrou mais vantajosa que as outras duas (DORIGO e STÜTZLE, 2003).

A primeira variante do AS original foi o EAS - *Elitist Ant System*, o qual valoriza ainda mais os melhores resultados através de um incremento extra na atualização dos feromônios no percurso responsável pelo melhor resultado (*best-so-far*), como se fosse uma formiga extra no algoritmo. Essa formiga adiciona uma quantidade de feromônio proporcional ao adicionado pelas outras por um fator “e” que define o peso desse incremento (DORIGO e STÜTZLE, 2003).

Na variante AS-*rank*, cada formiga deposita uma quantidade de feromônio que é inversamente proporcional a sua classificação no *ranking* de melhores resultados da função objetivo; empates no *ranking* podem ser resolvidos aleatoriamente. Em cada iteração, somente as melhores formigas do *ranking* e a formiga que produziu a melhor FO até agora dentre todas as iterações, depositam feromônio; essa última sempre contribui com a maior quantidade. Na solução do problema do caixeiro viajante, entre outros, um número adequado de formigas que depositam feromônio em cada iteração é de 6 (DORIGO e STÜTZLE, 2003).

Outra variante do AS, muito estudada e adaptada, é o MMAS - *Max-Min Ant System*, da qual apenas a melhor formiga (*best-so-far*) ou a melhor da iteração deposita o feromônio, aumentando o risco de estagnação em mínimos locais. Esse efeito é contornado limitando o valor máximo de feromônios depositado em cada trecho. O valor inicial dos feromônios é o limite superior permitido e, se o algoritmo alcança estagnação, nenhuma resposta melhor é encontrada em um certo número de iterações, ele é reinicializado (DORIGO e STÜTZLE, 2003).

Uma diferença crucial entre as variantes corresponde ao modo de construção de soluções, que pode ser em paralela ou sequencial. Na implementação paralela, a cada passo da construção da solução, todas as formigas realizam sua decisão e escolhem para qual nó seguir. Enquanto na implementação sequencial, apenas após uma formiga encontrar uma solução completa, a próxima começa a construir sua solução. Se tratando de AS, os dois modos de construção da solução acarretam em resultados bastante similares, enquanto que, em outros algoritmos ACO, como o ACS - *Ant Colony System* por exemplo, isso não ocorre (DORIGO e STÜTZLE, 2003).

O ACS é um tipo de ACO que possui três principais diferenças em relação ao AS (DORIGO e STÜTZLE, 2003):

- O ACS valoriza mais a informação de experiências acumuladas pelas formigas do que a AS, através do uso de uma regra de escolha mais agressiva.
- A evaporação e o depósito de feromônios ocorrem apenas nos arcos pertencentes à melhor rota até o momento.
- Quando uma formiga utiliza um arco, ela remove parte do feromônio do arco para aumentar a exploração de caminhos alternativos para as outras formigas desta iteração. Consequentemente, o algoritmo nunca converge para apenas um resultado.

3 MATERIAIS E MÉTODOS

Apresentam-se neste capítulo alguns relevantes algoritmos baseados na ACO para Seleção de Atributos em ordem cronológica, destacando-se suas tecnologias desenvolvidas em cada proposta. Dentre os detalhados a seguir, nota-se na literatura relacionado ao tema da pesquisa um destaque a uma das primeiras abordagens não-supervisionadas ACO propostas para Seleção de Atributos, o algoritmo do tipo *Filter* denominado UFSACO, proposto por Tabakhi *et al.* (2014). O algoritmo WFACOFS - *Wrapper-Filter ACO Feature Selection*, Ghosh *et al.* (2019) também chama atenção, pois além de ser um aprimoramento do UFSACO, comprovado experimentalmente seu melhor desempenho, apresenta algumas tecnologias que foram consideradas ao longo da pesquisa para implementação. Outra motivação referente ao algoritmo de Ghosh consiste na inovadora abordagem do tipo *Filter-Wrapper* proposto para ACO. Ambos algoritmos destacados foram utilizados para avaliar o desempenho do método proposto neste trabalho, detalhado no capítulo 4. Em seguida, um breve estudo da arte do método proposto é descrito, apresentando-se seu fluxograma, equações e conceitos do qual o atribui a classificação do tipo *Filter-Wrapper*. Os bancos de dados que estiveram disponíveis para análises ao longo do projeto são categorizados e encerra-se descrevendo a metodologia adotada para avaliar o algoritmo proposto. O código do algoritmo proposto assim como outros utilizados como referência e artigos que contribuíram para este projeto encontram-se disponíveis no *Github* do autor Ayres.

3.1. Algoritmos ACO utilizados para Seleção de Atributos

Da literatura, observa-se que algoritmos baseados em ACO tem sido aplicado para o domínio de Seleção de Atributos, superando muitos métodos contemporâneos em termos de acurácia e redução da dimensionalidade, (GHOSH *et al.*, 2019). Fallahzadeh *et al.* (2018) utiliza uma abordagem ACO para determinar o sub-conjunto de atributos extraído de imagens pré-processadas para classificação de câncer de mama. Sweetlin *et al.* (2018) utiliza a partir de imagens de tomografia computadorizada para diagnóstico de doenças pulmonares.

Embora Seleção de Atributos seja um domínio de aplicações importantes para ACO, outros trabalhos também tem focado em outros domínios. Em Yin *et al.* (2017), ACO é utilizado para controle de rejeição de auto-perturbação do motor de indução. Já Sabino *et al.* (2010) propõe ACO aplicado ao planejamento de manobras de locomotivas em pátios ferroviários. Outra aplicação ACO encontrada é no ramo de *e-commerce* para sistemas de recomendação Parvin *et al.* (2019), utilizado para aproximação de público-alvo. Também encontrado aplicações no domínio da economia, onde Uthayakumar *et al.* (2020) propõe uma abordagem ACO utilizada para previsão de crises financeiras. A tabela 3.1 apresenta os algoritmos baseados em ACO para Seleção de Atributos mais relevantes na literatura e um breve descritivo das tecnologias inovadoras propostas por cada um.

É notado que a maioria das versões existentes da ACO para Seleção de Atributos são

Tabela 3.1: Exemplos mais destacados de algoritmos baseados em ACO para Seleção de Atributos.

Algoritmo/Autor/Ano	Informações relevantes
AS Ant System Dorigo <i>et al.</i> (1996)	Introduziu para a escolha do caminho uma função probabilística envolvendo a distância dos caminhos e a quantidade de feromônio
ACS Ant Colony System Gambardella e Dorigo (1996)	Modelo adaptado do AS para resolução do problema caixeiro viajante Apenas formiga “best-so-far” deposita feromônio
MMAS MAX-MIN Ant System Stützle e Hoos (2000)	Exploração por método guloso. Somente as melhores formigas atualizam o feromônio, que possuem um valor min e max
TFSACO Text Feature Selection ACO Aghdam <i>et al.</i> (2009)	As regras de atualização global e local alteram de acordo com nível de feromônio dos caminhos após cada iteração
TSIACO Two-Stage Invariant ACO Zhang e Feng (2012)	Atualização do feromônio é realizada em 2 estágios
UFSACO Unsupervised Feature Selection ACO Tabakhi <i>et al.</i> (2014)	Primeiro método Não-Supervisionado proposto Seleciona sub-conjuntos baseado na similaridade entre os atributos. O feromônio é atualizado com base na frequência dos atributos selecionados
FS based on the graph clustering and ACO Moradi e Rostami (2015)	Agrupam os atributos usando o método Louvain para detecção de comunidades A similaridade é calculada através do coeficiente de correlação de Pearson
C-RACOFS Capability enRiched ACO Forsati <i>et al.</i> (2014)	A atualização do feromônio local previne que o algoritmo convirja prematuramente Formulações diminuem os níveis de feromonio nos nós percorridos
ACOFs Feature Selection ACO Kabir <i>et al.</i> (2011)	Modelo híbrido, combina as tecnologias do tipo <i>Filter</i> e <i>Wrapper</i> . Incorpora o IG do atributo no cálculo da informação eurística e avalia subconjuntos após adicionar novo atributo

Fonte: Solorio-Fernández *et al.* (2020), Ghosh *et al.* (2019).

Algoritmo/Autor/Ano	Informações relevantes
WFACOFs Wrapper-Filter ACO Feature Selection Ghosh <i>et al.</i> (2019)	Modelo híbrido, A regra de atualização do feromônio é balanceada [0,1] após cada iteração. Introduzido o conceito de <i>fitness-based memory</i>

Fonte: Solorio-Fernández *et al.* (2020), Ghosh *et al.* (2019).

do tipo *Filter*, entretanto literaturas recentes evidenciam uma convergência para o desenvolvimento de modelos híbridos, contendo as características de ambos.

3.1.1. UFSACO

O algoritmo *Unsupervised Feature Selection Ant Colony Optimization* foi sugerido por Tabakhi *et al.* (2014) sendo um dos primeiros métodos não-supervisionados baseados em ACO propostos para Seleção de Atributos. Atualmente encontram-se variações deste mesmo algoritmo como proposto pelo mesmo autor em Tabakhi e Moradi (2015) e modelos híbridos como em Ghosh *et al.* (2019).

Seu principal objetivo é selecionar subconjuntos com baixa similaridade entre os atributos (baixa redundância). O espaço das soluções é representado por um grafo completo, não direcionado com pesos onde os nós representam as variáveis do problema inicial e os pesos das arestas as semelhanças entre cada par destas variáveis. Essa similaridade é calculada usando a função de similaridade de cossenos. Tabakhi propõe que se duas variáveis são semelhantes, logo elas são redundantes para o sistema. Cada nó (atributo), conforme ilustrado na Figura 3.1 está associado a uma quantidade de feromônio, a qual é atualizada pelos agentes (formigas) em função de seu valor atual a uma taxa de decaimento pré-especificada e o número de vezes que uma determinada variável é selecionada por um agente. As formigas percorrem o grafo dando preferência a altos valores de feromônios e baixas similaridades até que um critério de parada pré-especificado (número de iterações) seja alcançado. Ao final do processo, aquelas variáveis com maior incidência de feromônio serão as que trazem mais informações ao sistema, possibilitando descartar as demais. Portanto espera-se escolher subconjuntos de variáveis com baixa redundância, (SOLORIO-FERNÁNDEZ *et al.*, 2020).

O algoritmo UFSACO é disponibilizado dentre um dos algoritmos presentes na *Kurdistan Feature Selection Tool*, KFST (2020), um código aberto desenvolvido em JAVA para Seleção de Atributos em diferentes áreas de pesquisas. Uma aplicação prática do algoritmo UFSACO é descrita conforme artigo “Estudo de Caso: TML”, detalhado no capítulo 4. Na ocasião deste experimento, sua aplicação possibilitou converter o estudo teórico em informações relevantes para suporte de decisão na indústria mineradora.

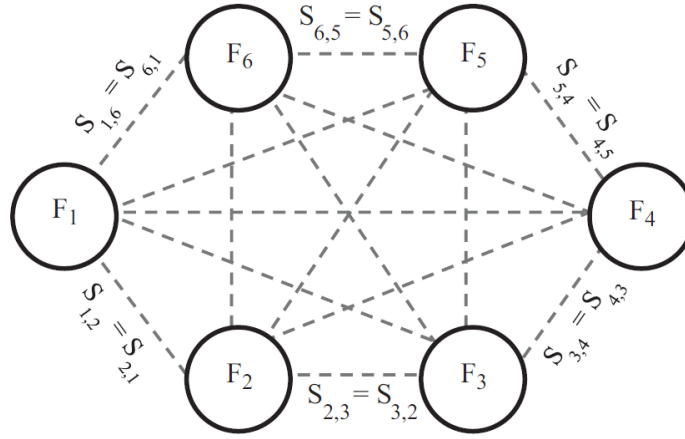


Figura 3.1: Representação gráfica de um problema de Seleção de Atributos onde $S_{i,j}$ representa a similaridade associada a cada nó entre as variáveis i e j . Em outras palavras, $S_{i,j} = \text{sim}(F_i, F_j)$. Fonte: Tabakhi *et al.* (2014).

O pseudo-código do método de seleção de atributos UFSACO e suas propriedades é apresentado na página seguinte.

3.1.2. WFACOFs

O algoritmo WFACOFs, *Wrapper-Filter ACO Feature Selection*, do tipo híbrido proposto por Ghosh *et al.* (2019) foi implementado de maneira a combinar as melhores vantagens dos métodos do tipo *Filter* e *Wrapper*. Para o seu desenvolvimento foi considerado como base os algoritmos já citados UFSACO e TFSACO, propondo técnicas de sucesso para contornar as deficiências observadas em cada. Ao final do artigo, uma exaustiva análise de desempenho entre algoritmos ACO utilizando bandos de dados UCI de vários tamanhos é apresentado onde nota-se um destaque para os índices alcançados pelo WFACOFs, principalmente nas análises em relação ao esforço computacional onde observam-se tempos significativos menores que seus anteriores.

Em um ACO, as formigas movem-se através de um grafo conectado depositando feromônio ao longo dos caminhos percorridos. Essa abordagem sofre uma limitação a partir de que mesmos subconjuntos percorridos podem ser construídos percorrendo caminhos diferentes. Por exemplo, como mostrado na Fig 3.2, considerando os nós (atributos) $\{A, B, C, D, E\}$, o caminho do qual está incluso estes nós pode ser construído a partir de $\{A \rightarrow B \rightarrow C \rightarrow D \rightarrow E\}$ ou $\{A \rightarrow C \rightarrow D \rightarrow B \rightarrow E\}$ dentre outros. Isso significa que os depósitos de feromônios não ocorrem corretamente, sendo distribuído de maneira desigual. Para superar este problema, é proposto que o depósito do feromônio ocorra no nó (atributo), ao invés do caminho (aresta) ou quina entre os nós, (GHOSH *et al.*, 2019).

Algorithm 2 Unsupervised Feature Selection based on Ant Colony Optimization (UFSACO)

Entrada X : $p \times n$ matriz, de dimensão n e p amostras

$m(\leq n)$: número de variáveis a serem mantidas no sub-conjunto final.

NC_{max} : Número máximo de ciclos de repetição do algoritmo.

N_{ant} : Define o número de agentes (formigas).

NF : Número de variáveis selecionadas por cada agente em cada ciclo.

ρ : Define a taxa de decaimento do feromônio sobre cada variável.

@sim: Função que calcula a similaridade entre variáveis.

Saída \tilde{X} : $p \times m$ matriz, conjunto de treinamento de dimensionalidade reduzida.

1: **procedure** : UFSACO

2: *Aplica* @sim para calcular a similaridade $S_{i,j}$ entre variáveis

3: $\tau_i(1) = c, \forall i = 1..n$, feromônio inicial = constante

4: **for** $t = 1$ to NC_{max} **do**

5: $FC[i] = 0, \forall i = 1..n$, *Define* valor inicial do contador de variáveis = 0

6: *Posiciona* os agentes aleatoriamente no nós do grafo.

7: **for** $i = 1$ to NF **do**

8: **for** $k = 1$ to N_{ant} **do**

9: *Escolhe* a próxima variável f não visitada conforme regra proporcional pseudo-aleatória

10: *Move* o k -ésimo agente para a nova variável f selecionada

11: $FC[f] = FC[f] + 1$, atualiza o contador associado à variável f

12: **end for**

13: **end for**

14: $\tau_i(t+1) = (1 - \rho)\tau_i(t) + \frac{FC[i]}{\sum_{j=1}^n FC[j]}$; $\forall i = 1..n$, regra de atualização global

15: **end for**

16: *Classifica* as variáveis por ordem decrescente de seus feromônios (τ_i)

17: *Constrói* o sub-conjunto \tilde{X} a partir de X selecionando as m variáveis com mais feromônio

Fim

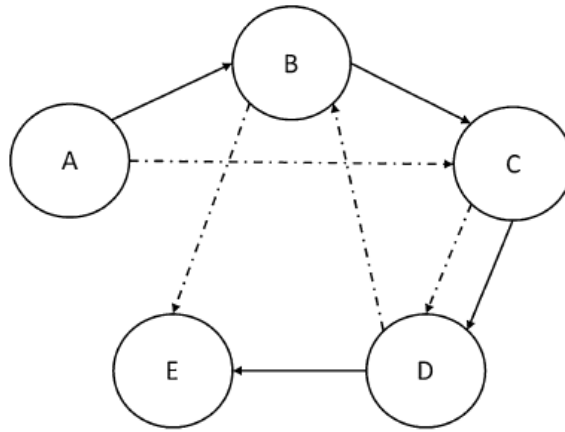


Figura 3.2: Representação gráfica de múltiplo caminho através de grafo. Dois caminhos, conforme ilustrado em negrito e tracejado, possuem o mesmo sub-conjunto $\{A, B, C, D, E\}$. Fonte: Ghosh *et al.* (2019).

Nos algoritmos TFSACO e UFSACO, onde a atualização do feromônio é realizada globalmente e localmente porém o nível de feromônio não é limitado, é observado um enviesamento na seleção de atributos para aqueles escolhidos mais vezes devido ao seu alto nível de feromônio depositado, atrapalhando na exploração de todos os demais atributos. Para isto, o algoritmo propõe uma normalização do nível de feromônio $[0,1]$ após cada iteração, (GHOSH *et al.*, 2019).

A falta de memória para armazenar as melhores soluções é um problema observado nos algoritmos TFSACO e UFSACO. Especificamente esta tecnologia, relacionada à memória, foi considerada ao longo da pesquisa como sugestão de implementação após contato com o algoritmo de Tabakhi utilizado no Estudo de Caso. Para solucionar isto, é introduzido de acordo com o autor o que ele chamou de “*Fitness-based memory*”, permitindo salvar os melhores sub-conjuntos de atributos em relação à acurácia obtida na classificação e número de atributos (em cada sub-conjunto) presentes em cada iteração. O fluxograma do algoritmo WFACOFs proposto é descrito através da Figura 3.3.

O conceito de similaridade entre 2 atributos, apresentado inicialmente no algoritmo UFSACO, foi mantido como cálculo. De acordo com o autor, o uso da similaridade ajuda a diminuir o esforço computacional em grande medida. O valor da similaridade é igual a “1” se os atributos estiverem completamente correlacionados e “0” caso os atributos são completamente independentes. O algoritmo adota característica do tipo *Filter* para a criação de sub-conjuntos de atributos e através de um método *Wrapper* mede sua qualidade para atualização dos valores de feromônio.

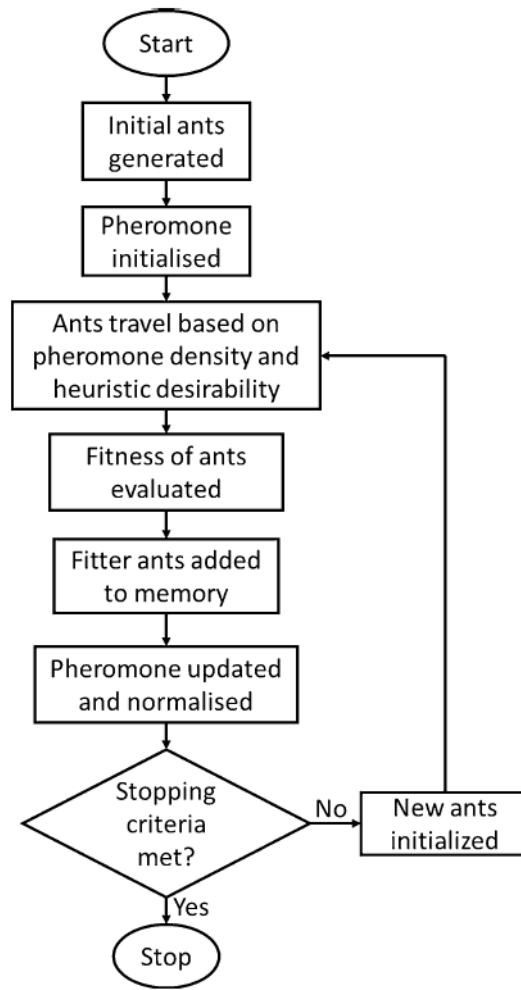


Figura 3.3: Fluxograma WFACOFS. Fonte: Ghosh *et al.* (2019).

O algoritmo WFACOFS é disponibilizado no Github do autor, (GHOSH, 2021), desenvolvido no formato *.mat*. É o algoritmo para Seleção de Atributos com melhores índices na literatura se tratando da ACO e foi utilizado para comparações do desempenho do método proposto neste trabalho.

3.2. Método *Filter-Wrapper* proposto: FS_ACO

Inicialmente proposto para resolução de problemas relacionados ao caixeiro-viajante, a ACO pode ser aplicada também para o domínio de seleção de atributos, encontrando-se na literatura índices que superam outros métodos contemporâneos em termos de acurácia, taxa de erro de classificação e redução da dimensionalidade. No artigo de Rosa (2017), é apresentada uma comparação entre o tema da pesquisa com o método SVD - *Decomposição em Valores Singulares*. Kanan *et al.* (2007) apresenta um estudo no campo de sistemas de reconhecimento facial do qual leva em consideração o algoritmo GA nas análises, incluindo a ACO, para a seleção de atributos. Já mencionado anteriormente, Tabakhi *et al.* (2014) em seu artigo considera em suas

análises métodos como *Information gain*, *Fischer score*, *Laplacian score* dentre outros para avaliar o desempenho de seu algoritmo proposto.

O método proposto neste projeto baseia-se no desenvolvimento de um algoritmo de seleção de atributos do tipo *Filter-Wrapper* baseado na ACO, observando-se a tendência de recentes literaturas propostas na área. Foi adotado como estrutura de código para a otimização por colônia de formigas o proposto no algoritmo *Ant Systems* e a partir dele realizado as alterações de forma que o método apresente características do tipo *Filter* e *Wrapper*.

Por questão de simplicidade, foi adotado o pseudônimo FS_ACO (Feature Selection based on Ant Colony Optmization) para o método. O fluxograma do algoritmo é descrito através da Figura 3.4. Para a avaliação do desempenho do algoritmo, foi considerado análises em banco de dados da vida real classificados pelo seu tamanho, amostras e número de atributos, e comparações com outros algoritmos referenciais de mesma área.

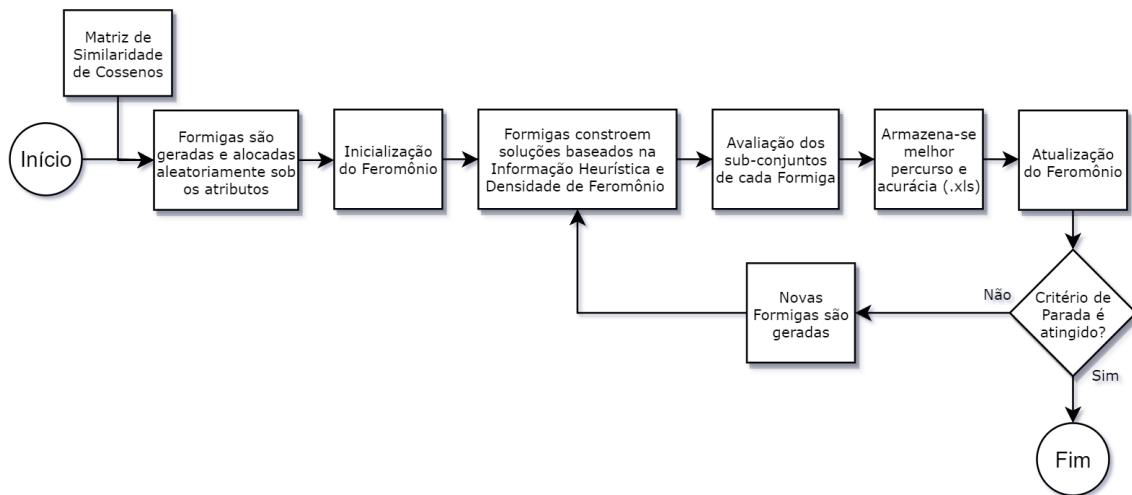


Figura 3.4: Fluxograma FS_ACO. Fonte: O autor.

O algoritmo foi desenvolvido adotando-se a linguagem Python 3.8 com o auxílio de ferramentas disponibilizadas pelas bibliotecas relacionadas à Preparação de Dados, Estatísticas e Machine Learning conforme descritos na Tabela 3.2.

A metaheurística ACO pode ser considerada uma melhoria do processo de construção de solução, buscando obter, pela influência dos feromônios, um efeito tanto de diversificação quanto de intensificação. O processo de construção de solução no ACO é estocástico. Ele constroi uma solução iterativamente adicionando componentes de solução a uma solução parcial levando em consideração a informação heurística e informação de feromônio, que muda dinamicamente para refletir a experiência da formiga (DORIGO e STÜTZLE, 2003).

Tabela 3.2: Bibliotecas e funções Python utilizadas no algoritmo FS_ACO.

Biblioteca	Função	Domínio
<i>NumPy</i>		Preparação dos Dados
<i>Pandas</i>		Preparação dos Dados
<i>sklearn.model_selection</i>	<code>train_test_split</code>	Machine Learning
<i>sklearn.metrics</i>	<code>accuracy_score</code>	Classificação multi-rótulo
<i>scipy</i>	<code>spatial.distance.cosine</code>	Funções de correlação
<i>sklearn.neighbors</i>	<code>KNeighborsClassifier</code>	Classificadores
<i>sklearn.neural_network</i>	<code>MLPClassifier</code>	Classificadores

Fonte: O autor.

3.2.1. Preparação dos dados

Antes do processo de Seleção de Atributos iniciar, o espaço de busca deve estar representado através de um grafo unidirecional totalmente conectado $G = \langle F, E \rangle$, onde $F = \{F_1, F_2, \dots, F_n\}$ representa o conjunto de atributos do qual cada atributo representa um nó no grafo e $E = \{(F_i, F_j) : F_i, F_j \in F\}$ representa o conjunto de arestas do grafo. O peso da aresta $(F_i, F_j) \in E$ é configurado para assumir o valor da similaridade entre F_i e F_j .

Conforme mencionado, na etapa de Preparação dos Dados é adotado o conceito de similaridade de cossenos, apresentado inicialmente pelo algoritmo UFSACO, Tabakhi *et al.* (2014). Uma matriz de similaridade de cossenos é processada levando em consideração pares de atributos de acordo com a Equação 3.1:

$$sim(X, Y) = \frac{\sum_{i=1}^a X_i * Y_i}{\sum_{i=1}^a X_i^2 * \sum_{i=1}^a Y_i^2}; \quad (3.1)$$

aonde X e Y representam 2 atributos p -dimensionais, vetores $(X = x_1, x_2, \dots, x_p$ e $Y = y_1, y_2, \dots, y_p)$. De acordo com a equação, é esperado que o valor da similaridade encontrado esteja compreendido entre $[0 - 1]$.

Tabakhi *et al.* (2014) diz que o uso da similaridade ajuda a diminuir o esforço computacional em grande medida. A matriz de similaridade para o banco de dados *Wine* do repositório UCI, Dua e Graff (2017), com 13 atributos é exibida na Figura 3.5.

Quanto mais próximo de “1”, mais correlacionados são os respectivos atributos. Logo, tende a ser escolhido ao longo da ACO aqueles atributos com baixos valores de correlação, uma vez que pares de atributos altamente correlacionados não contribuirão de mesma maneira para

o sistema (SOLORIO-FERNÁNDEZ *et al.*, 2020).

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1	0.904514	0.992594	0.979658	0.992133	0.966025	0.890448	0.940864	0.93984	0.918957	0.968621	0.961775	0.935841
1	0.904514	1	0.91446	0.914049	0.890767	0.847711	0.739721	0.89388	0.838772	0.86703	0.852456	0.839847	0.828573
2	0.992594	0.91446	1	0.985493	0.990135	0.95788	0.852235	0.946272	0.930389	0.913873	0.960946	0.950186	0.92991
3	0.979658	0.914049	0.985493	1	0.975503	0.932763	0.853739	0.95158	0.909001	0.899622	0.947042	0.937961	0.901121
4	0.992133	0.890767	0.990135	0.975503	1	0.96462	0.898453	0.9313	0.94068	0.911421	0.963842	0.958199	0.937889
5	0.966025	0.847711	0.95788	0.932763	0.96462	1	0.925143	0.878085	0.956063	0.880203	0.963845	0.976468	0.909129
6	0.890448	0.739721	0.852235	0.853739	0.898453	0.925143	1	0.700934	0.906046	0.80595	0.884228	0.925523	0.851205
7	0.940864	0.89388	0.946272	0.95158	0.9313	0.878085	0.700934	1	0.833879	0.860014	0.892522	0.865307	0.869832
8	0.93984	0.838772	0.930389	0.909001	0.94068	0.956063	0.906046	0.833879	1	0.849093	0.928263	0.94856	0.883343
9	0.918957	0.86703	0.913873	0.899622	0.911421	0.880203	0.80595	0.860014	0.849093	1	0.857727	0.845226	0.885621
10	0.968621	0.852456	0.960946	0.947042	0.963842	0.963845	0.884228	0.892522	0.928263	0.857727	1	0.972952	0.891476
11	0.961775	0.839847	0.950186	0.937961	0.958199	0.976468	0.925523	0.865307	0.94856	0.845226	0.972952	1	0.891921
12	0.935841	0.828573	0.92991	0.901121	0.937889	0.909129	0.851205	0.869832	0.883343	0.885621	0.891476	0.891921	1

Figura 3.5: Matriz de similaridade para banco de dados *Wine*. Fonte: O autor.

3.2.2. Adaptação do *Ant System* para método de Seleção de Atributos do tipo *Filter - Wrapper*

Ant System - AS foi o primeiro exemplo de algoritmo ACO, desenvolvido por Marco Dorigo, Vittorio Maniezzo e Alberto Colorni em 1996, Dorigo *et al.* (1996). Desde então formulações a partir desta proposta foram apresentadas mantendo sua estruturação conforme apresentado no pseudo-código Meta-heurística ACO. É usado como ponto de partida para a formulações de outros algoritmos ACO (DORIGO e STÜTZLE, 2003). No desenvolvimento do FS_ACO, o Ant System foi projetado de forma a considerar como peso o valor da similaridade de cossenos entre os atributos ao invés da distância entre cidades, originalmente proposto para tratativas relacionados ao caixeiro viajante.

Na fase de construção, cada formiga escolhe qual atributo ir com uma probabilidade que é uma função do valor da similaridade de cossenos entre os atributos e da quantidade de feromônio presente na aresta que conecta a esse atributo. Para não permitir passos inválidos, a formiga tem uma memória de quais atributos ela já visitou. Assim, enquanto constrói uma solução, a formiga é forçada a não visitar um mesmo atributo duas vezes até que a solução esteja completa. Quando a formiga termina um caminho completo, ela deposita uma certa quantidade de feromônio em cada aresta (i, j) que visitou, (DORIGO *et al.*, 1996). No AS, não há depósito de feromônio durante a fase de construção de solução. Ao dar um passo, a formiga calcula a probabilidade de todos os atributos que ela ainda não visitou e escolhe ir para aquele de maior probabilidade. Essa probabilidade é definida pela equação 3.2:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \quad (3.2)$$

onde α e β são parâmetros para indicar respectivamente a importância do feromônio e da informação heurística. τ_{ij} é a quantidade de feromônio na aresta ij , η_{ij} é o inverso da similaridade de cossenos e N_i^k o conjunto dos atributos ainda não visitados pela formiga k .

O AS não realiza nenhuma ação global, nem busca local e isso foi mantido nessa primeira proposta apresentada neste projeto. Logo, a próxima fase, após a construção da solução, é a atualização do feromônio, que envolve tanto o incremento do feromônio quanto a sua evaporação e é realizado por todas as formigas.

A atualização do feromônio no FS_ACO é definida pela equação 3.3:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k \quad (3.3)$$

onde ρ é um parâmetro para regular a taxa de evaporação do feromônio, m o número de formigas e $\Delta \tau_{ij}^k$ é definida por:

$$\begin{cases} \frac{1}{L^k}, & \text{se aresta } ij \text{ faz parte do caminho da formiga } k \\ 0, & \text{caso contrário} \end{cases}$$

onde L^k é o custo do caminho percorrido pela formiga k .

Para se obter um bom desempenho, alguns parâmetros devem ser ajustados. Além do número de atributos a serem selecionados informado na execução do algoritmo, um parâmetro para controlar a importância do feromônio, outro para a informação heurística, um quarto para regular a taxa de evaporação e um quinto para indicar o tamanho da população de formigas. O ajuste correto desses parâmetros é fundamental para se obter um bom equilíbrio entre intensificação e diversificação. Como por exemplo mencionado por Dorigo, ρ deve ser ajustado para um valor menor que 1, pois, do contrário, haverá uma acumulação ilimitada do feromônio, prejudicando a diversificação. A sugestão do autor é algo em torno de 0.5. Já para α , deve ser maior que 0. Quando o seu valor é alto, isso significa que o feromônio é muito importante e as formigas tendem a escolher arestas escolhidas por outras formigas no passado, apresentando um efeito de intensificação. Se o valor for baixo, a construção se comporta de maneira semelhante a uma construção gulosa aleatória, (DORIGO *et al.*, 1996). Entretanto, valores altos demais podem provocar a estagnação precoce do algoritmo. Dorigo propõe que α deve ficar em torno de 1. Um valor muito baixo de β provocaria estagnação precoce do algoritmo e um valor alto demais o aproxima de uma construção gulosa. O ajuste deste parâmetro é importante para se obter uma boa diversificação. Dorigo propõe que β fique em torno de 5. Geralmente, $m = n$, isto é, a quantidade de formigas é igual a quantidade de atributos. Por fim, Dorigo propõe inicializar o feromônio de todas as arestas por $\frac{1}{n \cdot L_{nn}}$ onde L_{nn} é o custo de uma construção puramente gulosa.

Abaixo é exibido o prompt de saída do FS_ACO para uma seleção de 8 atributos com os parâmetros ACO ajustados diante de um banco de dados com 13 atributos.

1 Informacoes do Bando de Dados (Amostras, Atributos): (177, 13)

```

3  Selecao de Atributos baseado na Otimizacao por Colonia de Formigas:
4  Projetado para selecionar atributos de determinado BD atraves da ACO adotando
5  a similaridade de cosseno entre pares de atributos como peso. A performance dos
6  subsets (acuracia) atraves de uma modelagem sera avaliada e ao final apresentado o
7  conjunto de maior valor. A atualizacao do feromonio e regra de probabilidade foram
8  desenvolvidas conforme algoritmo Ant-System
9  _____
10 Parametros ACO:
11 Numero de Formigas: 13
12 Taxa de evaporacao: 0.5
13 Alpha Heuristic(importancia do feromonio): 1
14 Beta Heuristic(importancia da informacao heuristica): 5
15 Numero de Iteracoes: 15
16 Numero de Atributos a serem selecionados: 8
17 _____
18 Solucao(sub-set) de atributos que apresentaram maior acuracia ao longo de 15 \
   iteracoes:
19 9 -> 6 -> 11 -> 8 -> 10 -> 7 -> 1 -> 5 | Acuracia: 0.73333
20 _____
21 Tempo de execucao do codigo: 0:00:16.187000

```

3.2.3. Componente *Filter*

Ao considerar um elemento estatístico, a similaridade de cossenos entre pares de atributos para o cálculo da probabilidade, o componente $[\eta_{ij}]$ na equação 3.2, denota um conceito do qual o método possa ser considerado do tipo *Filter*.

A informação heurística η_i , citado em artigos como “*heuristic desirability*” é calculado conforme Equação 3.4 e possui objetivo de incluir atributos não correlacionados visando aumentar a capacidade de reconhecimento do sistema.

$$\eta_i = \frac{1}{1 + \text{sim}(i, j)} \quad (3.4)$$

Os valores de similaridade de cossenos são processados multiplas vezes ao longo da otimização. Como são calculadas previamente, auxilia a reduzir a complexidade do método consideravelmente e tornar eficiente o método de seleção da próxima formiga (GHOSH *et al.*, 2019).

O algoritmo adota característica do tipo *Filter* para a criação de sub-conjuntos de atributos e através de um método *Wrapper* mede sua qualidade para atualização dos valores de feromônio e seleção de atributos.

3.2.4. Componente *Wrapper*

Ao considerar uma modelagem para avaliar a qualidade do sub-conjunto de atributos selecionados por cada formiga ao longo da ACO, denota um conceito do qual o método possa ser considerado do tipo *Wrapper*.

Ao fim de uma iteração, cada formiga apresenta uma lista de soluções, construída através do método *Filter*, contendo os atributos selecionados a partir de seu ponto de partida e da

quantidade informada de atributos a serem selecionados. Este sub conjunto é avaliado através de uma modelagem de dados, sendo a métrica acurácia sua função de avaliação.

Afim de realizar análises do desempenho do algoritmo FS_ACO, foram considerados 2 classificadores para esta etapa do método, o KNN - *K-Nearest Neighbors* e o MLP - *Multi-layer Perceptron*.

O KNN é um algoritmo bem conhecido para classificar objetos em relação aos seus vizinhos mais próximos. Sua resposta é uma classe de vizinhança obtida através de votos. Para determinar a classe de um elemento que não pertença ao conjunto de treinamento, o classificador KNN procura K elementos do conjunto de treinamento que estejam mais próximos deste elemento desconhecido, que tenham a menor distância. Estes K elementos são chamados de K -vizinhos mais próximos. Verifica-se quais são as classes desses K vizinhos e a classe mais freqüente será atribuída à classe do elemento desconhecido. É um classificador muito popular devido a sua simplicidade e eficiência ao mesmo tempo (GHOSH *et al.*, 2019).

Sobre o MLP, é um classificador popularmente utilizado e eficiente. É uma rede neural artificial de alimentação que consiste em um conjunto de três redes de camadas. As camadas formam um grafo conectado e são atribuídos aleatoriamente pesos que são modificados durante o treinamento utilizando o algoritmo de aprendizado *back-propagation* (GHOSH *et al.*, 2019).

Diante das propriedades dos classificadores utilizados, os resultados apenas foram possíveis devido à baixa complexidade do algoritmo proposto. Abaixo é exibido parte do script relacionado à etapa de classificação dos dados através do classificador MLP e seus parâmetros:

```
1 #DIVIDINDO-SE ENTRE TREINAMENTO / TESTE
2 X_train, X_test, y_train, y_test = train_test_split(banco_dados, target, test_size=
    =0.30, random_state=42)
3
4 #CLASSIFICACAO REALIZADA ATRAVES DO CLASSIFICADOR MLP, ADOTANDO-SE AS COLUNAS \
    SELECIONADAS NO SUBSET(NUM_FS)
5 #accuracy=[] # Cria uma lista vazia
6 for x in range(self.num_formigas): # Quantidade de formigas
7     #Create a MPL Classifier
8     model = MLPClassifier(hidden_layer_sizes=(30,30), max_iter=30)
9     #model = KNeighborsClassifier(n_neighbors=5) #Utilizado N = 5, padrao adotado
10    model.fit(X_train.iloc[:, Lista_FS[x]], y_train) # Aprender a base de treino, \
        usar variaveis TREINOS
11    #Predict the response for test dataset
12    y_pred = model.predict(X_test.iloc[:, Lista_FS[x]]) # Preve a base de teste
13    # Model Accuracy: how often is the classifier correct?
14    # Atualiza a SOLUCAO E ACURACIA encontrada pela formiga
15    self.formigas[x].setSolucao(Lista_FS[x], accuracy_score(y_test, y_pred)) #\
        Acuracia entre 2 variaveis
```

3.3. Bancos de Dados

Para a avaliação do desempenho e comparação do FS_ACO, foram selecionados diferentes bancos de dados frequentemente citados na literatura, disponibilizados através do repositório UCI, Dua e Graff (2017), NIPS2003, Isabelle Guyon (2003), e o banco de dados utilizado no Estudo de Caso. Diferentes tipos de campos são observados nos bancos e suas amostras variadas. A Tabela 3.3 descreve as propriedades dos bancos de dados utilizados e foram categorizados de acordo com a quantidade de atributos:

1. Número de Atributos < 15 .

- Vowel
- Wine
- Breast Cancer

2. $15 \leq$ Número de Atributos ≤ 100 .

- Zoo
- Lymphography
- TML
- Horse
- Soybean-small

Bancos de Dados com número de atributos acima de 100 requerem uma análise de seu desempenho em relação ao tempo de processamento e geralmente são bancos com características de dados provenientes de atividades como Reconhecimento de Emoção por Expressão Facial, *Facial Emotion Recognition* (FER) e Classificação de Câncer através de Microarray Datasets.

Geralmente, arquivos de bancos de dados são mantidos em seus repositórios através do formato *.dat*, que são mais difíceis de prepará-los. Os bancos de dados neste projeto foram transformados para os formatos *.csv* e *.mat* na intenção de facilitar seu acesso e implementação na etapa de Preparação dos Dados dos algoritmos. A maioria das linguagens de programação possuem bibliotecas para processar arquivos do tipo *.csv*. O formato *.mat* se deve ao fato do algoritmo de Ghosh *et al.* (2019), considerado para análises, ter sido desenvolvido no software *Matlab*. Cada banco possui um conjunto de atributos sendo que cada coluna representa um atributo, exceto a última, da qual representa a sua resposta, *class label*.

Tabela 3.3: Descrição dos Bancos de Dados utilizados nas análises

Banco de Dados	Amostras	Nº Classe	Nº Atributos
Vowel	528	11	10
Wine	178	3	13
Breast Cancer	699	2	10
Zoo	101	7	17
Lymphography	148	4	18
Horse	368	2	27
Soybean-small	46	4	35
TML	72	11	80

Fonte: O autor.

3.4. Metodologia

Atualmente um dos principais meios para avaliar os resultados de uma Seleção de Atributos não-supervisionados é através da avaliação em termos da qualidade do sub-conjunto de dados selecionados através de um classificador. Esta avaliação é a mais amplamente utilizada, e tornou-se a forma mais aceite para avaliar métodos de seleção de atributos não-supervisionados. Dentro deste tipo de avaliação, distinguem-se duas maneiras padrões, (GHOSH *et al.*, 2019):

1. Avaliação da acurácia da classificação ou da taxa de erro médio por classificadores supervisionados como KNN, SVM, *Nayve Bayes* (NB) entre outros. Este tipo de avaliação é mais utilizado para métodos como *Spectral Feature Selection* e *Bio-inspired*
2. Avaliação através de classificadores de agrupamento, como por exemplo K-means. Para avaliar a qualidade da agregação, medidas como *Normalized Mutual Information* (NMI) e *Clustering Accuracy* (ACC) são normalmente utilizadas.

A metodologia adotada para avaliar o desempenho do método proposto consiste no registo da média da função de avaliação obtida ao longo de 15 execuções do algoritmo, em relação à dimensão de atributos informados a serem selecionados. O número de execuções informado baseou-se no observado em experimentos descritos nos artigos considerados referenciais para este projeto e estão disponíveis no *Github* do autor.

Além da acurácia, adotado como função de avaliação, o desvio padrão da média de acurácia, uma medida de dispersão de um conjunto de dados é considerado. Nesta primeira etapa, da qual é avaliado exclusivamente o desempenho do método, é realizado através dos di-

ferentes bancos de dados abordados no capítulo anterior. O tempo médio das execuções não foi considerado em decorrência do pequeno tempo de processamento apresentado para os bancos de dados utilizados nestas análises.

Para o desempenho do método proposto em relação a outro algoritmo, foi considerado para comparação o proposto por *Ghosh*, uma vez que atualmente na literatura acerca do tema da pesquisa é o que apresenta melhores índices. Análises simulando as mesmas condições, tais como número de atributos selecionados, parâmetros ACO e bancos de dados foram avaliados e seus desempenhos comparados.

Em cada iteração, os bancos de dados foram divididos em 70% dos dados para treinamento e 30% para dados de teste. Todos os experimentos neste trabalho foram implementados nas linguagens *Python* e *Matlab*, e processados através de um Intel Core-i7 CPU com 8 GB de RAM. As análises são apresentadas no capítulo 4 adiante.

4 RESULTADOS E DISCUSSÕES

Neste capítulo apresentam-se os resultados gerados neste trabalho. Após revisão bibliográfica e contato com os primeiros algoritmos para seleção de atributos, com foco maior naqueles dos quais consideram em sua estruturação de código a otimização por colônia de formigas, um estudo de caso a partir de um banco de dados real foi realizado. Na ocasião, a aplicação prática do algoritmo UFSACO, Tabakhi *et al.* (2014), resultou no artigo intitulado “Seleção de Variáveis Baseado no Algoritmo Otimização Colônia de Formigas: Estudo de Caso na Indústria de Mineração”, do qual foi apresentado no XXIII *Congresso Brasileiro de Automática* (CBA 2020) é descrito.

Na sequência, uma avaliação do desempenho do método proposto a partir de bancos de dados, frequentemente citados na literatura e com propriedades diferentes, é abordado. Após, é analisado seu desempenho em relação a outros algoritmos referenciais da área, adotando-se os mesmos parâmetros ACO de configuração.

Finaliza-se o capítulo com alguns comentários e sugestões para trabalhos futuros. O código do FS_ACO.py assim como os principais algoritmos, artigos e referências que contribuíram para este projeto encontram-se disponibilizados no *GitHub* do autor, (AYRES).

4.1. Estudo de Caso: TML

Dentre uma das implementações de algoritmos para seleção de atributos relacionados ao tema da pesquisa, o programa proporcionou uma aplicação prática da qual resultou no artigo descrito a seguir. Neste experimento, um algoritmo disponibilizado em *.python* foi utilizado para a seleção de atributos e seu desempenho comparado com o sub-conjunto de mesma dimensão selecionado através do nó *Feature Selection* do software IBM SPSSModeler. Foi considerado o Erro Médio Absoluto (MAE - *Mean absolute error*) para a avaliação de desempenho dos cenários. A metodologia de dados CRISP-DM auxiliou nessa experimentação visto que o software mencionado assume em sua estruturação esta metodologia para a análise dos dados.

A possibilidade de aplicação prática converte a teoria em informações relevantes para a tomada de decisões do negócio.

“Seleção de Variáveis Baseado no Algoritmo Otimização Colônia de Formigas: Estudo de Caso na Indústria de Mineração”

O presente artigo foca na aplicação, em um processo da indústria de mineração, do algoritmo denominado UFSACO (*An unsupervised feature selection algorithm based on ant colony optimization*), Tabakhi *et al.* (2014), um algoritmo de enxame pertencente a classe Otimização Colônia de Formigas (*ACO - Ant Colony Optimization*) para a seleção de variáveis. A aplicação da tecnologia no estudo de caso envolve a predição da medida de segurança crítica Limite de Umidade Transportável, TML (*Transportable Moisture Limit*) para o transporte de cargas a

granel, incluindo finos de minério de ferro, que possuem tendências à liquefação colocando em risco as embarcações. Nos últimos anos foram observados acidentes com vítimas fatais atribuído a ocorrência deste fenômeno tornando-se então esta medida de segurança uma realidade regulatória no dia a dia das operações.

Anualmente milhões de toneladas de minério de ferro são transportadas por via marítima impulsionadas principalmente pela alta demanda vivenciado nos últimos anos. Estas cargas, materiais sólidos a granel contendo umidade, podem estar sujeitos a rupturas, deslizamentos e liquefação. A ocorrência desses fenômenos colocam em risco as embarcações sendo que nos últimos 30 anos ocorreram pelo menos 24 acidentes marítimos tendo como causa atribuída a liquefação da carga, somando mais de 177 vítimas fatais.

A Organização Marítima Internacional (IMO – *International Maritime Organization*), órgão que regulamenta o transporte marítimo de cargas, estabelece critérios para segurança das operações e relacionados ao grupo de cargas que apresentam riscos de liquefação, incluso finos de minério de ferro, é estabelecido uma medida de segurança crítica denominada Limite de Umidade Transportável. Esta medida, informação da qual a IMO exige que seja informada ao comandante do navio antes de iniciar o carregamento, consiste no limite máximo de umidade da qual a carga pode ser transportada com segurança. Caso a carga esteja com umidade igual ou superior ao TML não deve ser embarcada, a não ser que o navio seja especialmente projetado para transporte de cargas liquefeitas, Ferreira *et al.* (2017).

Dentro desse contexto, o estudo de caso proposto neste artigo apresenta uma aplicação na indústria da mineração através de uma FS de um banco de dados real com informações relacionadas à medição do TML em pátios e navios. A maioria das variáveis são categorizadas em grupos no que diz respeito de suas propriedades de granulometria, umidade, produtos e análise química.

Um banco de dados contendo 88 variáveis foi analisado em um primeiro momento. Depois de uma análise da qualidade dos dados, foram eliminados 7 variáveis que apresentavam valores nulos em mais de 70% dos registros, totalizando 81 variáveis a serem utilizadas, sendo parte delas categóricas e parte nominais.

Inicialmente, foi realizada uma seleção de variáveis utilizando o software IBM SPSS Modeler versão 18.1 através do nó *feature selection* que utiliza um método *filter*. Seus critérios de seleção são baseados na qualidade e na importância dos dados. O resultado deste processo foi a seleção de 21 variáveis. Adotamos essa referência de variáveis selecionadas na intenção de comparar com o sub-conjunto de variáveis de mesma quantidade processados a partir do algoritmo UFSACO.

A simulação do algoritmo UFSACO foi executada em Python 2.7.11 com o auxílio da biblioteca *python-weka-wrapper* instalada. As configurações dos parâmetros com suas propriedades descritas no pseudo-código do capítulo 3.1.1 foram:

$Nants = 81$, definido para ser o mesmo da quantidade de variáveis do Banco de Dados original
 $NC = 50$, Número de ciclos, iterações internas do algoritmo

$NF = 21$, Número de variáveis a serem selecionadas

$\rho = 0.2$, Taxa de decaimento, relativo a evaporação do feromônio

$\beta = 1$, Parâmetro usado para controle de importância entre *feromônio* x *similaridade*, ($\beta > 0$)

$\mathcal{T} = 0.2$, valor de feromônio inicial

$q0 = 0.7$, coeficiente [Exploration-Exploitation], um número real no intervalo de [0, 1].

Parâmetro que define a regra de transição de estado, entre uma busca gulosa ou probabilística.

O objetivo da via probabilística é evitar ficar preso em um local ótimo. A combinação de ambos são chamadas de “regra proporcional pseudo-aleatória”.

Diante dos procedimentos e tecnologias descritas anteriormente, chegamos ao total de 2 cenários de momento, que são sub-conjuntos processados a partir de diferentes FS conforme detalhado na Tabela 4.1. Também foi inserido para efeitos de comparação de desempenho o cenário 1 sem FS, ou seja, todas as variáveis foram consideradas na etapa de modelagem. A coluna ‘cor’ corresponde aos respectivos cenários apresentados no Gráfico 1 a seguir.

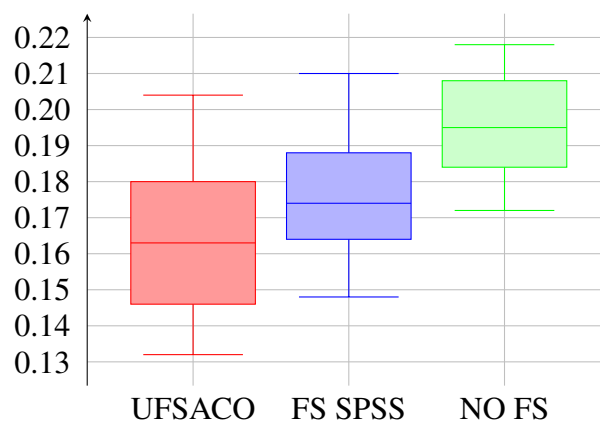
Tabela 4.1: Cenários obtidos após execuções

Cenário	Algoritmo FS	Cor
1	-	Verde
2	SPSS Modeler	Azul
3	UFSACO	Vermelho

Fonte: O autor.

Na intenção de medir o desempenho de cada um desses cenários, foram modelados a cada conjunto uma rede neural e executado 10 vezes a modelagem afim de se obter o Erro Médio Absoluto em relação à predição assumindo como target o índice TML. O desempenho dos cenários podem ser avaliados através do gráfico 1:

Gráfico 1: Cenários x Erro Médio Absoluto (MAE)



Fonte: O autor.

A partir do experimento realizado, foi observado que o sub-conjunto de variáveis selecionadas a partir do algoritmo em foco apresentou o menor Erro Médio Absoluto após modelagem através de uma rede neural se comparado com os demais cenários, incluindo o próprio FS do software de análise avançada utilizado. Diante de um banco de dados com 81 variáveis, reduzimos essa dimensão para 21 e comprovou-se dentre os cenários que as mais susceptíveis a carregarem informações intrínseca do sistema foram as variáveis selecionadas a partir do cenário 3. A qualidade da predição realizada com a rede neural atende os requisitos de negócio ao passo que sem o FS ou com o FS do SPSS Modeler o resultado não fica dentro dos requisitos estabelecidos para o MAE. Atualmente constam na literatura adaptações do algoritmo em estudo, como por exemplo em Ghosh *et al.* (2019) que propõe um método *embedded* demonstrando resultados contundentes. Segue como sugestão para estudos futuros a análise de algoritmos atuais baseados no método *embedded* com bom desempenho e sua implementação incorporando as técnicas de mais sucesso. A aplicação prática do experimento possibilitou converter o estudo teórico em informações para suporte à decisão que realmente agreguem valor aos processos da indústria de mineração.

4.2. Análise do FS_ACO através dos classificadores MLP e KNN para diferentes Bancos de Dados

A partir da metodologia descrita no capítulo 3.4, a seguir é apresentado as tabelas para avaliação do desempenho do método FS_ACO proposto neste trabalho. As análises foram categorizadas de acordo com as propriedades dos bancos de dados e avaliadas adotando-se a média da acurácia % obtidas através da classificação de sub-conjunto com dimensões variadas de atributos selecionados ao longo de 15 execuções. As tabelas apresentam a quantidade de atributos selecionados e sua performance, destacando-se em negrito o melhor resultado de acurácia obtido entre os dois classificadores para cada dimensão avaliada.

É importante salientar que, por se tratar de um método de seleção de atributos não-supervisionado, os rótulos dos atributos foram removidos para o processo de seleção de atributos e apenas atributos numéricos são considerados. Os campos não numéricos foram transformados em numerais, mapeando suas categorias e representando-as através de números inteiros. A Tabela 4.2 apresenta a descrição dos parâmetros e valores ACO configurados para as análises.

As Tabelas 4.3, 4.4 e 4.5 referem-se à categoria de Banco de Dados com Número de Atributos < 15 . Nesta categoria, é observado melhores índices através da classificação KNN, mediante um maior valor do desvio padrão. Para a classificação através do MLP, o desvio padrão é baixo, apresentando resultados mais coesos, porém seus resultados para o banco *Vowel* foram consideráveis baixos. Os valores obtidos em relação ao banco *Breast Cancer*, para ambas as classificações, são considerados baixos. Este banco de dados em específico apresenta apenas 2 classes como resposta. O tempo de processamento para cada um destes bancos de dados, ao longo das 15 execuções, não ultrapassaram 2 minutos no total.

Tabela 4.2: Descrição dos parâmetros e valores ACO.

Símbolo	Descrição do parâmetro	Valor
m	Quantidade de Formigas	Número de Atributos
α	Importância do Feromônio	1
β	Importância da Informação Heurística	5
Iterações	Número de Iterações	15
ρ	Taxa de Evaporação	0.5

Fonte: O autor.

Tabela 4.3: Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados *Wine*.

Classificadores:						
Nº Atributos Selecionados	KNN			MLP		
	Acurácia (%)	Melhor Acurácia (%)	Desvio Padrão (%)	Acurácia (%)	Melhor Acurácia (%)	Desvio Padrão (%)
4	81.234	88.867	0.5318	75.259	88.889	0.1027
6	80.687	92.592	0.4159	65.962	83.344	0.1410
8	79.166	91.358	0.7897	68.518	87.037	0.1642

Fonte: O autor.

Tabela 4.4: Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados *Vowel*.

Classificadores:						
Nº Atributos Selecionados	KNN			MLP		
	Acurácia (%)	Melhor Acurácia (%)	Desvio Padrão (%)	Acurácia (%)	Melhor Acurácia (%)	Desvio Padrão (%)
4	70.701	80.471	0.3178	33.670	36.700	0.0092
6	80.695	83.867	0.1468	37.374	41.414	0.0236
8	76.703	86.532	0.5934	39.393	43.771	0.0277

Fonte: O autor.

Tabela 4.5: Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados **Breast Cancer**.

Classificadores:						
Nº Atributos Selecionados	KNN			MLP		
	Acurácia (%)	Melhor Acurácia (%)	Desvio Padrão (%)	Acurácia (%)	Melhor Acurácia (%)	Desvio Padrão (%)
4	60.259	61.276	0.3547	61.111	64.285	0.0643
6	61.524	62.251	0.4785	62.428	66.567	0.0873
8	61.572	62.857	0.4268	63.464	64.814	0.0876

Fonte: O autor.

As Tabelas 4.6, 4.7, 4.8 e 4.9 referem-se à categoria de banco de dados com Número de Atributos entre 15 e 100. Nesta categoria, os melhores índices foram observados em ambas classificações, com destaque ao obtido através do MLP para o banco *Soybean-small*, registrando uma acurácia de 100% para uma seleção de 20 atributos. Para estes casos, deve ser considerado o risco de *overfitting* dos dados. Os resultados apresentados através da classificação KNN para o banco de dados *Zoo* também são significativos. Embora o banco de dados *Horse* apresente apenas 2 classes como resposta, possui um número significativo de atributos, amostras e seus resultados foram satisfatórios. O tempo de processamento para cada um destes bancos de dados, ao longo das 15 execuções, não ultrapassaram 4 minutos no total.

Tabela 4.6: Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados **Zoo**.

Classificadores:						
Nº Atributos Selecionados	KNN			MLP		
	Acurácia (%)	Melhor Acurácia (%)	Desvio Padrão (%)	Acurácia (%)	Melhor Acurácia (%)	Desvio Padrão (%)
5	84.205	95.643	0.0629	61.322	64.514	0.0720
8	88.258	96.774	0.0328	64.516	67.741	0.0278
11	87.096	90.322	0.0469	67.741	70.967	0.0430

Fonte: O autor.

De acordo com os resultados, nota-se um melhor desempenho para os bancos de dados compreendidos com números de atributos entre 15 e 100. Observa-se também que o desempenho do algoritmo para o banco de dados com pequeno número de classes e amostras é baixo.

Tabela 4.7: Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados *Lymphography*.

Nº Atributos Selecionados	Classificadores:					
	KNN			MLP		
	Acurácia (%)	Melhor Acuracia (%)	Desvio Padrão (%)	Acurácia (%)	Melhor Acurácia (%)	Desvio Padrão (%)
6	69.764	77.778	0.3768	62.637	73.353	0.6467
9	70.864	75.555	0.1954	62.222	73.912	0.3278
12	73.263	77.331	0.3486	65.057	84.444	0.7728

Fonte: O autor.

Tabela 4.8: Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados *Horse*.

Nº Atributos Selecionados	Classificadores:					
	KNN			MLP		
	Acurácia (%)	Melhor Acuracia (%)	Desvio Padrão (%)	Acurácia (%)	Melhor Acurácia (%)	Desvio Padrão (%)
7	73.669	76.576	0.4253	63.603	69.369	0.2457
12	75.963	86.486	0.3796	70.297	91.891	0.5914
17	74.259	75.675	0.4159	68.082	73.873	0.3260

Fonte: O autor.

Tabela 4.9: Valores da média de Acurácia% do método proposto, após 15 execuções, obtidas da classificação dos sub-conjuntos de atributos selecionados para o banco de dados *Soybean-small*.

Nº Atributos Selecionados	Classificadores:					
	KNN			MLP		
	Acurácia (%)	Melhor Acuracia (%)	Desvio Padrão (%)	Acurácia (%)	Melhor Acurácia (%)	Desvio Padrão (%)
10	71.428	78.571	0.2457	76.728	85.714	0.0803
15	76.428	92.857	0.3148	75.428	92.142	0.0736
20	79.647	85.285	0.2960	80.612	1	0.0735

Fonte: O autor.

4.3. Análise entre FS_ACO e WFACOFS

Neste capítulo é avaliado o desempenho do método proposto em relação à outro algoritmo referencial do tema na literatura. Conforme mencionado anteriormente, o WFACOFS apresentado em 2019 é considerado. Exclusivo a este algoritmo, alguns outros parâmetros de configuração para a execução da ACO são requeridos em relação ao apresentado neste projeto. Uma destas variáveis por exemplo se diz respeito à normalização dos valores de feromônio após cada iteração. Todavia, dentre os parâmetros em comum, foi estabelecido os mesmos valores afim de simular o quanto mais próximo os cenários. A Tabela 4.10 apresenta a descrição e valores dos parâmetros configurados no algoritmo WFACOFS.

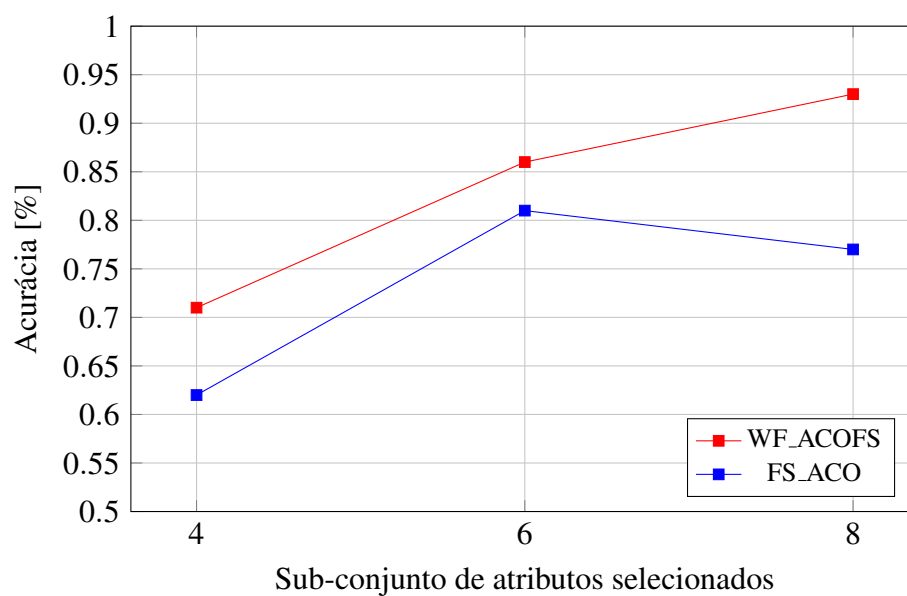
Tabela 4.10: Descrição dos parâmetros e valores ACO do algoritmo WFACOFS

Símbolo	Descrição do parâmetro	Valor
m	Quantidade de Formigas	10
α	Importância do Feromônio	1
β	Importância da Informação Heurística	1
Iterações	Número de Iterações	15
ρ	Taxa de Evaporação	0.15
θ	Taxa de Avaliação do Feromônio	0.8
$\omega 1$	Peso da Acurácia	100
$\omega 2$	Peso do Número de Atributos	1

Fonte: O autor.

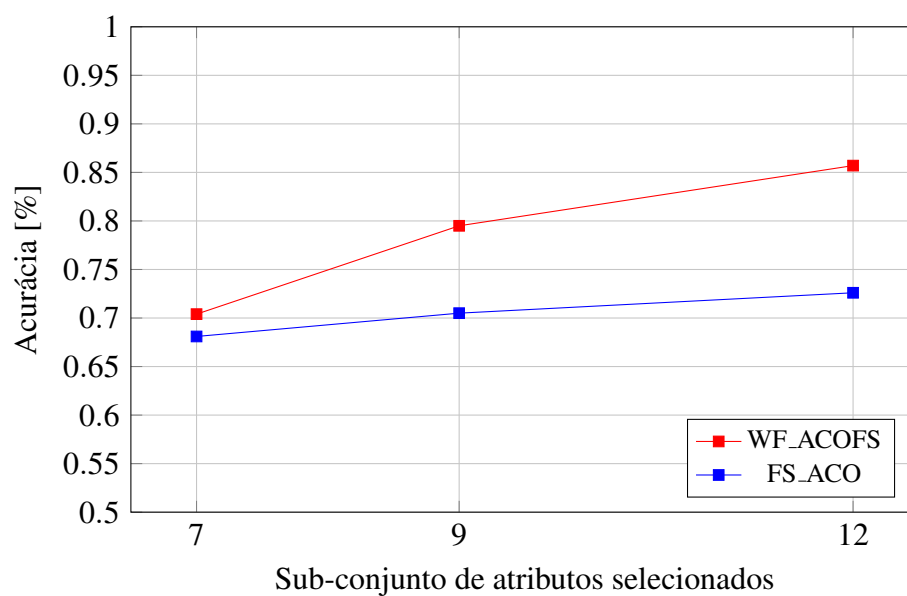
Nestas análises, para os resultados de ambos os algoritmos, foram considerados os classificadores KNN e MLP e os dados divididos entre 70% para treinamento e 30% para teste. Os Gráficos 4.1 e 4.2 apresentam os valores da média de acurácia, após 10 execuções, para os bancos de dados *Vowel* e *Lymphography* respectivamente e foram obtidos através da classificação KNN. Os resultados dos Gráficos 4.3 e 4.4 referem-se aos valores da média de acurácia, após 10 execuções, para os bancos de dados *Zoo* e *TML* obtidos através do classificador MLP. O eixo X denota o sub-conjunto de atributos selecionados enquanto o eixo Y a porcentagem da média de acurácia.

Gráfico 4.1: Valores da Média de Acurácia %, após 10 execuções, obtidos através da classificação KNN para o banco de dados *Vowel*:



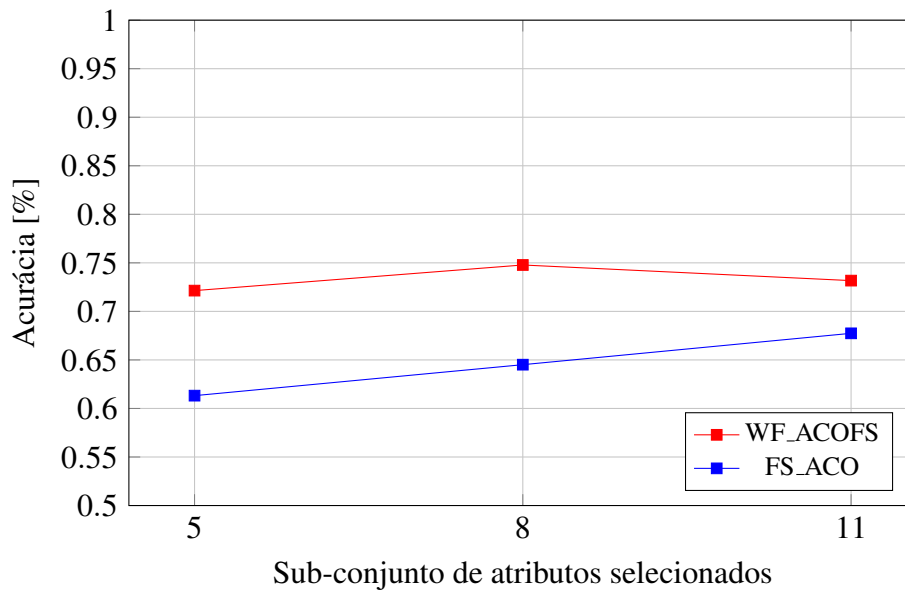
Fonte: O autor.

Gráfico 4.2: Valores da Média de Acurácia %, após 10 execuções, obtidos através da classificação KNN para o banco de dados *Lymprography*:



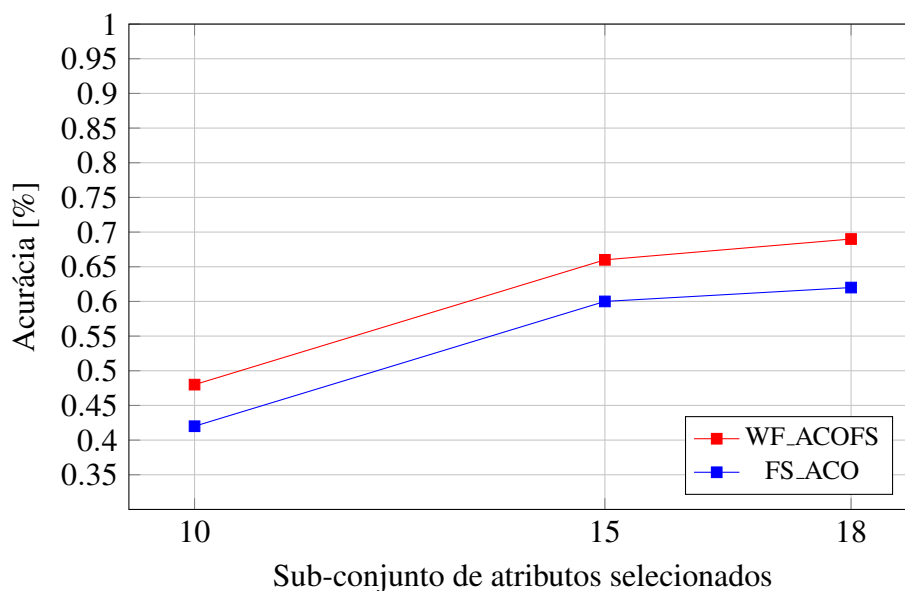
Fonte: O autor.

Gráfico 4.3: Valores da Média de Acurácia %, após 10 execuções, obtidos através da classificação MLP para o banco de dados *Zoo*:



Fonte: O autor.

Gráfico 4.4: Valores da Média de Acurácia %, após 10 execuções, obtidos através da classificação MLP para o banco de dados *TML*:



Fonte: O autor.

Conforme nota-se nos gráficos, o desempenho do WFACOFS apresentou-se superior em todos os cenários para os bancos de dados analisados em relação ao FS_ACO. Os resultados foram obtidos levando em consideração os classificadores MLP e KNN. Este desempenho era esperado diante de tecnologias presentes ao longo do algoritmo de *Ghosh* sugeridas para contornar enviesamentos observados na Otimização por Colônia de Formiga. A estrutura da

ACO implementada no FS_ACO é uma adaptação do algoritmo *Ant System*, uma das primeiras propostas sugeridas acerca do tema e indicada como ponto de partida para novas abordagens. A partir desta versão, otimizações podem ser desenvolvidas visando contornar as inconformidades observadas e alcançar melhores índices nas avaliações. O estudo do balanceamento dos parâmetros ACO também pode proporcionar melhoria nos resultados.

O método proposto neste trabalho apresentou bom desempenho, com seus valores não distanciando muito dos apresentados pelo WFACOFs, com destaque para o subconjunto de 6 atributos selecionados para o banco *Vowel* da qual apresentou uma diferença de 5% e para o subconjunto de 7 atributos selecionados do banco *Lymphography* da qual a diferença ficou de apenas 2%. Em relação ao banco de dados utilizado no estudo de caso, *TML*, os valores distinguem-se do apresentado no capítulo 4.1 em virtude do critério adotado para a função de avaliação. Na ocasião do estudo, foi considerado o Erro Médio Absoluto enquanto para esta etapa da dissertação a Acurácia.

5 CONCLUSÃO

O projeto apresentou os desafios no que se refere aos métodos de seleção de atributos não-supervisionados e preparação dos dados. A teoria introduzida através da metodologia de Mineração de Dados CRISPDM com foco na etapa de preparação dos dados, possibilitou a execução de procedimentos de obtenção e transformação dos dados, os adequando para as execuções dos algoritmos e posteriormente suas avaliações. Ao lidar com diferentes algoritmos, dentre os quais eram disponibilizados nos formatos *.java*, *.m* e *.py*, foi necessário em grande parte a adaptação dos códigos para a etapa de leitura dos dados dos mesmos. Após pesquisa na literatura a respeito de métodos para seleção de atributos baseados na otimização por colônia de formigas e entendimento de suas etapas presentes ao longo da otimização, foi proposto um método de características *Filter* e *Wrapper*, adotando-se o direcionamento atual na literatura. Nesta etapa foi adaptado a proposta do algoritmo *Ant Systems*, codificando-a para assumir as características projetadas. Condições de se avaliarem estes algoritmos foram desenvolvidas e seus resultados apresentados. O registro da média após 15 execuções da métrica acurácia obtidas através dos classificadores KNN e MLP foi adotado como metodologia para avaliação dos algoritmos. Os valores do método proposto não se encontram distantes dos destacados atualmente na literatura, ao ponto que otimizar suas funções inseridas ao longo da ACO aumentaria seus índices. Uma aplicação prática de um dos algoritmos referenciais do estudo foi realizada, com o auxílio do software estatístico SPSS Modeler, resultando na apresentação do artigo no CBA 2020. A maioria dos métodos de seleção não-supervisionados requer a especificação de hiperparâmetros, inerentes a cada método, e o estudo do balanceamento desses parâmetros diante de bancos de dados com características similares é um espaço em aberto para desenvolvimento.

6 SUGESTÕES PARA DESENVOLVIMENTO E DISCUSSÕES

Uma das motivações desta pesquisa baseia-se na importância que o algoritmo de Tabakhi (TABAKHI *et al.*, 2014), proposto em 2014, contribui para o tema de seleção de atributos baseados na otimização por colônia de formigas. Na ocasião, sua proposta ao considerar o valor da função de similaridade de cossenos para avaliar a semelhança entre variáveis foi significativamente importante. Desde então, seu algoritmo é usualmente citado em pesquisas, destacando-se seu bom desempenho nos experimentos.

Porém, atualmente na literatura a proposta apresentada por Ghosh e seus co-autores, proposto em 2019, se destaca em termos dos resultados alcançados e ao novo método proposto, do tipo *Filter* e *wrapper*. Em seu artigo, ele evidencia através de uma exaustiva análise em relação a outros algoritmos, inclusive o proposto por Tabakhi, um melhor desempenho alcançando resultados significativos. Logo, para os próximos anos, o método assume relevância na área da pesquisa e passa a ser considerado para análises e experimentos.

O algoritmo proposto nesse trabalho, o FS_ACO, possui seus módulos definidos, comentados e programados através do fluxo de uma otimização por colônia de formigas para seleção de atributos. A partir desta versão do algoritmo, disponível no *Github*, pode-se implementar a proposta de Ghosh. Ambos adotam-se características do tipo *Filter* e *Wrapper*, sendo que as diferenças mais significativas recaem sobre quando as formulações envolvendo as equações da regra de probabilidade e atualização do feromônio são executadas e como a realizam ao longo da otimização. A maneira como é armazenado os resultados do valor da função de avaliação é outro diferencial do algoritmo WFACOFs. Uma vez que o algoritmo de Ghosh encontra-se disponível apenas no formato *.mat*, sua implementação através da linguagem *python* contribuiria para os estudos de pesquisadores e interessados na área. Compreendido e implementadas suas tecnologias, seria um bom começo para sugestões de novos algoritmos visando alcançar e superar os índices observados pelo WFACOFs.

A calibração de hyper-parâmetros inerentes à otimização por colônia de formigas é uma área de estudos a ser considerada para trabalhos futuros, visando determinar os valores ótimos dos métodos. Frente a bancos de dados maiores, esse ajuste contribuiria para uma melhor performance. Existem estudos publicados relacionados a esse tema.

A maioria dos algoritmos de seleção de atributos são desenvolvidos para analisar dados no formato numérico. E atualmente, principalmente em aplicações biomédicas, bancos de dados apresentam características com dados mistos. Existe um espaço para desenvolvimento de métodos que realizem a seleção de atributos de bancos de dados com estas características. Outra sugestão é a utilização de outro classificador na modelagem da etapa *wrapper*, como por exemplo o SVM - *Support Vector Machine* na intenção de contribuir com outros índices para a etapa de avaliação.

Referências Bibliográficas

- AGHDAM, M. H., GHASEM-AGHAEE, N., BASIRI, M. E. “Text feature selection using ant colony optimization”, **Expert systems with applications**, v. 36, n. 3, pp. 6843–6853, 2009.
- ALOISE, D., NORONHA, T., MAIA, R., BITTENCOURT, V. G., ALOISE, D. J. “Heurísticas de colônia de formigas com path-relinking para o problema de otimização da alocação de sondas de produção terrestre–SPT”, **XXXIV SBPO**, 2002.
- AYRES, P. “**Seleção de Atributos baseado no algoritmo de Otimização por Colônia de Formigas para processos mineradores**”. Disponível em: <https://github.com/PedroAyres99/Master_FS_ACO.git>. Acesso em 10-09-2021.
- BEDI, M. K., SINGH, S. “Comparative study of two natural phenomena based optimization techniques”, **International Journal of Scientific & Engineering Research**, v. 4, n. 3, pp. 1–4, 2013.
- BLUM, C., ROLI, A. “Metaheuristics in combinatorial optimization: Overview and conceptual comparison”, **ACM computing surveys (CSUR)**, v. 35, n. 3, pp. 268–308, 2003.
- CAMILO, C. O., SILVA, J. C. D. “Mineração de dados: Conceitos, tarefas, métodos e ferramentas”, **Universidade Federal de Goiás (UFG)**, pp. 1–29, 2009.
- CARVALHO, E. M. D., RAMOS, G. S. “**Otimização por Colônia de Formigas (Ant Colony Optimization-ACO)**”, .
- CHAPMAN, P., CLINTON, J., KERBER, R., KHABAZA, T., REINARTZ, T., SHEARER, C., WIRTH, R. “The CRISP-DM user guide”. Em: **4th CRISP-DM SIG Workshop in Brussels in March**, v. 1999, 1999.
- DONG, G., LIU, H. **Feature engineering for machine learning and data analytics**. CRC Press, 2018.
- DORIGO, M. “Optimization, learning and natural algorithms”, **Ph. D. Thesis, Politecnico di Milano**, 1992.

- DORIGO, M., STÜTZLE, T. “The ant colony optimization metaheuristic: Algorithms, applications, and advances”. Em: **Handbook of metaheuristics**, Springer, pp. 250–285, 2003.
- DORIGO, M., STÜTZLE, T. “Ant colony optimization: overview and recent advances”. Em: **Handbook of metaheuristics**, Springer, pp. 311–351, 2019.
- DORIGO, M., MANIEZZO, V., COLORNI, A. “Ant system: optimization by a colony of cooperating agents”, **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, v. 26, n. 1, pp. 29–41, 1996.
- DORIGO, M., BIRATTARI, M., STUTZLE, T. “Ant colony optimization”, **IEEE computational intelligence magazine**, v. 1, n. 4, pp. 28–39, 2006.
- DUA, D., GRAFF, C. “**UCI Machine Learning Repository**”. 2017. Disponível em: <<http://archive.ics.uci.edu/ml>>. Acesso em 10-09-2021.
- FALLAHZADEH, O., DEHGHANI-BIDGOLI, Z., ASSARIAN, M. “Raman spectral feature selection using ant colony optimization for breast cancer diagnosis”, **Lasers in medical science**, v. 33, n. 8, pp. 1799–1806, 2018.
- FERREIRA, R. F., POLICARPO, D. L. V., PADULA, V. P., FERREIRA, M. T. S. “Limite de umidade transportável de minérios de ferro: aspectos regulatórios e técnicos”, **Tecnologia em Metalurgia, Materiais e Mineração**, v. 14, n. 1, pp. 16–23, 2017.
- FOGEL, D. B. “An introduction to simulated evolutionary optimization”, **IEEE transactions on neural networks**, v. 5, n. 1, pp. 3–14, 1994.
- FORSATI, R., MOAYEDIKIA, A., JENSEN, R., SHAMSFARD, M., MEYBODI, M. R. “Enriched ant colony optimization and its application in feature selection”, **Neurocomputing**, v. 142, pp. 354–371, 2014.
- GAMBARDELLA, L. M., DORIGO, M. “Solving symmetric and asymmetric TSPs by ant colonies”. Em: **Proceedings of IEEE international conference on evolutionary computation**, pp. 622–627. IEEE, 1996.
- GARCÍA, S., LUENGO, J., HERRERA, F. **Data preprocessing in data mining**. Springer, 2015.
- GHOSH, M. “**Feature-Selection-Algorithm**”. 2021. Disponível em: <<https://github.com/ManosijGhosh/Feature-Selection-Algorithm.git>>. Acesso em 10-09-2021.

- GHOSH, M., GUHA, R., SARKAR, R., ABRAHAM, A. “A wrapper-filter feature selection technique based on ant colony optimization”, **Neural Computing and Applications**, pp. 1–19, 2019.
- GLOVER, F., SÖRENSEN, K. “Metaheuristics”, **Scholarpedia**, v. 10, n. 4, pp. 6532, 2015. doi: 10.4249/scholarpedia.6532. revision #149834.
- GOLDSCHMIDT, R., PASSOS, E., BEZERRA, E. **Data Mining**. Elsevier Brasil, 2015.
- GUTJAHR, W. J. “Mathematical runtime analysis of ACO algorithms: Survey on an emerging issue”, **Swarm Intelligence**, v. 1, n. 1, pp. 59–79, 2007.
- GUTJAHR, W. J., RAUNER, M. S. “An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria”, **Computers & Operations Research**, v. 34, n. 3, pp. 642–666, 2007.
- GUYON, I., ELISSEEFF, A. “An introduction to variable and feature selection”, **Journal of machine learning research**, v. 3, n. Mar, pp. 1157–1182, 2003.
- ISABELLE GUYON, A. E. “**NIPS feature selection challenge, 2003**”. 2003.
- JUNIOR, G., CARVALHO, L. C., RODRIGUES, J. R., PROENÇA, M. P. “Network anomaly detection using IP flows with Principal Component Analysis and Ant Colony Optimization”, **Journal of Network and Computer Applications**, v. 64, n. 1, pp. 1–11, April 2016. ISSN: 1095-8592. doi: 10.1016/j.jnca.2015.11.024.
- KABIR, M. M., SHAHJAHAN, M., MURASE, K. “A new local search based hybrid genetic algorithm for feature selection”, **Neurocomputing**, v. 74, n. 17, pp. 2914–2928, 2011.
- KALRA, M., SINGH, S. “A review of metaheuristic scheduling techniques in cloud computing”, **Egyptian informatics journal**, v. 16, n. 3, pp. 275–295, 2015.
- KANAN, H. R., FAEZ, K., TAHERI, S. M. “Feature selection using ant colony optimization (ACO): a new method and comparative study in the application of face recognition system”. Em: **Industrial conference on data mining**, pp. 63–76. Springer, 2007.
- KAR, A. K. “Bio inspired computing—a review of algorithms and scope of applications”, **Expert Systems with Applications**, v. 59, pp. 20–32, 2016.
- KFST. “**Kurdistan Feature Selection Tool**”. Disponível em <https://kfst.uok.ac.ir/index.html>, 2020. Acesso em 10-09-2021.
- KOTSIANTIS, S. “Feature selection for machine learning classification problems: a recent overview”, **Artificial Intelligence Review**, v. 42, n. 1, pp. 157–176, 2011.

- LIU, H., MOTODA, H. “On issues of instance selection”, **Data Mining and Knowledge Discovery**, v. 6, n. 2, pp. 115, 2002.
- MANIKANDAN, G., ABIRAMI, S. “Feature Selection Is Important: State-of-the-Art Methods and Application Domains of Feature”, **Applications in Ubiquitous Computing**, p. 177, 2020.
- MORADI, P., ROSTAMI, M. “Integration of graph clustering with ant colony optimization for feature selection”, **Knowledge-Based Systems**, v. 84, pp. 144–161, 2015.
- MULLEN, R. J., MONEKOSSO, D., BARMAN, S., REMAGNINO, P. “A review of ant algorithms”, **Expert systems with Applications**, v. 36, n. 6, pp. 9608–9617, 2009.
- NIIJIMA, S., OKUNO, Y. “Laplacian linear discriminant analysis approach to unsupervised feature selection”, **IEEE/ACM transactions on computational biology and bioinformatics**, v. 6, n. 4, pp. 605–614, 2008.
- OLARIU, S., ZOMAYA, A. Y. **Handbook of bioinspired algorithms and applications**. Chapman and Hall/CRC, 2005.
- OTARIS. “**Data Analysis, Modeling and Reporting - Gaining knowledge with CRISP-DM**”. 2018. Disponível em: <<http://www.otaris.de/gb/datenanalysen-modellbildung-reporting/>>. Acesso em 10-09-2021.
- PARVIN, H., MORADI, P., ESMAEILI, S. “TCFACO: Trust-aware collaborative filtering method based on ant colony optimization”, **Expert Systems with Applications**, v. 118, pp. 152–168, 2019.
- ROSA, E. “Redução da dimensão de um conjunto de variáveis: decomposição em valores singulares versus algoritmo colônia de formigas”, **Revista Eletrônica Paulista de Matemática**, v. 10, 2017.
- SABINO, J. A., LEAL, J. A. E. A., STÄ, T., BIRATTARI, M. “A multi-objective ant colony optimization method applied to switch engine scheduling in railroad yards”, **Pesquisa Operacional**, v. 30, pp. 486 – 514, 08 2010. ISSN: 0101-7438. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382010000200013&nrm=iso>.
- SIMON, D. **Evolutionary optimization algorithms Biologically-Inspired and Population-Based Approaches to Computer Intelligence**. John Wiley & Sons, 2013.
- SOLORIO-FERNÁNDEZ, S., CARRASCO-OCHOA, J. A., MARTÍNEZ-TRINIDAD, J. F. “A review of unsupervised feature selection methods”, **Artificial Intelligence Review**, v. 53, n. 2, pp. 907–948, 2020.

- STÜTZLE, T., HOOS, H. H. “MAX–MIN ant system”, **Future generation computer systems**, v. 16, n. 8, pp. 889–914, 2000.
- SUTHA, K., TAMILSELVI, J. J. “A review of feature selection algorithms for data mining techniques”, **International Journal on Computer Science and Engineering**, v. 7, n. 6, pp. 63, 2015.
- SWEETLIN, J. D., NEHEMIAH, H. K., KANNAN, A. “Computer aided diagnosis of pulmonary hamartoma from CT scan images using ant colony optimization based feature selection”, **Alexandria engineering journal**, v. 57, n. 3, pp. 1557–1567, 2018.
- TABAKHI, S., MORADI, P. “Relevance–redundancy feature selection based on ant colony optimization”, **Pattern recognition**, v. 48, n. 9, pp. 2798–2811, 2015.
- TABAKHI, S., MORADI, P., AKHLAGHIAN, F. “An unsupervised feature selection algorithm based on ant colony optimization”, **Engineering Applications of Artificial Intelligence**, v. 32, pp. 112–123, 2014.
- TANG, J., ALELYANI, S., LIU, H. “Feature selection for classification: A review”, **Data classification: Algorithms and applications**, p. 37, 2014.
- THEODORIDIS, S., KOUTROUMBAS, K. “Pattern Recognition, Waltham, MA”. 2008.
- UTHAYAKUMAR, J., METAWA, N., SHANKAR, K., LAKSHMANAPRABU, S. “Financial crisis prediction model using ant colony optimization”, **International Journal of Information Management**, v. 50, pp. 538–556, 2020.
- WEISE, T. “Global optimization algorithms-theory and application”, **Self-Published Thomas Weise**, 2009.
- WITTEN, I., FRANK, E., HALL, M., PAL, C. “**DataMining**: Practical machine learning tools and techniques”. November 2016.
- YIN, Z., DU, C., LIU, J., SUN, X., ZHONG, Y. “Research on autodisturbance-rejection control of induction motors based on an ant colony optimization algorithm”, **IEEE Transactions on Industrial Electronics**, v. 65, n. 4, pp. 3077–3094, 2017.
- ZHANG, Z., FENG, Z. “Two-stage updating pheromone for invariant ant colony optimization algorithm”, **Expert Systems with Applications**, v. 39, n. 1, pp. 706–712, 2012.