UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

LUCAS DE ALMEIDA RIBEIRO

Algoritmo Evolutivo Multi-objetivo em Tabelas para Seleção de Variáveis em Classificação Multivariada

Goiânia
2014

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR AS TESES E DISSERTAÇÕES ELETRÔNICAS (TEDE) NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: **[x] Dissertação** **[ ] Tese**

2. Identificação da Tese ou Dissertação

Autor (a):	Lucas de Almeida Ribeiro						
E-mail:	lucas.ar@live.com						
Seu e-mail pode ser disponibilizado na página? [x]Sim [ ] Não							
Vínculo empregatício do autor	Professor Universitário						
Agência de fomento:	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior				Sigla:	CAPES	
País:	BRASIL	UF:	GO	CNPJ:	00889834/0001-08		
Título:	Algoritmo Evolutivo Multi-objetivo em Tabelas para Seleção de Variáveis em Classificação Multivariada						
Palavras-chave:	seleção de variáveis, classificação multivariada, análise discriminante linear, algoritmo evolutivo multi-objetivo em tabelas						
Título em outra língua:	Multi-objective evolutionary algorithm on tables for variable selection in multivariate classification						
Palavras-chave em outra língua:	variable selection, multivariate classification, linear discriminant analysis, multi-objective evolutionary algorithm on tables						
Área de concentração:	Ciência da Computação						
Data defesa: (dd/mm/aaaa)							
Programa de Pós-Graduação:	PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO - MESTRADO EM CIÊNCIA DA COMPUTAÇÃO						
Orientador (a):	Anderson da Silva Soares						
E-mail:	anderson@inf.ufg.br						
Co-orientador (a):*							
E-mail:							

*Necessita do CPF quando não constar no SisPG

3. Informações de acesso ao documento:

Concorda com a liberação total do documento [x] SIM [ ] NÃO[1]

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF ou DOC da tese ou dissertação.

O sistema da Biblioteca Digital de Teses e Dissertações garante aos autores, que os arquivos contendo eletronicamente as teses e ou dissertações, antes de sua disponibilização, receberão procedimentos de segurança, criptografia (para não permitir cópia e extração de conteúdo, permitindo apenas impressão fraca) usando o padrão do Acrobat.

_____ Data: _____ / _____ / _____

Assinatura do (a) autor (a)

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

LUCAS DE ALMEIDA RIBEIRO

Algoritmo Evolutivo Multi-objetivo em Tabelas para Seleção de Variáveis em Classificação Multivariada

Dissertação apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação.

Orientador: Prof. Dr. Anderson da Silva Soares

Goiânia
2014

Ficha catalográfica elaborada automaticamente
com os dados fornecidos pelo(a) autor(a), sob orientação do Sibi/UFG.

de Almeida Ribeiro, Lucas
Algoritmo Evolutivo Multi-objetivo em Tabelas para Seleção de Variáveis em Classificação Multivariada [manuscrito] / Lucas de Almeida Ribeiro. - 2014.
80 f.: il.

Orientador: Prof. Dr. Anderson da Silva Soares.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF) , Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2014.
Bibliografia.
Inclui siglas, abreviaturas, símbolos, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. Seleção de variáveis. 2. Classificação multivariada. 3. Análise discriminante linear. 4. Algoritmo evolutivo multi-objetivo em tabelas. I. da Silva Soares, Anderson, orient. II. Título.

Lucas de Almeida Ribeiro

Algoritmo Evolutivo Multi-objetivo em Tabelas para Seleção de Variáveis em Classificação Multivariada

Dissertação defendida no Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Mestre em Ciência da Computação, aprovada em 29 de outubro de 2014, pela Banca Examinadora constituída pelos professores:

Prof. Dr. Anderson da Silva Soares
Instituto de Informática – UFG
Presidente da Banca

Prof. Dr. Clarimar José Coelho
Pontifícia Universidade Católica de Goiás – PUC/GO

Prof. Dr. Fernando Marques Federson
Instituto de Informática – UFG

Lucas de Almeida Ribeiro

Graduou-se em Sistemas de Informação na UEG - Universidade Estadual de Goiás.

A Deus, a minha família.

Agradecimentos

Agradeço a Deus e a espiritualidade amiga por me auxiliarem a vencer mais uma caminhada, mantendo em mim o ensejo por dias melhores e por contribuir com os conhecimentos alcançados durante a execução deste trabalho.

Agradeço de forma especial ao meu saudoso pai, Alonço Ribeiro de Oliveira, que não mediu esforços para me formar e me tornar um ser humano melhor a cada dia. Espero que de onde esteja possa ver neste trabalho mais uma contribuição dele à minha jornada evolutiva.

Agradeço à minha mãe, Gerusa de Almeida Ribeiro Oliveira, pela dedicação, pelos conselhos e pela compreensão que sempre demonstrou para comigo. Aos meus irmãos Gerlon, Marcos e Leandra por serem um ponto de apoio e de referência quando me encontro em momentos difíceis. Ao Gerlon pela postura altruísta e carinhosa com que me tratou desde minha infância. Ao Marcos pelo companheirismo e pela amizade com que sempre me acolheu e me orientou. À Leandra por ver minhas qualidades antes dos meus defeitos e ter sempre desvelo quando se trata de minha imagem.

Agradeço aos meus amigos e primos por fazerem a diferença em minha vida sempre me auxiliando e sendo pacientes com minhas limitações. Um agradecimento especial ao Adorando e à Larissa por me darem forças e confiarem em mim em tantos momentos.

Agradeço aos meus tios e tias que sempre me trataram com amor, sem a confiança e o amor das pessoas que nos rodeiam qualquer caminhada se torna mais difícil e qualquer obstáculo se torna mais complexo. De forma especial, agradeço à madrinha Shirley por ter me alfabetizado e sempre me oferecer seus melhores conselhos, também por sempre acreditar em minha capacidade e me motivar em minhas escolhas.

Ao orientador, Professor Doutor Anderson da Silva Soares, por ter aceitado me instruir nesta jornada, por sua postura ética, por confiar e me auxiliar em momentos difíceis. Por sua dedicação e atenção desde o meu ingresso neste programa. Também agradeço à sua esposa, Telma, por me auxiliar no cumprimento dos trabalhos e pela compreensão nos momentos necessários.

Agradeço também a cada um dos colegas que fiz durante o mestrado: à Mariana, à Carina, ao Roussian, ao Leonardo, ao Lauro e a cada um dos que fizeram parte deste

momento e desta jornada. Agradeço de forma especial ao Carlos por ter sido sempre tão prestativo, atencioso e amigo em tantos momentos do curso.

Agradeço aos professores do INF-UFG pelos conhecimentos adquiridos nas disciplinas estudadas. À equipe administrativa na pessoa da Mirian por sempre me atender de forma prestativa e atenciosa.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo fornecimento de minha bolsa de estudo.

À todas as pessoas que fizeram e farão parte do meu futuro muito obrigado. Hoje me sinto mais capaz e mais preparado para novos desafios graças à contribuição que cada um fez em minha vida.

"A pobreza é involuntária e debilitante, a simplicidade é voluntária e mobilizadora."

"Duane Elgin",
Simplicidade Voluntária.

Resumo

Ribeiro, Lucas de Almeida. **Algoritmo Evolutivo Multi-objetivo em Tabelas para Seleção de Variáveis em Classificação Multivariada**. Goiânia, 2014. 81p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

Este trabalho propõe o uso do algoritmo evolutivo multi-objetivo em tabelas (AEMT) para a seleção de variáveis em problemas de classificação, por meio de análise discriminante linear. O algoritmo proposto busca encontrar subconjuntos mínimos, das variáveis originais, que modelem classificadores robustos, sem perda significativa na capacidade de classificação. Os resultados dos classificadores modelados pelas soluções encontradas por este algoritmo são comparadas, neste trabalho, às encontradas por formulações mono-objetivo (como o PLS, o APS e implementações próprias de um Algoritmo Genético Simples) e formulações multi-objetivos (como algoritmo genético multi-objetivo simples - MULTI-GA - e o NSGA II). Como estudo de caso, o algoritmo foi aplicado na seleção de variáveis espectrais, para a classificação por análise discriminante linear (LDA - Linear Discriminant Analysis), de amostras de biodiesel/diesel. Os resultados obtidos mostraram que as formulações evolutivas encontram soluções com um menor número de variáveis (em média) e uma melhor taxa de erros (média) se comparadas ao PLS e o APS. A formulação do AEMT proposta com as funções de aptidão: risco médio de classificação, número de variáveis selecionadas e quantidade de variáveis correlacionadas presentes no modelo, encontrou soluções com uma média de erros inferior as encontradas pelo NSGA II e pelo MULTI-GA, e também uma menor quantidade de variáveis se comparado ao MULTI-GA. Em relação à sensibilidade a ruídos a solução encontrada pelo AEMT se mostrou menos sensível que as outras formulações comparadas, mostrando assim que o AEMT encontra classificadores mais robustos. Por fim, são apresentadas as regiões de separação das classes, com base na dispersão das amostras, em função das variáveis selecionadas por uma das soluções do AEMT, nota-se que é possível determinar regiões de separação a partir das variáveis selecionadas.

Palavras–chave

seleção de variáveis, classificação multivariada, análise discriminante linear, algoritmo evolutivo multi-objetivo em tabelas

Abstract

Ribeiro, Lucas de Almeida. **Multi-objective evolutionary algorithm on tables for variable selection in multivariate classification**. Goiânia, 2014. 81p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

This work proposes the use of multi-objective evolutionary algorithm on tables (AEMT) for variable selection in classification problems, using linear discriminant analysis. The proposed algorithm aims to find minimal subsets of the original variables, robust classifiers that model without significant loss in classification ability. The results of the classifiers modeled by the solutions found by this algorithm are compared in this work to those found by mono-objective formulations (such as PLS, APS and own implementations of a Simple Genetic Algorithm) and multi-objective formulations (such as the simple genetic algorithm multi -objective - MULTI-GA - and the NSGA II). As a case study, the algorithm was applied in the selection of spectral variables for classification by linear discriminant analysis (LDA) of samples of biodiesel / diesel. The results showed that the evolutionary formulations are solutions with a smaller number of variables (on average) and a better error rate (average) and compared to the PLS APS. The formulation of the AEMT proposal with the fitness functions: medium risk classification, number of selected variables and number of correlated variables in the model, found solutions with a lower average errors found by the NSGA II and the MULTI-GA, and also a smaller number of variables compared to the multi-GA. Regarding the sensitivity to noise the solution found by AEMT was less sensitive than other formulations compared, showing that the AEMT is more robust classifiers. Finally shows the separation regions of classes, based on the dispersion of samples, depending on the selected one of the solutions AEMT, it is noted that it is possible to determine variables of regions split from the selected variables.

Keywords

variable selection, multivariate classification, linear discriminant analysis, multi-objective evolutionary algorithm on tables

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Símbolos

Lista de Abreviaturas e Siglas

AEMT	Algoritmo Evolutivo Multiobjetivo em Tabelas 20
NIR	Infravermelho Próximo (Near-Infrared) 20
$IV-FT$	Intravermelho de transformada de Fourier 23
LDA	Análise Discriminante Linear (Linear Discriminant Analysis 29
EC	Computação Evolutiva (Evolutionary Computation) 30
EA	Algoritmos Evolutivos (Evolutionary Algorithm) 30
GA	Algoritmo Genético (Genetic Algorithm) 36
ES	Estratégias Evolutivas (Evolutionary Strategy) 39
VEGA	Algoritmo Genético de Vetor Valorado (Vector Evaluated Genetic Algorithm) 48
*SPEA*2	Algoritmo Evolutivo da Força de Pareto 2 (Strength Pareto Evolutionary Algorithm 2) 50
NSGAII	Algoritmo Genético por Ordenação de Não Dominância II (Non dominated Sorting Genetic Algorithm II) 53
$MONO-GA$	Algoritmo Genético Simples Mono-objetivo 63
$MULTI-GA$	Algoritmo Genético Simples de Objetivos Múltiplos 63

Capítulo 1 - Introdução

As técnicas de classificação multivariada tem por objetivo a formação de grupos comuns entre objetos (ou amostras) e a alocação de objetos, nos grupos pré-estabelecidos, com base nas múltiplas variáveis medidas para cada amostra [35]. Assim, o objetivo da classificação multivariada é alocar um objeto ao seu grupo a partir da mensuração de suas variáveis [54] [36].

O uso das técnicas de classificação multivariada pode ser observado nas mais diversas áreas: na saúde pública [66], em pesquisas históricas [46] [27], em pesquisas sobre a produção agrícola [20] [48], em antropologia forense [75], na área de economia e negócios [45], na área de turismo [64], na área química [28] [53], entre outras. Dentre as áreas citadas, a que busca prover informações sobre a relação, ou resultantes da relação, de centenas a milhares de variáveis de dados químicos são chamada quimiometria [30]. A técnica de classificação multivariada mais utilizada em dados químicos é a análise discriminante [55].

As variáveis utilizadas nos problemas de quimiometria são mensuradas de forma simultânea sobre cada amostra e a correlação entre elas produz colinearidade nas informações contidas nestas variáveis [60]. A presença de colinearidade e quase colinearidade prejudicam a capacidade de generalização dos modelos de classificação [55]. Como as variáveis colineares possuem dados redundantes, e podem trazer instabilidade (tendências) aos métodos de classificação, é necessário descartá-las [60], o que torna os métodos de seleção de variáveis uma prioridade [35].

Dentre os métodos de seleção de variáveis tem-se: o método *stepwise*, análise de todos os possíveis subconjuntos, regressão $Forwards - Stagewise$, e a regressão $Least - Angle$ [35]. A análise de todos os possíveis subconjuntos tem complexidade exponencial, sendo necessário um algoritmo de busca para tratar este problema. Neste contexto, pode se fazer uso de heurísticas como estratégia de busca, do subconjunto de variáveis, para otimizar a classificação ou regressão dos dados. Dentre estas heurísticas destacam-se os algoritmos evolutivos [40].

Algoritmos evolutivos (EAs) são uma heurística de busca de solução moderna, baseados no princípio da evolução pela sobrevivência do mais apto [50]. O tipo mais

conhecido de algoritmos evolutivos é o algoritmo genético [38] e seu uso mais comum, em problemas de classificação, está relacionado à seleção das variáveis (ou atributos) [77].

Um algoritmo evolutivo mantém um conjunto de soluções chamado população, cada solução deve possuir uma representação genética, chamada gene, e um nível de aptidão, chamado "valor de aptidão" (ou $fitness$). Os algoritmos evolutivos dão preferência aos indivíduos mais aptos (com melhores valores $fitness$). Os mecanismos responsáveis pela busca de solução, em um algoritmo evolutivo, são os operadores de transformação (operadores genéticos) que criam novas soluções a partir de modificações realizadas em uma ou mais soluções [79]. O valor de aptidão, de uma solução, pode apresentar a qualidade da solução ou pode ser a proximidade da solução em relação à solução esperada [13].

Em relação a problemas de seleção de variáveis para classificação, a qualidade de uma solução (conjunto de variáveis selecionadas) pode ser avaliada com base nas características do classificador modelado pelo subconjunto de variáveis presente nesta solução. Uma medida de qualidade que pode ser utilizada neste contexto é o "risco médio de classificação". O "risco de classificação" de uma amostra é resultante da relação entre a proximidade da amostra às amostras de sua classe e a proximidade da amostra às amostras de uma outra classe (a classe mais próxima, diferente da que ela pertence). O "risco médio de classificação"de uma solução é obtido pela média dos "riscos de classificação"da coleção que está sendo classificada (coleção de testes) [58] [65].

Algoritmos de busca que têm como objetivo único o "risco médio de classificação", são limitados, pois não avaliam as características dos subconjuntos de variáveis, avaliam apenas os classificadores modelados por estes subconjuntos. Neste trabalho partese da premissa que a inclusão dos objetivos: remoção das variáveis correlacionadas da solução e redução do subconjunto de variáveis, no algoritmo de busca, pode melhorar o desempenho do classificador no que se refere a robustez, e proximidade das soluções encontradas ao subconjunto ideal.

O método clássico de resolver problemas multi-objetivo é convergir los à um único objetivo, por uma função de agregação, porém essa solução é totalmente dependente dos valores escalares escolhidos [69]. Outra solução é proposta por Deb [14], onde se ordena as soluções segundo seu nível de dominância. Tais soluções possuem um bom desempenho para problemas com poucos objetivos, preferencialmente com 2 objetivos [15]. Tal restrição ocorre em razão de que em poucas em poucas iterações o algoritmo genético consegue encontrar muitas soluções não dominadas e o processo de escolha entre estas soluções se torna estocástico [15] [33].

Sanches [62] [63] e Vargas [76] propõem um algoritmo evolutivo multi-objetivo que não está restrito às soluções não dominadas ou a um vetor escalar e funciona com múltiplos objetivos. O algoritmo proposto divide a população de cada geração em subpopulações de acordo com as funções objetivo ou uma mesclagem de funções de aptidão.

Cada uma destas subpopulações, também chamadas de tabelas, armazena os indivíduos avaliados por uma função de aptidão, lidando assim com múltiplos critérios de seleção de indivíduos de maneira simultânea, por isso o algoritmo é chamado Algoritmo Evolutivo Multi-objetivo em Tabelas (AEMT). Uma das tabelas deve funcionar como um decisor, sendo que o seu valor objetivo é a agregação dos valores de aptidão das demais tabelas. Esta formulação foi utilizada por Brasil [6], Santos [19], e Sanches [62] e obteve resultados positivos em predição *abinitio* de estrutura de proteínas [6] e para reconfiguração de sistemas de distribuição de energia elétrica [62] [19], utilizando múltiplos objetivos. Os trabalhos citados utilizam 4 objetivos ou mais e o uso do AEMT possibilitou encontrar soluções de pareto mais extremas que os algoritmos comparados possibilitando mais opções para o decisor. Jorge [39] utilizou de uma implementação do AEMT para a seleção de variáveis em problemas de calibração multivariada com 3 objetivos, e obteve resultados melhores que as abordagens clássicas. Em relação a abordagens multi-objetivo que utilizam conceitos de dominância a implementação do AEMT obteve resultados similares ao NSGA II, porém, obteve uma melhor sensibilidade a ruído [39].

Este trabalho propõe o uso de uma formulação do algoritmo evolutivo multi-objetivo em tabelas (AEMT), na seleção de variáveis para classificação, com o uso de análise discriminante linear. A implementação do AEMT por Brasil [6] utilizava a representação do indivíduo com cromossomos de codificação de números reais e a por Santos [19] e Sanches [62] [63] uma representação com cromossomos de codificação de números naturais. Neste contexto a formulação do AEMT para um problema de seleção de variáveis para a classificação utilizando codificação binária é uma contribuição nova para o AEMT. Com relação ao problema de classificação espera-se que a abordagem multi-objetivo apresente um desempenho melhor em relação a modelagem mono-objetivo. Em resumo, a proposta tem como objetivo principal encontrar subconjuntos mínimos de variáveis que possibilitem a construção de classificadores robustos e com alta acurácia. Como funções de avaliação serão utilizadas: (i) o risco médio de classificação, (ii) o número de variáveis correlacionadas utilizadas, (iii) o número de variáveis, e (iv) uma função de agregação das anteriores.

Como estudo de caso, o algoritmo proposto é aplicado ao mesmo conjunto de Pontes et al. [59] para fins de comparação. Tal conjunto envolve o problema de classificação de amostras de diesel e biodiesel obtidas por espectroscopia no infravermelho próximo (NIR'). Utiliza-se como base de comparação mono-objetivo, além dos resultados obtidos por Pontes et al. [59], uma implementação própria de um algoritmo evolutivo (MONO-GA-R) onde a função de avaliação é o risco médio de classificação em um conjunto de testes, e uma formulação cuja função de avaliação é o número de erros do classificador no conjunto de testes (MONO-GA-E). Para formulações multi-objetivos são utilizados para comparação, o algoritmo MULTI-GA com a ponderação dos objeti-

vos como função de aptidão; e uma implementação do algoritmo genético multi-objetivo NSGA II para cada uma das combinações de objetivos (risco médio de classificação (R), número de variáveis correlacionadas (C), e número de variáveis selecionadas para a construção do modelo (V)).

1.1 Organização do Trabalho

O Capítulo 2 apresenta uma revisão de classificação multivariada, o problema de colinearidade e quase colinearidade, e seleção de variáveis. O Capítulo 3 aborda os conceitos pelos quais se baseiam os algoritmos evolutivos e suas principais abordagens. O capítulo 4 mostra uma fundamentação teórica sobre os algoritmos evolutivos multi-objetivos. O Capítulo 5 apresenta o algoritmo proposto neste trabalho e os experimentos realizados. O Capítulo 6 apresenta os resultados obtidos com o uso do algoritmo proposto comparando-os com os obtidos pelos outros algoritmos. O Capítulo 7 conclui, apresenta as considerações finais e sugere trabalhos futuros.

CAPÍTULO 2

Capítulo 2 - Classificação Multivariada

As técnicas de classificação multivariada tem por objetivo a formação de grupos comuns entre objetos (ou amostras) e a alocação de objetos, nos grupos pré-estabelecidos, com base nas múltiplas variáveis medidas para cada amostra [35]. Assim, o objetivo da classificação multivariada é alocar um objeto ao seu grupo a partir da mensuração de suas variáveis [54] [36]. O uso de técnicas de classificação se destaca na área de quimiometria, área que busca prover informações sobre a relação, ou resultantes da relação, de centenas a milhares de variáveis de dados de origem química [30]. Uma técnica versátil para a obtenção destes dados, que não destrói a amostra, é a espectroscopia no infravermelho próximo [72].

2.1 Espectroscopia do Infravermelho Próximo (NIR)

A espectroscopia no infravermelho é uma técnica experimental baseada na vibração dos átomos de uma molécula, quando submetidos à radiação infravermelha. Um espectro infravermelho é obtido incidindo-se radiação infravermelha em uma amostra e analisando a fração de radiação que foi absorvida pela amostra [72]. O espectro é normalmente representado por um gráfico, onde a abscissa é o comprimento de onda ao qual a amostra é submetida, e a ordenada é um valor que representa a radiação absorvida pela amostra, naquele comprimento de onda [41]. Esta técnica é versátil e é relativamente fácil obter espectros de amostras em soluções, ou nos estados sólidos, líquidos ou gasosos [72].

A radiação eletromagnética na região do infravermelho com comprimentos de onda de 10000 à 100 cm^{-1} é absorvida e convertida pela molécula orgânica em energia interna de vibração molecular. O nível de absorção é dependente da massa relativa, das forças de ligação, e da geometria dos átomos [67].

A absorção de energia no infravermelho é um processo quantizado. Pode-se selecionar frequências de radiação do infravermelho as quais a molécula absorverá. O padrão de absorção no infravermelho, ou espectro infravermelho, em duas moléculas diferentes nunca serão iguais, mesmo que a frequência absorvida seja a mesma os padrões

de absorção não o serão. Como moléculas iguais têm o mesmo padrão de absorção, a análise dos espectros pode trazer essa informação de equivalência ou informações sobre a estrutura interna da molécula [41].

A espectroscopia no infravermelho próximo (NIR) é um tipo de espectroscopia vibracional que emprega energias em uma faixa de $2,65 \times 10^{-19}$ à $7,96 \times 10^{-20}$J, que corresponde a um faixa de comprimentos de onda de 750 a 2500 nm (frequências: 13000 à 4000cm^{-1}). Esta ordem de energia é maior que a necessária para levar as moléculas apenas aos seus estados vibracionais excitados e menor que valores típicos necessários para a excitação dos eletrons nas moléculas [57]. Os métodos de análise resultantes do uso da espectroscopia no infravermelho próximo são rápidos (menos de um minuto por amostra), não destroem a amostra, não são invasivos, dão alta possibilidade de sondagem no feixe de radiação, são adequados para linha de produção, têm aplicação quase universal (qualquer molécula contendo CH, NH, SH ou ligações OH), e exigem pouca preparação das amostras [57].

O equipamento utilizado para se obter o espectro de absorção de um composto é chamado espectrômetro de infravermelho. Um dos tipos de espectrômetros de infravermelho bastante usado em laboratórios químicos é o espectrômetro de infravermelho dispersivo, que está representado na Figura 2.1 e seu funcionamento é apresentado a seguir[41]. Existem outros tipos de espectrômetros de infravermelho, como é o caso do espectrômetro de intravermelho de transformada de Fourier (IV-FT) que consegue obter um padrão (interferograma) em menos de um segundo, sendo assim possível coletar dezenas de interferogramas da mesma amostra e gravá-los no computador. Quando se realiza a técnica de transformada de Fourier nos interferogramas armazenados, pode-se obter um espectro com um menor ruído em relação ao sinal, e por isso, os IV-FTs possuem uma maior velocidade e maior sensibilidade em relação a equipamentos dispersivos [41]. Outros tipos de espectrômetros de infravermelho são apresentados em [68], e uma visão mais detalhada do IV-FT é dada em [41].

Nos espectrômetros de infravermelhos dispersivos a amostra é colocada em uma cela, e é utilizada, também, uma cela sem amostra (cela de referência) para se verificar a frequência absorvida pela amostra. O instrumento então, produz um feixe de radiação no infravermelho a partir de um resistor aquecido que é dividido por dois espelhos. Os feixes originados desta divisão são enviados para as celas, um à cela que contém a amostra, outro à referência. Os feixes então chegam, de forma alternada, por um monocromador que dispersa, à uma rede de difração (nos instrumentos mais antigos, um prisma). A rede gira lentamente variando a frequência, ou o comprimento de onda da radiação, que chega ao detector de termopar. Este último, sente a razão entre as intensidades dos feixes de referência e de amostra, analisando quais frequências foram absorvidas e quais não foram absorvidas pela amostra, depois de o sinal do detector ser ampliado, o registrador registra

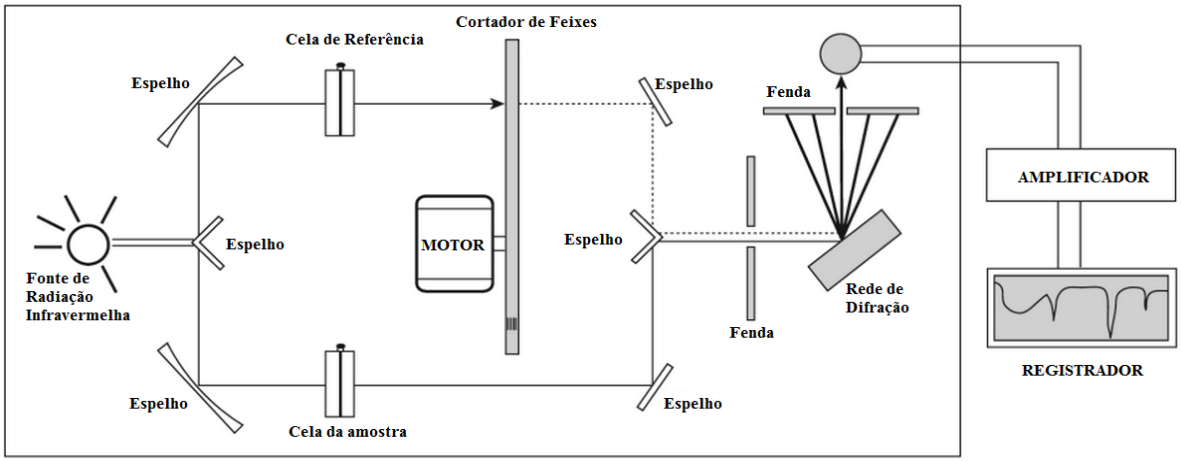

Figura 2.1: *Diagrama esquemático de um espectrômetro de infravermelho dispersivo [41].*

o espectro resultante. O espectro é registrado à medida que a frequência da radiação no infravermelho é alterada pela rotação da rede de difração. O espectro resultante é um gráfico com o número de onda sendo a abscissa e a luz transmitida sendo a ordenada, ou seja, a transmitância percentual($\%T$), pois o detector registra a razão entre as intensidades dos dois feixes e [41]:

$$\%T = \frac{intensidade\ do\ feixe\ de\ amostragem}{intensidade\ do\ feixe\ de\ referncia} \times 100. \tag{2-1}$$

Os valores de máximo representados no gráfico então, são os comprimentos de onda que não são alterados quando passados pela amostra. Ou seja, os pontos próximos à 100% de transmitância são os comprimentos que não são absorvidos pela amostra. Já os pontos mínimos no gráfico são os comprimentos onde houve a absorção, mesmo assim, a absorção é tradicionalmente chamada de pico [41]. O químico então, analisa os picos para determinar quais compostos estão presentes na amostra.

Sendo assim, as variáveis que compõem um espectro (a transmitância percentual, em cada um dos comprimentos de onda) podem ser utilizadas para separar compostos de interesse, ou formar classes de moléculas que formam determinado padrão, de acordo com os níveis de absorção em comprimentos de onda diferentes. Como cada uma das amostras é representada por um vetor de valores (absorção, em um determinado comprimento de onda) é necessário o uso de técnicas de análise que observem mais de uma das variáveis para se fazer a separação das classes ou as análises que interessam ao químico.

2.2 Análise Multivariada

A análise multivariada é a coleção de métodos e técnicas utilizadas quando são medidas diversas características de um mesmo indivíduo ou amostra. Essas diversas

medidas são chamadas de variáveis, e os indivíduos, "unidades" ou amostras. Os métodos de análise multivariada têm sido utilizados nas mais diversas áreas abrangendo as ciências humanas, exatas e da saúde [61].

As técnicas de análise multivariada foram criadas em uma época em que era possível processar apenas pequenas ou médias coleções de dados, comparadas às atuais. Porém com a evolução computacional, o armazenamento e processamento, de uma quantidade de dados cada vez maior se tornou viável. Com isso, o novo foco para as técnicas de análise multivariada é obter resultados satisfatórios com os dados medidos, eficientemente, a partir do uso da capacidade computacional [35]. A computação, então, é utilizada como uma área macro, que permite que as outras áreas implementem as técnicas multivariadas para a solução de seus problemas.

Para que a capacidade computacional seja usada, as técnicas devem ser implementadas em algoritmos computáveis e os dados representados em tipos computacionais. Muitas técnicas de análise multivariada exigem que as variáveis sejam mensuradas em um único tipo e escala [61]. A representação dos dados é feita em notação matricial, usualmente, com a_{ij} para indicar um valor particular da i-ésima amostra na j-ésima variável medida, e os valores são normalmente numéricos na matriz de dados A.

A análise multivariada pode ser utilizada com os seguintes intuitos: redução dos dados, ou simplificação estrutural; ordenação e agrupamento de amostras semelhantes; investigação da dependência das variáveis; predição; e construção e teste de hipóteses [21]. Rencher divide as atividades e técnicas de análise multivariada em duas grandes áreas: estatística descritiva e estatística inferencial, onde na estatística descritiva são obtidas informações dos dados medidos e a estatística inferencial apresenta provas de hipóteses a partir de testes [61]. Uma das técnicas mais importantes e mais utilizadas no ramo da quimiometria (que é a aplicação de métodos estatísticos na área química) é a classificação multivariada de dados [54].

2.3 Classificação e Discriminação Multivariada

Suponha um conjunto de dados Z (representado pela matriz A) de observações multivariadas (amostras com medições analisadas no conjunto dos números reais), suponha também que cada amostra X pertença a uma (e somente uma) de K classes pré-definidas e que os elementos de cada classe possuem características em comum. X pode ser visto como um vetor linha de A. Estas classes podem ser identificadas, e rotuladas de forma a se tornarem distinguíveis, associando-se um único identificador a cada classe. Cada amostra pode ser rotulada como pertencente à uma, ou várias, das classes identificadas [35].

As técnicas multivariadas, que atendem a estas situações, possuem duas finalidades principais, a primeira é separar as amostras de uma população em conjuntos pré-definidos, a segunda é alocar novos objetos nos grupos definidos anteriormente [36]. Os principais objetivos das técnicas de discriminação e classificação multivariada são: apresentar uma descrição da discriminância dos grupos com funções lineares das variáveis para se descrever, ou elucidar, as diferenças entre dois ou mais grupos; e alocar amostras em grupos pré-definidos com funções lineares ou quadráticas [60].

Uma das metas das funções discriminantes (discriminação) é encontrar o quanto uma variável contribui, relativamente, para se construir um modelo de separação, ou agrupamento, e encontrar o plano ótimo em que os pontos podem ser projetados para melhor apresentar como os grupos estão configurados no espaço. Já na classificação, as amostras são alocadas nos grupos, por funções de alocação, com base em suas variáveis [60]. É necessário enfatizar que as funções de classificação devem também atender a novas amostras não categorizadas [36]. Uma função que separa as amostras pode servir como um classificador, e reciprocamente, uma regra de associação usada para se classificar as amostras pode servir como procedimento discriminatório, com isso a diferença entre separação e alocação não é totalmente delimitada [36]. Não há consenso entre os autores sobre qual o termo mais adequado para cada um dos objetivos, a união dos dois objetivos é normalmente chamado de análise discriminante [60]. A análise discriminante é geralmente utilizada para atividades de predição de classes [35].

Para que os objetivos da análise discriminante sejam realizáveis, é necessário que os valores contidos nas variáveis que formam uma amostra apresentem informações relevantes para tal fim [36]. Assim, as variáveis devem apresentar algum padrão em comum nas amostras de uma mesma classe e diferir quando comparado às amostras de outras classes. Quando os valores observados pelo classificador para se fazer a classificação não são suficientemente diferentes, as classes não são distinguíveis e uma nova amostra poderá ser rotulada como pertencente à qualquer uma delas, sem um critério válido [36].

A função de classificação, ou classificador, funciona como uma combinação das variáveis de entrada. Para se realizar a classificação em uma coleção que possua duas classes ($K = 2$), é necessário um único classificador, já para uma coleção que possua K classes ($K > 2$) necessita-se de no mínimo dois (e no máximo $K - 1$) classificadores para discriminar a coleção e classificar uma amostra (prever à qual classe uma amostra pertence) [35].

Para ser possível a classificação de uma amostra em classes diferentes, as p variáveis randômicas de X devem diferir de alguma forma, de uma classe à outra [36]. Para se classificar deve-se obter um modelo dos vetores de observação de cada grupo, e comparar a amostra a cada um destes modelos [60]. Assim, as p variáveis de X são

suscetíveis a serem escolhidas para auxiliar à distinção entre as classes [35]. Uma das abordagens para a classificação é comparar a amostra (vetor X) que se deseja classificar com os vetores $\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_k$ ($\bar{X}_i$ - vetores obtido pela média dos indivíduos da i-ésima classe) e classificá-lo ao grupo em que o $\bar{X}_i$ é o mais próximo de X [60].

Considerando o problema em classificação de amostras em uma de K classes, onde uma amostra pertence a uma (e somente uma) das K classes ($classe_1$, $classe_2$, ..., $classe_K$). E seja [35]:

$$\rho(X \in classe_i) = \pi_i, i = 1, 2, \ldots, K, \tag{2-2}$$

a probabilidade inicial da amostra selecionada aleatoriamente $x = X$ pertencer à cada uma das classes. As classes podem ser descritas com a seguinte função de densidade [35]:

$$\rho(x = X | x \in classe_i) = h_i(X), i = 1, 2, \ldots, K \tag{2-3}$$

Pode-se então obter da equação (2-2) e (2-3) a seguinte probabilidade posterior [35]:

$$\rho(classe_i|x) = \rho(x \in classe_i | x = X) = \frac{h_i(X)\pi_i}{\sum_{k=1}^{K} h_k(X)\pi_k}, i = 1, 2, \ldots, K \tag{2-4}$$

Para um dado X, é razoável classificar-lo à classe com maior probabilidade posterior (*Teorema de Bayes*). Assim, como o denominador da equação 2-4 é igual para todas as classes, X será classificado à $classe_i$ se [35]:

$$h_i(X)\pi_i = \max_{1 \leq j \leq K} h_j(X)\pi_j \tag{2-5}$$

Assim, x é associado à $classe_i$, se $h_i(x)\pi_i > h_j(x)\pi_j$ para todo $j \neq i$.

2.4 Análise Discriminante Linear

A classificação de um elemento pode ser dada com base na comparação do vetor X com os vetores $\bar{X}_1$, $\bar{X}_2$, ..., $\bar{X}_k$ (centróides de cada uma das classe - vetores média) e classificar a amostra X ao grupo i em que $\bar{X}_i$ é mais próximo de x [60], mapeando a distância à função de probabilidade. A chave da abordagem de Mahalanobis tem a ver com a sua definição de distância, em que ele sugere que no lugar de ser usada a distância euclidiana, seja usada uma métrica ajustada à covariância das medidas. E a distância, de mahalanobis, ao quadrado da amostra X à $classe_i$ ajustada à covariância é dada por [7]:

$$D_i^2 = (x - \bar{x}_i)' \Sigma^{-1} (x - \bar{x}_i) \tag{2-6}$$

onde X é o vetor que representa a amostra, $\bar{x}_i$ é o vetor média na i-ésima classe, e Σ é a matriz de covariância conjunta de todas as classes, que pode ser obtida através de uma média das matrizes de covariância de cada classe. A amostra então é classificada ao grupo mais próximo, ou seja quando D_i^2 é minimizado. Essa regra de classificação pode ser modificada para levar em conta conhecimentos prévios desiguais. Se os dados estão distribuídos de acordo com uma distribuição normal multivariada, a densidade de probabilidade é proporcional à distância exponenciada de Mahalanobis, ou seja [7]:

$$\rho(classe_i|X) \propto \exp(\frac{D_i^2}{2}) \tag{2-7}$$

Da equação (2-7) e da (2-4) pode se obter [7]:

$$\rho(classe_i|X) = P(X \in classe_i|X = x) = \frac{\exp(\frac{D_i^2}{2})\pi_i}{\sum_{k=1}^{K}\exp(\frac{D_k^2}{2})\pi_k}, i = 1,2,\ldots,K \tag{2-8}$$

E de 2-9 tem-se:

$$h_i(x)\pi_i = \max_{1\leq j\leq K}\exp(\frac{D_j^2}{2})\pi_j \tag{2-9}$$

Assim, x é associado à $classe_i$, se $\exp(\frac{D_i^2}{2})\pi_i > \exp(\frac{D_j^2}{2})\pi_j$ para todo $j \neq i$.

O termo quadrático $x'\Sigma^{-1}x$ é independente da classe e a superfície de decisão se torna linear em X. Por isso, essa técnica de classificação é chamada Análise Discriminante Linear (LDA - Linear Discriminant Analysis) [9].

2.4.1 Risco de classificação

Na análise discriminante linear a probabilidade de determinado indivíduo pertencer a uma classe é totalmente relacionada a distância entre este indivíduo e a classe a que ele pertence, e à distância entre ele e as outras classes (aos centróides das outras classes). Assim, o risco de se classificar errado uma amostra advém da relação entre a distância dela ao centróide de sua classe e a distância dela ao centróide da classe mais próxima à ela. Em termos matemáticos o risco g_i da amostra i ser classificada erroneamente é dada pela equação (2-10):

$$g_i = \frac{D^2(X^i,\mu_{j'})}{min_{j\neq k}D^2(X^i,\mu_j)} \tag{2-10}$$

onde o numerador é a distância de Mahalanobis entre o objeto X^i e a média $\mu_{j'}$ dos indivíduos da classe a que X^i realmente pertence (ambos vetores linha). O denominador

é a distancia entre o objeto X^i e a média μ_j dos indivíduos da classe a que X^i está mais próximo (diferente da classe a que ele pertence). Assim, se g_i é maior que 1 indica que ele foi classificado erroneamente e quanto mais próximo de 1, maior é a chance desta má-classificação ter occorido.

Risco médio de classificação

Um valor que pode ser utilizado como métrica de avaliação de uma solução é a média do risco de classificação que o modelo apresenta em cada uma amostra em uma coleção de testes, chamada aqui de risco médio de classificação. O risco médio de classificação pode ser obtido a partir da equação (2-11):

$$f_R(ind) = \frac{1}{N}\sum_{i=1}^{N} g_i \tag{2-11}$$

onde N é o tamanho da coleção, e g_i é o risco de classificação da i-ésima amostra. Em estratégias evolutivas este valor pode ser utilizado como função aptidão (quanto menor melhor é a solução).

2.4.2 Seleção de Variáveis

Conjuntos de dados que possuem grandes dimensões frequentemente contém pares de variáveis altamente correlacionados e por isso colineares [35]. As variáveis colineares apresentam covariância nula e introduzem colinearidade também à matriz de covariâncias, tornando-a singular (determinante nulo e por isso sem matriz inversa). Já variáveis quase colineares introduzem à matriz de covariância autovalores muito baixos, e autovalores instáveis.

Como visto, a função discriminante responsável pela classificação na LDA, contém a matriz inversa da covariância das classes (Σ^{-1}). Isso implica que a matriz Σ não deve ser singular, e seus autovalores devem ser estáveis. O menor autovetor e autovalor, se instáveis, trarão muita influência à distância usada como estimativa de probabilidade. Assim, a estabilidade dos autovalores e autovetores são importantes para os métodos de classificação e predição [56]. Um outro limitante à função discriminante é que, para o cômputo da inversa, o número de amostras deve ser maior ou igual ao número de variáveis selecionadas [9], de outra forma, a matriz de covariâncias será singular.

Uma técnica utilizada para amenizar a influência, advinda da instabilidade dos menores autovalores e autovetores, é a regularização. A regularização consiste na substituição da matríz Σ por $(\Sigma + \gamma I)$, com o intuito de diminuir a proporção entre o maior e menor autovalor de Σ [51]. Porém segundo Friedman, esta regularização pode não ser suficiente quando o número de variáveis excede ao número de amostras, e a regularização

é totalmente dependente do fator γ, que quando mal ajustado pode introduzir tendências ao classificador [24].

Outra abordagem para o problema de colinearidade é a análise de quais variáveis são realmente importantes para o classificador, pois, do conjunto total de variáveis, existem as que ajudam na classificação e variáveis desnecessárias, que não contribuem na alocação [60]. Assim, a redução de dados, ou os métodos de seleção de variáveis, para estes problemas de classificação são uma prioridade [35]. A remoção das variáveis redundantes pode, de fato, ser responsável por melhorar o classificador. A redução de p (número de variáveis) pode, ainda, aumentar a robustez de classificadores que se utilizam de funções lineares ou quadráticas [60].

O número de subconjuntos possíveis com p variáveis é $(2^p - 1)$ [35], o que segundo [74] é um problema intratável, pois o tempo de execução do algoritmo para a análise de cada um dos caminhos cresce muito rapidamente à medida que p cresce. Torna-se então necessária uma heurística de busca, no espaço de solução, que otimize a classificação. As heurísticas apresentadas por Izenman [35] são: os algoritmos *stepwise*, o algoritmo *Least-angle Regression*, os algoritmos *Branch-and-bound* e o algoritmo *Forward-Stagewise Regression*. Outra abordagem pode ser otimizar a escolha do subconjunto de variáveis com técnicas de inteligência artificial, incluindo métodos baseados nos processos naturais.

Os algoritmos que compõem a computação evolutiva (EC - Evolutionary Computation), também chamados algoritmos evolutivos (EA - Evolutionary Algorithms), são os algoritmos utilizados para otimização, ou aprendizagem, modelados para possuir a habilidade de evoluir [79]. Os algoritmos evolutivos buscam a otimização das soluções baseados em um poderoso princípio: a sobrevivência do mais apto e são modelados de acordo com o fenômeno natural da herança genética e evolução darwiniana [49]. O tipo mais conhecido de algoritmos evolutivos é o algoritmo genético [38] e seu uso mais comum, em problemas de classificação, está relacionado à escolha das variáveis (ou atributos) [77].

Os algoritmos genéticos propostos por Holland, na década de 1960, enfrentaram uma barreira tecnológica para serem utilizados. Os computadores da época não possuíam a capacidade computacional requerida para serem executados em tempo razoável [43]. Por isso, só começaram a ser aplicados de fato na década de 1990, quando o tempo computacional gasto parou de ser um empecilho [44]. A vantagem no uso de técnicas heurísticas em relação aos métodos tradicionais aumenta de acordo com a complexidade do problema [44].

2.5 Trabalhos Relacionados

Lavine usa algoritmos evolutivos para a seleção de variáveis, buscando o subconjunto que otimize a classificação das amostras de madeira em três classes distintas (Macia, Dura e Tropical) por meio de análise de componentes principais em dados espectrais obtidos por espectroscopia de Raman. Essa otimização ocorre com a busca do subconjunto de variáveis menos semelhantes para a formação de dois componentes principais, que serão utilizados para a separação das amostras por análise de componentes principais (PCA - *Principal Component Analysis*). A função de aptidão do algoritmo evolutivo é baseada em uma pontuação para cada amostra e classe, e o número de "vizinhos mais próximos"de cada amostra (vetores mais semelhantes às amostras) dados pelo algoritmo K-NN (K-*Nearest Neighbors*) [42].

O algoritmo evolutivo conseguiu reduzir a dimensionalidade dos dados de 670 variáveis à 8, e possibilitou a classificação nas 3 classes, o que não era possível de ser efetuado com 2 componentes principais e 670 variáveis. A função de aptidão para o algoritmo genético é um escore de locação, baseado em uma pontuação para cada classe, uma pontuação para cada amostra e no número de vizinhos próximos de cada classe, utilizando-se de valores ajustados à cada geração para a formação destas pontuações. Percebe-se claramente que a estratégia utilizada tem dois objetivos bem definidos, para uma amostra, agrupados em uma única função de aptidão: diminuir a distância entre os indivíduos de uma mesma classe, e aumentar a distância entre as classes. Assim uma estratégia multi-objetivo poderia ajudar na solução deste problema sem o uso das constantes [42].

Silva [65] compara o uso de algoritmos evolutivos, algoritmo *stepwise*, e algoritmo de projeções sucessivas, para a escolha de variáveis, com intuito de prover a classificação de canetas azuis, em marcas e tipos, por meio de análise discriminante linear. Os dados são obtidos por espectroscopia no infravermelho. O algoritmo evolutivo utiliza-se do "risco médio de classificação" como função de aptidão e a melhor de 3 execuções do algoritmo seleciona 15 variáveis para a classificação de tipos, e 59 para a classificação de marcas. Os resultados são analisados em 3 tipos de papeis. Os algoritmos analisados apresentam mais de 91% de correção nos 3 tipos de papéis para a classificação de tipos e marcas sendo que o uso de algoritmos genéticos para a seleção de variáveis apresentou resultados (na predição de novas amostras) no mínimo iguais às outras abordagens. Pontes [59] utiliza-se do algoritmo das projeções sucessivas como meio de seleção de variáveis para a classificação por análise discriminante linear e análise discriminante por mínimos quadrados parciais, para fazer a detecção de adulteração de diesel/biodiesel com dados de infravermelho próximo. A função objetivo do algoritmo das projeções sucessivas é também o "risco médio de classificação", função que otimiza a separação entre as

classes.

Silva [65] e Pontes [59] utilizam a análise discriminante linear como classificador e avaliam as soluções de acordo com o risco médio de classificação. Este método de avaliação, das soluções, ignora o tamanho do subconjunto escolhido e a quantidade de variáveis correlacionadas incluídas ao modelo. Segundo Dash [11], a solução ideal é constituída pelo subconjunto mínimo de variáveis que não pioram significativamente o classificador.

Capítulo 3 - Algoritmos Evolutivos

Algoritmos evolutivos (EAs) são uma heurística de busca de solução moderna, baseados no princípio da evolução pela sobrevivência do mais apto [50]. Tiveram origem em 1930, quando Wright [78] apresentou uma abordagem dos sistemas evolutivos naturais como algoritmos de exploração de múltiplos picos com uma função objetivo única [25], porém só começaram a existir na década de 1960 quando a tecnologia digital dos computadores deixou de ser tão cara [13].

Os EAs buscam imitar o processo de evolução natural apresentada por Darwin em "A Origem das Espécies" [10], com o intuito de desenvolver algoritmos para problemas de adaptação (aprendizagem [79]) e otimização[3]. O processo de evolução das espécies pode ser visto como um processo de otimização do nível de adaptação da espécie ao ambiente [79]. O problema de busca de solução é um tipo de otimização, pois seu intuito é encontrar a "melhor solução"ao final do processo de busca [50].

Os problemas de aprendizagem e de otimização normalmente estão relacionados. De um lado, a função que necessita ser otimizada pode ser tão complexa a ponto de não ser possível calcular o valor objetivo para cada possível solução, assim os algoritmos de aprendizado podem ser usados para calcular um valor aproximado, utilizando-se de funções simplistas, para o cálculo do valor objetivo, baseadas em soluções avaliadas anteriormente. Por outro lado, nos problemas de aprendizagem (agrupamento ou regressão) pode se tomar uma função que apresenta a performance do classificador (nos problemas de agrupamento) ou uma função que apresente o erro de predição (nos problemas de regressão) e otimizar o valor objetivo desta função [79].

O problema de otimização pode ser visto como a busca do conjunto $\bar{s} = (s_1, \ldots, s_n) \in M$ de parâmetros do sistema que está sendo considerado, tais que um certo critério de qualidade $f : M \longrightarrow \Re$, chamado normalmente de função objetivo, é maximizado (ou minimizado, conforme a aplicação) [3], ou seja:

$$f(\bar{s}) \longrightarrow \max \tag{3-1}$$

Os algoritmos evolutivos buscam encontrar o vetor $\bar{s}$ que otimiza a equação 3-

1. Embora não haja um consenso, os elementos que compõem um sistema evolutivo englobam: uma, ou mais populações de indivíduos competindo por recursos limitados; o conceito de aptidão que reflete na capacidade do indivíduo sobreviver e se reproduzir; a noção de mudanças populacionais ocorridas pela morte e nascimento de indivíduos; e o conceito de herança em que as proles são parecidas com seus pais, porém não idênticas [13]. Disso resultam três características principais [79]:

- **São baseados em população**. Os EAs mantém um grupo de soluções, chamados população, e buscam aprimorá-la em processo iterativo estocástico.
- **São orientados a adaptação**. Cada elemento da população é chamado indivíduo (que é um espécime da solução). Cada indivíduo tem uma representação genética, chamada gene, e um nível de aptidão, chamado "valor aptidão" (ou $fitness$). Os algoritmos evolutivos dão preferência aos indivíduos mais aptos e isto é a base da otimização e o motivo de se haver convergência.
- **São guiados pela variação**. A modificação dos indivíduos (ou recombinação entre eles) será efetuada por meio de operadores que realizam trasnformações na solução (operadores genéticos), imitando assim as mudanças genéticas ocorridas no meio natural, esta mudança é quem provê a busca no espaço de soluções.

Em geral, não há necessidade de igualdade entre os valores dados pela função objetivo e o valor aptidão de um indivíduo, mas a função objetivo deve sempre compor a função aptidão [4]. A função de aptidão pode apresentar a qualidade da solução representada pelo indivíduo ou pode ser a proximidade do indivíduo à solução esperada [13].

3.1 Funcionamento de um EA

O EA mantém uma população de indivíduos, $P(t) = I^t_1, \ldots, I^t_n$ na iteração t, em que cada indivíduo mantém uma representação de uma solução potencial para o problema em questão, e deve ser implementado em alguma estrutura de dados [50]. A população tem um tamanho fixado, e cada indivíduo é representado por um vetor (cromossomo) de valores reais (genes), e possuem um nível de aptidão dado pela função de aptidão. O vetor é utilizado para representar as características do indivíduo que são potencialmente relevantes para estimar o nível de aptidão do indivíduo. Outras representações podem ser necessárias em problemas cujo espaço de busca de solução é não linear, com soluções de tamanho variável, ou ainda não linear de tamanho variável [13].

Cada indivíduo I^t_i é avaliado pela função de aptidão e é medido o nível de aptidão para cada solução do problema. A nova população (iteração t+1), então, será constituída pelos indivíduos com maior nível de aptidão (seleção) na população atual [50], a população inicial é gerada de forma aleatória [13]. A cada geração são escolhidos alguns

indivíduos que serão submetidos à transformações (variação) por meio de operadores genéticos para a formação de novas soluções (indivíduos) [50]. A transformação se dá pelo processo de mutação: escolhe-se um membro da população atual, chamado de pai, e este produz uma cópia, chamada de prole (ou criança), que é um clone do pai; escolhe-se então um, ou mais, genes aleatórios no cromossomo e modifica-o pela soma ou subtração de um valor ao valor atual; o novo indivíduo (criança) então é avaliado e é forçado à competir com um membro existente da população (adulto), se a criança apresentar um nível de aptidão maior que o adulto o adulto morre e a criança o substitui na população, senão, a criança morre e não é agregada à população [13]. Outra transformação chamada cruzamento seleciona dois ou mais pais para a formação de um único indivíduo pela combinação de partes de seus cromossomos [50].

O algoritmo 3.1 representa o processo de um EA comum [3].

Algoritmo 3.1: EA genérico.

1. $t \leftarrow 0$; `// Inicialização do contador de gerações`
2. $Inicialize(P(t))$ `// gera aleatoriamente os indivíduos da população inicial`
3. $Avalia(P(t))$; `// é aplicada a função` $fitness$ `na população`
4. **Enquanto não** condição-de-finalização **faça** `// teste do critério de parada (um critério de saída deve ser definido)`
   $t \leftarrow t+1$ `// contador de gerações é atualizado`
   $P(t) \leftarrow Seleciona(P(t-1))$ `// seleção da nova população, ou criação dos clones`
   $Altera(P(t))$ `// são aplicados os operadores genéticos`
   $Avalia(P(t))$; `// avaliação dos novos indivíduos de P segundo a função adequação.`
   $eliminar\ os\ inaptos(P(t))$; `// morte dos novos indivíduos ou dos indivíduos que já compunham a população segundo o nível de aptidão`
5. **Fim Enquanto**

3.2 Abordagens de EA

As três correntes principais de métodos (ou algoritmos, ou ainda metaheurísticas), estabelecidos nestes últimos 50 anos são: os algoritmos genéticos, a programação evolutiva e as estratégias evolutivas [3]. Ainda existem outras abordagens, como é caso da programação genética e algoritmos híbridos, que mesclam as características de mais de uma técnica [50].

Cada um dos três algoritmos têm se mostrado capaz de produzir soluções aproximadamente ótimas dados espaços de busca complexos, multimodais, não diferenciais e

descontínuos. Eles têm também alcançado sucesso em panoramas dependentes do tempo e sensíveis a ruído [38]. Abaixo são apresentadas cada uma das abordagens de EA, tendo um maior nível de detalhes nos algoritmos genéticos.

3.2.1 Algoritmos Genéticos - GAs

Os algoritmos genéticos, foram inventados por Holland na década de 1960, e desenvolvidos por ele, seus alunos, e colegas, na Universidade de Michingan nas décadas de 1960 e 1970 [52]. Ele enfatizou em seus trabalhos a necessidade de sistemas com a capacidade de se auto adaptar, ao passar do tempo, em função da experiência adquirida pela interação com o ambiente em que eles operam [13]. O intuito desta abordagem de algoritmos evolutivos era a criação de algoritmos que não fossem modelados à problemas específicos, mas que o estudo da capacidade dos processos evolutivos naturais propiciassem um caminho para a "importação" destes processos pelo computador [52]. Essa abordagem levou a família inicial de "planos reprodutivos" que são a base dos algoritmos genéticos simples de hoje [13].

Representação e avaliação

Os algoritmos genéticos codificam o problema com indivíduos no formato de vetores binários de tamanho fixo [38]. Assim os cromossomos são codificados como cadeias de bits, cada gene é um bit e é uma instância particular de um alelo (isto é, 0 ou 1) [52]. Outras representações podem ser utilizadas mas a proposta original de GAs foi com o código binário, imitando o código genético dos organismos naturais [79], e a força da preferência por esta representação está na teoria esquemática de algoritmos genéticos, que tentam analisar os GAs em termos de seu comportamento amostrado [3].

Para funções objetivo pseudo-booleanas a notação binária pode ser utilizada diretamente, já para problemas de otimização com funções objetivo formadas por parâmetros contínuos a cadeia de bits é logicamente dividida em segmentos de mesmo tamanho, e cada segmento é interpretado como o código binário correspondente a variável objetivo (que está no intervalo de valores dos parâmetros, muitas vezes em intervalos de números reais) [4]. Fixada a notação binária, o uso dos algoritmos genéticos frequentemente obrigam a utilização de funções de codificação e decodificação. A decodificação de cada segmento é obtida transformando o código binário em um número inteiro no intervalo entre 0 e $2^{(tamanho\ da\ string)}$-1 e fazendo um mapeamento linear deste inteiro à um intervalo real $[u_i, v_i]$; a codificação é obtida pelo mapeamento inverso [2]. É importante notar que um intervalo real não pode ser representado com vetores binários de tamanhos limitados (fixados), então deve-se levar em conta que haverá uma aproximação na conversão dos vetores ao intervalo real [79].

A função de aptidão têm como entrada os parâmetros representados na solução, ou seja cada indivíduo representa uma lista de parâmetros de uma solução, e a função aptidão mede a qualidade desta solução. Na notação apresentada em GAs, a função de aptidão decodifica inicialmente os genes nos parâmetros da função (salvo em funções pseudo-booleanas), e resulta num número real que representa o valor objetivo, este valor então é analisado para se verificar a qualidade da solução [4]. Normalmente é necessário escalar o valor obtido na função de aptidão, visto que os valores obtidos pela função objetivo devem ser mapeados a valores reais, e indivíduos mais aptos devem possuir maiores valores de aptidão (*fitness*). Bäck (1996) [2] apresenta os métodos que são mais utilizados para escalar os valores objetivos para as funções de aptidão, e são eles: escalamento estático linear, escalamento dinâmico linear, escalamento logarítmico, e escalamento exponencial [2].

Inicialização

Devem ser definidos inicialmente o critério de parada, o número de indivíduos,e as restrições de cada indivíduo. Quando são conhecidas boas soluções, e a população inicial pode ser constituída de indivíduos que representem estas soluções. Porém normalmente os indivíduos são gerados de forma aleatória para que a busca global não seja prejudicada pelas restrições e pelos tipos de soluções previamente estabelecidas [79]. Assim cada gene que forma os cromossomos (que representam os indivíduos da população) é gerado de forma aleatória [2].

Operadores Genéticos

Os operadores genéticos em GAs são a recombinação e a mutação [3]. A recombinação ocorre quando existe troca de informação entre dois ou mais indivíduos, ou seja há uma troca de genes entre os indivíduos, a recombinação também é chamada cruzamento (crossover), e a mutação quando o gene de um indivíduo muda por si próprio [79]. Há uma ênfase no operador de recombinação [3], e este é o maior mecanismo de exploração de um GA e é o responsável pela busca global no espaço de soluções [79].

A mutação ocorre quando um operador aleatório escolhe uma posição de um único indivíduo e a inverte [52], ou cria-se um vetor de bits aleatórios do tamanho do cromossomo e inverte-se a posição do cromossomo dependendo do valor encontrado no vetor criado [2]. As pequenas mudanças ocorridas no indivíduo após a mutação são responsáveis pela busca local no espaço de soluções [79].

Como no mundo real a probabilidade de mutação por bit deve ser bem pequena [52]. Essas probabilidades tem configurações que variam, normalmente, entre $0,1$ e 1%, em populações grandes (mais que 200 indivíduos) é sugerida uma probabilidade de

mutação baixa (menor que 5%), e em populações pequenas (menos que 20 indivíduos) é sugerido que a probabilidade de mutação não seja muito baixa (menor que 0,2%) [2]. A mutação não só aprimora a busca da solução, quanto também provê uma segurança contra uma população muito uniforme, a ponto de não evoluir, com novas gerações [31]. Porém essa mutação deve ser pequena, para que o indivíduo obtido pela mutação não seja muito diferente geneticamente de seu ancestral [2]. Após a mutação o indivíduo é chamado de mutante [79].

A recombinação nos algoritmos genéticos, ocorre com o cruzamento, que é o operador responsável por recombinar segmentos úteis de indivíduos diferentes [4]. O *crossover* é o operador sexual que a partir de uma probabilidade (ou outro método de escolha) escolhe, na população, dois indivíduos que serão os pais, e recombina-os formando dois novos indivíduos [2]. Os valores típicos para a probabilidade de *crossover* variam entre 60 e 100% [4].

Nem todos os indivíduos da população participam do processo de reprodução. Primeiramente são selecionados alguns indivíduos da população (de acordo com a probabilidade de cruzamento), e estes são atribuídos ao "grupo de acasalamento", depois é criada uma associação de pares entre os indivíduos, e essa associação pode se dar de duas formas: a primeira é "misturar" os pais e fazer associação de cada indivíduo da população à outro, de forma que o indivíduo de índice 1 seja associado ao de índice 2, o de índice 3 ao de índice 4 e assim por diante, e este par será utilizado para o *crossover*; a segunda é gerar uma função de permutação aleatória, *per*, tal que $per(i) = j$, i e j pertencem ao intervalo $[1,d]$ (onde d é o tamanho do grupo de acasalamento), associando o i-ésimo elemento do grupo ao j-ésimo. E assim é mapeado $ind_{per(1)}$ e $ind_{per(2)}$ sem reposição como o primeiro par, $ind_{per(3)}$ e $ind_{per(4)}$ sem reposição como o segundo par, e assim por diante [79]..

O *crossover* trabalha totalmente na representação binária (no cromossomo), ignorando se a representação dos indivíduos é dada por segmentos, ou outra forma de representação [4]. O algoritmo padrão para *crossover* é o *crossover* em um único ponto onde é gerado aleatoriamente um inteiro w, no intervalo $[1,L-1]$ onde L é o tamanho do cromossomo [79], esse ponto é chamado ponto de cruzamento [3]. Os genes após o ponto w são trocados pelos pais e os cromossomos resultantes são chamados de filhos, ou proles [79]. Existem outras possibilidades, onde as principais são: a troca com mais de um ponto de cruzamento (geralmente com 2 pontos de cruzamento) e o o crossover uniforme [73], onde cada alelo é selecionado de um pai, ou outro, com uma probabilidade uniforme [1] [37]. Na seção Operadores Reais (Seção 3.3), são apresentadas outras formas de cruzamento e mutação com números reais, é também apresentada outra forma de recombinação com mais de 2 indivíduos que [1] denomina como forma panmítica.

Seleção

O mecanismo de seleção nos algoritmos genéticos é a regra de sobrevivência probabilística [2]. A probabilidade de sobrevivência é baseada no valor de *fitness* relativo dado pela seguinte fórmula [79]:

$$\nu_i = \frac{f_i}{\sum_{j=1}^{l} f_j} \tag{3-2}$$

onde ν_i é a probabilidade de sobrevivência do i-ésimo indivíduo e f^i e f^j são os valores dados pela função *fitness* no i-ésimo e j-ésimo indivíduos respectivamente. Essa é probabilidade de um indivíduo sobreviver em uma geração, ou por [79] a probabilidade de um indivíduo fazer parte do grupo de acasalamento.

O algoritmo mais comumente utilizado para fazer a seleção é o método da roleta viciada [2]. Primeiramente é construída uma roleta com setores de tamanhos proporcionais à probabilidade de sobrevivência de cada indivíduo [47]. A maior probabilidade corresponde ao maior setor na roleta [79]. Na seleção da roleta viciada, é construída uma linha para representar a roleta, e esta linha tem tamanho igual a soma de todos os valores *fitness*, após isso é gerado de forma aleatória um número entre 0 e o tamanho da roleta, e se localiza o setor correspondente, selecionando assim o respectivo indivíduo [47].

A forma apresentada por Yu [79] é a simulação de uma roleta real, com a diferença de que os setores possuirão tamanhos diferentes - de acordo com a probabilidade de sobrevivência. Cada setor da roleta identifica um indivíduo e tem um ângulo central dependente de sua probabilidade de sobrevivência. A seleção é realizada simulando o processo de rotação do ponteiro da roleta: um número aleatório (entre 0 e 1) é gerado e multiplicado por 2π; é então simulada a escolha do setor; o indivíduo correspondente ao setor é selecionado (o setor é encontrado pela acumulação das probabilidades da origem até a posição em $2\pi rand$) e colocado no grupo de acasalamento. Assim, quem tem maior probabilidade de sobrevivência tem maior probabilidade de ser selecionado.

Outra forma de seleção é a seleção por torneio, onde são sorteados m (usualmente $m = 2$) indivíduos que concorrerão para se tornarem possíveis pais. Assim são selecionados aleatoriamente dois indivíduos, o que possuir maior fitness será escolhido como primeiro pai, e então são selecionados ι indivíduos que também concorrerão entre si, e o com melhor *fitness* será o segundo [25].

3.2.2 Estratégias Evolutivas

Os estudos com estratégias evolutivas (Evolutionary Strategy - ES) foram iniciados por Rechenberg e Schwefel, na década de 1960 na Universidade Técnica de Berlin, Alemanha [1]. O foco inicial desta abordagem é a otimização de funções com múltiplos

parâmetros reais [13], e são comumente utilizadas em problemas de otimização "caixa-preta"contínuos [29].

As Estratégias Evolutivas representam um indivíduo, por três componentes: o primeiro é o vetor de parâmetros da função objetivo; o segundo é o vetor de variâncias (armazenado para auxiliar nos operadores genéticos); e o último é um vetor de rotação (utilizado para se garantir que a matriz de covariâncias, também utilizada para fins de operações genéticas, seja invertível) [4]. A função objetivo e a função fitness expressam o mesmo resultado, em um indivíduo [4]. Assim a medida de qualidade de uma solução é totalmente dependente do valor obtida pela função objetivo.

Os operadores genéticos usados nas Estratégias Evolutivas são a mutação e a recombinação, havendo uma ênfase na mutação [79]. O operador de mutação, chamado mutação normal, efetua uma pequena diferenciação em cada gene de cada indivíduo a ser mutado, com distribuição de probabilidade Gaussiana, com média (esperança) zero e com desvio padrão do gene correspondente no pai [25].

A recombinação é utilizada para a geração de todos os indivíduos da nova população, e após esta geração são efetuados os operadores de mutação [79]. A recombinação pode agir não só no vetor de parâmetros da função objetivo, como também nos parâmetros estratégicos, e esta abordagem tem obtido melhores resultados [4]. A seleção é completamente elitista (ou seja, são selecionados apenas os melhores indivíduos de cada população - crianças e pais - dependendo do tipo de estratégia evolutiva utilizada) [25], e por isso é determinística [4]. Os mecanismos de seleção dependem da estratégia evolutiva utilizada, apresentada por sua notação. Nas Estratégias Evolutivas (μ+λ)-ESs, os λ novos indivíduos são agregados a população anterior e os λ piores indivíduos são descartados. Nas Estratégias Evolutivas (μ,λ)-ESs, λs piores pais morrem e a prole é agregada a nova população sem competição [25] [4] [2].

3.2.3 Programação Evolutiva

A programação evolutiva teve início na década de 1960 [22], sendo que os primeiros trabalhos de Fogel tiveram o intuito de utilizar os conceitos de evolução para solucionar problemas de aprendizado [79] no desenvolvimento da Inteligência Artificial [25], com uso de máquinas de estados finitos para resolver tarefas de predição [4].

O indivíduo é representado por uma máquina de estados finitos, que processa uma sequência de símbolos [25]. A qualidade de uma máquina representada pode ser medida com base da capacidade de predição da máquina, ou seja, compara-se cada símbolo de saída dado com o símbolo esperado e mede-se a pior predição pela função payoff acumulada [1]. O único operador genético utilizado é a mutação, e todos os indivíduos geram novos descendentes [25]. A mutação foi implementada como uma

mudança aleatória da descrição da máquina e pode se dar por cinco tipos de mudanças: mudança em um símbolo de saída, mudança em um estado de transição, criação de um novo estado, remoção de um estado ou mudança do estado inicial [1].

O processo de mutação tem similaridades com o ocorrido nas estratégias evolutivas e as versões "mais elaboradas"de Programação Evolutiva mantém as variâncias no cromossomo para facilitar a auto-adaptação dos parâmetros [2]. A seleção é também totalmente estocástica e funciona mesma forma que em uma $(\mu + \mu)$-EE, ou seja todos os pais e todos os filhos competem para a sobrevivência, em cada geração, e apenas metade deles sobrevivem [4].

3.3 Operadores Reais

Os operadores genéticos de mutação e crossover foram detalhados anteriormente para os algoritmos genéticos, porém os GAs codificam os genes de uma solução com um conjunto muito restrito, o conjunto binário.

Para problemas do mundo real, de otimização, é natural expressar o cromossomo com as entradas da função à ser otimizada, ou seja genes reais [79]. No caso de otimização por Programação Evolutiva ou nas Estratégias Evolutivas, um dos componentes da representação genotípica são os parâmetros ($\in \Re$) da função a ser otimizada [1] [79].

3.3.1 Mutação

O processo de mutação ocorre em duas etapas: o pai é clonado, e são feitas alterações em seu cromossomo pela modificação de um ou mais genes na estrutura do filho (mutante) [13]. Esta mutação, assim como nos Algoritmos Genéticos, é condicionada a um fator chamado taxa de mutação [25], e no caso de números reais também está condicionada ao montante de mutação que será aplicado em um gene, esse montante normalmente é associado à uma métrica de distância, tal como a distância Euclidiana [13].

São exemplos de mutações que podem ocorrer em cromossomos com base nos números reais [79]:

- **Mutação Uniforme**. São definidos os limites (superior e inferior) e então é gerado um número aleatório entre o limite inferior e superior para cada gene - de acordo com a probabilidade de mutação.
- **Mutação por Fronteira**. Similarmente a mutação uniforme são definidos os limites de um gene e então é escolhido ou o limite superior ou o inferior para substituir o valor atual de forma aleatória.
- **Mutação não Uniforme**. É similar ao método da têmpera simulada, onde ocorrem mutações e essas são verificadas se melhoram ou pioram o indivíduo.

- **Mutação Normal**. Na mutação normal a mutação ocorre de forma dependente ao desvio padrão, sendo este, único para a variável ou geral. Na mutação normal o desvio padrão deve ser atualizado à cada geração.

3.3.2 Recombinação

O mecanismo clássico de reprodução é a recombinação de dois pais, em que subcomponentes dos cromossomos (genomas) dos pais são clonados e remontados para criar o cromossomo de um filho (ou vários) [13]. A taxa de reprodução diz respeito à quantidade de indivíduos da população atual que farão parte da população de pais (ou grupo de acasalamento) [79].

Na reprodução por recombinação em conjuntos reais, dois princípios devem ser satisfeitos: a preservação das estatísticas dos pais, pela prole, como a média, variância e covariância; e a diversidade da prole, onde um bom operador de cruzamento terá como resultado uma prole tão diversa quanto possível, e que satisfaça o primeiro princípio [79]. Os métodos de recombinação podem ser classificados com relação aos valores, com relação aos pontos de cruzamento e com relação ao número de pais.

Com relação aos valores os métodos podem ser classificados em recombinação discreta e recombinação intermediária. A recombinação discreta é similar à realizada em cromossomos binários (o gene de um dos pais é escolhido para o filho), e a recombinação intermediária (ou aritmética) cria novos alelos nos descendentes com valores intermediários aos encontrados nos pais [25]. O valor do gene do cromossomo filho θ_i, obtido pela associação do gene de cada um de dois pais ($\Theta 1_i$ e $\Theta 2_i$) é dado pela seguinte fórmula:

$$\theta_i = r \times (\Theta 1_i - \Theta 2_i) + \Theta 2_i \tag{3-3}$$

$$\theta_i = \chi \times (\Theta 1_i - \Theta 2_i) + \Theta 2_i \text{ [4]} \tag{3-4}$$

onde r é um binário aleatório, e $\chi \in [0,1]$ é uma variável randômica uniforme conforme Bäck [4]. A equação (3-3) é para a recombinação discreta, e a equação (3-4) é para a intermediária. Yu [79] utiliza χ fixado como $\frac{1}{2}$.

Com relação aos pontos de cruzamento a recombinação pode ser simples, individual, ou total. Na recombinação simples, assim como o crossover de único ponto (nos algoritmos genéticos), é definido um ponto de corte w e o Filho 1 é construído com os primeiros w genes do Pai 1 e com os demais $L-w$ genes restantes obtidos do valor intermediário entre o gene do Pai 1 e do Pai 2 (equação(3-4)) [25] ou os genes do Pai 2 (discreta). Na recombinação individual apenas o alelo w escolhido é modificado. Na recombinação total cada gene do filho é gerado pela associação dos genes dos 2 pais, a sua

fórmula é descrita pelas equações 3-3 e 3-4 (discreta e intermediária, respectivamente) com o *i* variando de 1 ao tamanho do cromossomo [4] [25].

Com relação ao número de pais existe o cruzamento de dois pais, que é o padrão, e o cruzamento global (forma panmítica [2]) dado pela seguinte equação:

$$\theta_i = \rho \times (\Theta_{j,i} - \Theta_{m,i}) + \Theta_{m,i} \tag{3-5}$$

$$\theta_i = \chi \times (\Theta_{j,i} - \Theta_{j,i}) + \Theta_{m,i}\ [4] \tag{3-6}$$

onde *j* e *m* são escolhidos, de forma aleatória, para cada gene (*i*) que será modificado [2].

Capítulo 4 - Algoritmos Evolutivos Multi-objetivo

Os algoritmos evolutivos multi-objetivo são uma das áreas de pesquisa mais ativas no campo da computação evolutiva [34]. Para problemas com objetivos múltiplos existe um conjunto de soluções que não são piores que nenhuma outra solução no espaço de soluções, as soluções ótimas de pareto [79]. Um algoritmo evolutivo pode encontrar um grande número de soluções ótimas de pareto de forma simultânea em uma única execução [26] por manter uma população de soluções [23]. Os algoritmos evolutivos podem, ainda, explorar características similares entre as soluções pelo cruzamento [81], se tornando uma escolha natural em otimizações multi-objetivo [26].

4.1 Otimização Multi-objetivo

A tarefa de encontrar uma, ou várias soluções, que minimiza(am) (ou maximizam) um ou mais objetivos específicos, e que satisfaz(em), quando existe(em), à(s) restrição(ões) de um determinado problema é chamada otimização [5]. De forma geral um problema de otimização pode ser dividido em diferentes categorias, dependendo do tipo de variáveis de decisão, função(ões) objetivo, e restrições, são elas [18]:

- **Programação Linear (LP - *Linear Programming*)**. As funções objetivo e restrições são lineares. Os parâmetros são escalares e contínuos.
- **Programação Não Linear (NLP - *Non Linear Programming*)**. As funções objetivo e restrições não são lineares. Os parâmetros são escalares e contínuos.
- **Programação Inteira (IP - *Integer Programming*)**. Os parâmetros são escalares e inteiros.
- **Programação Inteira e Linear (MILP - *Mixed Integer Linear Programming*)**. As funções objetivo, e restrições são lineares. Os parâmetros são escalares, alguns deles são inteiros e outros são variáveis contínuas.

- **Programação Inteira e Não Linear (MINLP - *Mixed Integer Nonlinear Programming*)**. Um problema de programação não linear envolvendo parâmetros inteiros e contínuos.
- **Otimização discreta**. Problemas envolvendo parâmetros discretos (inteiros). O que inclui: IP, MILP, MINLPs.
- **Controle otimizado**. Os parâmetros são vetores.
- **Programação estocástica, ou otimização estocástica**. Também conhecida como otimização sob incerteza. Existe a presença de variáveis aleatórias nas funções objetivo e/ou parâmetros de entrada. Todas as categorias citadas podem ser implementadas de maneira estocástica.
- **Otimização Multi-objetivo**. Problemas envolvendo mais que um objetivo. Todas as categorias citadas podem contemplar múltiplos objetivos.

Um problema de otimização multi-objetivo, com κ objetivos, pode ser representado como [79]:

$$\begin{aligned} \min\{z_1 = f_1(s), z_2 = f_2(s), \ldots, z_k = f_\kappa(s))\} \\ sujeito\ a\ \ c_i(s) \leq 0, i = 1, \ldots, q \\ \xi_j(s) = 0, j = q+1, \ldots, \kappa, \\ s \in \Re^n \\ e\ s_i^{(l)} \leq s_i \leq s_i^{(u)}\ [16] \end{aligned}$$

onde s é uma solução, f_i é o i-ésimo objetivo, c_i é a i-ésima restrição de desigualdade, ξ_j é a j-ésima restrição de igualdade [79], e $s^{(l)}$ e $s^{(u)}$ são os limites inferior e superior respectivamente. Pode-se além de existir as restrições citadas, existir ainda mínimos locais ($\overrightarrow{s}'$) com [1]:

$$\exists \varepsilon > 0 : \forall \overrightarrow{s} \in M, \|\overrightarrow{s} - \overrightarrow{s}'\| < \varepsilon \Rightarrow f(\overrightarrow{s'}) \leq f(\overrightarrow{s}) \tag{4-1}$$

Problemas de otimização de objetivo único requerem encontrar um vetor $\overrightarrow{s}^*$ tal que $\forall \overrightarrow{s} \in M$ (M é o espaço de soluções): $f(\overrightarrow{s}) \geq f(\overrightarrow{s}^*)$ [1], e portanto $\overrightarrow{s}$ é a solução ótima. No caso de otimização multi-objetivo, podem ser considerados objetivos conflitantes e pode não haver uma única solução que seja ótima para todos os objetivos. Há todavia um conjunto de soluções com diferentes compensações (são melhores em um objetivo mas piores em outro), chamadas soluções ótimas de pareto, ou soluções não dominadas [5].

4.1.1 Abordagens não baseadas em preferências: soluções ótimas de pareto

Pareto aborda o conceito de dominância, onde duas soluções podem ter três tipos de relação: uma solução é pior que a outra (inferior/é dominada), uma solução é melhor que a outra (superior/domina), ou não é comparável à outra (não inferior/não dominada) [79]. Estas relações são representadas matematicamente, em problemas de minimização (sem perda de generalidade), como [23]:

Inferioridade: Uma solução $\overrightarrow{s1} = (s1_1, \ldots, s1_n)$ é dita inferior à $\overrightarrow{s2} = (s2_1, \ldots, s2_n)$, se e somente se, $\overrightarrow{s2}$ é parcialmente menor que $\overrightarrow{s1}$ ($\overrightarrow{s2} p < \overrightarrow{s1}$), ou seja,

$$\begin{aligned} &\forall i = 1, \ldots, n, s2_i \leq s1_i \wedge \\ &\exists i = 1, \ldots, n : s2_i < s1_i \end{aligned} \tag{4-2}$$

o que é representado por $\overrightarrow{s1} \succ \overrightarrow{s2}$ ($\overrightarrow{s2}$ domina $\overrightarrow{s1}$)

Superioridade: Uma solução $\overrightarrow{s1} = (s1_1, \ldots, s1_n)$ é dita superior à $\overrightarrow{s2} = (s2_1, \ldots, s2_n)$, se e somente se, $\overrightarrow{s2}$ é inferior à $\overrightarrow{s1}$. O que é representado por $\overrightarrow{s1} \prec \overrightarrow{s2}$ ($\overrightarrow{s1}$ domina $\overrightarrow{s2}$)

Não inferioridade: Uma solução $\overrightarrow{s1} = (s1_1, \ldots, s1_n)$ é dita não inferior à $\overrightarrow{s2} = (s2_1, \ldots, s2_n)$, se e somente se, $\overrightarrow{s2}$ não é nem inferior, nem superior, à $\overrightarrow{s1}$. O que é representado por $\overrightarrow{s1} \sim \overrightarrow{s2}$.

Uma solução $\overrightarrow{s1}$ faz parte das soluções ótimas de pareto, se e somente se, não há nenhuma solução, no espaço de soluções que a domina [79]. Ou matematicamente:

$$\nexists \overrightarrow{s2} | \overrightarrow{s2} \prec \overrightarrow{s1} \tag{4-3}$$

Embora existam múltiplas soluções ótimas de pareto, na prática, apenas uma dessas soluções será escolhida [5]. A escolha, da solução realizada por um decisor, deverá se basear em um conhecimento prévio sobre o problema de otimização em questão [82]. Assim é matematicamente mais favorável a escolha de soluções, dentre as soluções ótimas de pareto, que oferecem o mínimo conflito entre os objetivos [71].

No contexto de múltiplos objetivos, o caso extremo de encontrar a única solução ótima não pode ser realizada quando os outros objetivos são também importantes, e conflitantes. A escolha de uma solução deve garantir que os ganhos desta, com relação a um objetivo, devem ser mais relevantes que as perdas em outros objetivos (dependendo da prioridade de cada um dos objetivos) e essa permuta é um empecilho para se escolher uma solução baseado em um único objetivo [16].

Estas idéias sugerem que dois objetivos devem ser almejados em uma otimização multi-objetivo [16] [81]:

- Buscar o conjunto ótimo de pareto global;
- Manter a diversidade da população a fim prevenir uma convergência prematura e alcançar uma fronteira bem distribuída de soluções com diferentes compensações

Além destes objetivos deve-se ainda escolher um único objetivo ao final do processo com um conhecimento de alto nível, o conhecimento do problema. Sendo assim o processo de busca dos algoritmos evolutivos multi-objetivo devem se atentar a duas etapas: encontrar múltiplas soluções não dominadas, tão próximos da fronteira ótima de pareto quanto possível; e escolher entre essas soluções uma única, com base nos conhecimentos de alto nível [16]. O decisor pode ser usado em três momentos distintos: antes da busca, após a busca, ou durante a busca [82].

4.1.2 Abordagens baseadas em preferências

Se existe alguma informação de preferência antes da otimização (tais como o objetivo mais importante ao decisor, ou o nível de importância de cada objetivo) pode se incluir essas preferências como requisitos de usuário nos algoritmos e gerar soluções de alta qualidade mais rapidamente [79]. A solução é então a criação de uma função de utilidade, que é uma combinação dos objetivos em um único [23].

A função utilidade padrão é formada pela combinação de alguma função pontual dos múltiplos atributos para formar uma função fitness. As estratégias combinam os múltiplos atributos com restrições pela associação com penalidades e combinações lineares com pesos sobre os atributos/objetivos. Porém essa estratégia torna a solução totalmente dependente dos pesos e das penalidades o que torna a muito sensível à mudanças nestes parâmetros [32].

O procedimento de transformação de múltiplos objetivos em um único pode seguir os seguintes métodos [79] [8]:

- **Método da soma dos pesos**. O valor é uma soma ponderada com pesos para cada objetivo definidos de forma fixa, randômica ou adaptativa. Na forma fixa é definido um peso constante para cada objetivo; na forma randômica é gerado um vetor randômicos de pesos entre 0 e 1; na forma adaptativa são definidos pontos ideais positivos e negativos e assim o valor de cada objetivo é ponderado por esses pontos.
- **Método de compromisso**. O valor é a distância entre o ponto (o vetor objetivo) e o ponto ideal.
- **Método de programação de metas**. O valor é a soma dos desvios entre o valor de cada objetivo e um valor de meta.

Os valores gerados no método da soma dos pesos deve ser maximizado em problemas de maximização, e minimizados em problemas de minimização. Já os valores gerados pelo método de compromisso e programação de metas devem ser minimizados, mesmo em problemas de maximização.

Outro método que pode ser utilizado é apresentado por Coello [8] o método ε-restrições. Onde uma única função objetivo representa o valor aptidão de um indivíduo e os outros objetivos são tratados como restrições. É então estabelecido um limite superior, para problemas de minimização, para cada objetivo um ε_i. O problema passa então a ser de minimização de objetivo único. Múltiplas configurações de ε podem ser utilizadas para uma melhor tomada de decisão.

As funções de utilidade citadas podem ser utilizadas no decisor ou podem servir como função fitness para estratégias evolutivas [79]. Assim, a agregação de múltiplos objetivos em um único critério de otimização tem a vantagem de que as estratégias de otimização de objetivo único podem ser aplicadas sem nenhuma modificação adicional, porém são totalmente dependentes do critério estabelecido [82].

4.2 Algoritmos Genéticos Multi-objetivo

Coello divide os algoritmos genéticos multi-objetivos atuais em 3 categorias: abordagens baseadas em funções de utilidade, outras abordagens que não são baseadas em preferência e abordagens baseadas em preferência. As abordagens não baseadas em preferência são subdividas em uma categoria baseada em funções de utilidade e em uma que não é baseada em função de utilidade, ou seja que não agrupam os objetivos em um único. Dentro desta categoria (que não fazem uso de função de utilidade) se encontra o Algoritmo Genético de Vetor Valorado. As abordagens atuais de algoritmos que não são baseados em preferência usam o conceito de dominância para avaliar as soluções, dentro destas abordagens se encontra o Algoritmo Evolutivo da Força de Pareto II e o Algoritmo Genético por Ordenação de Não-Dominância II [8].

4.2.1 Algoritmo Genético de Vetor Valorado - VEGA

O algoritmo genético de vetor valorado (VEGA - *Vector Evaluated Genetic Algorithm*) foi o primeiro EA para otimização multi-objetivo. Schaffer, seu criador, modificou o operador de seleção de um algoritmo genético simples de forma a ser efetuada a seleção para cada objetivo [25] [16] [26].

Na população atual, cada indivíduo é avaliado em cada um dos objetivos. O método de seleção então é realizado para cada objetivo de forma individual para preencher o grupo de acasalamento [16], selecionando, ao final, $\frac{N}{\kappa}$ indivíduos com relação a cada

objetivo (onde N é o tamanho da população e κ é o número de objetivos), resultando em κ subpopulações de mesmo tamanho, uma para cada objetivo [82]. As subpopulações então são misturadas e são aplicados os operadores genéticos (cruzamento e mutação) da mesma maneira que nos algoritmos genéticos simples [82] [16]. O algoritmo é ilustrado na Figura 4.1:

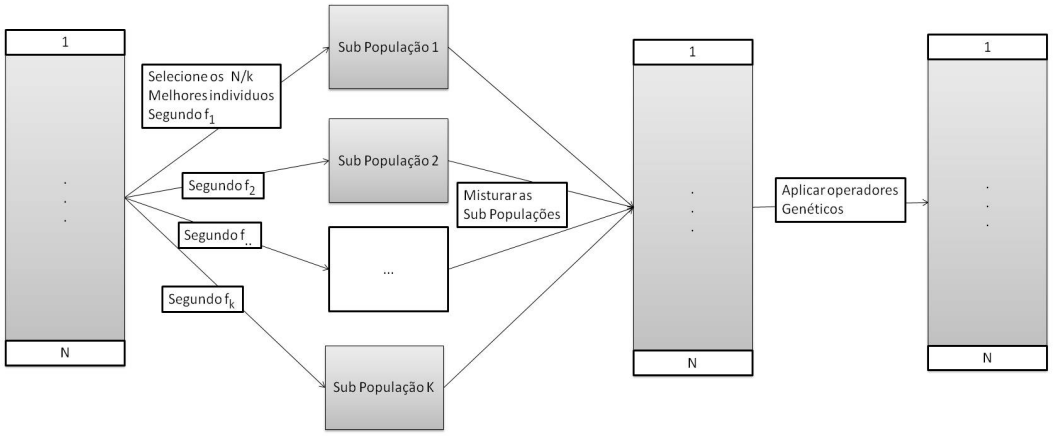

Figura 4.1: *Funcionamento esquemático do VEGA.*

onde κ é a quantidade de objetivos, N é o tamanho da população, $\frac{N}{\kappa}$ é o tamanho de cada subpopulação, f_1 é a primeira função objetivo, ..., f_κ é a última função objetivo.

Em um primeira análise o algoritmo parece resolver os problemas dos algoritmos predecessores sendo o ideal [79]. Porém, em alguns casos ele começa a tender para alguns indivíduos ou regiões (principalmente os melhores indivíduos de cada objetivo) [16]. Essa tendência pode fazer com que a população gerada fique circunscrita a determinados objetivos, propriedade que Schaffer nomeou como especiação [82]. Esse problema acontece por que essa técnica seleciona indivíduos que sobressaem em um objetivo sem olhar os outros objetivos [8] e vai contra um dos objetivos almejados nos algoritmos evolutivos, que é manter a população com a máxima diversidade possível [16].

O resultado deste processo de especiação é a convergência da população inteira a regiões ótimas individuais (associadas à poucos objetivos), depois de um grande número de gerações. Uma das metas dos algoritmos evolutivos multi-objetivos é encontrar o maior número de soluções não dominadas o possível, com a maior diversidade, sobre todos os objetivos. São então desenvolvidas duas heurísticas para tentar minimizar este problema: a seleção não dominada, que busca penalizar soluções dominadas, diminuindo o número de cópias destas e aumentando proporcionalmente o número de cópias das soluções não dominadas; e a heurística do parceiro, que busca cruzar os indivíduos com os indivíduos mais diferentes (diferença relativa à distância Euclidiana). Porém a primeira heurística falha quando há poucos indivíduos não dominados, trazendo uma tendência à eles, e a

segunda falha por trazer ao grupo de acasalamento indivíduos com nível de aptidão muito ruim (com baixo *fitness*) [70] [26].

4.2.2 Algoritmo Evolutivo da Força de Pareto II - SPEA 2

O Algoritmo Evolutivo da Força de Pareto (*Strength Pareto Evolutionary Algorithm - SPEA*) surgiu da mescla de técnicas recentes (à seu tempo) e antigas, formando um único algoritmo, a fim de encontrar múltiplas soluções ótimas de pareto, de forma paralela [84] [80].

Das técnicas antigas as seguintes características foram agregadas: armazenamento externo das soluções ótimas de pareto encontradas, uso de um escalar para os valores de aptidão baseado no conceito de dominância, e agrupamento de soluções para reduzir o número de soluções não dominadas armazenadas conservando as características da fronteira de pareto [84] [80].

São características peculiares do SPEA: a combinação das três técnicas apresentadas em um único algoritmo; o valor de aptidão de um indivíduo ser determinado unicamente das soluções armazenadas no conjunto externo; todas as soluções no conjunto externo são submetidas ao operador de seleção; e um novo método de agrupamento de soluções (método de nicho) é implementado para preservar a diversidade na população baseado na dominância de pareto da solução. E suas potenciais fraquezas são: o seu processo de associação de valor de aptidão (em casos de indivíduos diferentes que são dominados pelos mesmos indivíduos), a necessidade de uma estimação de densidade (pois indivíduos muito similares não contribuem para o processo evolutivo) e em seu sistema de truncamento [84] [80].

O SPEA 2 implementa melhorias ao seu predecessor, o SPEA, com uma estratégia de associação de *fitness* mais aprimorada [83]. Além de um melhor cálculo de "*fitness*", o SPEA 2 busca reduzir as fraquezas de seu predecessor com uma técnica de estimação de densidade (análise da proximidade do "vizinho mais próximo"), que permite um melhor guia de precisão ao processo de busca; e um novo método de truncamento que garante a preservação das soluções que fazem parte da fronteira de pareto [82].

Tendo como entrada: o tamanho da população (N), o tamanho do arquivo externo (quantidade de indivíduos - $\overline{N}$), e o máximo de gerações (T), os seguintes passos são realizados com intuito de se obter um conjunto de soluções não dominadas [83]:

- **Passo 1 - Inicialização**: É gerada uma população inicial, P_0, e um arquivo vazio (conjunto externo), $\overline{P}_0$. O contador de gerações é inicializado em 0: t = 0.
- **Passo 2 - Associação de *fitness***: Os valores de *fitness* são calculados para cada indivíduo em P_t e $\overline{P}_t$.

- **Passo 3 - Seleção "ambiental"**: Todos os indivíduos não dominados de P_t e $\overline{P}_t$ são copiados para $\overline{P}_{t+1}$. Talvez seja necessário reduzir $\overline{P}_{t+1}$ (caso o tamanho de $\overline{P}_{t+1}$ seja maior que $\overline{N}$), nesse caso, é aplicado um operador de truncamento. Por outro lado pode ser necessário completar $\overline{P}_{t+1}$ com soluções dominadas(caso $\overline{N}$ seja maior que o tamanho de $\overline{P}_{t+1}$).
- **Passo 4 - Critério de parada**: Caso t $\geq$ T, ou seja definido e satisfeito outro critério de parada o conjunto de soluções não dominadas presentes em $\overline{P}_{t+1}$ é dado como saída, e o algoritmo é encerrado. Caso contrário o Passo 5 é executado.
- **Passo 5 - Seleção de pais**: o método de torneio binário é executado, com reposição, em $\overline{P}_{t+1}$ e o grupo de acasalamento é gerado.
- **Passo 6 - Operadores genéticos**: Efetuar os operadores de mutação e recombinação para gerar uma nova população. Incrementar o contador de gerações. Voltar ao **Passo 2**.

A população é gerada pelo seguinte processo: são gerados indivíduos pertencentes ao conjunto de soluções possíveis (frequentemente de maneira aleatória), e são escolhidos N indivíduos de acordo com uma distribuição de probabilidade [80].

O valor de aptidão de um indivíduo é calculado da seguinte forma: cada indivíduo i é associado a um valor de "força", $S(i) \in [0,1)$, que também é utilizado para o cálculo de seu valor de aptidão $F(i)$. $S(i)$ é o número de indivíduos que são dominados ou são iguais ao indivíduo i (com respeito aos valores objetivo) dividido pelo tamanho da população mais 1. O valor de aptidão $F(j)$ então é calculado pela soma dos valores de "força"(S(i)) de todos os indivíduos (que pertencem ao arquivo) que dominam ou são iguais ao indivíduo mais 1 [83].

Uma das fraquezas do SPEA é que indivíduos que são dominados pelos mesmos indivíduos tem valores de aptidão idênticos. O que significa que se existe apenas uma solução não dominada na população, o arquivo externo possuirá um único indivíduo e todas as outras soluções possuirão o mesmo valor de aptidão, independente se há níveis de dominância diferentes entre essas soluções. Isso faz da seleção dos indivíduos que preencherão a população uma seleção aleatória [83].

Para resolver essa fraqueza, no SPEA 2, tanto as soluções dominadas quanto as que dominam uma solução são consideradas no cálculo do valor de aptidão. Assim, para cada indivíduo i no arquivo externo e na população atual é atribuído o valor de força $S(i)$(quantidade de soluções que i domina). O valor de aptidão bruto ($B(i)$), então, é obtido pelo somatório da força das soluções que dominam i. Porém, indivíduos que não dominam nenhum indivíduo ainda tem o mesmo valor de aptidão, e então é adicionada uma função de densidade para discriminar indivíduos que possuem mesmo $B(i)$. A função de densidade $D_s(i)$ calcula a distancia D_s entre i e o k-ésimo vizinho mais próximo. Para isso, o algoritmo calcula a distância entre i e todas as demais soluções e armazena em

uma lista, a lista é ordenada e seu k-ésimo termo é a distância escolhida (k é usualmente $\sqrt{N+\overline{N}}$, onde N é o tamanho da população e $\overline{N}$ é o tamanho do arquivo). A distância então é calculada conforme a equação 4-4 [83]:

$$D_s(i) = \frac{1}{L_s^i + 2} \tag{4-4}$$

E o valor aptidão do i-ésimo indivíduo é dado pela equação 4-5, onde se nota que as soluções não dominadas tem $F(i) < 1$:

$$F(i) = B(i) + D_s(i) \tag{4-5}$$

A Figura 4.2 apresenta a comparação, dos esquemas de associação de valor aptidão, entre o valor de aptidão calculado pelo procedimento do SPEA (à esquerda) e o "*fitness*" bruto do SPEA 2, para uma mesma população, com relação à duas funções objetivo (f_1 e f_2) [83].

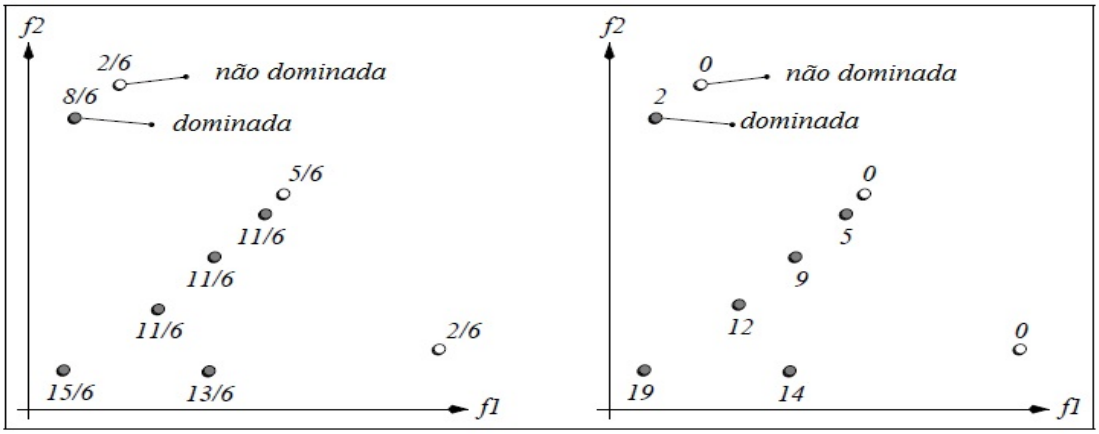

Figura 4.2: *Comparação dos esquemas de associação de aptidão entre SPEA e o valor bruto do SPEA2.*

A operação de atualização do arquivo possui duas diferenças com relação ao SPEA [83]. As mudanças na seleção ambiental são: o arquivo externo tem um tamanho fixo de população, que é mantido adicionando, se necessário, soluções dominadas, ou realizando uma poda pelo algoritmo de agrupamento; uma modificação no algoritmo de agrupamento de maneira a preservar as soluções da fronteira de pareto com maior diversidade [16].

Na seleção ambiental, se o tamanho do arquivo externo é menor que a constante ($\overline{N}$, são adicionados na população os indivíduos dominados com maior $F(i)$. O algoritmo de poda remove indivíduos do arquivo de forma iterativa até que o tamanho da população do arquivo, na nova geração, seja igual a $\overline{N}$. Em cada iteração é removido o indivíduo com menor distância L_s, se houver mais de um indivíduo com esta distância, então L_s é

atualizada para a segunda distância (k-1), e assim por diante [83]. O algoritmo de poda descrito é apresentado na Figura 4.3, onde à esquerda é apresentada a população $\overline{P}_{t+1}$ antes da poda e a direita o resultado da remoção das soluções.

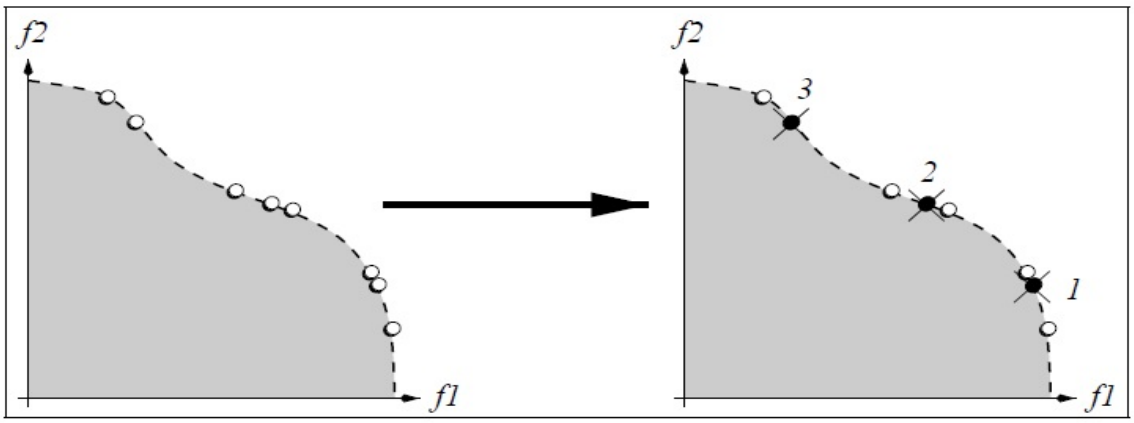

Figura 4.3: *Algoritmo de corte do SPEA 2.*

4.2.3 Algoritmo Genético por Ordenação de Não Dominância - NSGA II

O Algoritmo Genético por Ordenação de Não Dominância (NSGA) é uma variação de um algoritmo genético simples, proposta por Deb, com respeito à forma que operador de seleção funciona. A ordenação por não dominância se baseia num método de ranquear as soluções de modo a enfatizar bons pontos (as soluções não dominadas) e um método de agrupamento é utilizado para manter subpopulações estáveis destes bons pontos [69].

As principais críticas relacionadas ao NSGA foram: a alta complexidade computacional para a ordenação de não dominância; a falta de elitismo; e a necessidade de especificação do parâmetro compartilhado (embora já existiam trabalhos na área de se utilizar um parâmetro dinâmico, um método sem parâmetros que preservasse a diversidade da população é desejável) [17]. Para resolver os problemas supracitados um novo algoritmo foi proposto por Deb, o NSGA II [17]. O NSGA II é caracterizado pelos seguintes recursos [16]:

- ele usa o princípio de elitismo;
- ele usa um mecanismo de preservação de diversidade explícito;
- ele valoriza as soluções não dominadas.

O NSGA II, como o SPEA 2, utiliza de uma população regular e um arquivo. No NSGA II, a capacidade de ambas as populações são iguais, ou seja, $N = \overline{N}$ [79].

Inicialmente a população $\overline{P}_0$ é criada, de forma aleatória. Cada solução é avaliada, em cada um dos objetivos, e associada ao seu nível de dominância (1 é o melhor nível). O método de seleção, por torneio binário, é utilizado para se gerar uma população de filhos ζ_0, de tamanho N [17].

Em cada geração t, a população de filhos ζ_t é criada a partir da população de pais e os operadores genéticos usuais. Depois, as duas populações são combinadas para formar uma nova população P_t ($P_t = \overline{P}_t \bigcup \zeta_t$) de tamanho $2N$. Então, a população P_t tem seus indivíduos classificados em diferentes classes de não-dominância. Após isso, a nova população é preenchida pelos pontos de diferentes fronteiras de não dominância, uma por vez. O preenchimento começa com a primeira fronteira de não dominância, e segue para as demais. Uma vez que o tamanho de P_t é diferente de $\overline{P}_t$ nem todas as fronteiras serão alocadas na nova população, e as que não são alocadas são removidas (todos os indivíduos). Quando a última fronteira está sendo considerada, há a possibilidade de se existir mais indivíduos nesta fronteiras que espaços vazios da nova população, assim os pontos menos similares devem ser escolhidos, para manter a diversidade da população [16].

A diversidade no NSGA II, é dado por uma medida de distância que é independente de valores compartilhados (uma melhoria em relação ao seu predecessor), essa distância é chamada distância de agrupamento (Distância crowding), e é dada pela equação 4-7:

$$c^{[i]} = \sum_{a=1}^{kappa} c_a^{[i]} \tag{4-6}$$

onde i é o indivíduo que se quer calcular a distancia, c_a^i é a distância de agrupamento do i-ésimo indivíduo, com relação ao objetivo a. Para se calcular essa distancia, com relação ao objetivo a, é necessário primeiro ordenar os indivíduos da fronteira, à qual ele pertença, em ordem ascendente de f_a (f_a é o valor objetivo da função a) e a distância do indivíduo $[j]$ na sequência é dada pela equação 4-7

$$c_a^{[j]} = \frac{f_a^{[j+1]} - f_a^{[j-1]}}{f_a^{\max} - f_a^{\min}} \tag{4-7}$$

A ordenação por não dominância é dada pelo algoritmo 4.1 [17]:

onde F_1 é o conjunto formado pelos indivíduos da primeira fronteira de pareto, F_2 da segunda, e assim por diante; S_{I^p} são subpopulações utilizadas para armazenar os indivíduos que I^p domina; e n_{I^p} é o número de soluções que dominam I^p [17].

Assim a ordenação rápida por não dominância busca resolver o problema da complexidade computacional, a distância de agrupamento resolve a necessidade de um

Algoritmo 4.1: Ordenação rápida por não dominância

1. **Seja P a população que se deseja ordenar**
2. **Para cada** $I^p \in P$
   Para cada $I^q \in P$
   Se $(I^p \prec I^q)$ **então** // Se I^p domina I^q então
   $S_{I^p} \leftarrow S_{I^p} \bigcup \{I^p\}$ // adiciona I^q em S_{I^p}
   Senão Se $(I^q \prec I^p)$ **então** // Se I^q domina I^p então
   $n_{I^p} \leftarrow n_{I^p} + 1$ // incrementa n_{I^p}
   Fim se
   Fim se
   Fim para
   Se $n_{I^p} = 0$ **então** // Se ninguém domina I^p
   $F_1 \leftarrow F_1 \bigcup \{I^p\}$ // adiciona I^p em F_1
   Fim se
3. **Fim para**
4. $i \leftarrow 1$
5. **Enquanto** $F_i \neq \emptyset$ **faça**
   $H = \emptyset$
   Para cada $I^p \in F_i$
   Para cada $I^q \in S_p$
   $n_{I^q} = n_{I^q} - 1$
   se $n_{I^q} = 0$ **then**// Se ninguém mais o domina
   $H \leftarrow H \bigcup \{I^p\}$ // Ele é incluído aos elementos desta fronteira
   Fim se
   Fim para
   Fim para
6. **Fim enquanto**
   $i \leftarrow i + 1$
   $F_i \leftarrow H$

parâmetro compartilhado, e o sistema de elitismo é baseado na inclusão da população externa [79].

Capítulo 5 - Metodologia Proposta e Experimentos

Este Capítulo apresenta o EA multi-objetivo proposto para a seleção de variáveis em problemas de classificação multivariada de dados químicos. O EA utilizado se baseia na proposta implementada por Sanches [62] [63] e Vargas [76], que utiliza uma estrutura, para armazenamento da população, diferente dos algoritmos convencionais, com o intuito de tratar múltiplos objetivos e restrições [19] [62] [76]. A estrutura proposta divide a população do EA em subpopulações, de acordo com as características do problema, e as armazena em tabelas. Por isso o EA foi nomeado Algoritmo Evolutivo Multi-objetivo em Tabelas (AEMT) [19].

5.1 Algoritmo Genético Multi-objetivo de Tabelas

O AEMT foi utilizado para reconfiguração de sistemas de distribuição de energia elétrica [19] [62]. E foi expandido para otimização em problemas de predição de estruturas de proteínas [6], problemas de calibração multivariada [39], e problemas de alocação de monitores de qualidade da energia em sistemas de distribuição de energia elétrica [12].

Cada tabela, que faz parte da estrutura de um AEMT, contém a subpopulação com os indivíduos que possuem os melhores valores de aptidão para uma característica do sistema. O valor de aptidão de um indivíduo em relação a uma característica pode ser dado por: uma função objetivo, uma função de agregação de múltiplos valores objetivo, uma função proveniente da força de pareto da solução, uma função de agregação de múltiplos valores de aptidão(múltiplas tabelas). As funções de agregação podem ser baseadas nos métodos de transformação de múltiplos objetivos em um único (método da soma ponderada, método do compromisso, ou método de programação de metas) [79] [8] citados no Capítulo 4.

Uma tabela pode conter também uma subpopulação baseada em outras abordagens evolutivas, uma das formas já implementadas é o armazenamento das soluções não dominadas [63]. As características da solução podem incluir funções de densidade, força

de pareto, nível de dominância, entre outras características, para o uso destas características no AEMT é necessário que o valor aptidão possibilite elistismo.

5.1.1 Funcionamento do AEMT

Após a modelagem do problema, os seguintes parâmetros populacionais devem ser estabelecidos: o tamanho da população inicial (N); o tamanho de cada uma das subpopulações(SP[i].tamanho), pode ser utilizado o mesmo tamanho para todas as subpopulações; quantidade de indivíduos que serão criados em cada geração $\overline{N}$, valor que pode ser uma constante fixa τ (τ=1), $\tau \times N$ (estar em função do tamanho da população inicial), ou $\tau\times$ SP.tamanho (estar em função da quantidade de indivíduos vigentes); número máximo de gerações (*maxGen*); probabilidade de mutação do indivíduo (*pm*); probabilidade de mutação do gene em um indivíduo (*pmg*); probabilidade de cruzamento (*pc*).

Após a definição dos parâmetros uma população é gerada (de forma aleatória, ou baseada em soluções pré-escolhidas). O processo de inicialização das tabelas é descrito no Algoritmo 5.1.

Algoritmo 5.1: Algoritmo Evolutivo Multi-objetivo em Tabelas - Inicialização das tabelas

1. **Seja P uma população inicial**
2. **Para cada** $I^p \in P$
   Para $i \leftarrow 1,\ldots,k$
   $I^p.valorAptidao[i] \leftarrow f[i](I^p)$;// Avaliação do indivíduo por $f[i]$
   Se ($tamanho(SP[i]) < \overline{SP}[i]$) **então** // SP[i] não está cheia
   $SP[i] \leftarrow SP[i] \bigcup \{I^p\}$ // adiciona I^p em $SP[i]$
   Senão Se ($I^p >_p PIOR(SP[i])$) **então** // Se sobrevive em SP[i]
   $SP[i] \leftarrow SP[i] \bigcup \{I^p\}$ // adiciona I^p em $SP[i]$
   $SP[i] \leftarrow SP[i] - \{PIOR(SP[i])\}$ // remove a pior solução de $SP[i]$
   Fim se
   Fim se
   Fim para
3. **Fim para**

Onde P é a população inicial, $SP[i]$ é a i-ésima tabela (subpopulação), k é o número de objetivos, $I^p.valorAptidao[i]$ é o i-ésimo valor aptidão do indivíduo I^p, $f[i](o)$ é a função aptidão para a tabela i. A função $PIOR(SP[i])$, retorna o indivíduo menos apto de $SP[i]$.

Com as tabelas construídas, são selecionados os indivíduos que poderão se reproduzir, essa seleção se dá em dois passos. O primeiro passo é a seleção da tabela (usualmente de forma aleatória), no segundo passo é efetuada a escolha do indivíduo, por um operador de seleção normal. Após a seleção do indivíduo, um operador é escolhido

e aplicado ao indivíduo gerando um novo indivíduo [19] [62]. Assim, o operador de reprodução, da abordagem inicial, é exclusivamente o de mutação.

Novas implementações do AEMT utilizam o operador de cruzamento [6] ou, também, o operador de mutação [12] [39]. Para se realizar o cruzamento cada um dos indivíduos passam pelo processo de seleção descrito anteriormente. O novo indivíduo é então gerado com um operador de recombinação idêntico aos descritos para cromossomos binários e/ou reais. O processo de seleção de pais e reprodução é descrito pela Figura 5.1 onde $SP[1], SP[2], \ldots, SP[Q]$, são as subpopulações, que podem ter tamanhos diferentes, dependendo da abordagem.

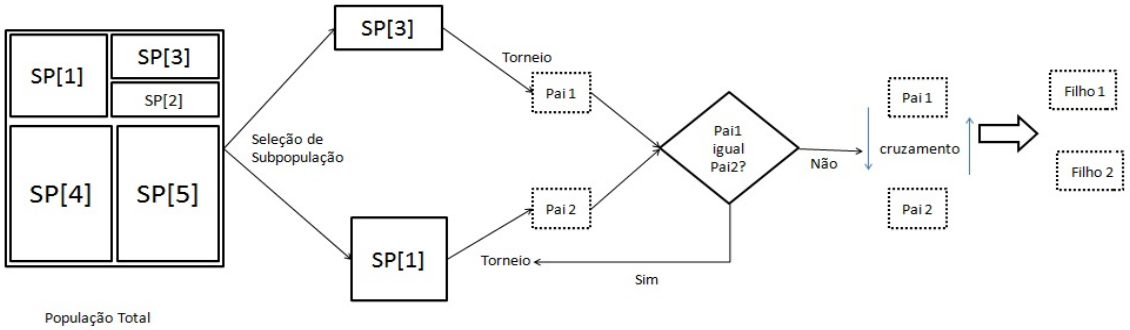

Figura 5.1: *Processo de seleção e cruzamento no AEMT.*

O indivíduo criado é avaliado para cada uma das tabelas. Em cada uma delas, o indivíduo é comparado com o pior indivíduo da tabela (pior valor aptidão), o pior destes dois indivíduos é descartado e o outro sobrevive. Este processo pode preservar o mesmo indivíduo em diversas tabelas, porém deve-se fazer um filtro para que uma tabela só possua indivíduos diferentes. Neste caso duas situações são possíveis: considera-se iguais indivíduos com cromossomos idênticos, ou que sejam muito próximos (qualquer métrica de distância, simples, pode ser utilizada).

O processo completo do AEMT é representado pelo Algoritmo 5.2. Onde P é a população inicial, SP é a coleção de tabelas, $SP[i]$ é a i-ésima tabela (subpopulação), k é o número de objetivos, $I^p.valorAptidao[i]$ é o i-ésimo valor aptidão do indivíduo I^p, $f[i](o)$ é a função aptidão para a tabela i, $\overline{N}$ é a quantidade de indivíduos que serão criados em cada geração. A função $PIOR(SP[i])$, retorna o indivíduo menos apto de $SP[i]$, e a função $MELHOR(SP[i])$, retorna o indivíduo mais apto de $SP[i]$, sendo *tabelaDecisao* a tabela agregada.

Cada tabela, ao final do processo, contém os melhores indivíduos encontrados pelo algoritmo, de acordo com as funções *fitness*. E servem como opção de solução para quem utiliza o classificador.

Algoritmo 5.2: Algoritmo Evolutivo Multi-objetivo em Tabelas

1. **Seja P uma população inicial**
2. $SP = InicializaTabelas(P)$;
3. **Para** $t = 1, \ldots, maxGen$
   Para $n \leftarrow 1, \ldots, \overline{N} * 2$
   $I^p \leftarrow seleoECruzamentoAEMT(SP)$;
   $I^p \leftarrow mutar(I^p)$;
   $I^p \leftarrow validar(I^p)$;// p é modificado, se necessário, para atender às restrições
   $I^p.valorAptido[i] \leftarrow f[i]$;// avalia a i-ésima característica
   Se ($tamanho(SP[i]) < \overline{SP}[i]$) **então** // Se a tabela ainda não foi preenchida
   $SP[i] \leftarrow SP[i] \bigcup \{I^p\}$ // adiciona I^p em $SP[i]$
   Senão Se ($I^p >_{f[i]} PIOR(SP[i])$) **então** // Se I^p é melhor que o pior de SP[i]
   $SP[i] \leftarrow SP[i] \bigcup \{p\}$ // adiciona I^p em $SP[i]$
   $SP[i] \leftarrow SP[i] - \{PIOR(SP[i])\}$ // remove a pior solução de $SP[i]$
   Fim se
   Fim se
   Fim para
4. **Fim para**
5. retornar MELHOR(SP[tabelaDecisao]);

5.1.2 Funções Objetivo Consideradas

As funções objetivo consideradas para comparação entre os algoritmos evolutivos são: o risco médio de classificação (R), o número de variáveis selecionadas (V) e o número de variáveis correlacionadas selecionadas para a construção do modelo (C). A função número de erros em uma coleção de testes (E) foi utilizada apenas na implementação mono-objetivo e comparada apenas à implementações que otimizam somente o risco médio de classificação (R), assim ela é apresentada neste tópico mas não é utilizada nas abordagens multi-objetivo.

Erros em uma coleção de Teste (E)

Para julgar a capacidade de classificação de um classificador normalmente utilizada-se a taxa de erros. A taxa de erros aparente pode ser calculada em uma coleção de testes onde se verifica a relação entre a quantidade de indivíduos classificados erroneamente em todas as classes (somatório dos erros em cada classe) e a quantidade de indivíduos classificados [61].

Como a quantidade de indivíduos classificados é a mesma para todos os experimentos, o valor responsável pela avaliação do classificador é unicamente o número de erros na coleção de testes. Sendo assim, o número de erros em uma coleção de classifica-

ção é um tipo de valor objetivo a ser otimizado. Ou sendo R o vetor resposta obtido pelo classificador e TESTE as classes à que a coleção é realmente classificada.

$$f_E(ind) = |R_i \neq TESTE_i| \tag{5-1}$$

Risco médio de Classificação (R)

É utilizada também o risco médio de classificação pelo LDA em um conjunto de testes. Dada pela equação (5-2):

$$f_R(ind) = \frac{1}{N}\sum_{i=1}^{N} g_i \tag{5-2}$$

onde N é o tamanho da coleção, e g_i é o risco de classificação da i-ésima amostra. O risco g_i é definido pela equação (5-3):

$$g_i = \frac{D^2(X^i,\mu_k)}{min_{j\neq k}D^2(X^i,\mu_j)} \tag{5-3}$$

Na equação (5-3), o numerador é a distância de Mahalanobis entre o objeto X^i e a média μ_k da classe a que o individuo realmente pertence (ambos vetores linha).

Quantidade de variáveis correlacionadas (C)

O uso de variáveis correlacionadas pode introduzir tendências ao classificador. Correlação é uma medida de relação linear entre variáveis, invariante à mudanças de escala (a matriz de covariâncias é dependente da escala), obtida normalizando-se a covariância das variáveis. A normalização ocorre pela divisão entre a covariância e o desvio padrão entre elas [61].

A correlação entre o vetor X^1 e o vetor X^2 pode ser representada pela equação 5-4:

$$r_{X^1Y^2} = \frac{\sum_{i=0}^{p}(X_i^1 - \overline{X^1})(X_i^2 - \overline{X^2})}{\sum_{i=0}^{p}(X_i^1 - \overline{X^1})^2 \sum_{i=0}^{n}(Y_i^1 - \overline{Y^2})^2} \tag{5-4}$$

onde p é o tamanho dos vetores, $\overline{X^1}$ e $\overline{X^2}$ são as médias dos vetores X^1 e X^2 respectivamente. Assim, a correlação entre duas variáveis escolhidas é feita a partir das medições em diversas amostras. E a matriz de correlação entre p' variáveis é uma matriz COR (p' x p') onde a_{ij} é a correlação entre a variável i e a variável j.

E $f_C(ind)$ é a quantidade de variáveis que possuem uma correlação máxima (com pelo menos 1 variável) maior que $0,8$. Apresentada pela equação (5-5).

$$f_C(ind) = |max(Cor) > 0,8| \tag{5-5}$$

Número de variáveis (V)

Segundo Dash [11] a seleção de variáveis se esforça para encontrar o subconjunto com o menor número de variáveis (recursos/atributos) que não piore a classificação. Assim o número de variáveis deve ser considerado, a equação (5-6) apresenta o número de variáveis.

$$f_V(ind) = TAMANHO(ind) \tag{5-6}$$

Função de Agregação (O)

A função de agregação utilizada foi uma soma simples entre os valores objetivo considerados. Com os seguintes valores para c: 1, 32 e 32 para os objetivos risco médio de classificação, número de variáveis selecionadas e número de variáveis correlacionadas. O número de erros na coleção de testes não foi pontuado por não estar presente em nenhuma implementação multi-objetivo. A estratégia pode ser representada pela equação:

$$f_O = \Sigma_{i=0}^{k} \frac{f_i(x)}{c_i} \tag{5-7}$$

5.1.3 Implementação

Para o problema de seleção de variáveis os seguintes objetivos foram otimizados:

- **Risco médio de classificação (R)**. A probabilidade de erro do classificador. Equação 5-2.
- **Número de Variáveis Correlacionadas (C)**. A quantidade de variáveis que possuam correlação máxima maior que 0,8. Equação 5-5.
- **Número de Variáveis (V)**. A quantidade de variáveis do selecionadas. Equação 5-5.
- **Função de Agregação (O)**. Soma ponderada dos objetivos. Equação 5-7.

Como comparação foi ainda utilizada a equação (5-1), que minimiza a quantidade de erros no conjunto de testes. Este atributo é similar ao risco médio de classificação e não foi utilizado como objetivo.

Para este trabalho foram postas restrições sobre o número de variáveis, onde limite superior é o tamanho da coleção de treinamento (por utilizar análise discriminante linear) e o limite inferior é uma única variável. Para tratamento das restrições duas

estratégias foram utilizadas: poda aleatória nas variáveis selecionadas, para o limite superior; criação aleatória de uma variável.

Em todas as implementações utilizadas neste trabalho, o cromossomo utiliza a notação binária para a seleção de variáveis. Um cromossomo é representado por $\Theta = \Theta_1, \Theta_2, \ldots, \Theta_L$, onde L é o tamanho do cromossomo. Na notação binária o gene θ_i = 1 indica que a *i*-ésima variável foi selecionada, caso contrário ($\Theta_i = 0$) a variável não faz parte da solução. Nos indivíduos são armazenados, também, os valores de aptidão para cada objetivo, com o intuito de realizar o calculo da função aptidão uma única vez para cada solução. Na representação utilizada L (tamanho do cromossomo) é igual a p (número de variáveis medidas).

Foi utilizado o mesmo tamanho para todas as tabelas com o valor igual a 10% do tamanho da população inicial. O tamanho da população, e o número de gerações, foi escolhido após uma série de configurações distintas.

5.2 Materiais e Métodos

5.2.1 Dados - Adulteração de Biodiesel

O conjunto de dados utilizado é constituído por dados espectrais obtidos de cento e quarenta amostras de quatro diferentes classes de combustível, são elas: diesel livre de biodiesel e óleos vegetais crus, D (35); amostras contendo diesel, biodiesel e óleos vegetais, OBD (38); amostras de diesel e óleo vegetal, OD (35); e amostras que são compostas de 5% de biodiesel, com variação de $\pm 0,5\%$, B5(32). Os dados foram pré-processados por primeira derivada pelo método de Savitzky-Golay, empregando um polinômio de segunda ordem e janela de 7 pontos. A coleção foi dividida no conjunto de treinamento, validação, e testes, pelo algoritmo Kennard-Stone.

As misturas analisadas foram preparadas utilizando diferentes sementes oleaginosas, gordura animal e seus respectivos ésteres. O biodiesel (B100) e as amostras de óleo foram adquiridas no mercado de diferentes marcas. As amostras de B100 foram analisadas segundo as especificações brasileiras. Com o intuito de incluir variedade às amostras de diesel puro, foram misturadas diferentes amostras de óleo diesel.

A mistura modelo com ponto central para cada tipo de biodiesel, incluindo pontos externos, foram utilizadas para preparar as soluções. O nível máximo de biodiesel e óleo foram de 10%. A razão de concentração éster/óleo variou de 0,25 a 6,0.

Os espectros foram registrados em um espectrômetro com transformada de Fourier (FT-IR) modelo FTLA $2000-160$ da marca *Bomen*®, e comprimento óptico de 1,0 mm. Cada espectro foi obtido pela média de 32 scans no intervalo de $8,814-3,799 cm^{-1}$ com resolução de $8 cm^{-1}$. O espectro de fundo foi obtido usando uma célula

limpa. A temperatura foi controlada em 23 ± 1^{o}C por meio do processo de aquisição do espectro.

Esta coleção foi analisada por [59] utilizando o algoritmo das projeções sucessivas como método de seleção de variáveis e análise discriminante linear como classificador (SPA-LDA), em comparação a classificação do classificador por análise discriminante dos mínimos quadrados parciais (PLS-DA).

5.2.2 Ferramentas e Ambiente

A configuração computacional deste trabalho consiste em um computador de mesa equipado com processador Intel® Core i5, 6 GB de memória RAM. Matlab 8.0.0.783 (R2012b) foi o software utilizado para implementação desse trabalho.

5.2.3 Algoritmos de Comparação

Foram utilizados os resultados obtidos pelos algoritmos SPA-LDA e PLS-DA [59], e implementações próprias de algoritmos genéticos mono-objetivos (MONO-GA), algoritmos genéticos multi-objetivos por função de agregação (MULTI-GA) e do algoritmo NSGA-II (NSGA2). Sendo assim os seguintes algoritmos foram construídos: MONO-GA-R, MONO-GA-E, MULTI-GA-RC, MULTI-GA-RV, MULTI-GA-RCV; NSGA2-RC, NSGA2-RV, NSGA2-RCV; e os algoritmos AEMT-RC, AEMT-RV, AEMT-RCV.

Onde os algoritmos genéticos multi-objetivos otimizam a função de agregação apresentada na equação 5-7. Os algoritmos MONO-GA-R e MONO-GA-E otimizam as funções apresentadas nas equações 5-2 e 5-1 respectivamente. Os algoritmos com terminação RC otimizam as equações 5-2 e 5-5; com terminação RV, as equações 5-2 e 5-6; e com terminação RCV otimizam as equações 5-2, 5-5 e 5-6.

Os algoritmos MULTI-GAs utilizam a equação 5-7 como função de aptidão de um algoritmo genético simples. Os algoritmos NSGA2 usam a equação 5-7 como decisor. E os algoritmos AEMTs utilizam a equação 5-7 como uma das tabelas e como decisor.

As configurações de população utilizadas foram escolhidas após uma série de combinações, as populações variaram de 50 à 200 indivíduos, e o número de gerações variou de 50 a 150 indivíduos. Foi escolhida a configuração 100 indivíduos (População) e 150 gerações(condição de parada).

Capítulo 6 - Resultados e Discussões

Neste capítulo são apresentados os resultados obtidos na execução das implementações do algoritmo NSGA-II (NSGA2-RC, NSGA2-RV, NSGA2-RCV), das implementações multi-objetivo de algoritmos genéticos simples (MULTI-GA-RC, MULTI-GA-RV, MULTI-GA-RCV) e dos algoritmos AEMT(AEMT-RV, AEMT-RC, AEMT-RCV). Os algoritmos foram aplicados para a seleção de variáveis em problemas de classificação multivariada de amostras de diesel e biodiesel. Para a execução dos algoritmos foram utilizadas as configurações descritas na seção (5.2.3).

6.1 Desempenho dos Classificadores

6.1.1 Algoritmos mono-objetivos

A Tabela 6.1 apresenta o desempenho, em termos do número de erros no conjunto de validação, dos classificadores que utilizam um único objetivo.

Tabela 6.1: *Desempenho dos classificadores.*

	Número de Erros	Variáveis Selecionadas
PLS-DA	8	7
SPA-LDA	12	6
MONO-GA-E	$4,7^{1^*,8^{**}}$	$23,8^{16^*,32^{**}}$
MONO-GA-R	$2,8^{0^*,7^{**}}$	$6,6^{4^*,9^{**}}$

*Mínimo. **Máximo.

Por serem soluções não determinísticas, os resultados apresentados na Tabela 6.1, para os algoritmos os algoritmos MONO-GA-E e MONO-GA-R, são os valores médios encontrados em 20 execuções. São apresentados também, os resultados máximos e mínimos para os parâmetros analisados. Os classificadores obtidos pelos algoritmos MONO-GA-E e MONO-GA-R conseguiram melhor desempenho médio no conjunto de validação, quando comparado aos algoritmos PLS-DA e SPA-LDA. Vale citar que os resultados dos algoritmos PLS-DA e SPA-LDA por Pontes et. al [59]. O MONO-GA-E apresentou melhor desempenho médio que as soluções apresentadas por Pontes et al. [59]

com relação ao número médio de erros, porém com um número maior de variáveis. O algoritmo MONO-GA-R apresentou melhor desempenho, em média, nos dois parâmetros analisados quando comparado aos demais algoritmos mono-objetivo citados.

Dos resultados apresentados nota-se que em alguns casos o algoritmo genético, que otimiza apenas a capacidade de classificação de um subconjunto de variáveis, seleciona uma quantidade maior de variáveis que os encontrados por Pontes et al. [59]. A seleção de variáveis tem por objetivo a obtenção de um subconjunto de variáveis que discrimine as classes consideradas. Além da capacidade de discriminação é razoável considerar o número de variáveis selecionadas e a correlação entre as variáveis selecionadas. A inclusão de variáveis correlacionadas ao modelo de classificação pode resultar em um classificador com baixa capacidade de generalização. A Tabela 6.2 apresenta o número médio de variáveis selecionadas e a quantidade média de variáveis correlacionadas incluídas no modelo. São consideradas variáveis correlacionadas, as que apresentam uma correlação máxima maior que 0,8 com outra variável.

Tabela 6.2: *Quantidade média de variáveis correlacionadas,*

	Variáveis Selecionadas	Variáveis Correlacionadas
MONO-GA-R	6,6	1,6
MONO-GA-E	23,8	11,25

Nota-se que o MONO-GA-R seleciona, em média, um subconjunto significativamente menor de variáveis quando comparado ao MONO-GA-E. O algoritmo seleciona cerca de 72% menos variáveis que o MONO-GA-E. Verifica-se também que cerca de 24% das variáveis selecionadas pelo MONO-GA-R são correlacionadas e cerca de 47% das variáveis selecionadas pelo MONO-GA-E o são. As variáveis correlacionadas não melhoram a capacidade do classificador e podem introduzir problemas na sua capacidade de generalização. Logo, nos algoritmos multi-objetivos serão considerados, além do risco de classificação, o número de variáveis e o número de variáveis correlacionadas.

A implementação do MONO-GA-R (algoritmo genético que otimiza o risco médio de classificação) apresentou melhores resultados, em média, que o MONO-GA-E (que otimiza a acurácia do classificador na coleção de testes). Como ambos avaliam a capacidade de classificação de um subconjunto de variáveis, deste ponto em diante do trabalho será considerado apenas a função objetivo risco médio de classificação (como função aptidão relacionada a capacidade de classificação).

6.1.2 Algoritmos multi-objetivos

Os resultados dos algoritmos multi-objetivo são divididos de acordo com as combinações de objetivos e são comparados pelas abordagens multi-objetivo (MULTI-GA, NSGA II e AEMT).

Risco Médio de Classificação e Número de Variáveis Correlacionadas

Na Tabela 6.3 são apresentados os resultados, médios, de erros (no conjunto de validação), risco de médio de classificação (no conjunto de validação), quantidade de variáveis selecionadas pelo algoritmo e quantidade de variáveis correlacionadas. Os algoritmos evolutivos tem como função objetivo minimizar o risco médio de classificação, na coleção de validação e a quantidade de variáveis correlacionadas. Por serem resultados médios, de 20 execuções, também são apresentados o maior e o menor resultado encontrado (a solução que encontrou o maior/menor resultado no parâmetro analisado) juntamente com os valores encontrados pelas outras funções objetivo naquela solução. Além disso são apresentados o desvio padrão dos resultados obtidos por cada função nas 20 execuções.

Tabela 6.3: *Resultados Obtidos Pelos Algoritmos Multi-objetivos Otimizando Risco de Classificação e Variáveis Correlacionadas,*

	MULTI-GA-RC			
	Erros	Risco	Número de Variáveis	Variáveis Correlacionadas
Média	2,85	0,49	5,5	0
Maior	$6^{(0,60^R,6^V,0^C)}$	$0,65^{(6^E,6^V,0^C)}$	$8^{(4^E,0,49^R,0^C)}$	$0^{(2^E,0,35^R,5^V)}$
Menor	$1^{(0,37^R,7^V,0^C)}$	$0,35^{(2^E,5^V,0^C)}$	$4^{(3^E,0,52^R,0^C)}$	$0^{(2^E,0,35^R,5^V)}$
Desvio	1,63	0,08	1,14	0
	NSGA2-RC			
	Erros	Risco	Número de Variáveis	Variáveis Correlacionadas
Média	2,6	0,5	8,95	0,85
Maior	$4^{(0,56^R,9^V,0^C)}$	$0,6^{(4^E,15^V,0^C)}$	$15^{(4^E,0,6^R,0^C)}$	$4^{(4^E,0,59^R,14^V)}$
Menor	$1^{(0,34^R,6^V,2^C)}$	$0,34^{(1^E,6^V,2^C)}$	$5^{(2^E,0,45^R,0^C)}$	$0^{(2^E,0,37^R,7^V)}$
Desvio	0,94	0,08	2,6	1,27
	AEMT-RC			
	Erros	Risco	Número de Variáveis	Variáveis Correlacionadas
Média	2,75	0,51	7,8	0
Maior	$7^{(0,71^R,10^V,0^C)}$	$0,8^{(4^E,5^V,0^C)}$	$12^{(2^E,0,51^R,0^C)}$	$0^{(1^E,0,35^R,7^V)}$
Menor	$1^{(0,35^R,7^V,0^C)}$	$0,35^{(1^E,7^V,0^C)}$	$4^{(2^E,0,40^R,0^C)}$	$0^{(1^E,0,35^R,7^V)}$
Desvio	1,55	0,1	1,91	0

^{E}: Número de Erros. ^{R}: Risco médio de classificação. ^{V}: Número de Variáveis. ^{C}: Variáveis correlacionadas.

O NSGA2-RC possui uma pequena vantagem na média de erros quando comparado ao AEMT-RC porém, o número médio de variáveis utilizado neste algoritmo é o maior dos 3 algoritmos analisados. Além disso, o NSGA2-RC apresenta um número médio de variáveis correlacionadas superior ao dos outros algoritmos. É possível observar que os algoritmos MULTI-GA-RC e AEMT-RC eliminam as variáveis fortemente correlacionadas do modelo de classificação. O MULTI-GA-RC apresentou resultado médio ligeiramente melhor no risco de classificação, e utilizou um número de variáveis menor na formação do modelo. O AEMT-RC apresentou uma melhor média de erros, quando comparado ao MULTI-GA-RC, e uma melhor média de variáveis, quando comparado ao

NSGA2-RC. Apesar do algoritmo NSGA2-RC ter encontrado uma solução com 1 erro, 0,34 de risco, 6 variáveis selecionadas, e 2 variáveis correlacionadas, a melhor solução obtida foi encontrada pelo AEMT-RC com 1 erro, 0,35 de risco, 7 variáveis selecionadas e nenhuma variável correlacionada na solução, visto que a solução encontrada pelo NSGA2-RC não elimina as variáveis correlacionadas.

As implementações que buscavam otimizar o risco médio de classificação e o número de variáveis correlacionadas reduziram o número de variáveis correlacionadas sem diminuir a qualidade do classificador, na maioria dos casos extinguindo-as. Porém, a quantidade de variáveis utilizadas para a construção do modelo se manteve estável, em alguns casos superando a quantidade utilizada pelos algoritmos PLS-DA e SPA-LDA. Assim, quando se objetiva encontrar um subconjunto ideal, a quantidade de variáveis também deve ser observada.

Risco Médio de Classificação e Número de Variáveis

Na Tabela 6.4 são apresentados os resultados, médios, de erros (no conjunto de validação), risco de médio de classificação (no conjunto de validação), quantidade de variáveis selecionadas pelo algoritmo e quantidade de variáveis correlacionadas. Os algoritmos evolutivos tem como função objetivo minimizar o risco médio de classificação, na coleção de validação, e a quantidade de variáveis selecionadas. Por serem resultados médios, de 20 execuções, também são apresentados o maior e o menor resultado encontrado (a solução que encontrou o maior/menor resultado no parâmetro analisado) juntamente com os valores encontrados pelas outras funções objetivo naquela solução. Além disso são apresentados o desvio padrão dos resultados obtidos por cada função nas 20 execuções.

O algoritmo AEMT-RV apresenta uma melhor média de erros no conjunto de validação, o algoritmo MULTI-GA-RV inclui menos variáveis, em média, ao subconjunto e o algoritmo NSGA2-RV apresenta um menor risco médio de classificação. Utilizando o risco médio de classificação e o número de variáveis, como objetivos, os algoritmos MULTI-GA-RV, NSGA2-RV e AEMT-RV conseguiram reduzir o tamanho do subconjunto de variáveis em 48, 32 e 44 % respectivamente, em relação ao MONO-GA-R, e apresentaram em média uma taxa de erros de 10, 12 e 11 % enquanto o MONO-GA-R apresentou uma taxa de erros média de 8 %, ou seja, 4 % maior no pior caso. Sendo assim os algoritmos reduziram o número de variáveis significativamente, com uma qualidade um pouco inferior, de forma a se aproximarem da solução ideal.

Como apresentado na Tabela 6.4 os algoritmos se mantiveram com um baixo número de erros, e conseguiram diminuir o número de variáveis selecionadas, porém as variáveis correlacionas não são removidas com esta abordagem. Assim se torna

Tabela 6.4: *Resultados Obtidos Pelos Algoritmos Multi-objetivos Otimizando Risco de Classificação e Número de Variáveis.*

	MULTI-GA-RV			
	Erros	Risco	Número de Variáveis	Variáveis Correlacionas
Média	3,3	0,59	3,4	0,4
Maior	$6^{(0,78^R,2^V,0^C)}$	$1,18^{(6^E,3^V,2^C)}$	$5^{(1^E,0,42^R,2^C)}$	$2^{(1^E,0,42^R,5^V)}$
Menor	$1^{(0,35^R,4^V,0^C)}$	$0,3^{(3^E,3^V,0^C)}$	$2^{(2^E,0,35^R,0^C)}$	$0^{(2^E,0,35^R,2^V)}$
Desvio	1,49	0,25	0,75	0,82
	NSGA2-RV			
	Erros	Risco	Número de Variáveis	Variáveis Correlacionadas
Média	3,95	0,55	4,5	0,9
Maior	$6^{(0,62^R,3^V,0^C)}$	$0,86^{(6^E,2^V,0^C)}$	$8^{(4^E,0,52^R,2^C)}$	$5^{(4^E,0,47^R,7^V)}$
Menor	$2^{(0,41^R,4^V,0^C)}$	$0,41^{(2^E,4^V,0^C)}$	$2^{(6^E,0,82^R,0^C)}$	$0^{(2^E,0,41^R,4^V)}$
Desvio	1,39	0,14	1,57	1,55
	AEMT-RV			
	Erros	Risco	Número de Variáveis	Variáveis Correlacionadas
Média	3,15	0,56	3,7	0,4
Maior	$10^{(1,19^R,3^V,0^C)}$	$1,2^{(10^E,3^V,2^C)}$	$6^{(1^E,0,40^R,2^C)}$	$2^{(4^E,0,36^R,3^V)}$
Menor	$1^{(0,31^R,4^V,0^C)}$	$0,31^{(1^E,4^V,0^C)}$	$2^{(6^E,0,75^R,0^C)}$	$0^{(1^E,0,41^R,4^V)}$
Desvio	2,18	0,25	0,92	0,82

^{E}: Número de Erros. ^{R}: Risco médio de classificação. ^{V}: Número de Variáveis. ^{C}: Variáveis correlacionadas.

necessária uma abordagem que otimize o risco de classificação, o número de variáveis correlacionadas e o número de variáveis.

Risco Médio de Classificação, Número de Variáveis Correlacionadas e Número de Variáveis

Na Tabela 6.5 são apresentados os resultados, médios, de erros (no conjunto de validação), risco médio de classificação (no conjunto de validação), quantidade de variáveis selecionadas pelo algoritmo e quantidade de variáveis correlacionadas. Os algoritmos evolutivos tem como objetivo o risco médio de classificação, na coleção de validação, a quantidade de variáveis que são correlacionadas e a quantidade de variáveis selecionadas. Por serem resultados médios, de 20 execuções, também são apresentados o maior e o menor resultado encontrado (a solução que encontrou o maior/menor resultado no parâmetro analisado) juntamente com os valores encontrados pelas outras funções objetivo naquela solução. Além disso são apresentados o desvio padrão dos resultados obtidos por cada função nas 20 execuções.

Com esta configuração de objetivos o AEMT-RCV apresentou melhores resultados, em média, no número de variáveis quando comparado ao MULTI-GA-RCV. Apresentou também resultados, em média, melhores em relação ao risco médio de classificação, quando comparado ao NSGA2-RCV. Os classificadores modelados pelas variáveis selecionadas pelo AEMT-RCV possuem uma taxa de erros mais baixa, em média, que

Tabela 6.5: *Resultados Obtidos Pelos Algoritmos Multi-objetivos Otimizando Risco de Classificação, Número de Variáveis Correlacionadas e Número de Variáveis.*

	MULTI-GA-RCV			
	Erros	Risco	Número de Variáveis	Variáveis Correlacionadas
Média	3,85	0,70	3,8	0
Maior	$6^{(0,48^R,3^V,0^C)}$	$1,70^{(4^E,4^V,0^C)}$	$5^{(0^E,0,3^R,0^C)}$	$0^{(0,3^R,5^V,0^C)}$
Menor	$0^{(0,3^R,5^V,0^C)}$	$0,30^{(0^E,5^V,0^C)}$	$2^{(4^E,0,69^R,0^C)}$	$0^{(0,3^R,5^V,0^C)}$
Desvio	1,39	0,14	1,57	0
	NSGA2-RCV			
	Erros	Risco	Número de Variáveis	Variáveis Correlacionadas
Média	4,1	3,04	3,1	0
Maior	$12^{(43,53^R,1^V,0^C)}$	$43,53^{(12^E,1^V,0^C)}$	$7^{(3^E,0,50^R,0^C)}$	$0^{(0^E,0,25^R,3^V)}$
Menor	$0^{(0,25^R,3^V,0^C)}$	$0,25^{(0^E,3^V,0^C)}$	$1^{(12^E,43,53^R,0^C)}$	$0^{(0^E,0,25^R,3^V)}$
Desvio	2,69	9,6	1,55	0
	AEMT-RCV			
	Erros	Risco	Número de Variáveis	Variáveis Correlacionadas
Média	3,4	0,75	3,35	0
Maior	$6^{(0,74^R,2^V),0^C}$	$2,45^{(3^E,3^V,0^C)}$	$5^{(3^E,0,52^R,0^C)}$	$0^{(0^E,0,29^R,4^V)}$
Menor	$0^{(0,29^R,4^V,0^C)}$	$0,29^{(0^E,4^V,0^C)}$	$2^{(6^E,0,74^R,0^C)}$	$0^{(0^E,0,29^R,4^V)}$
Desvio	1,76	0,52	0,74	0

^{E}: Número de Erros. ^{R}: Risco médio de classificação. ^{V}: Número de Variáveis. ^{C}: Variáveis correlacionadas.

os obtidos pelo NSGA2-RCV e pelo MULTI-GA-RCV. A melhor solução foi encontrada pelo NSGA2-RCV ela possui 3 variáveis, 0 erros no conjunto de validação, 0,25 de risco médio e elimina as variáveis correlacionadas. Porém, a pior solução também foi encontrada pelo NSGA2-RCV possuindo 1 variável, 12 erros no conjunto de validação, risco médio de classificação igual a 43,53.

6.1.3 Sensibilidade a ruídos

Para verificar a robustez dos classificadores, o modelo construído pela melhor solução de cada um dos algoritmos multi-objetivo (com melhores resultados para os objetivos analisados - RCV) foi avaliado na coleção modificada por um ruído simulado. Foi analisado o comportamento (taxa de erros) do classificador à medida que a taxa de ruído percentual crescia, este ruído foi gerado com distribuição normal, média variando entre 0 e 15% e desvio padrão igual ao da coleção. As variáveis selecionadas pelas soluções de cada um dos algoritmos é apresentada na figura 6.1.

Como os comprimentos de onda são expressos em centímetros recíprocos o eixo da abscissa é apresentado em ordem decrescente. A Tabela 6.6 apresenta o índice de variável selecionada (pelo cromossomo) e o comprimento de onda aproximado em centímetros recíprocos. Exemplo: a variável 415 equivale ao comprimento de onda de cerca de 7201 centímetros recíprocos, e a variável 1137 equivale ao comprimento de onda

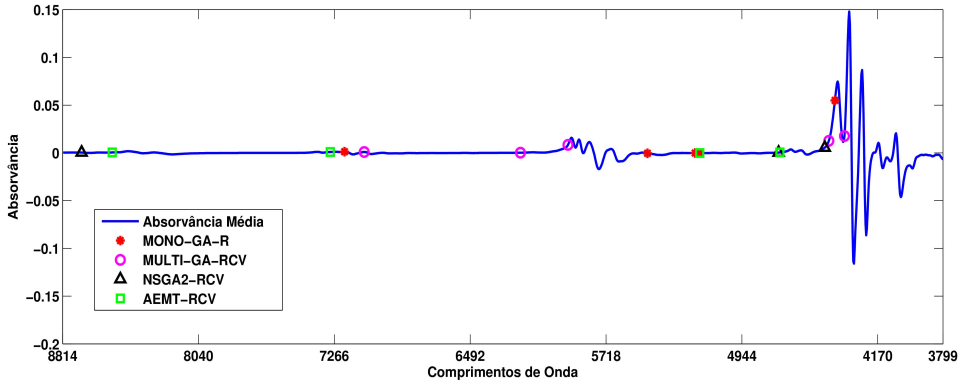

Figura 6.1: *Variáveis selecionadas pelas melhores soluções de cada um dos algoritmos.*

4414.

Tabela 6.6: *Variáveis selecionadas pelos algoritmos evolutivos.*

	Variáveis	Comprimentos de onda aproximados
MONO-GA-R	415, 861, 931, 938, 1137	7201, 5483, 5211, 5184, 4414
MULTI-GA-RCV	444, 674, 744, 1128, 1151	7095, 6205, 5935, 4449, 4360
NSGA2-RCV	28, 1054, 1122	8705, 4735, 4472
AEMT-RCV	73, 394, 938, 1056	8531, 7289, 5184, 4727

Foram aplicadas 50 configurações distintas de ruído branco (de forma aleatória) e a média de erros em função da porcentagem de ruído é apresentada na figura Figura 6.2. Uma solução será considerada útil enquanto a taxa de erros resultantes de sua classificação for menor que a apresentada por Pontes et al. [59], ou seja, 12 erros.

Da Figura 6.2 nota-se que as soluções encontradas pelos algoritmos NSGA2-RCV e AEMT-RCV são mais robustas que a solução encontrada pelo MULTI-GA-RCV. Nota-se também que ao reduzir o número de variáveis e de variáveis correlacionadas, a sensibilidade a ruído foi aprimorada, possibilitando encontrar soluções menos restritas aos conjuntos de validação e teste. As soluções encontradas pelo AEMT-RCV se mantiveram úteis por mais tempo que as soluções encontradas pelos outros algoritmos (MONO-GA-R, MULTI-GA-RCV e NSGA2-RCV). Apesar de inicialmente a solução encontrada pelo NSGA2-RCV se manter com uma média inferior aos demais algoritmos, o AEMT-RCV se manteve útil até uma porcentagem de ruído aproximada de 8,5%, enquanto que as soluções encontradas pelo MONO-GA-R e pelo MULTI-GA-RCV permaneceram úteis até um ruído aproximado de 2% e o NSGA2-RCV até um percentagem média de 6%.

6.1.4 Dispersão das amostras

Para possibilitar a visualização das superfícies de separação, do LDA, foi utilizada a solução com menor número de variáveis com uma boa acurácia (menos que 4

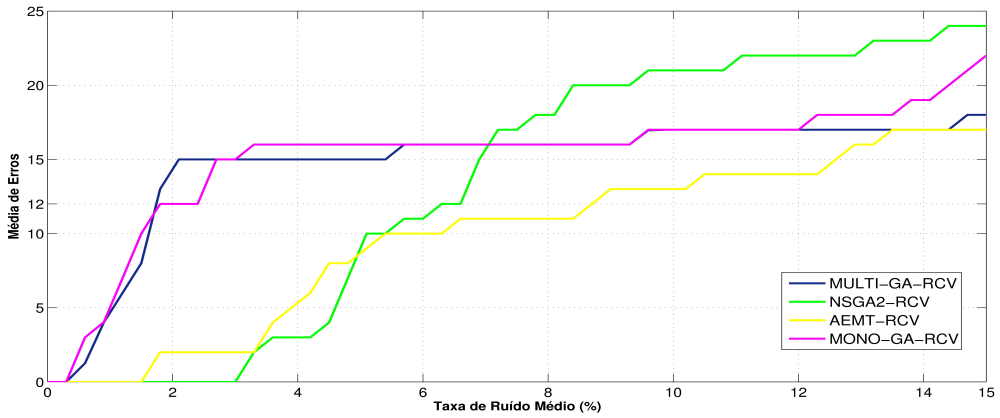

Figura 6.2: *Robustez dos algoritmos modelados nos objetivos RCV.*

erros). Desta solução foram selecionadas as duas variáveis com maior discriminância. O LDA separa as classe com base em superficies lineares, ou seja, as fronteiras de decisão entre duas classes são equações lineares. A figura 6.3 apresenta as variáveis selecionadas.

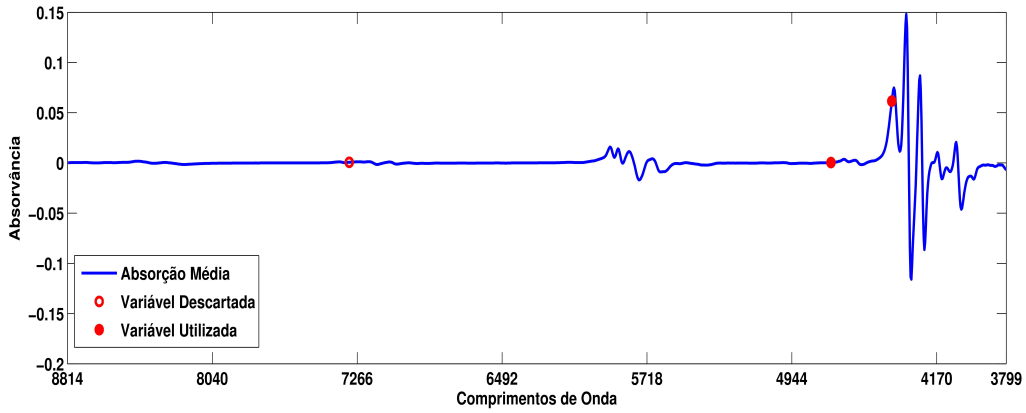

Figura 6.3: *Comprimentos de Onda Selecionados Para a Formação do Modelo de Separação.*

Na Figura 6.3 observa-se que a solução é composta pelas seguintes variáveis: 389, 1054, 1138. Estas variáveis correspondem aos comprimentos de onda: 7308, 4735, 4410, respectivamente. Quando modelado com as 3 variáveis o classificador tem uma acurácia de 91%. Com as 2 variáveis (1054 e 1138), o classificador obteve uma acurácia de 88%. As fronteiras de separação são apresentadas na Figura 6.4.

Nota-se que existem 3 amostras na região da classe 3 (OD) que pertencem a classe 2 (OBD), e uma na região da classe 2 que pertence a classe 1. O classificador não errou, na coleção de validação, na classificação de amostras da classe D e B5. Assim, o classificador obtido, classificou errado 4 amostras num conjunto total de 32 (conjunto de validação).

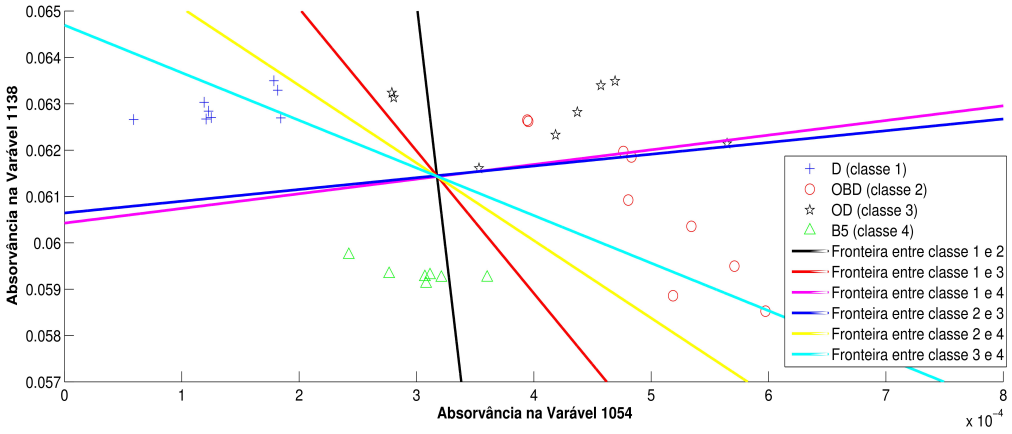

Figura 6.4: *Fronteiras de separação para o classificador LDA.*

6.2 Considerações Finais

As soluções encontradas pelo AEMT-RCV apresentam melhores desempenhos médios que as encontradas pelos NSGA2-RCV e MULTI-GA-RCV. Embora o MULTI-GA-RCV apresente uma ligeira vantagem no risco médio de classificação intuindo que ele encontra melhores classificadores quando comparado ao AEMT-RCV, é possível notar que o AEMT-RCV encontra soluções menos sensíveis a ruídos. Quando comparado ao NSGA2-RCV, o AEMT-RCV apresenta desempenho médio superior nos objetivos analisados, porém o NSGA2-RCV encontrou a melhor solução, com base nos objetivos almejados. Deve-se destacar que apesar de ter encontrado a melhor solução, o NSGA2-RCV também encontrou a pior solução, e possui em média os maiores desvios padrão.

O AEMT-RCV apresenta uma média de erros superior ao MONO-GA-RCV, porém a diferença não é significativa baseado no teste t. O tamanho médio das soluções encontradas pelo AEMT-RCV é significativamente menor, e assim é possível notar que o AEMT-RCV encontra soluções mais próximas do ideal.

Capítulo 7 - Conclusões

Neste trabalho foi proposta uma formulação do algoritmo evolutivo multi-objetivo em tabelas para o problema de seleção de variáveis em classificação multivariada. Um estudo de caso baseado na verificação de adulteração de combustível foi apresentado. O conjunto possui inicialmente 1296 variáveis e 140 amostras, sendo que estas estão divididas em 4 classes e são elas: diesel livre de biodiesel e óleos vegetais crus, D (35); amostras contendo diesel, biodiesel e óleos vegetais, OBD (38); amostras de diesel e óleo vegetal, OD (35); e amostras que são compostas de 5% de biodiesel, com variação de $\pm 0,5\%$, B5(32).

Dos resultados obtidos foi possível observar que o uso de algoritmos multiojetivos melhorou a classificação (número de erros), reduziu o tamanho do subconjunto de variáveis e removeu as variáveis correlacionadas do modelo. Os modelos obtidos com estas formulações apresentaram uma melhor sensibilidade a ruídos, sendo assim, modelos mais robustos. Os algoritmos multi-objetivos propostos usam uma menor quantidade de variáveis correlacionadas, possuem uma boa acurácia média e reduzem a média de variáveis utilizadas.

A contribuição que este trabalho apresenta em relação à algoritmos de seleção de variáveis para a classificação é o uso de outros objetivos como a redução das variáveis correlacionadas, e a busca por subconjuntos próximos do ideal. Neste cenário, foi proposto o uso de um algoritmo multi-objetivo em tabelas para selecionar um subconjunto mínimo de variáveis, removendo as variáveis correlacionadas, e promovendo uma maior capacidade discriminativa.

Quando comparado aos resultados publicados por Pontes et al [59], mesmo a formulação mono-objetiva (utilizando-se do risco de classificação) já apresenta uma melhor capacidade de classificação, o que mostra que os algoritmos evolutivos são uma boa estratégia para a seleção de variáveis em problemas de classificação. Nas formulações multi-objetivo o AEMT se mostrou melhor em pelo menos um dos objetivos analisados.

O AEMT-RCV apresentou uma melhor sensibilidade a ruídos, quando comparado ao MULTI-GA-RCV, e uma sensibilidade equivalente ao NSGA2-RCV. Se comparado aos outros algoritmos o AEMT-RCV apresentou melhores médias nos objetivos V

(número de variáveis), e C (número de variáveis correlacionadas), quando comparado ao MULTI-GA-RCV; em relação ao NSGA2-RCV se mostrou melhor no R (risco médio de classificação) e utiliza de subconjuntos de variáveis com tamanho mais estáveis (menores intervalo de dados, e desvio padrão).

7.1 Trabalhos Futuros

Como estudos futuros, sugere-se adicionar tabelas baseadas em dominância de pareto para todos os objetivos em questão e uma tabela sobre a força de pareto das soluções; o uso de funções de distâncias diferentes para a melhoria do classificador; o uso de mais de um método de classificação para a otimização do classificador.

Em relação a busca de variáveis espectrais em problemas de classificação de dados químicos, nem todas as regiões do espaço de busca contém variáveis significantes. Ao se modelar o espaço de busca pode-se optar apenas por regiões relevantes. Assim o local da variável espectral no espectro pode ser utilizado como função de pontuação/penalidade às funções de avaliação, ou servir como base para novas funções de aptidão de soluções. Sugere-se então encontrar uma métrica de avaliação, para ponderar a solução, de acordo com o grau de relevância (baseado na posição, no espectro) das variáveis contidas na solução.

Em relação aos algoritmos multi-objetivo é sugerido que sejam analisadas métricas de qualidade como o hipervolume.

Referências Bibliográficas

[1] BACK, T.; FOGEL, D.; MICHALEWICZ, Z. **Evolutionary Computation 1: Basic Algorithms and Operators**. Basic algorithms and operators. Taylor & Francis, 2000.

[2] BÄCK, T. **Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms**. Oxford University Press, Oxford, UK, 1996.

[3] BACK, T.; SCHWEFEL, H.-P. **Evolutionary computation: An overview**. In: *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, p. 20–29. IEEE, 1996.

[4] BÄCK, T.; SCHWEFEL, H.-P. **An overview of evolutionary algorithms for parameter optimization**. *Evolutionary computation*, 1(1):1–23, 1993.

[5] BRANKE, J.; DEB, K.; MIETTINEN, K.; SLOWINSKI, R. **Multiobjective optimization: Interactive and evolutionary approaches**, volume 5252. Springer, 2008.

[6] BRASIL, C. R. S. **Algoritmo evolutivo de muitos objetivos para predição ab initio de estrutura de proteínas**. PhD thesis, Universidade de São Paulo, 2012.

[7] CARROLL, J.; GREEN, P.; LATTIN, J.; AVRITSCHER, H. **ANALISE DE DADOS MULTIVARIADOS**. CENGAGE.

[8] COELLO, C. A. C. **A comprehensive survey of evolutionary-based multiobjective optimization techniques**. *Knowledge and Information systems*, 1(3):269–308, 1999.

[9] DA SILVA SOARES, A. **Detecção e diagnóstico de falhas empregando o algoritmo das projeções sucessivas na seleção de atributos para classificação de padrões**, 2010.

[10] DARWIN, C. **On the origins of species by means of natural selection**. *London: Murray*, 1859.

[11] DASH, M.; LIU, H. **Feature selection for classification**. *Intelligent data analysis*, 1(3):131–156, 1997.

[12] DE ENERGIA ELÉTRICA, D. **Hermes Manoel Galvão Castelo Branco**. PhD thesis, Universidade de São Paulo, 2013.

[13] DE JONG, K. A. **Evolutionary computation: a unified approach**. MIT press, 2006.

[14] DEB, K.; AGRAWAL, S.; PRATAB, A.; MEYARIVAN, T. **A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II**. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.

[15] DEB, K.; JAIN, H. **Handling many-objective problems using an improved nsga-ii procedure**. In: *Evolutionary Computation (CEC), 2012 IEEE Congress on*, p. 1–8, June 2012.

[16] DEB, K. **Introduction to evolutionary multiobjective optimization**. In: *Multiobjective Optimization*, p. 59–96. Springer, 2008.

[17] DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. **A fast elitist multi-objective genetic algorithm: Nsga-ii**. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2000.

[18] DIWEKAR, U. **Introduction to applied optimization**, volume 22. Springer, 2008.

[19] DOS SANTOS, A. C. **Algoritmo evolutivo computacionalmente eficiente para reconfiguração de sistemas de distribuição**, 2009.

[20] ESCUDERO-GILETE, M.; GONZÁLEZ-MIRET, M.; HEREDIA, F. **Application of multivariate statistical analysis to quality control systems. relevance of the stages in poultry meat production**. *Food Control*, 40(0):243 – 249, 2014.

[21] FERREIRA, D. F. **Análise multivariada**. *Lavras: UFLA*, 22, 1996.

[22] FOGEL, L.; OWENS, A.; WALSH, M. **Artificial Intelligence Through Simulated Evolution**. John Wiley & Sons, 1966.

[23] FONSECA, C. M.; FLEMING, P. J.; OTHERS. **Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization.** In: *ICGA*, volume 93, p. 416–423, 1993.

[24] FRIEDMAN, J. H. **Regularized discriminant analysis**. *Journal of the American statistical association*, 84(405):165–175, 1989.

[25] GABRIEL, P. H. R.; DELBEM, A. C. B. **Fundamentos de algoritmos evolutivos**, 2008.

[26] GHOSH, A.; DEHURI, S. **Evolutionary algorithms for multi-criterion optimization: A survey**. *International Journal of Computing & Information Sciences*, 2(1):38–57, 2004.

[27] GOLEŽ, M.; HLADNIK, A. **Interpreting the age of the ruins of st. john the baptist's church with multivariate analysis**. *Journal of Cultural Heritage*, 14(4):354 – 358, 2013.

[28] GRASSI, S.; AMIGO, J. M.; LYNDGAARD, C. B.; FOSCHINO, R.; CASIRAGHI, E. **Beer fermentation: Monitoring of process parameters by ft-nir and multivariate data analysis**. *Food Chemistry*, 155(0):279 – 286, 2014.

[29] HANSEN, N.; ARNOLD, D. V.; AUGER, A. **Evolution strategies**. 2013.

[30] HASWELL, S. **Practical Guide to Chemometrics**. Taylor & Francis, 1992.

[31] HOLLAND, J. H. **Genetic algorithms**. *Scientific american*, 267(1):66–72, 1992.

[32] HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. E. **A niched pareto genetic algorithm for multiobjective optimization**. In: *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, p. 82–87. Ieee, 1994.

[33] ISHIBUCHI, H.; TSUKAMOTO, N.; NOJIMA, Y. **Evolutionary many-objective optimization: A short review**. In: *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, p. 2419–2426, June 2008.

[34] ISHIBUCHI, H.; TSUKAMOTO, N.; NOJIMA, Y. **Evolutionary many-objective optimization: A short review.** In: *IEEE Congress on Evolutionary Computation*, p. 2419–2426, 2008.

[35] IZENMAN, A. **Modern multivariate statistical techniques: regression**. *Classification, and Manifold Learning Springer Texts in Statistics, New York*, 2008.

[36] JOHNSON, R. A.; WICHERN, D. W. **Applied multivariate statistical analysis**, volume 5. Prentice hall Upper Saddle River, NJ, 2002.

[37] JONES, A. J. **Genetic algorithms and their applications to the design of neural networks**. *Neural Computing & Applications*, 1(1):32–45, 1993.

[38] JONES, G. **Genetic and evolutionary algorithms**. *Encyclopedia of Computational Chemistry. John Wiley and Sons*, 1998.

[39] JORGE, C. A. C. **Algoritmo evolutivo multi-objetivo de tabelas para seleção de variáveis em calibração multivariada**. Master's thesis, Universidade Federal de Goiás, Goiás, april 2014.

[40] KOHAVI, R.; JOHN, G. H. **Wrappers for feature subset selection**. *Artificial intelligence*, 97(1):273–324, 1997.

[41] LAMPMAN, G.; PAVIA, D.; KRIZ, G.; VYVYAN, J. **Introdução a espectroscopia**. CENGAGE.

[42] LAVINE, B. K.; DAVIDSON, C.; MOORES, A. J.; GRIFFITHS, P. **Raman spectroscopy and genetic algorithms for the classification of wood types**. *Applied Spectroscopy*, 55(8):960–966, 2001.

[43] LEARDI, R. **Genetic algorithms in chemometrics and chemistry: a review**. *Journal of Chemometrics*, 15(7):559–569, 2001.

[44] LEARDI, R. **Genetic algorithms in chemistry**. *Journal of Chromatography A*, 1158(1–2):226 – 233, 2007. Data Analysis in Chromatography.

[45] LEE, S.; CHOI, W. S. **A multi-industry bankruptcy prediction model using back-propagation neural network and multivariate discriminant analysis**. *Expert Systems with Applications*, 40(8):2941 – 2946, 2013.

[46] LEROY, S.; COHEN, S.; VERNA, C.; GRATUZE, B.; TÉREYGEOL, F.; FLUZIN, P.; BERTRAND, L.; DILLMANN, P. **The medieval iron market in ariège (france). multidisciplinary analytical approach and multivariate analyses**. *Journal of Archaeological Science*, 39(4):1080 – 1093, 2012.

[47] LIPOWSKI, A.; LIPOWSKA, D. **Roulette-wheel selection via stochastic acceptance**. *Physica A: Statistical Mechanics and its Applications*, 391(6):2193–2196, 2012.

[48] LONGOBARDI, F.; VENTRELLA, A.; CASIELLO, G.; SACCO, D.; CATUCCI, L.; AGOSTIANO, A.; KONTOMINAS, M. **Instrumental and multivariate statistical analyses for the characterisation of the geographical origin of apulian virgin olive oils**. *Food Chemistry*, 133(2):579 – 584, 2012.

[49] MICHALEWICZ, Z.; HINTERDING, R.; MICHAELEWICZ, M. **Fuzzy evolutionary computation**. chapter Evolutionary Algorithms, p. 3–31. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[50] MICHALEWICZ, Z.; SCHOENAUER, M. **Evolutionary algorithms**. *Wiley Encyclopedia of Operations Research and Management Science*.

[51] MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. C. **Machine learning, neural and statistical classification**. 1994.

[52] MITCHELL, M. **An introduction to genetic algorithms**. MIT press, 1998.

[53] MOBARAKI, N.; HEMMATEENEJAD, B. **Structural characterization of carbonyl compounds by {IR} spectroscopy and chemometrics data analysis**. *Chemometrics and Intelligent Laboratory Systems*, 109(2):171 – 177, 2011.

[54] NÆS, T.; ISAKSSON, T.; FEARN, T.; DAVIES, T. **A user-friendly guide to multivariate calibration and classification**, volume 6. NIR publications Chichester, 2002.

[55] NAES, T.; MEVIK, B.-H. **Understanding the collinearity problem in regression and discriminant analysis**. *Journal of Chemometrics*, 15(4):413–426, 2001.

[56] NAES, T.; MEVIK, B.-H. **Understanding the collinearity problem in regression and discriminant analysis**. *Journal of Chemometrics*, 15(4):413–426, 2001.

[57] PASQUINI, C. **Near Infrared Spectroscopy: fundamentals, practical aspects and analytical applications**. *Journal of the Brazilian Chemical Society*, 14:198 – 219, 04 2003.

[58] PONTES, M. J. C.; GALVAO, R. K. H.; ARAUJO, M. C. U.; MOREIRA, P. N. T.; NETO, O. D. P.; JOSE, G. E.; SALDANHA, T. C. B. **The successive projections algorithm for spectral variable selection in classification problems**. *Chemometrics and Intelligent Laboratory Systems*, 78(1):11–18, 2005.

[59] PONTES, M. J. C.; PEREIRA, C. F.; PIMENTEL, M. F.; VASCONCELOS, F. V. C.; SILVA, A. G. B. **Screening analysis to detect adulteration in diesel/biodiesel blends using near infrared spectrometry and multivariate classification**. *Talanta*, 85(4):2159–2165, 2011.

[60] RENCHER, A. **Methods of Multivariate Analysis**. Wiley Series in Probability and Statistics. Wiley, 2003.

[61] RENCHER, A. C.; CHRISTENSEN, W. F. **Methods of multivariate analysis**, volume 709. John Wiley & Sons, 2012.

[62] SANCHES, D. S. **Algoritmos evolutivos multi-objetivo para reconfiguração de redes em sistemas de distribuição de energia elétrica**, Dec. 2012.

[63] SANCHES, D.; MAZUCATO, S.; CASTOLDI, M.; DELBEM, A.; LONDON, J. **Combining subpopulation tables, non-dominated solutions and strength pareto of moeas to treat service restoration problem in large-scale distribution systems**. In:

Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE, p. 1986–1991, Nov 2013.

[64] SHOVAL, N.; RAVEH, A. **Categorization of tourist attractions and the modeling of tourist cities: based on the co-plot method of multivariate analysis**. *Tourism Management*, 25(6):741 – 750, 2004.

[65] SILVA, C. S.; BORBA, F. D. S. L.; PIMENTEL, M. F.; PONTES, M. J. C.; HONORATO, R. S.; PASQUINI, C. **Classification of blue pen ink using infrared spectroscopy and linear discriminant analysis**. *Microchemical Journal*, 2012.

[66] SILVA, G. D. D.; ACÚRCIO, F. D. A.; CHERCHIGLIA, M. L.; GUERRA JÚNIOR, A. A.; ANDRADE, E. I. G. **Medicamentos excepcionais para doença renal crônica: gastos e perfil de utilização em minas gerais, brasil; dispensing of exceptional drugs for chronic renal failure: expenditures and patients' profile in minas gerais state, brazil**. *Cad. saúde pública*, 27(2):357–368, 2011.

[67] SILVERSTEIN, R.; WEBSTER, F. **Spectrometric identification of organic compounds**. John Wiley & Sons, 2006.

[68] SKOOG, D.; HOLLER, F.; NIEMAN, T.; CARACELLI, I. **Príncipios de análise instrumental**. Bookman, 2002.

[69] SRINIVAS, N.; DEB, K. **Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms**. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[70] SRINIVAS, N.; DEB, K. **Multiobjective optimization using nondominated sorting in genetic algorithms**. *Evolutionary Computation*, 2:221–248, 1994.

[71] SRINIVAS, N.; DEB, K. **Muiltiobjective optimization using nondominated sorting in genetic algorithms**. *Evolutionary computation*, 2(3):221–248, 1994.

[72] STUART, B. H. **Infrared Spectroscopy: Fundamentals and Applications**. Analytical Techniques in the Sciences (AnTs) *. Wiley, 2004.

[73] SYSWERDA, G. **Uniform crossover in genetic algorithms**. In: *Proceedings of the 3rd International Conference on Genetic Algorithms*, p. 2–9, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[74] TOSCANI, L.; VELOSO, P. **Complexidade de Algoritmos: Série Livros Didáticos Informática UFRGS - Vol. 13**. Bookman.

[75] VACCA, E.; VELLA, G. D. **Metric characterization of the human coxal bone on a recent italian sample and multivariate discriminant analysis to determine sex**. *Forensic Science International*, 222(1–3):401.e1 – 401.e9, 2012.

[76] VARGAS, D. V.; MURATA, J.; TAKANO, H.; DELBEM, A. C. B. **General subpopulation framework and taming the conflict inside populations**. *Evolutionary computation*, p. 1–36, 2014.

[77] WANG, P.; WEISE, T.; CHIONG, R. **Novel evolutionary algorithms for supervised classification problems: an experimental study**. *Evolutionary Intelligence*, 4(1):3–16, 2011.

[78] WRIGHT, S. **The roles of mutation, inbreeding, crossbreeding, and selection in evolution**, volume 1. na, 1932.

[79] YU, X.; GEN, M. **Introduction to evolutionary algorithms**. Springer, 2010.

[80] ZITZLER, E. **Evolutionary algorithms for multiobjective optimization: Methods and applications**, 1999.

[81] ZITZLER, E.; DEB, K.; THIELE, L. **Comparison of multiobjective evolutionary algorithms: Empirical results**. *Evolutionary computation*, 8(2):173–195, 2000.

[82] ZITZLER, E.; LAUMANNS, M.; BLEULER, S. **A tutorial on evolutionary multiobjective optimization**. In: *Metaheuristics for multiobjective optimisation*, p. 3–37. Springer, 2004.

[83] ZITZLER, E.; LAUMANNS, M.; THIELE, L. **Spea2: Improving the strength pareto evolutionary algorithm**. 2001.

[84] ZITZLER, E.; THIELE, L.; ZITZLER, E.; ZITZLER, E.; THIELE, L.; THIELE, L. **An evolutionary algorithm for multiobjective optimization: The strength pareto approach**, volume 43. Citeseer, 1998.