

COMPARAÇÃO ENTRE TÉCNICAS DE OTIMIZAÇÃO MULTIOBJETIVOS PARA TREINAMENTO DE CLASSIFICADORES ADABOOST PARA DETECÇÃO DE PLAQUETAS

Trabalho de Conclusão de Curso

Engenharia da Computação

Lizandra Raflesia Monteiro de Lira

Orientador: Carmelo José Albanez Bastos Filho

Lizandra Raflesia Monteiro de Lira

***COMPARAÇÃO ENTRE TÉCNICAS DE
OTIMIZAÇÃO MULTIOBJETIVOS PARA
TREINAMENTO DE CLASSIFICADORES
ADABOOST PARA DETECÇÃO DE
PLAQUETAS***

Monografia apresentada para obtenção do
Grau de Bacharel em Engenharia da Com-
putação pela Universidade de Pernambuco

Orientador:
Prof. Dr. Carmelo José Albanez Bastos Filho

GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO
ESCOLA POLITÉCNICA DE PERNAMBUCO
UNIVERSIDADE DE PERNAMBUCO

**Recife - PE, Brasil
Junho de 2014**

Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação
Projeto de Final de Curso



MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 2 de 7 de 2014, às 15:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente LIZANDRA RAFLESIA MONTEIRO DE LIRA, orientado pelo professor Carmelo José Albanez Bastos Filho, sob título COMPARAÇÃO ENTRE TÉCNICAS DE OTIMIZAÇÃO MULTIOBJETIVOS PARA TREINAMENTO DE CLASSIFICADORES ADABOOST PARA DETECÇÃO DE PLAQUETAS, a banca composta pelos professores:

Bruno José Torres Fernandes

Carmelo José Albanez Bastos Filho

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada Aprovada com Restrições* Reprovada

e foi-lhe atribuída nota: 9,0 (NOVE)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 10 dias para entrega da versão final da monografia a contar da data deste documento.

BRUNO JOSÉ TORRES FERNANDES

CARMELO JOSÉ ALBANEZ BASTOS FILHO

Agradecimentos

Primeiramente gostaria de agradecer a meus pais, Cláudio Roberto e Denise Monteiro, por sempre terem me apoiado e ajudado em minha longa jornada, me dando forças e coragem para enfrentar os novos desafios e vencer os antigos. Ao meu irmão, Felipe Luan, por sempre ter compreendido meus momentos de estudo e aceitado meus pedidos de “abaixa o som, por favor! ”. Gostaria de agradecer também a meu namorado, Diógenes Diniz, por sua paciência e por sempre ter estado ao meu lado me deixando mais calma nos momentos de estudos.

Queria agradecer também ao meu orientador, Carmelo Filho, por seu tempo e conselhos desde a época de Iniciação Científica (IC), construindo a base de conhecimento para a elaboração deste trabalho de conclusão de curso. Além do meu orientador, pessoas participaram do meu aprendizado e desenvolvimento desta monografia. Por isso gostaria de agradecer a Dennis Cunha por disponibilizar seu tempo e ensinamento durante meu período de IC, assim como a Willamos de Aguiar por te me “socorrido” na elaboração deste trabalho.

Por fim, quero agradecer aos amigos conhecidos nesses anos de faculdade, tornando a jornada mais alegre, fazendo das noites de estudos algo menos cansativo. Pessoas que se tornaram muito especiais, propiciando momentos únicos e inesquecíveis. Agradecer ao meu amigo John Kennedy, por sua amizade deste meu primeiro dia na faculdade, amizade que será para o resto da vida. Como também aos vários amigos adquiridos, como Alexandre Azevedo, Bruno Yamashita, Geraldo Seabra, Humberto Beltrão, Rodrigo Augusto, Victor Vilmarques, entre outros. Agradecer também aos amigos fora da faculdade, por seus apoios e conselhos durante minha jornada.

Enfim, gostaria de agradecer aos todos que fizeram dessa conquista algo especial para mim. À todos vocês, muito obrigada!

Resumo

O sangue humano é constituído por três tipos de células, onde as plaquetas têm uma atuação importante na coagulação sanguínea e sua contagem é necessária para que sejam identificadas alterações plaquetárias, tanto de forma quantitativa quanto qualitativa. *Boosting* é um método para aprimorar o desempenho de algoritmos de aprendizado, usando classificadores fracos para compor classificadores fortes, sendo o *AdaBoost* um dos mais influentes algoritmos de *boosting* existentes. Este trabalho tem como objetivo analisar e comparar o desempenho entre técnicas de Otimização Multiobjetivo para o treinamento de classificadores *AdaBoost* para realizar a detecção e contagem automática de plaquetas. Para otimizar o treinamento dos *AdaBoost*, são aplicados um Algoritmo Evolucionário, o SPEA 2 (Algoritmo Evolutivo da Força de Pareto 2), e um Algoritmo baseado em Enxame de Partículas, o DFMOPSO (Fator de Diversidade para Otimização Multiobjetiva por Enxame de Partículas). Foram realizados trinta processos de treinamento independentes usando mil iterações com cem partículas em cada técnica para realizar a otimização do *AdaBoost* com cinco, dez e vinte classificadores fracos. Os resultados obtidos foram comparados com técnicas mono-objetivas e mostraram que o SPEA 2 apresentou melhores resultados com dez classificadores, possuindo convergência mais lenta para vinte classificadores fracos, enquanto o DFMOPSO apresentou melhores resultados com cinco classificadores fracos.

Abstract

The human blood is composed of three types of cells, where the platelets have an important role in blood clotting and your count it's required so that platelets changes to be identified, both quantitatively and qualitatively. Boosting is a method to enhance the performance of learning algorithms, using weak classifiers to form strong classifiers, where the AdaBoost is one of the most influential boosting algorithms existing. This work aims to analyze and compare the performance among Multiobjective Optimization techniques for training classifiers AdaBoost to perform automatic detection and platelet count. To optimize the training of AdaBoost are applied one Evolutionary Algorithm, the SPEA 2 (Strength Pareto Evolutionary Algorithm 2), and na Algorithm based on Particle Swarm, the DFMOPSO (Diversity Factor Multi-Objective Particle Swarm Optimization). Thirty independent training procedures were performed using thousand iterations with one hundred particles to perform AdaBoost optimization with five, ten and twenty weak classifiers. The results achieved were compared with mono-objective techniques and showed that the SPEA 2 showed better results with ten classifiers, having convergence more slow with twenty weak classifiers, while the DFMOPSO showed better results with five weak classifiers.

“It happens more than you think, because this is not just engineering and science but also art. Sometimes when you’re in the middle of a crisis and you’re not sure if you will overcome it. But we Always did and so we have a certain level of confidence in ourselves, even if we question the times. I believe the key is that the fear comes in during difficult times. We must consider that we put our heart and soul into the products”

– Steve Jobs

Sumário

Índice de Figuras

Índice de Tabelas

Índice de Algoritmos

Lista de Abreviaturas e Siglas

1	Introdução	1
1.1	Motivação e Caracterização do Problema	1
1.2	Objetivos e Metas	2
1.3	Organização do Documento	3
2	Revisão Bibliográfica: Plaquetas	4
2.1	Morfologia das Plaquetas	4
2.2	Alterações Qualitativas e Quantitativas	5
2.3	Métodos de Contagem de Plaquetas	7
2.3.1	Método Manual	7
2.3.2	Método Automatizado	8
3	AdaBoost	10
3.1	Modelo de Viola-Jones (Método de Detecção de Faces)	10
3.2	Imagem Integral	11
3.3	Características de Haar	12
3.4	Classificador Fraco	13
3.5	Algoritmo de Aprendizado	14
3.5.1	AdaBoost	15
3.6	Cascata de Classificadores	17
3.6.1	Treinamento da Cascata de Classificadores	18
4	Otimização Multi-Objetivos	20

4.1	Conceitos de Otimização Multiobjetivos	20
4.2	Algoritmos Evolucionários	23
4.2.1	SPEA 2	23
4.2.1.1	Função da Força do Pareto	25
4.2.1.2	Algoritmo de Corte de Excedentes do Arquivo Externo	26
4.3	Algoritmos baseados em Enxame de Partículas para Otimização Multiobjetiva	27
4.3.1	<i>Multi-Objective Particle Swarm Optimization</i> (MOPSO)	27
4.3.2	MOPSO-DFR.....	28
4.3.2.1	Seleção do Líder Cognitivo	30
4.3.2.2	Seleção do Líder Social	31
4.3.3	DFMOPSO	32
4.3.3.1	Equação da Posição e Velocidade	33
4.4	Métrica para avaliação da Frente de Pareto.....	35
4.4.1	<i>Hypervolume</i> (HV)	36
5	Arranjo Experimental e Resultados	37
5.1	Arranjo Experimental	37
5.1.1	Treinamento do <i>AdaBoost</i> utilizando o SPEA 2	38
5.1.2	Treinamento do <i>AdaBoost</i> utilizando o DFMOPSO	39
5.1.3	Ambiente de Desenvolvimento	37
5.1.4	Imagens e Conjunto de Treinamento.....	37
5.2	Análise dos Resultados	41
5.2.1	Experimento utilizando 5 Classificadores Fracos	42
5.2.2	Experimento utilizando 10 Classificadores Fracos	43
5.2.3	Experimento utilizando 20 Classificadores Fracos	46
5.2.4	Experimento Geral.....	48

6 Conclusões e Trabalhos Futuros 51

6.1 Conclusões e Contribuições51

6.2 Trabalhos Futuros52

Referências Bibliográficas

Índice de Figuras

Figura 2.1 Megacariócito formando plaquetas fora da periferia.....	5
Figura 2.2 Plaqueta Gigante.....	7
Figura 3.1 Estrutura proposta por Viola e Jones	11
Figura 3.2 Cálculo da Imagem Integral.....	12
Figura 3.3 Características de Haar.....	13
Figura 3.4 Cascata de Classificadores.....	17
Figura 4.1 Mapeamento de uma solução no espaço de variáveis para o espaço de objetivos	21
Figura 4.2 Conceito de Dominância	22
Figura 4.3 Demonstração de como será realizado o cálculo do DF para um problema com dois objetivos.....	29
Figura 4.4 Demonstração da Seleção do Líder Cognitivo	30
Figura 4.5 Demonstração da Seleção do Líder Social	32
Figura 4.6 Cálculo do <i>Crowding Distance</i>	33
Figura 4.7 União de retângulos para o cálculo do <i>hypervolume</i> em um problema com dois objetivos.....	36
Figura 5.1 Análise do <i>hypervolume</i> para o SPEA 2.....	42
Figura 5.2 Análise do <i>hypervolume</i> para o DFMOPSO	42
Figura 5.3 Comparativo entre as duas técnicas implementadas analisando os dois objetivos conflitantes utilizando 5 classificadores.....	43
Figura 5.4 Comparativo entre as técnicas, utilizando 5 classificadores: O gráfico esquerdo representa a taxa de <i>miss rate</i> e o direito a taxa de falso positivo	44

Figura 5.5 Resultados das métricas utilizadas por Silva [38], utilizando 5 classificadores.....	45
Figura 5.6 Comparativo entre as duas técnicas implementadas analisando os dois objetivos conflitantes utilizando 10 classificadores.....	45
Figura 5.7 Comparativo entre as técnicas, utilizando 10 classificadores: O gráfico esquerdo representa a taxa de <i>miss rate</i> e o direito a taxa de falso positivo	46
Figura 5.8 Resultados das métricas utilizadas por Silva [38], utilizando 10 classificadores.....	46
Figura 5.9 Comparativo entre as duas técnicas implementadas analisando os dois objetivos conflitantes utilizando 20 classificadores.....	47
Figura 5.10 Comparativo entre as técnicas, utilizando 20 classificadores: O gráfico esquerdo representa a taxa de <i>miss rate</i> e o direito a taxa de falso positivo	48
Figura 5.11 Resultados das métricas utilizadas por Silva [38], utilizando 20 classificadores.....	48
Figura 5.12 Comparação entre as Frentes de Paretos para 5,10 e 20 classificadores no SPEA 2.....	49
Figura 5.13 Comparação entre as Frentes de Paretos para 5,10 e 20 classificadores no DFMOPSO	50

Índice de Tabelas

Tabela 2.1 Principais causas de Plaquetopenias e Plaquetoses	6
Tabela 5.1 Distribuição da Quantidade de Imagens	41

Índice de Algoritmos

Algoritmo 1. Pseudocódigo do <i>AdaBoost</i>	16
Algoritmo 2. Pseudocódigo do treinamento de uma cascata de classificadores	19
Algoritmo 3. Pseudocódigo do SPEA 2	24
Algoritmo 4. Pseudocódigo do DFMOPSO.....	35
Algoritmo 5. Pseudocódigo do Treinamento do <i>AdaBoost</i> utilizando o SPEA 2	38
Algoritmo 6. Pseudocódigo do Treinamento do <i>AdaBoost</i> utilizando o DFMOPSO	40

Lista de Abreviaturas e Siglas

- AdaBoost** – Do inglês, *Adaptive Boosting* (Impulso/Estímulo Adaptativo);
- AE** – Arquivo Externo;
- ASH** – Do inglês, *American Society of Hematology* (Sociedade Americana de Hematologia);
- DF** – Do inglês, *Diversity Factor* (Fator de Diversidade);
- DFMOPSO** – Do inglês, *Diversity Factor Multi-Objective Particle Swarm Optimization* (Fator de Diversidade para Otimização Multiobjetiva por Enxame de Partículas);
- EA** – Do inglês, *Evolutionary Algorithm* (Algoritmo Evolucionário);
- FSS** – Do inglês, *Fish School Search* (Busca por Cardumes);
- GA** – Do inglês, *Genetic Algorithm* (Algoritmo Genético);
- Java** – Linguagem de Programação Orientada a Objetos;
- MOP** – Do inglês, *Multi-Objective Problems* (Problemas Multiobjetivos);
- MOPSO-DFR** – Do inglês, *Multi-Objective Particle Swarm Optimization using Diversity Factor and Roulette Wheel* (Otimização Multiobjetiva por Enxame de Partículas usando Fator de Diversidade e Método da Roleta);
- OpenCV** – Do inglês, *Open Source Computer Vision Library* (Biblioteca de Código Aberto para a área de Visão Computacional);
- Pixel** – Do inglês, *Picture Element* (Elemento da Imagem);
- PSO** – Do inglês, *Particle Swarm Optimization* (Otimização por Enxame de Partículas);
- RAM** – Do inglês, *Random Access Memory* (Memória de Acesso Aleatório);
- SPEA-2** – Do inglês, *Strength Pareto Evolutionary Algorithm 2* (Algoritmo Evolutivo da Força de Pareto 2);
- VPM** – Volume Plaquetário Médio;

1 *Introdução*

“A única forma de descobrir os limites do possível é aventurar-se um pouco além deles, para o impossível”

– Arthur Clarke

Neste trabalho de conclusão de curso serão utilizadas técnicas de otimização de problemas multiobjetivos para otimizar o treinamento do AdaBoost, técnica de classificação baseada em Classificadores Fracos, para auxiliar na identificação e contagem automática de células sanguíneas.

Neste capítulo é apresentada a introdução desta monografia, e está organizado em 3 seções. Na Seção 1.1 é apresentada a motivação para a realização deste documento assim como a caracterização do problema abordado. Em seguida, na Seção 1.2 são apresentados os objetivos gerais e específicos e a hipótese de como o problema poderá ser solucionado. Por fim, na Seção 1.3 é descrita a estrutura do restante da monografia.

1.1 **Motivação e Caracterização do Problema**

O sangue humano é composto por duas frações combinadas, sendo 34% para os elementos figurados e 66% para o plasma [1]. Os elementos figurados referem-se a três tipos de células: os eritrócitos (glóbulos vermelhos ou hemácias), os leucócitos (glóbulos brancos) e os trombócitos (plaquetas) [2].

As plaquetas têm como função principal atuar na coagulação sanguínea, evitando vazamento espontâneo de sangue em processos hemorrágicos [3].

Considerando sua importância, a contagem de plaquetas torna-se necessária para que sejam identificadas alterações plaquetárias, tanto de forma quantitativa quanto qualitativa [4]. Trombocitose é o aumento do número de plaquetas no sangue e pode ser ocasionado por Leucemia Mielóide, doenças infecciosas crônicas,

1.2 Objetivos e Metas

períodos pós-hemorragicos, dentre outras complicações, levando a sintomas como náuseas, enjoos, labirintite, dentre outros. Por outro lado, a redução do número de plaquetas, conhecida por plaquetopenia, tem como causas principais a redução da produção medular, viroses em geral, rubéola congênita, dentre outras, ocasionando sangramentos [5].

A contagem das plaquetas pode ser realizada de duas formas: o método manual, através da microscopia, por exemplo, onde serão adicionadas informações úteis àquelas obtidas pelos analisadores hematológicos, ou automatizada, utilizando técnicas como tecnologia de impedância, dispersão de luz, ou ambas [6].

A contagem das plaquetas através do método manual fica custoso devido ao grande volume de exames feitos por um laboratório. Apesar de produzirem resultados mais precisos, as contagens automatizadas ainda assim estão sujeitas a falhas [4]. Podem ocorrer casos onde a presença de fragmentação das hemácias resulte em um diagnóstico equivocado de trombocitose, assim como aglomerados plaquetários ou plaquetas gigantes podem levar a um falso cenário de plaquetopenia [7].

Para ajudar na identificação e contagem das plaquetas, técnicas de reconhecimento de padrões podem ser aplicadas [6]. Para aperfeiçoar o treinamento e aumentar a eficiência dessa técnica de reconhecimento de padrão, é possível fazer uso de técnicas de inteligência computacional. Dessa forma, procura-se soluções para os objetivos conflitantes deste problema, ou seja, soluções que aumentem a taxa de classificação (de acertos) e diminuam a taxa de falsos positivos (indicação de plaquetas onde não existe), contribuindo para resultados mais precisos da contagem.

1.2 Objetivos e Metas

O objetivo principal deste trabalho é desenvolver um sistema computacional que seja capaz de auxiliar na identificação e contagem automática de plaquetas,

1.3 Organização do Documento

evitando a presença de agregados plaquetários, macro plaquetas, plaquetas gigantes, assim como fragmentos celulares nas amostras sanguíneas.

Para o desenvolvimento do sistema proposto, será utilizada uma técnica de classificação baseada em Classificadores fracos, chamada de *AdaBoost*. Para aumentar a eficiência e otimizar o treinamento do *AdaBoost*, serão implementadas técnicas de otimização de problemas multiobjetivos (MOP, *Multi-Objective Problems*) por apresentarem bons resultados para problemas com 2 ou mais objetivos conflitantes (no caso da proposta desse trabalho, é a taxa de acertos e a taxa de falsos positivos). Dentre essas técnicas, serão utilizadas um algoritmo evolucionário, o SPEA-2 (*Strength Pareto Evolutionary Algorithm 2*) e um algoritmo baseado em enxames, o DFMOPSO (*Diversity Factor Multi-Objective Particle Swarm Optimization*). Ambos os algoritmos fazem uso de uma população externa, conseguindo obter resultados mais eficientes.

1.3 Organização do Documento

Este trabalho está organizado em seis capítulos. No Capítulo 2, serão abordados aspectos relativos à morfologia das plaquetas e suas anomalias, assim como as técnicas existentes relacionadas à contagem de plaquetas. Em seguida, no Capítulo 3, será explanado o mecanismo de classificação e detecção de objetos utilizado neste trabalho, o *AdaBoost*. O Capítulo 4 demonstrará as técnicas multiobjetivas utilizadas para a detecção de plaquetas, o SPEA 2 (do inglês, *Strength Pareto Evolutionary Algorithm 2*) e o DFMOPSO (do inglês, *Diversity Factor Multi-Objective Particle Swarm Optimization*). Os experimentos e os resultados serão apresentados no Capítulo 5. Por fim, no Capítulo 6, serão abordadas as principais conclusões dessa monografia, assim como as propostas para trabalhos futuros.

2 Revisão Bibliográfica: Plaquetas

“Uma das causas mais comuns de todas as doenças é o diagnóstico.”

– Karl Kraus

Neste capítulo serão abordados os aspectos morfológicos das plaquetas, assim como anomalias relacionadas, respectivamente Seção 2.1 e 2.2. Em seguida, na Seção 2.3, serão mostradas algumas técnicas para a contagem de plaquetas, tanto pelo método manual quanto pelo automatizado. O objetivo deste capítulo é apresentar uma visão geral sobre o problema abordado, de modo que as necessidades sejam identificadas e os parâmetros para a construção da solução sejam mapeados.

2.1 Morfologia das Plaquetas

As plaquetas são produzidas em uma das maiores células do organismo, os megacariócitos. Sua produção ocorre através da fragmentação dessas células, na medula óssea. Cada megacariócito dá origem de 1000 a 5000 plaquetas [8]. As plaquetas apresentam formato discoide, com diâmetro de 2 a 3 μm e espessura em torno de 1 μm . Sua vida média no sangue circulante é de 8 a 12 dias [9]. A Figura 1 mostra plaquetas sendo formadas em um megacariócito.

Essas células sanguíneas têm como função principal a formação de coágulos, gerando um processo fundamental para a cicatrização de ferimentos. Isso se deve ao fato delas armazenarem grande quantidade de tromboplastina, que é uma enzima relacionada à coagulação sanguínea [10].

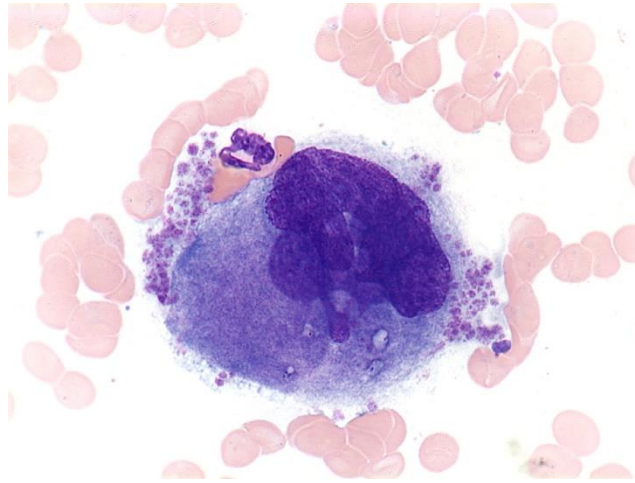


Figura 2.1: Megacariócito formando plaquetas fora da periferia.

2.2 Alterações Qualitativas e Quantitativas

Em condições normais, as plaquetas encontram-se dispostas separadamente uma das outras e não apresentam alterações de tamanho.

- **Anomalias quantitativas** – Esses tipos de anomalias referem-se à contagem de plaquetas. Em casos normais, uma pessoa possui entre 200.000 a 400.000 plaquetas por mm^3 de sangue. Quando é detectado uma diminuição na quantidade de plaquetas, diz-se que ocorre uma plaquetopenia (ou trombocitopenia), podendo levar a sangramentos. No caso contrário, o aumento do quantitativo de plaquetas, diz-se que ocorre plaquetose (ou trombocitose), podendo aumentar o risco de trombose [4]. As principais doenças causadoras de anomalias quantitativas podem ser observadas na Tabela 2.1.
- **Anormalidades quanto à distribuição das plaquetas** – Outro tipo de anomalia que pode acometer as plaquetas é a formação de agregados plaquetários. Essa agregação refere-se às plaquetas aderidas umas às outras, formando grupos de tamanhos variados, que serão visualizados ao microscópio óptico. A presença desse tipo de anomalia impede a contagem real de plaquetas e pode gerar uma falsa plaquetometria. Outro conceito bastante conhecido na área médica refere-

2.2 Alterações Qualitativas e Quantitativas

se ao satelitismo plaquetário, onde as plaquetas aderem *in vitro*¹ aos neutrófilos envolvendo-os como uma coroa.

- **Anormalidades quanto ao tamanho das plaquetas** – Essas células sanguíneas são muito pequenas e discoides, com diâmetros de 1,3 a 3,5 μm e volume médio de 7 a 11 fL. Os tamanhos das plaquetas são avaliados microscopicamente comparando os seus diâmetros dos mesmos com o diâmetro das hemácias. O tamanho das plaquetas varia inversamente com a contagem plaquetária em pacientes saudáveis, ou seja, quanto maior o número de plaquetas, menor será seu tamanho. Durante a formação de plaquetas, algumas anormalidades, como a trombopoiese, faz com que o aumento do tamanho das plaquetas seja suficientemente grande para ser detectado microscopicamente. As plaquetas com o diâmetro um pouco maior do que o diâmetro normal são denominadas de macrotrombócitos enquanto as que tem o diâmetro próximo ao das hemácias são denominadas de plaquetas gigantes (ver Figura 2.2). O aumento no tamanho das plaquetas pode ocorrer em decorrência de doenças como a Leucemia.

Tabela 2.1: Principais causas de Plaquetopenias e Plaquetoses.

Plaquetopenias	Plaquetoses
Leucemias Agudas	Policitemia Vera
Anemia Aplásica	Leucemia Mielóide Crônica
Esplenomegalia	Período pós-hemorragico
Grandes hemorragias	Após esplenectomia
Viroses em geral	Doenças infecciosas crônicas
AIDS	Trombocitemia essencial
Púrpuras imunológicas	
Quimioterapia e radioterapia antitumoral	

Fonte: Renato Failace, Hemograma – Manual de Interpretação, 2003 [8].

¹ *In Vitro*: é uma expressão latina que designa todos os processos biológicos que têm lugar fora dos sistemas vivos, no ambiente controlado e fechado de um laboratório e que são feitos normalmente em recipientes de vidro.

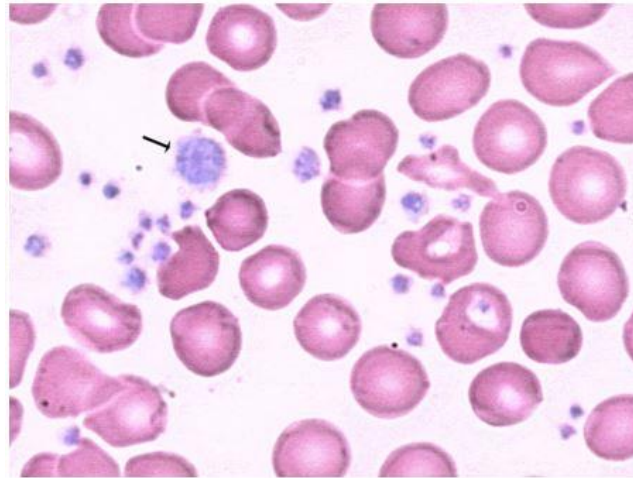


Figura 2.2: Plaqueta Gigante.

2.3 Métodos de Contagem de Plaquetas

A contagem de plaquetas torna-se problemático devido ao seu pequeno tamanho, a sua rápida desintegração e a sua tendência a aderir a superfícies estranhas ao endotélio vascular [2]. O termo plaquetograma (ou trombocitograma) refere-se aos contadores eletrônicos que contam e medem sistematicamente as plaquetas no conjunto do hemograma.

A contagem das plaquetas pode ser realizada de duas formas: o método manual, através da microscopia, por exemplo, onde serão adicionadas informações úteis àquelas obtidas pelos analisadores hematológicos, ou automatizada, utilizando técnicas como tecnologia de impedância, dispersão de luz, ou ambas [6]. Esses métodos não apresentam valores exatos e sim uma estimativa.

2.3.1 Método Manual

Antes da automação, os estudos plaquetários limitavam-se à contagem de plaquetas e às referências morfológicas sobre plaquetas, através da microscopia. As plaquetas eram então contadas por métodos diretos e indiretos, o método Rees-Ecker e o método de Fônio, respectivamente [9].

2.3 Métodos de Contagem de Plaquetas

No método Rees-Ecker, as plaquetas são contadas por microscopia ótica comum na câmara de Neubauer, no sangue diluído. A Câmara de Neubauer é usada em medicina para realizar a contagem de células em um meio líquido. É uma lâmina grossa de vidro composta por duas câmaras independentes, uma superior e uma inferior. Cada câmara possui uma grade no centro onde a contagem celular é realizada [11]. Através desse método, os elementos sanguíneos podem ser contados em conjunto. Esse método apresenta limitações como erros inerentes devido à distribuição irregular de células na câmara de contagem e por ser trabalhoso.

Já pelo método indireto de Fônio, a execução técnica apresenta mais facilidade e é mais econômica, podendo defeitos qualitativos das plaquetas, como formas gigantes, serem identificados. O método utiliza esfregaço² para a contagem de plaquetas em relação ao número de hemácias. As plaquetas existentes em cinco campos, que devem conter cerca de 200 hemácias cada campo, irão ser contadas e uma regra de três em relação ao número global de hemácias irá ser calculada.

2.3.2 Método Automatizado

Atualmente o plaquetograma é composto por contagem de plaquetas automatizadas de vários parâmetros. Entre esses índices, vale ressaltar o Volume Plaquetário Médio (VPM), que mede o tamanho da plaqueta determinando sua função e atividade plaquetária [6 e 9]. Por isso esse parâmetro possui grande importância, principalmente no que diz respeito à detecção de anormalidades clínicas na contagem de plaquetas, como trombocitose e trombocitopenia. O plaquetograma realiza a contagem automática de plaquetas através de amostras sanguíneas do paciente e é composto por dois sensores principais: um detector de luz e um de impedância elétrica. As amostras de sangue que possuírem um número

² Esfregaço: é uma leve camada de matéria orgânica sobre uma lâmina de vidro, para exame microscópico.

2.3 Métodos de Contagem de Plaquetas

de plaquetas abaixo do valor normal serão examinadas ao microscópio para excluir possíveis agregações plaquetárias e satelitismo.

A contagem das plaquetas através do método manual fica custoso devido ao grande volume de exames feitos por um laboratório. Apesar de produzirem resultados mais precisos, as contagens automatizadas ainda assim estão sujeitas a falhas [4]. Os erros mais comuns referem-se a problemas na coleta do sangue e aparelhos mal calibrados. A agitação errada do sangue colhido, uma coleta muito lenta, entre outros fatores, podem fazer com que as plaquetas se agrupem, formando aglomerados plaquetários ou plaquetas gigantes que podem levar a um falso cenário de plaquetopenia, assim como pode ocorrer a presença de fragmentação das hemácias resultando em um diagnóstico equivocado de trombocitose [7].

3 *AdaBoost*

“*Nada é difícil se for dividido em pequenas partes*”

– Henry Ford

Neste Capítulo é abordado o mecanismo de classificação e detecção de objetos utilizado neste trabalho. A Seção 3.1 apresenta os conceitos propostos por Viola-Jones. Em seguida, na Seção 3.2, explane o conceito de Imagem Integral. A explicação dos métodos para a detecção de objetos segue na Seção 3.3 e 3.4 e Seção 3.5 o conceito do *AdaBoost* é introduzido. Por fim, a Seção 3.6 conceitua o método utilizado para o conjunto de treinamento.

3.1 Modelo de Viola-Jones (Método de detecção de faces)

Violas e Jones [12] propõem um modelo onde é introduzido um classificador para detecção de objetos, dando ênfase à detecção de faces. Este processo pode ser aplicado sobre aplicações em tempo real, e ocorre sobre imagens em escala de cinza e estáticas. Três contribuições principais surgiram deste trabalho, tornando-o computacionalmente mais eficiente que outros algoritmos. Elas são:

1. Introdução a uma nova representação de imagens, a Imagem Integral [13];
2. Utilização do algoritmo de aprendizado do tipo *boosting* [14], denominado *AdaBoost*, para classificação a partir da seleção de características de Haar [15];
3. Criação de uma estrutura em árvore, formada de vários classificadores fracos interligados em cascata, chamada de Cascata de Classificadores.

A Figura 3.1 mostra a estrutura proposta por Viola e Jones.

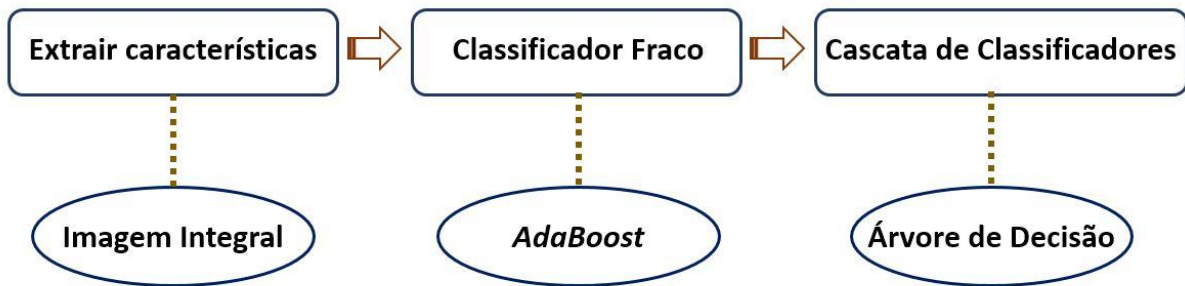


Figura 3.1: Estrutura proposta por Viola e Jones.

3.2 Imagem Integral

Corresponde à representação da imagem original, onde o somatório da intensidade de *pixels* da imagem original estará contido sobre cada ponto desta representação. Ou seja, essa representação corresponde como a localização (x, y) contendo a soma dos valores de todos os *pixels* à esquerda e acima deste ponto, incluindo-o, conforme é mostrado na Equação (3.1). Essa representação do cálculo da imagem integral é a composição de dois cálculos: (1) somatório dos pontos ao longo das linhas (Equação 3.2) e (2) somatório ao longo das colunas (Equação 3.3) [13].

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (3.1)$$

onde $ii(x, y)$ é a integral da imagem e $i(x', y')$ é a imagem original.

$$s(x, y) = s(x, y - 1) + i(x, y), \quad (3.2)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y), \quad (3.3)$$

onde $s(x, y)$ é o somatório acumulado de uma linha, $s(x, -1) = 0$, e $ii(-1, y) = 0$.

A utilização desta representação intermediária aumenta a velocidade de extração de características da imagem permitindo que qualquer retângulo em uma imagem possa ser calculado pela imagem integral, numa única interação. A Figura 3.2 representa um exemplo do cálculo da imagem integral. A posição 1 representa o somatório dos *pixels* do retângulo A. O valor referente à posição 2 é igual ao somatório de $A + B$, para o ponto 3 a soma é igual a $A + C$ e, por fim, a posição 4 é a soma de $A + B + C + D$. Conclui-se, portanto, que a soma dos valores dos *pixels* do retângulo D pode ser calculada por $4 + 1 - (2 + 3)$.

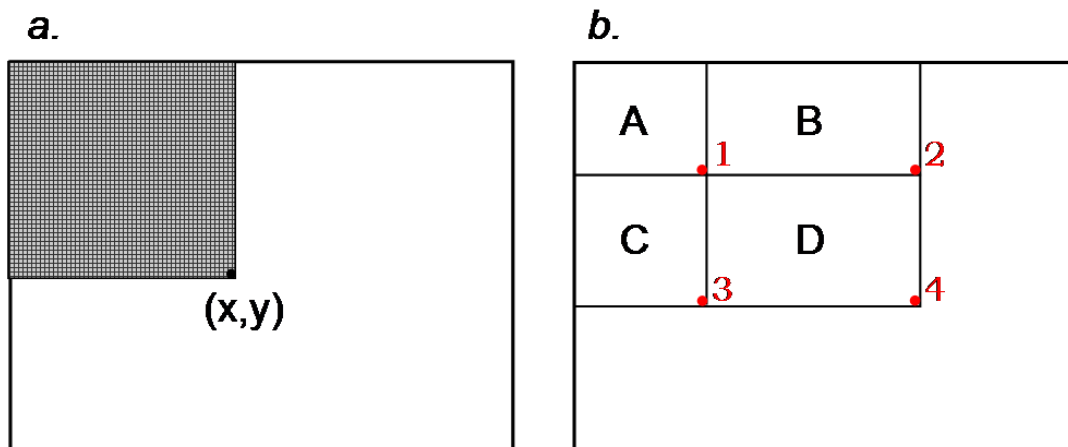


Figura 3.2: Cálculo da Imagem Integral: (a) Após integração, o pixel na posição indicada contém a soma de todos os valores de pixel no retângulo sombreado (b) Exemplo do cálculo da imagem integral.

3.3 Características de Haar

Na proposta de Viola e Jones [12], o sistema usado para detecção de faces é baseado na classificação de características simples. As características são baseadas nas funções de Haar, cujo estudo foi realizado previamente por Papageorgiou e Poggio [16].

As características do tipo Haar (do inglês, *Haar-like features*) são representações retangulares baseadas em *Haar wavelets*, e são compostas por um retângulo com pelo menos uma região clara e uma ou mais regiões escuras. O

3.4 Classificador Fraco

cálculo do valor correspondente à característica é feito através da subtração da média dos valores dos *pixels* da região escura com a região clara. A existência de uma característica será confirmada se essa diferença estiver acima de um limiar, ou seja, existe uma região mais escura e outra mais clara [15]. Na Figura 3.3a, a característica é responsável por detectar bordas verticais enquanto que a característica da Figura 3.3b detecta bordas horizontais. As linhas verticais são detectadas pela característica da Figura 3.3c, enquanto que as linhas diagonais são detectadas pela característica da Figura 3.3d.

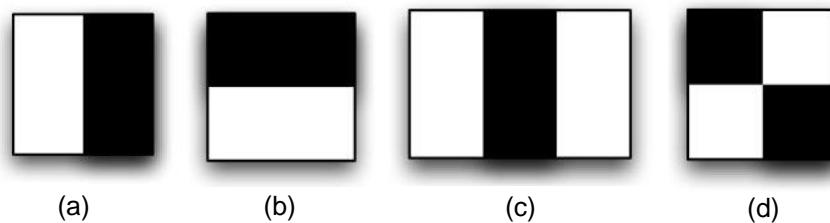


Figura 3.3: Características de Haar.

3.4 Classificador Fraco

O modelo de Viola e Jones é montado utilizando-se classificadores fracos. Um classificador fraco h_j é uma estrutura simples contendo um vetor de características f_j , um limiar θ_j e uma paridade p_j . O objetivo desse classificador é encontrar o melhor limiar que separa os exemplos das imagens estabelecidas como positivas das negativas da base de dados. Para cada característica, deverá ser encontrado um limiar ótimo que possua um número maior de exemplos classificados de maneira correta [17].

A Equação (3.4) demonstra a função do classificador fraco. Caso os exemplos positivos sejam classificados abaixo do limiar θ_j a paridade p_j receberá valor “1” enquanto que se forem classificados acima deste limiar receberá valor “-1”.

3.5 Algoritmo de Aprendizado

$$\hat{h}_j(x) = \begin{cases} 1, & \text{se } p_j f_j(x) < p_j \theta_j \\ 0, & \text{caso contrário} \end{cases}. \quad (3.4)$$

A determinação do limiar não é objetiva e assume-se apenas que favorece a definição do valor que possua menos exemplos classificados erroneamente. Existem várias propostas para o cálculo do limiar, dentre elas merece destaque a de Carbonetto [17] que considera tanto os valores dos exemplos positivos quanto dos negativos em relação à distribuição desses valores associados às características. A Equação abaixo apresenta o cálculo do limiar:

$$\hat{h}_j(x) = \frac{1}{2} \left(\frac{1}{|C_0|} \sum_{x \in C_0} f_j(x) + \frac{1}{|C_1|} \sum_{x \in C_1} f_j(x) \right), \quad (3.5)$$

onde C_0 é o conjunto de exemplos de imagens negativas, C_1 é o conjunto de exemplos das imagens positivas e $f_j(x)$ é o valor da característica. Os algoritmos de aprendizado *boosting* compensam as deficiências da não existência de um limiar ótimo.

3.5 Algoritmo de Aprendizado

Importantes para aperfeiçoar a precisão de uma classificação, os algoritmos de aprendizados são baseados em informações passadas [18]. Eles podem ser divididos em 2 grandes grupos: os supervisionados e os não supervisionados.

Neste trabalho será abordado um algoritmo de aprendizado supervisionado baseado em *Boosting*, o *AdaBoost*.

3.5.1 AdaBoost

Boosting é um método para aprimorar o desempenho de qualquer algoritmo de aprendizado, segundo Freund e Schapire [14]. Por isso, essa técnica não é usada de forma isolada, sendo combinada com outras técnicas, como Árvores de Decisão³ e Redes Neurais⁴, com o objetivo de transformar os classificadores fracos em classificadores fortes [19]. O *Boosting* utiliza a combinação de vários classificadores fracos para obter um classificador forte, ou seja, ele trabalha com os resultados de um mesmo algoritmo de aprendizado combinando os classificadores gerados e se ajustando de acordo com os erros cometidos pelo classificador anterior [14]. O objetivo em utilizar um conjunto de classificadores fracos ao invés de um classificador forte para a classificação é que esse conjunto irá obter uma classificação forte efetiva no final.

Na Equação abaixo, pode-se verificar a criação de um classificador forte através de um algoritmo *Boosting*:

$$\mathcal{H}(x) = a_1 h_1 + a_2 h_2 + \dots + a_n h_n(x), \quad (3.6)$$

onde $H(x)$ é um classificador forte, a_i é o peso associado ao classificador fraco h_i . Esse peso classifica de forma mais precisa dois conjuntos de dados pré-rotuladas, onde o peso maior é mais significativo para a classificação de exemplos de um conjunto determinado.

O *AdaBoost* (do inglês, *Adaptative Boosting*) é um dos mais influentes e conceituados algoritmos de boosting existentes. Sua principal característica é a

³ Árvores de Decisão: representação de uma tabela de decisão sob a forma de uma árvore, porém pode haver outras aplicações.

⁴ Redes Neurais: são modelos computacionais capaz de realizar o aprendizado de máquina bem como o reconhecimento de padrões.

3.5 Algoritmo de Aprendizado

distribuição e modificação de pesos nos conjuntos de exemplos com o decorrer das iterações do algoritmo [19]. O pseudocódigo do *AdaBoost* é descrito no Algoritmo 1. Durante o treinamento do *AdaBoost*, várias iterações são realizadas e as classificações de um classificador fraco são pesadas. Os pesos são inicializados de acordo com uma distribuição uniforme e em seguida são normalizados para, por fim, serem atualizados. Os classificadores fracos vão alterando a distribuição dos pesos e as hipóteses a cada iteração do algoritmo. O fator β_t representa o menor erro do classificador e, multiplicado pelo peso atual, resulta na alteração dos pesos. Se uma classificação é corretamente realizada sobre os exemplos de teste, ela recebe um peso menor enquanto que um peso maior será atribuído àquelas classificações realizadas incorretamente.

Algoritmo 1: Pseudocódigo do *AdaBoost*

Entrada: m = conjunto de exemplos negativos
 l = conjunto de exemplos positivos

- 1 Obtenha N exemplos de imagens $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, onde x corresponde a matriz de *pixels* de uma imagem, e $y = 0$, ou 1 para exemplos negativos e positivos, respectivamente;
- 2 Inicialize os pesos $\omega_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ para $y_i = 0$ e 1 respectivamente;
- 3 **para** $t = 1, \dots, T$ **faça**
- 4 Normalize os pesos: $\omega_{t,i} \leftarrow \frac{\omega_{t,i}}{\sum_{j=1}^N \omega_{t,j}}$, onde ω_t é uma distribuição de probabilidades;
- 5 **para cada característica** j **faça**
- 6 Treine um classificador h_j restrito ao uso de uma única característica;
- 7 Avalie o erro de acordo com: $\omega_j, \epsilon_j = \sum_i \omega_i |h_j(x_i) - y_i|$;
- 8 **fim**
- 9 Escolha o classificador h_t com o menor erro ϵ_t ;
- 10 Atualize os pesos: $\omega_{t+1,i} = \omega_{t,i} \beta_t^{(1-e_i)}$, onde $e_i = 0$, se o exemplo x_i for classificado corretamente, $e_i = 1$ caso contrário, e $\beta_t = \frac{e_t}{1-e_t}$;
- 11 **fim**
- 12 O classificador forte é definido por: $H(x) = \begin{cases} 1, & \text{se } \sum_{t=1}^T a_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T a_t \\ 0, & \text{caso contrário} \end{cases}$, onde $a_t = \log \frac{1}{\beta_t}$.

3.6 Cascata de Classificadores

Em busca de uma solução para solucionar a quantidade de tempo demandada para realizar o treinamento, Viola e Jones [12] propuseram um novo método, onde um encadeamento de classificadores do mais genérico ao mais específico formariam uma árvore de decisão ou cascata de classificadores.

A utilização de uma cascata de classificadores com N estágios tem como objetivo otimizar o reconhecimento de objetos. A combinação de funções de classificação oriundas do uso do algoritmo do *AdaBoost* cria estágios dentro da cascata. A exemplificação de uma cascata de classificadores pode ser observada na Figura 3.4.

A principal ideia da cascata é fazer com que seus estágios iniciais descartem um grande número de regiões que contém o objeto desejado enquanto os estágios mais avançados sejam cada vez mais precisos, evitando assim um falso positivo na região sendo analisada.

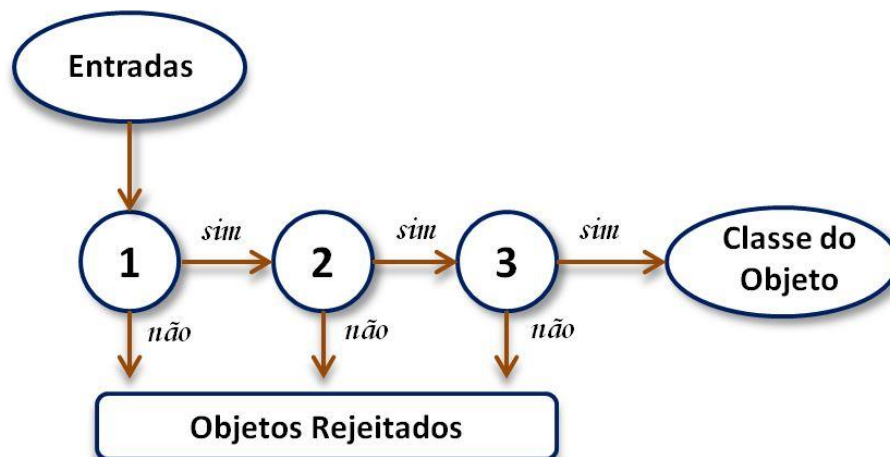


Figura 3.4: Cascata de Classificadores.

3.6 Cascata de Classificadores

3.6.1 Treinamento da Cascata de Classificadores

O processo para o treinamento da cascata é guiado visando atingir um conjunto de metas tanto de detecção quanto de desempenho. A quantidade de estágios na cascata deve ser o suficiente para garantir detecção de objetos a uma alta taxa, uma minimização da ocorrência de falsos positivos e diminuição do tempo de processamento durante o teste de uma região da imagem.

A taxa de falsos positivos de uma cascata de classificadores treinada pode ser verificada pela Equação abaixo:

$$\mathcal{F} = \prod_{i=1}^{\mathcal{K}} f_i, \quad (3.7)$$

onde F é a taxa de falsos positivos, K é o número de classificadores e f_i é a taxa de falso positivo do i – ésimo classificador. Já a taxa de detecção D pode ser verificada na Equação abaixo:

$$\mathcal{D} = \prod_{i=1}^{\mathcal{K}} d_i. \quad (3.8)$$

onde K é o número de classificadores e d_i é a taxa de detecção do i – ésimo classificador.

O sistema receberá como entrada dois conjuntos, um contendo exemplos de imagens positivas e outro contendo as negativas. As taxas mínimas aceitáveis para f_i e d_i serão passadas para o sistema. Cada estágio da cascata será então treinado pelo *AdaBoost* até atingir as taxas de detecção e falsos positivos previstos. Um novo estágio será adicionado na cascata à medida que a taxa de falso positivo não for

3.6 Cascata de Classificadores

atendida. Os estágios subsequentes serão treinados com imagens negativas obtidas de falsas detecções realizadas anteriormente.

O pseudocódigo do treinamento de uma cascata de classificadores está descrito mais detalhadamente no Algoritmo 2.

Algoritmo 2: Pseudocódigo do Treinamento de uma Cascata de Classificadores

Entrada: m = conjunto de exemplos negativos

l = conjunto de exemplos positivos

```

1 Seleccione os valores para:
   $f$  : a máxima taxa de aceitação de falso positivo por estágio da cascata
   $d$  : a taxa mínima de detecção, por estágio, aceita;
2  $F_0 = 1, 0; D_0 = 1, 0; i = 0;$ 
3 enquanto  $F_i > F_{alvo}$  faça
4    $i = i + 1;$ 
5    $n_i = 0; F_i = F_{i-1};$ 
6   enquanto  $F_i > f \cdot F_{i-1}$  faça
7     Use  $l$  e  $m$  para treinar um classificador com  $n_i$  características usando o AdaBoost;
8     Avalie a cascata de classificadores no conjunto de validação para determinar  $F_i$  e  $D_i$ ;
9     Diminua o limiar para o  $i$ -ésimo classificador até que a cascata atual tenha uma taxa
     de detecção de pelo menos  $d(D_{i-1})$ , o que também afeta  $F_i$ ;
10  fim
11   $m \leftarrow 0;$ 
12  se  $F_i > F_{alvo}$  então
13    Avalie o detector da cascata atual no conjunto de imagens negativas e coloque
    qualquer detecção falsa no conjunto  $m$ ;
14  fim
15 fim
```

Com o objetivo de otimizar o treinamento dos classificadores do *AdaBoost*, técnicas de Otimização Multiobjetivas serão aplicadas para encontrar soluções que resolvam os dois objetivos conflitantes do problema deste trabalho, a taxa de acertos e a taxa de falsos positivos. O próximo capítulo introduzirá conceitos de Problemas Multiobjetivos e apresentará os dois algoritmos que serão desenvolvidos neste trabalho.

4 Otimização Multi-Objetivos

*“Não temos nas nossas mãos as soluções para todos os problemas do mundo,
mas diante de todos os problemas do mundo temos as nossas mãos”*

– Friedrich Schiller

Neste Capítulo são abordados os aspectos teóricos referentes à otimização multiobjetivo. Na Seção 4.1 é definido um problema multiobjetivo. Em seguida, na Seção 4.2, é conceituado o Algoritmo Evolucionário utilizado neste trabalho, o SPEA 2. O algoritmo baseado em Enxames de Partículas utilizado neste trabalho, o DFMOPSO, é apresentado na Seção 4.3. Por fim, é apresentado na Seção 4.4 uma métrica que serve para avaliar a qualidade das soluções encontradas por técnicas de otimização multiobjetivo.

4.1 Conceitos de Otimização Multiobjetivo

Muitos problemas do mundo real possuem a necessidade de otimização de múltiplos objetivos simultaneamente, ou seja, um problema multiobjetivo (MOP, do inglês *Multi-Objective Problems*) consiste em um problema onde existem dois ou mais objetivos conflitantes a serem resolvidos. Por exemplo, ao pesquisar preços na compra de um computador novo, a aquisição ótima para o consumidor é aquela que fornece o custo mínimo enquanto maximiza o desempenho do equipamento. Nota-se, então, que esses objetivos são conflitantes entre si, onde a melhoria de um fator implica na deterioração do outro. O conceito de otimização num MOP baseia-se no estudo introduzido por Edgeworth [20] e que depois é generalizada por Pareto [21], denominado de Pareto-ótimo. Em um problema multiobjetivo, a solução é formada por um conjunto de soluções que será denominado conjunto Pareto-ótimo se, para cada solução do conjunto, não existir nenhuma outra solução que apresente resultado melhor que as demais soluções em pelo menos um objetivo do problema.

4.1 Conceitos de Otimização Multiobjetivo

Na otimização de um MOP, trabalha-se com dois espaços: o espaço de variáveis e o espaço de objetivos. A busca pelas soluções do problema é feita no espaço de variáveis enquanto que o espaço de objetivos é formado pelas funções-objetivo do problema. Um exemplo de mapeamento feito pelas funções-objetivo entre o espaço de variáveis e o de objetivos é mostrada na Figura 4.1.

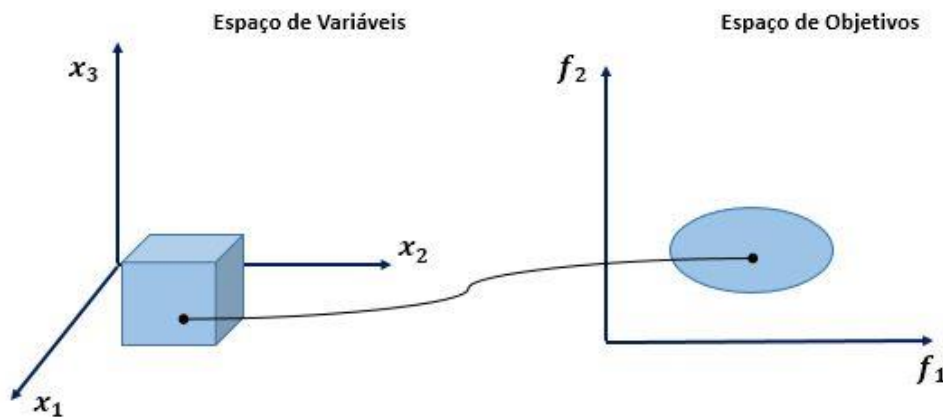


Figura 4.1: Mapeamento de uma solução no espaço de variáveis para espaço de objetivos.

Uma otimização multiobjetiva pode ser definida como a minimização ou maximização do vetor a seguir:

$$F(\vec{x}) = \{f_1(\vec{x}), \dots, f_k(\vec{x})\} \quad (4.1)$$

onde estará sujeita às restrições:

$$\begin{aligned} g_i(\vec{x}) &\leq 0, \quad i = 1, \dots, m; \\ h_j(\vec{x}) &= 0, \quad j = 1, \dots, p; \\ \vec{x} &\in \mathcal{D}. \end{aligned} \quad (4.2)$$

Uma solução do problema minimiza (ou maximiza) os componentes do vetor $F(\vec{x})$. A variável \vec{x} representa um vetor n-dimensional, $\vec{x} = \{x_1, \dots, x_k\}$, de algum

4.1 Conceitos de Otimização Multiobjetivo

universo D . Durante a minimização (ou maximização) de $F(\vec{x})$, as restrições precisam ser satisfeitas e D contém todos os possíveis \vec{x} que podem ser usados para satisfazer uma avaliação de $F(\vec{x})$.

As soluções ótimas, soluções equivalentes consideradas não-dominadas em relação às múltiplas funções objetivo, são chamadas de não-dominadas e são soluções pertencentes à Frente do Pareto. Cada solução apresenta melhor resultado, em pelo menos um objetivo, que as demais soluções. Um vetor $\vec{u} = (u_1, \dots, u_k)$ domina um vetor $\vec{v} = (v_1, \dots, v_k)$ se:

$$\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i, \quad (4.3)$$

A Figura 4.2 mostra o conceito de dominância.

Existem três metas importantes em uma otimização MOP (do inglês, *Multi-Objective Problems*, Problemas Multiobjetivos):

- Obter um conjunto de soluções que esteja o mais próximo possível da Frente de Pareto;
- Obter um conjunto de soluções que possua um espalhamento ao longo da Frente de Pareto, possuindo uma maior diversidade;
- Alcançar uma maior eficiência computacional na obtenção do conjunto de soluções.

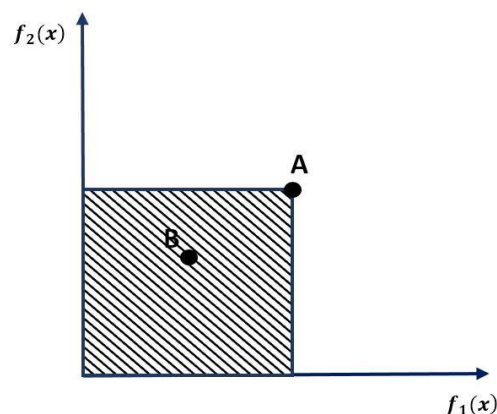


Figura 4.2: Conceito de dominância: o ponto B e todos os pontos na região hachurada dominam a solução representada pelo ponto A.

4.2 Algoritmos Evolucionários

São algoritmos baseados em conceitos da evolução biológica, apresentando facilidade e flexibilidade de modelagem. A possibilidade de não precisar de informações adicionais e de poder trabalhar com várias soluções simultaneamente faz do EA (do inglês, *Evolutionary Algorithm*) uma ótima técnica a ser empregada no problema multiobjetivo [22].

Os algoritmos evolucionários possuem dois tipos de classificação: o elitista e o não-elitista [23]. No tipo elitista, há o uso de uma população externa para manter os indivíduos não-dominados, ao contrário do que ocorre no tipo não-elitista, onde não existe essa população externa. Neste trabalho, é abordado um algoritmo evolucionário elitista, o SPEA 2.

4.2.1 SPEA 2

O SPEA 2 (do inglês, *Strength Pareto Evolutionary Algorithm 2*) foi proposto por Zitzler [24] e, como já foi mencionado, trata-se de um algoritmo elitista.

Esse método utiliza duas populações, uma interna e um arquivo externo, P e \bar{P} respectivamente. Em \bar{P} , serão armazenadas apenas as soluções não dominadas encontradas pelo algoritmo até o momento. O tamanho N da população interna e o tamanho \bar{N} do arquivo externo serão passados como parâmetros iniciais, assim como o número máximo de iterações T . A cada iteração $t = 0, 1, \dots, T$, as populações P e \bar{P} serão denotadas como P_t e \bar{P}_t , respectivamente [24]. Para melhor entendimento, o Algoritmo 3 descreve o fluxo básico do SPEA 2.

O SPEA 2 começa criando uma população interna inicial P_0 e um arquivo externo inicial \bar{P}_0 , inicializando essas duas populações como vazias. A cada iteração do algoritmo, a função da força do Pareto (ver Subseção 4.2.1.1) será calculada para cada solução $i \in P_t \cup \bar{P}_t$. O objetivo é minimizar o valor da função *fitness*, pois quanto menor melhor será a adaptação do indivíduo. Após essa etapa, serão

4.2 Algoritmos Evolucionários

verificadas quais soluções foram dominadas e quais pertencem ao grupo das não dominadas, sendo essas últimas copiadas para o novo arquivo externo \overline{P}_{t+1} .

O próximo passo será verificar o tamanho do arquivo externo. Se o conjunto de soluções não dominadas encontrado tiver o tamanho exato do arquivo externo então a seleção está encerrada. Caso contrário, pode-se ter duas situações: o tamanho do arquivo externo é menor que o número de soluções não dominadas, neste caso as soluções de $P_t \cup \overline{P}_t$ são ordenadas de acordo com a função *fitness* e o arquivo externo será preenchido com as soluções dominadas com maior valor da função *fitness*, ou o tamanho do arquivo externo excede o tamanho \overline{N} , neste caso será usado um algoritmo de corte (ver Subseção 4.2.1.2).

Algoritmo 3: Pseudocódigo do SPEA2

Entrada: N: Tamanho da população interna
 \overline{N} : Tamanho do Arquivo Externo
T: Número máximo de gerações;
Saída: A: Soluções não dominadas;

- 1 **Inicialização:**
- 2 | Gere uma população inicial P_0 e crie um Arquivo Externo vazio $\overline{P}_0 = \emptyset$;
- 3 | Atribua $t = 0$;
- 4 **fim**
- 5 **para** $t = 0, \dots, T$ **faça**
- 6 | **se** $t \geq T$ **então**
- 7 | | $A = \overline{P}_{t+1}$;
- 8 | | Pare o programa;
- 9 | **senão**
- 10 | | Calcule o *fitness* de P_t e \overline{P}_t ;
- 11 | | Copie todos os indivíduos não dominados de P_t e \overline{P}_t para \overline{P}_{t+1} ;
- 12 | | **se** $|\overline{P}_{t+1}| = \overline{N}$ **então**
- 13 | | | Teste condição de parada;
- 14 | | **senão se** $|\overline{P}_{t+1}| > \overline{N}$ **então**
- 15 | | | Aplique o truncamento a \overline{P}_{t+1} ;
- 16 | | **senão**
- 17 | | | Complete o Arquivo com elementos dominados de P_t e \overline{P}_t ;
- 18 | | **fim**
- 19 | | Execute seleção por torneio direto (binário) em \overline{P}_{t+1} . Aplique *crossover* em \overline{P}_{t+1} até completar o *mating pool*;
- 20 | | Aplique o operador de mutação ao *mating pool* e atribua a população resultante em P_{t+1} ;
- 21 | **fim**
- 22 **fim**

4.2 Algoritmos Evolucionários

4.2.1.1 Função da Força do Pareto

O cálculo da função da força do pareto é realizado usando conceitos de dominância e densidade, buscando minimização desse valor [25]. A força do indivíduo $S(i)$ corresponde ao número de soluções que ele consegue dominar:

$$S(i) = \left| \left\{ j \mid j \in P_t + \bar{P}_t \wedge i \succ j \right\} \right|, \quad (4.4)$$

onde o símbolo \succ representa a relação de dominância de pareto e o número de soluções em $P_t + \bar{P}_t$ dominadas pela solução i é representado pelo valor i , ou seja, $S(i)$ terá valor igual a zero para cada solução que não domina nenhuma outra. A soma das forças de todos os indivíduos que o dominam corresponde à força bruta do indivíduo, definido como $R(i)$, e tem seu cálculo realizado como demonstrado a seguir:

$$R(i) = \sum_{j \in P_t + \bar{P}_t, j \succ i} S(j). \quad (4.5)$$

As soluções não dominadas têm um valor de força bruta igual a zero, enquanto que as soluções dominadas têm esse valor alto.

Este mecanismo ordena as soluções por dominância, mas pode falhar se o número de soluções não dominadas for muito alto, tendo o valor $S(i)$ próximo a zero para todas as soluções. Por isso, torna-se necessário um mecanismo que privilegie soluções dentre as não dominadas, denominado Fator de Densidade. A densidade de um indivíduo corresponde à uma função decrescente em relação ao k -ésimo vizinho mais próximo a ele. O cálculo do fator de densidade $D(i)$ é demonstrado na Equação (4.6), onde as distâncias entre cada indivíduo i e todos os indivíduos j da população interna P_t e do arquivo externo \bar{P}_t são calculadas e serão armazenadas em uma lista. O termo $dist_{ij}(k)$ corresponde ao k -ésimo elemento da lista, ordenada em ordem crescente, criada. Os autores do SPEA 2 (*Strength Pareto Evolutionary*

4.2 Algoritmos Evolucionários

Algorithm 2) sugerem que o componente k receba um valor $k = \sqrt{N + \bar{N}}$, onde N e \bar{N} correspondem ao tamanho da população interna e ao tamanho do arquivo externo, respectivamente. A Equação abaixo exemplifica o cálculo do fator de densidade:

$$\mathcal{D}(i) = \frac{1}{\text{dist}_{ij}(\bar{k}) + 2}. \quad (4.6)$$

Logo, a função de força final $F(i)$ é dada por:

$$F(i) = \mathcal{R}(i) + \mathcal{D}(i). \quad (4.7)$$

4.2.1.2 Algoritmo de Corte de Excedentes do Arquivo Externo

Como já foi mencionado anteriormente (ver Subseção 4.2.1), o algoritmo de corte será utilizado quando o tamanho do arquivo externo exceder o tamanho \bar{N} , com o objetivo de restringir o tamanho de $\overline{P_{t+1}}$ para \bar{N} soluções. A cada iteração, a solução é removida de modo que a sua distância para o seu vizinho mais próximo seja a menor dentre as distâncias existentes. Se houver empate, pega-se a segunda menor distância para a realização do cálculo, e assim sucessivamente. A Equação (4.8) mostra como a solução i será escolhida para ser removida:

$$\begin{aligned} i \leq_d j \Leftrightarrow \forall 0 < \bar{k} < \left\lceil \overline{P_{t+1}} \right\rceil : \sigma_i^{\bar{k}} = \sigma_j^{\bar{k}} \\ \vee \\ \exists 0 < \bar{k} < \left\lceil \overline{P_{t+1}} \right\rceil : \left[\left(\forall 0 < l < \bar{k} : \sigma_i^l = \sigma_j^l \right) \wedge \sigma_i^{\bar{k}} = \sigma_j^{\bar{k}} \right], \end{aligned} \quad (4.8)$$

onde $\sigma_i^{\bar{k}}$ é a distância entre i e o \bar{k} -ésimo vizinho mais próximo em $\overline{P_{t+1}}$. σ_i^l é a distância entre i e o l -ésimo vizinho mais próximo. A solução i é escolhida para ser removida se $i \leq_d j$ para todo $j \in \overline{P_{t+1}}$ satisfazendo as condições apresentadas.

4.3 Algoritmos baseados em Enxame de Partículas para Otimização Multiobjetiva

Proposto por Kennedy e Eberthart [26], a otimização por Enxame de Partículas (do inglês PSO, *Particle Swarm Optimization*) é um ramo da Inteligência Computacional que utiliza o conceito de partículas para otimizar problemas do mundo real. O PSO baseia-se no comportamento coletivo de sistemas autônomos, flexíveis, sendo esses sistemas formados por uma população de agentes simples que possuem a capacidade de modificar o ambiente ao redor de maneira local. Um dos mais conhecidos desses sistemas diz respeito ao comportamento do voo dos pássaros, onde estudos foram realizados para observar as regras que guiavam o bando, assim como a sincronização na forma em que voavam.

Para serem aptos a resolver o MOP (do inglês, *Multi-Objective Problems*), os algoritmos baseados em enxames tiveram que ser adaptados, onde o objetivo é maximizar o número de soluções pertencentes à Frente de Pareto, aproximando a Frente de Pareto encontrada da verdadeira e, com isso, aumentar o espalhamento das soluções encontradas.

Em busca da solução desse problema, o algoritmo de Otimização Multiobjetiva por Enxame de Partículas (MOPSO, do inglês *Multi-Objective Particle Swarm Optimization*) foi proposto e é apresentado adiante (ver Subseção 4.3.1).

4.3.1 *Multi-Objective Particle Swarm Optimization* (MOPSO)

Proposto por Coello e colaboradores [27], o MOPSO é baseado em arquivo externo onde, em cada iteração, cada partícula deposita seu conhecimento. O arquivo externo (AE) servirá como um repositório, guardando as soluções não-dominadas encontradas pelas partículas durante o processo de otimização, e será dividido em hipercubos. Cada solução do AE é associada a um desses hipercubos,

4.3 Algoritmos baseados em Enxame de Partículas

onde cada um receberá um valor *fitness*, representando sua qualidade. Quanto mais soluções estiverem presentes em um hipercubo, maior será sua qualidade [28]. O líder de cada partícula do enxame será escolhido através de uma seleção por roleta, ou seja, um hipercubo será escolhido levando-se em conta seu valor de *fitness*. Após a seleção do hipercubo, uma solução será escolhida aleatoriamente.

O método da Roleta [29] ocasiona diversidade entre as soluções encontradas uma vez que força as partículas do enxame a explorar as regiões menos densas. Apesar da diversidade, o algoritmo talvez não seja capaz de escapar de mínimos locais, sendo necessário a utilização de um operador de mutação para manter essa diversidade no processo de busca.

4.3.2 MOPSO-DFR

O algoritmo baseado em Otimização Multiobjetiva por Enxame de Partículas usando Fator de Diversidade e Método da Roleta (do inglês, *Multi-Objective Particle Swarm Optimization using Diversity Factor and Roulette Wheel*), proposto por Cunha [30], tem o objetivo de encontrar uma Frente de Pareto mais espalhada e espaçada. Para tal, uma abordagem chamada Fator de Diversidade (DF, do inglês *Diversity Factor*) será utilizada com o intuito de analisar a distribuição das partículas ao longo da Frente de Pareto. A Figura 4.3 mostra uma Frente de Pareto e explicita como o cálculo do DF será realizado.

Para a realização do cálculo do DF, calcula-se a distância média de uma partícula i para as demais partículas da Frente de Pareto, como observado pela Equação (4.9), onde N representa a quantidade de partículas da Frente de Pareto e M a quantidade de objetivos do problema [31].

$$d_i = \frac{1}{N-1} \sum_{j=1, j \neq i}^N \sqrt{\sum_{k=1}^M (x_i^k - x_j^k)^2} \quad (4.9)$$

4.3 Algoritmos baseados em Enxame de Partículas

Depois de realizado o cálculo da distância média, o DF de cada partícula pertencente à Frente de Pareto será calculado através da Equação (4.10):

$$DF_i = \frac{d_i - d_{min}}{d_{max} - d_{min}}, \quad (4.10)$$

onde DF_i é o DF da partícula i , d_i é a distância média da partícula i para as demais, d_{max} e d_{min} são as distâncias médias máxima e mínima, respectivamente, de uma partícula para as demais.

As partículas que possuírem valores mais altos de DF terão uma maior probabilidade de serem escolhidas como líder e elas se encontram mais próximas às extremidades. Essa dinâmica talvez promova o aumento da diversidade do enxame, por se tratar de uma análise de toda a Frente de Pareto.

O operador de Mutação [25] combinado com o cálculo do DF mantêm a diversidade das soluções não-dominadas do Arquivo Externo. A seleção do líder cognitivo e social será abordada nas próximas seções.

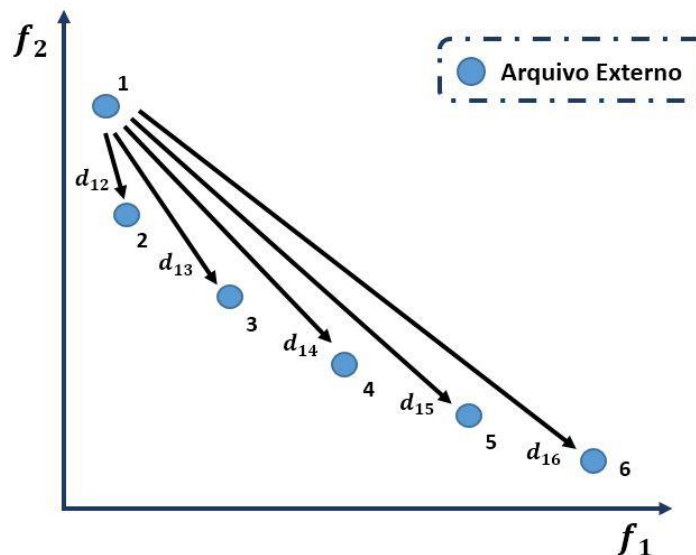


Figura 4.3: Demonstração de como será realizado o cálculo do DF para um problema com dois objetivos.

4.3 Algoritmos baseados em Enxame de Partículas

4.3.2.1 Seleção do Líder Cognitivo

Em um problema de otimização por enxame de partículas, o líder cognitivo (\vec{P}_{best}) representa o quanto a partícula foi capaz de aprender no processo de otimização até o momento e a seleção desse líder é muito importante para a convergência e eficiência do algoritmo.

No processo de seleção do líder cognitivo no algoritmo do MOPSO-DFR (do inglês, *Multi-Objective Particle Swarm Optimization using Diversity Factor and Roulette Wheel*), caso a solução atual dominar o \vec{P}_{best} , este será atualizado, caso contrário, a escolha será feita utilizando o Arquivo Externo (AE). É necessário o uso do AE para identificar as soluções presentes do AE que estão mais próximas da partícula atual e do \vec{P}_{best} . O próximo passo será verificar as soluções levando-se em consideração seus valores de DF (*Diversity Factor* – Fator de Diversidade). O \vec{P}_{best} tem seu valor atualizado se o valor de DF da solução mais próxima da posição atual tiver um valor superior, caso contrário seu valor não é atualizado.

A Figura 4.4 explicita como é feito a seleção do líder cognitivo no algoritmo do MOPSO-DFR.

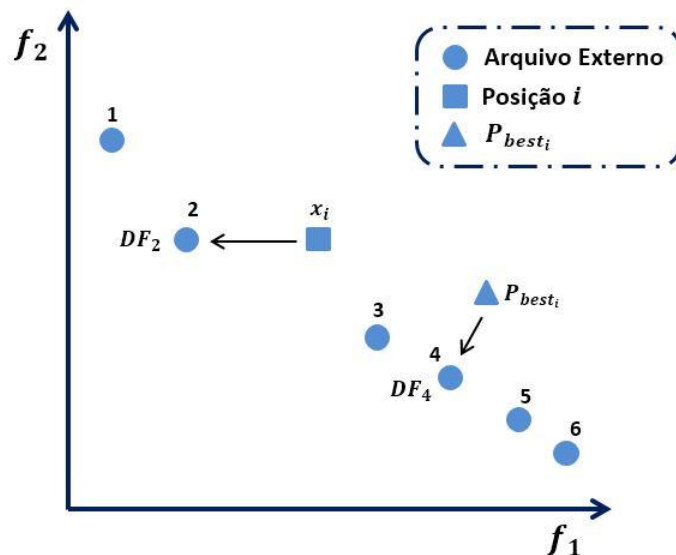


Figura 4.4: Demonstração da Seleção do Líder Cognitivo.

4.3 Algoritmos baseados em Enxame de Partículas

Na Figura 4.4, podemos ver as partículas da Frente de Pareto mais próximas da partícula i e o \vec{P}_{best_i} . A posição da partícula i , representada por x_i , e o \vec{P}_{best_i} são não-dominadas, por isso será feita uma análise para saber se o \vec{P}_{best_i} será atualizado ou não. A partícula 2 da Frente de Pareto é a mais próxima de x_i , possuindo um valor DF_2 , enquanto a partícula 4 é a mais próxima de \vec{P}_{best_i} , possuindo um valor DF_4 . O valor de \vec{P}_{best_i} será atualizado para x_i pois o valor de DF_4 é maior que o valor de DF_2 .

4.3.2.2 Seleção do Líder Social

O líder social, em um problema de otimização por enxame de partículas, corresponde ao componente social da partícula, ou seja, é a melhor posição encontrada até o momento pela vizinhança em que a partícula pertence. A seleção desse líder social é muito importante para a distribuição das soluções ao longo da Frente de Pareto assim como para a convergência do algoritmo.

No Arquivo Externo (AE), encontram-se todas as soluções não-dominadas do problema, como já foi mencionado anteriormente (ver subseção 4.3.1), e estas são as possíveis candidatas a serem escolhidas com líder social da população interna (*i.e.* enxame). O MOPSO-DFR realiza essa seleção através da ordenação, à cada iteração, dessas soluções presentes no AE. Essa ordenação é feita usando como referência o valor de DF (*Diversity Factor* – Fator de Diversidade) de cada partícula. O líder social será então selecionado através da utilização de uma roleta, onde as soluções do AE que tiverem maior valor de DF possuem maior probabilidade de serem escolhidas.

A Figura 4.5 demonstra mais exemplificado como é realizada a dinâmica da seleção do líder social no MOPSO-DFR, exibindo diversas partículas pertencentes à população interna e ao Arquivo Externo. As setas em direção à Frente do Pareto demonstram os líderes encontrados por cada partícula. A seleção através da utilização da roleta permite que partículas que possuem um valor de DF mais baixo possam ser escolhidas como líderes.

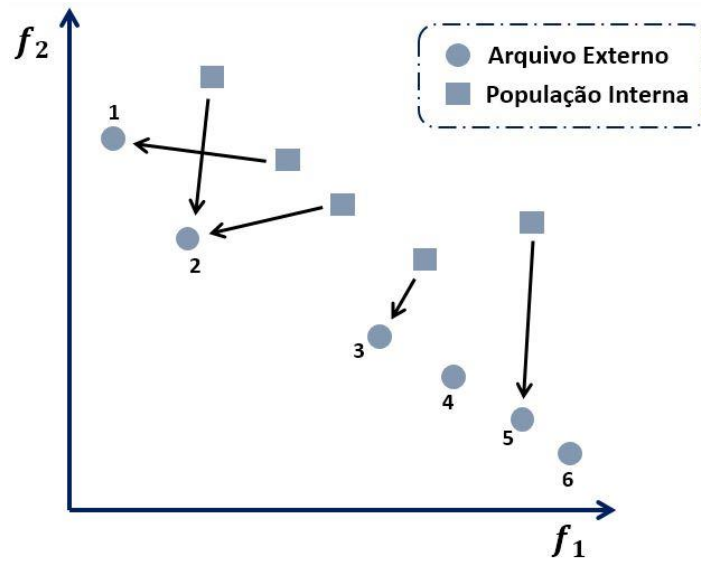


Figura 4.5: Demonstração da Seleção do Líder Social.

Essa dinâmica da seleção do líder social é um fator que influencia diretamente a movimentação do enxame para regiões menos populadas, uma vez que o líder social está presente em um dos termos da equação da velocidade, como será visto na próxima Subseção.

4.3.3 DFMOPSO

Do inglês, *Diversity Factor Multi-Objective Particle Swarm Optimization*, o DFMOPSO, proposto por Silva e Bastos-Filho [32], é uma versão melhorada do MOPSO-DFR (do inglês, *Multi-Objective Particle Swarm Optimization using Diversity Factor and Roulette Wheel*), utilizando Torneio Binário para a seleção dos líderes sociais.

O DFMOPSO também faz uso de um Arquivo Externo (AE) para armazenar as soluções não-dominadas, mas não usa o Método da Roleta para selecionar os líderes sociais. Em vez disso, o algoritmo faz uso de Torneio Binário entre duas soluções não-dominadas selecionadas do AE levando em consideração o DF (Fator

4.3 Algoritmos baseados em Enxame de Partículas

de Diversidade) de cada solução. O líder cognitivo só será atualizado se ele dominar a soluções atual.

À medida que as soluções forem sendo inseridas no Arquivo Externo, análises serão realizadas para evitar que a quantidade de soluções não-dominadas exceda o tamanho máximo do Arquivo Externo. Essas análises correspondem ao valor do *crowding distance* (CD) de uma solução, descartando as soluções com piores valores de CD do Arquivo Externo.

O valor de *crowding distance* [29] de uma solução corresponde em uma estimativa da densidade das soluções em relação às demais, ou seja, consiste em estimar o tamanho do maior cuboide em torno do ponto i sem que qualquer outro ponto seja incluído, como pode ser observado na Figura 4.6.

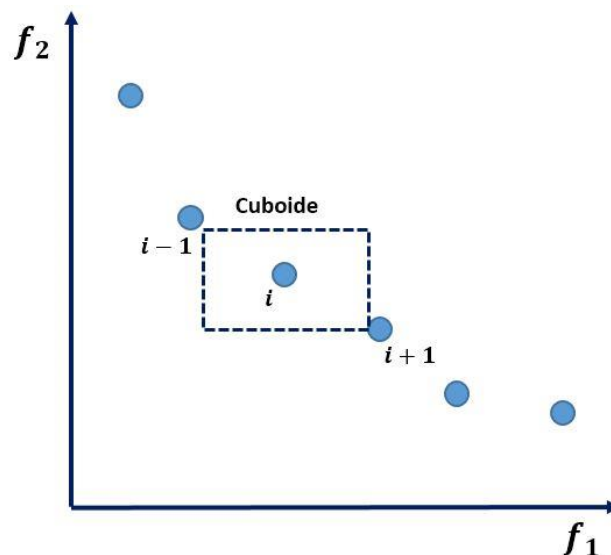


Figura 4.6: Cálculo do Crowding Distance.

4.3.3.1 Equação da Posição e Velocidade

Em um problema de otimização multiobjetivo, cada partícula é composta por 4 vetores: \vec{x}_i , sua posição atual no espaço de busca; \vec{p}_i , sua melhor posição encontrada até então (fator cognitivo); \vec{n}_i , melhor posição encontrada pelos seus vizinhos (fator social) e \vec{v}_i , sua velocidade.

4.3 Algoritmos baseados em Enxame de Partículas

A equação da velocidade da partícula em um algoritmo DFMOPSO [35] é mostrada na Equação (4.11):

$$\vec{v}_i(t+1) = X \left\{ \vec{v}_i(t) + c_1 r_1 [\vec{p}_i(t) - \vec{x}_i(t)] + c_2 r_2 [\vec{n}_{i_{DF}}(t) - \vec{x}_i(t)] \right\} \quad (4.11)$$

onde $\vec{n}_{best_{DF}}$ é o líder social escolhido baseado no DF , c_1 e c_2 correspondem ao coeficiente de aceleração associado ao líder cognitivo e social, respectivamente, e r_1 e r_2 são número aleatórios gerados no intervalo de $[0,1]$. X está relacionado ao Fator de Constrição, que será visto mais adiante.

A posição da partícula é calculada utilizando a Equação a seguir:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \quad (4.12)$$

Em determinados instantes de tempo a partícula adquire uma velocidade muito alta, oscilando entre os extremos do espaço de busca. Esse efeito é conhecido como um estado de “explosão” de velocidades. O Fator de Constrição, proposto por Clerc em 1999 [33] e redefinido em 2002 [34], será incorporado na equação da velocidade (Equação 4.11) fazendo com que não ocorra esse estado de “explosão” da velocidade, uma vez que o ajuste da velocidade das partículas de um enxame ocorrerá de forma estável. As Equações (4.13) e (4.14) mostram o cálculo do Fator de Constrição χ .

$$\varphi = c_1 + c_2, \quad (4.13)$$

$$\chi = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}. \quad (4.14)$$

4.4 Métrica para avaliação da Frente de Pareto

O pseudocódigo do DFMOPSO é descrito no Algoritmo 4.

Algoritmo 4: Pseudocódigo do DFMOPSO

```
1 Inicialize o enxame;  
2 Selecione as soluções para o Arquivo Externo;  
3 enquanto critério de parada não for alcançado faça  
4   para cada partícula faça  
5     Aplique o operador de turbulência utilizado no MOPSO;  
6     Selecione líderes sociais usando Torneio Binário com DF;  
7     Atualize a velocidade usando a Equação (4.11);  
8     Aplique Fator de Construção (Equação 4.14);  
9     Atualize a posição usando a Equação (4.12);  
10    Avalie a partícula;  
11    Atualize o líder cognitivo, se necessário;  
12  fim  
13  Atualize e ajuste o Arquivo Externo usando CD;  
14  Avalie o DF das soluções dentro do Arquivo Externo;  
15 fim  
16 Retorne o Arquivo Externo.
```

4.4 Métrica para avaliação da Frente de Pareto

Com o objetivo de avaliar a qualidade das soluções, algumas métricas são utilizadas para quantificar e qualificar a Frente de Pareto, tendo como foco o grau de convergência e a diversidade das soluções. O grau de convergência está relacionado ao quanto as soluções encontradas se aproximam da Frente de Pareto ideal. Já a diversidade refere-se ao quanto o conjunto de soluções encontrado está bem distribuído ao longo da Frente de Pareto.

4.4.1 Hypervolume (HV)

Essa métrica é definida pela área no espaço de objetivos coberta pela Frente de Pareto, ou seja, corresponde à área acima da curva [25].

Considerando-se dois objetivos, cada ponto $(f_1(\bar{x}), f_2(\bar{x}))$ pertencente à Frente de Pareto, juntamente com o ponto R, delimita um retângulo no espaço de objetivos. O *Hypervolume* corresponde à área formada pela união de todos os retângulos formados por cada ponto da Frente de Pareto, como pode ser visto na Figura 4.7.

Para melhor entendimento da métrica, a Equação (4.15) generaliza o conceito demonstrado para problemas com n-objetivos:

$$HV = \left\{ \bigcup_i a_i \mid x_i \in P_a^* \right\}, \quad (4.15)$$

onde x_i é um vetor de soluções não-dominadas presentes na Frente de Pareto P_a^* e a_i corresponde à métrica *Hypervolume* determinada pelos componentes de x_i e o ponto R.

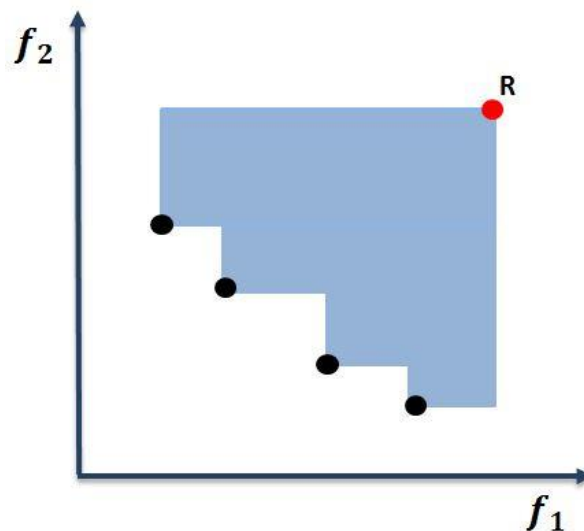


Figura 4.7: União de retângulos para o cálculo do *hypervolume* em um problema com dois objetivos.

5 *Arranjo Experimental e Resultados*

“Você nunca sabe que resultados virão da sua ação. Mas se você não fizer nada, não existirão resultados.”

– Mahatma Ghandi

Este capítulo tem como objetivo descrever como foi separado os conjuntos de treinamento da base de dados, como foi feito o arranjo experimental para detecção e contagem de plaquetas em imagens sanguíneas. Por fim, será mostrado como foi desenvolvida a etapa de teste.

5.1 Arranjo Experimental

5.1.1 Treinamento do *AdaBoost* utilizando o SPEA 2

Para a detecção de plaquetas utilizando o SPEA 2 (do inglês, *Strength Pareto Evolutionary Algorithm 2*), o tamanho da população interna (N) e do Arquivo Externo (\bar{N}) foi de 100 partículas. O número máximo de iterações (T) foi definido como 1000. A quantidade de repetições realizadas foi de 30.

À medida que uma solução consegue dominar outra, seu valor de força será incrementado. Ao final obtém-se sua força total, que será utilizada como parâmetro para ordenação das soluções dentro do Arquivo Externo.

As partículas receberam valores aleatórios dos classificadores e foram utilizados as quatro Características de Haar (ver Subseção 3.3). Além disso, foram passados a quantidade de objetivos conflitantes, no caso deste trabalho são dois, e para efeito de comparação nos testes a serem realizados foram usados 5, 10 e 20 classificadores.

5.1 Arranjo Experimental

Para analisar a Frente do Pareto do SPEA 2, foi realizado o cálculo do Hypervolume (ver Subseção 4.4.1) das soluções com o objetivo de verificar se o algoritmo estava convergindo. O resultado será mostrado mais adiante. O Algoritmo 5 descreve o pseudocódigo do treinamento do *AdaBoost* no SPEA 2.

Algoritmo 5: Pseudocódigo do treinamento do *AdaBoost* utilizando o SPEA2

Entrada: N : Tamanho da população interna
 \bar{N} : Tamanho do Arquivo Externo
 T : Número máximo de gerações;
Saída: A : Soluções não dominadas;

```

1 Inicialização:
2   Gere uma população inicial  $P_0$  e crie um Arquivo Externo vazio  $\bar{P}_0 = \emptyset$ ;
3   Atribua  $t = 0$ ;
4   Carregue a base de treino;
5   Inicialize as partículas com valores aleatórios para os classificadores;
6   para cada partícula faça
7     Calcule o limiar do AdaBoost;
8     Calcule a polaridade do classificador;
9   fim
10 fim
11 para  $t = 0, \dots, T$  faça
12   se  $t \geq T$  então
13      $A = \bar{P}_{t+1}$ ;
14     Pare o programa;
15   senão
16     Calcule o fitness de  $P_t$  e  $\bar{P}_t$ ;
17     Copie todos os indivíduos não dominados de  $P_t$  e  $\bar{P}_t$  para  $\bar{P}_{t+1}$ ;
18     se  $|\bar{P}_{t+1}| = \bar{N}$  então
19       Teste condição de parada;
20     senão se  $|\bar{P}_{t+1}| > \bar{N}$  então
21       Aplique o truncamento a  $\bar{P}_{t+1}$ ;
22     senão
23       Complete o Arquivo com elementos dominados de  $P_t$  e  $\bar{P}_t$ ;
24     fim
25     Execute seleção por torneio direto (binário) em  $\bar{P}_{t+1}$ . Aplique crossover em  $\bar{P}_{t+1}$  até completar o mating pool;
26     Aplique o operador de mutação ao mating pool e atribua a população resultante em  $P_{t+1}$ ;
27   fim
28 fim

```

5.1.2 Treinamento do *AdaBoost* utilizando o DFMOPSO

Para a detecção de plaquetas utilizando o DFMOPSO (do inglês, *Diversity Factor Multi-Objective Particle Swarm Optimization*), o tamanho da população e do

5.1 Arranjo Experimental

Arquivo Externo foram de 100 partículas. A quantidade de iterações foi definida como 1000 e a quantidade de repetições realizadas foi de 30. Os valores de c_1 e c_2 (coeficientes de aceleração do líder cognitivo e social, respectivamente) foram escolhidos randomicamente na faixa de valores entre 1,5 e 2,5. Através dos valores obtidos para os coeficientes de aceleração, é possível calcular o valor do fator de constrição (ver Equação 4.13 e 4.14). Clerc [34] demonstrou que só existe convergência quando $\varphi > 4$, caso contrário as partículas se movem de forma lenta executando movimento em espiral e não há garantia de convergência. Por isso, foi definido, durante desenvolvimento do DFMOPSO deste trabalho, que se $\varphi \leq 4$ o coeficiente de constrição retornaria valor igual a 1, caso contrário seu cálculo seria realizado pela Equação (4.14). O percentual de mutação do MOPSO foi definido como 0,5.

Em relação à integração do *AdaBoost* com o DFMOPSO, foram utilizados quatro Características de Haar (Ver Subseção 3.3) e são passados a quantidade de objetivos e a quantidade de classificadores a serem utilizados. Para o nosso problema, temos dois objetivos conflitantes e para efeito de comparação nos testes a serem realizados foram usados 5, 10 e 20 classificadores.

Para analisar a Frente do Pareto do DFMOPSO, foi realizado o cálculo do Hypervolume (ver Subseção 4.4.1) das soluções com o objetivo de verificar se o algoritmo estava convergindo. O resultado será mostrado mais adiante. O pseudocódigo da integração do *AdaBoost* no DFMOPSO pode ser visto no Algoritmo 6.

5.1.3 Ambiente de Desenvolvimento

O sistema proposto pelo trabalho foi desenvolvido utilizando a linguagem de programação Java, no ambiente de desenvolvimento Eclipse. Para o código do *AdaBoost*, foi utilizado o *OpenCV* [35] (do inglês, *Open Source Computer Vision Library*), uma biblioteca multiplataforma para o desenvolvimento de aplicativos na área de Visão Computacional, no ambiente Java.

5.1 Arranjo Experimental

As simulações foram realizadas em um computador com 4GB de memória RAM, processador *Intel Core i5 2,50 GHz*, utilizando o Sistema Operacional *Windows 7*.

Algoritmo 6: Pseudocódigo do treinamento do *AdaBoost* utilizando o DFMOPSO

```
1 Carregue a base de treino;
2 Inicialize as partículas com valores aleatórios para os classificadores;
3 para cada partícula faça
4   | Calcule o limiar do AdaBoost;
5   | Calcule a polaridade do classificador;
6 fim
7 Selecione as soluções para o Arquivo Externo;
8 enquanto critério de parada não for alcançado faça
9   | para cada partícula faça
10    | Aplique o operador de turbulência utilizado no MOPSO;
11    | Selecione líderes sociais usando Torneio Binário com DF;
12    | Atualize a velocidade usando a Equação (4.11);
13    | Aplique Fator de Constrição (Equação 4.14);
14    | Atualize a posição usando a Equação (4.12);
15    | Avalie a partícula;
16    | Atualize o líder cognitivo, se necessário;
17   | fim
18   | Atualize e ajuste o Arquivo Externo usando CD;
19   | Avalie o DF das soluções dentro do Arquivo Externo;
20 fim
21 Retorne o Arquivo Externo.
```

5.1.4 Imagens e Conjunto de Treinamento

As imagens sanguíneas contendo plaquetas e os demais elementos sanguíneos foram coletadas a partir do banco referenciado por Nascimento [36], o banco de imagens da Sociedade Americana de Hematologia [37] (ASH – do inglês, *American Society of Hematology*), para compor o Banco de Imagens a ser utilizado neste trabalho.

Após criado o banco de imagens, o conjunto de treinamento foi montado. O banco de imagens foi dividido em dois subgrupos: um grupo contendo 75% das imagens que foram utilizadas para fazer o treinamento e outro grupo contendo os 25% restantes das imagens que foram utilizadas como testes. A partir das imagens

5.2 Análise dos Resultados

de treino, foram construídos 2 grupos de imagens: um grupo contendo imagens de plaquetas (denominadas imagens positivas) e um grupo contendo imagens que não são plaquetas (denominadas imagens negativas). A quantidade de imagens utilizadas neste trabalho pode ser observada na Tabela 5.1.

Tabela 5.1: Distribuição da quantidade de imagens

Imagem	Quantidade
Positiva	110
Negativa	417
Teste Positiva	37
Teste Negativa	139

Após a divisão do banco de imagens, foi realizado o treinamento do *AdaBoost* para a construção do sistema de detecção de plaquetas. O treino foi realizado conforme o algoritmo original do *AdaBoost* (Algoritmo 1) demonstrado na Subseção 3.5.1. Os classificadores foram compostos pelas quatro Características de Haar (Ver Subseção 3.3; Imagem 3.3) durante o treinamento. Na próxima Subseção serão realizadas as análises dos resultados obtidos pela aplicação das técnicas utilizadas neste trabalho.

5.2 Análise dos Resultados

A obtenção dos resultados foi baseada na comparação realizada por Silva [38] entre a quantidade de classificadores fracos utilizada no treinamento do *AdaBoost*. Para efeito de comparação, as quantidades de classificadores fracos utilizadas neste trabalho serão de 5, 10 e 20. Além da comparação entre as técnicas multiobjetivas deste trabalho, outras comparações serão feitas levando em consideração os resultados obtidos pelo trabalho de Silva [38] com técnicas mono-objetivas. Os resultados dos experimentos serão observados nas próximas Subseções.

Para analisar a Frente de Pareto das técnicas de otimização multiobjetivo aplicadas neste trabalho, um gráfico da métrica *Hypervolume* foi gerado para cada

5.2 Análise dos Resultados

técnica. A Figura 5.1 representa o *hypervolume* do SPEA 2. Nota-se que o algoritmo evolucionário convergiu, mas possui convergência mais lenta para o treinamento com 20 classificadores, podendo ter estagnado em algum máximo local.

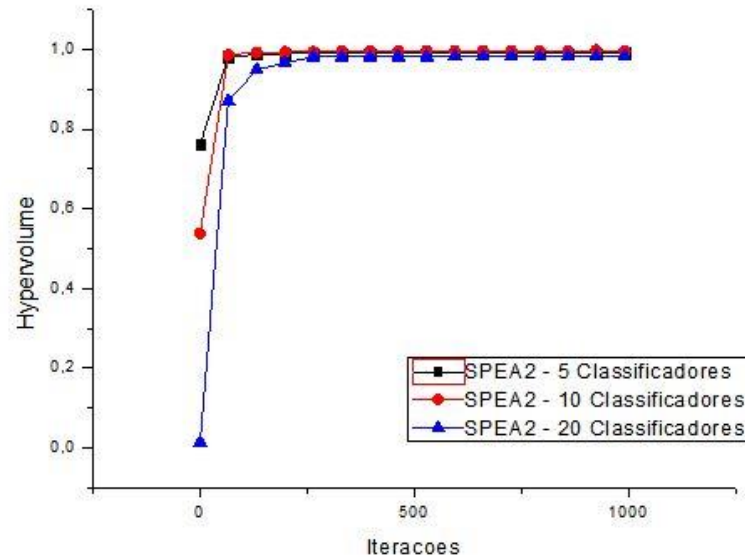


Figura 5.1: Análise do *Hypervolume* para o SPEA 2.

Assim como foi gerado para o SPEA 2, a Figura 5.2 mostra o gráfico gerado para a métrica do *Hypervolume* no algoritmo do DFMOPSO. O gráfico demonstra que o DFMOPSO convergiu satisfatoriamente para o problema proposto neste trabalho, apresentando melhores resultados para o treinamento com 5 classificadores.

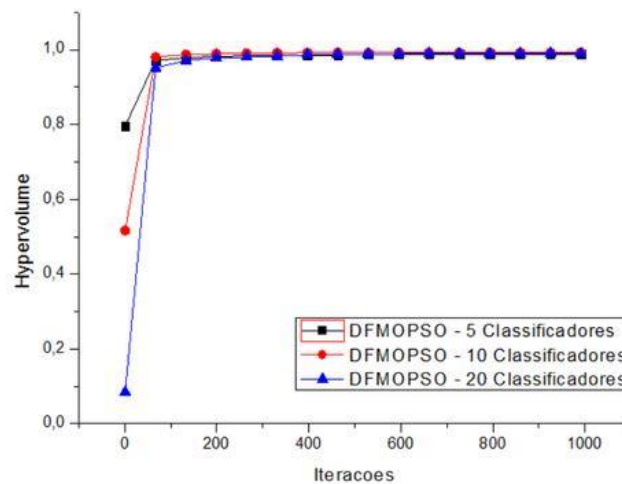


Figura 5.2: Análise do *Hypervolume* para o DFMOPSO

5.2.1 Experimento utilizando 5 Classificadores Fracos

A Figura 5.3 mostra uma comparação entre as Frentes de Pareto das técnicas do SPEA 2 (do inglês, *Strength Pareto Evolutionary Algorithm 2*) e do DFMOPSO (do inglês, *Diversity Factor Multi-Objective Particle Swarm Optimization*). Observa-se que o DFMOPSO apresenta maior convergência das soluções da Frente de Pareto.

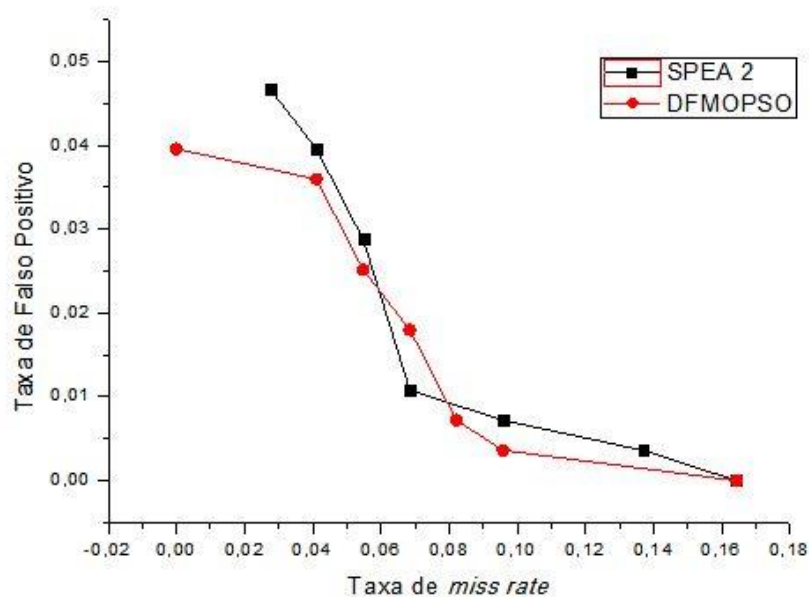


Figura 5.3: Comparativo entre as duas técnicas implementadas analisando os dois objetivos conflitantes utilizando 5 classificadores.

Para melhor entendimento e comparação, o gráfico esquerdo da Figura 5.4 representa a dispersão entre as taxas de *miss rate* encontradas em cada técnica. Observa-se que o DFMOPSO apresenta taxa de *miss rate* menor que o SPEA 2, possuindo uma diferença perceptível entre eles.

Analisando o gráfico direito da Figura 5.4, cujo mesmo representa a taxa de falso positivo obtida por cada técnica, pode-se observar que o SPEA 2 possui uma maior taxa de falso positivo, sendo o DFMOPSO a técnica com melhor resultado. As duas técnicas não apresentam muitas diferenças nos valores das taxas.

5.2 Análise dos Resultados

Ao analisar os três gráficos, pode-se observar que o DFMOPSO possui melhor desempenho no treinamento do *AdaBoost* com 5 classificadores fracos.

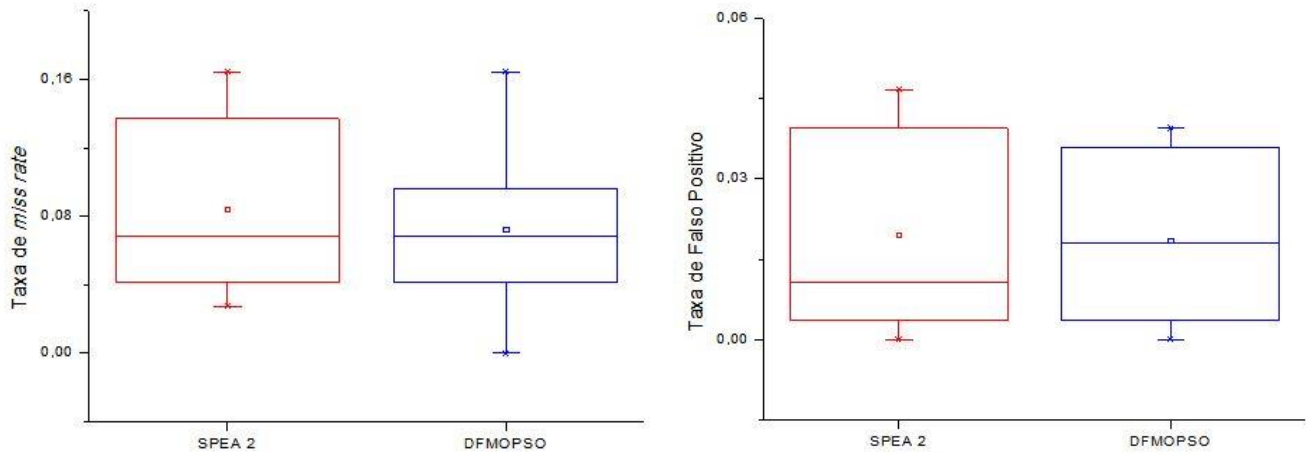


Figura 5.4: Comparativo entre as técnicas, utilizando 5 classificadores: O gráfico esquerdo representa a taxa de *miss rate* e o direito a taxa de falso positivo.

Analisando os resultados obtidos pelo trabalho de Silva [38], algumas comparações podem ser realizadas. Observando a Figura 5.5 observa-se que o GA (*Genetic Algorithm*) apresenta melhores resultados para o treinamento com 5 classificadores. O DFMOPSO apresenta resultados próximos ao do GA em relação à taxa de *miss rate*. Quanto à taxa de falso positivo, o GA possui resultados melhores do que as métricas aplicadas neste trabalho.

5.2.2 Experimento utilizando 10 Classificadores Fracos

A Figura 5.6 mostra uma comparação entre as Frentes de Pareto das técnicas do SPEA 2 (do inglês, *Strength Pareto Evolutionary Algorithm 2*) e do DFMOPSO (do inglês, *Diversity Factor Multi-Objective Particle Swarm Optimization*), notando-se o bom desempenho do SPEA 2 para o treinamento com 10 classificadores.

5.2 Análise dos Resultados

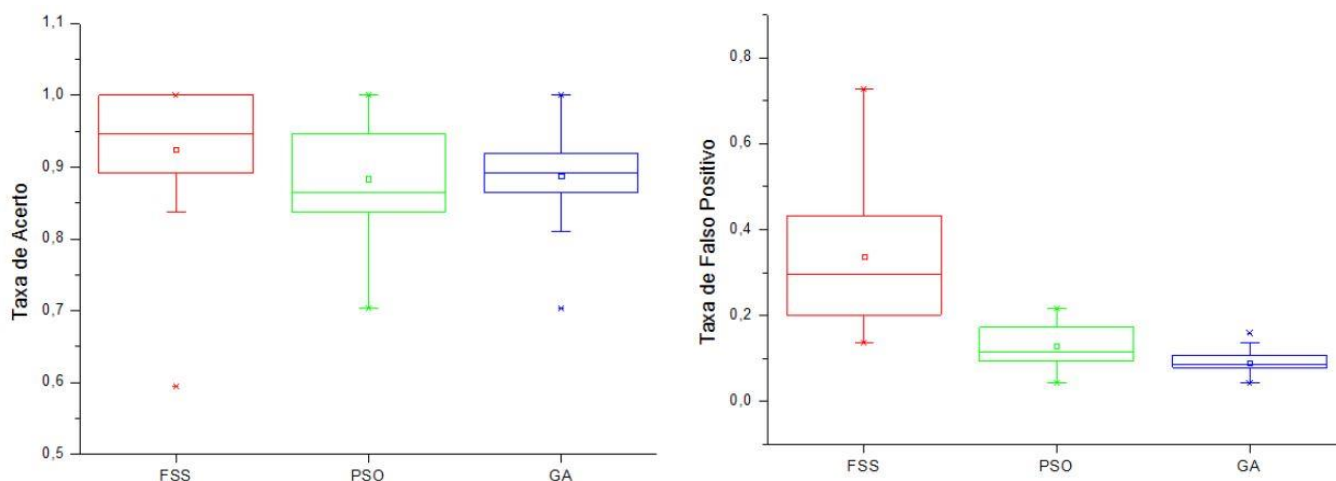


Figura 5.5: Resultados das métricas utilizadas por Silva [38], utilizando 5 classificadores.

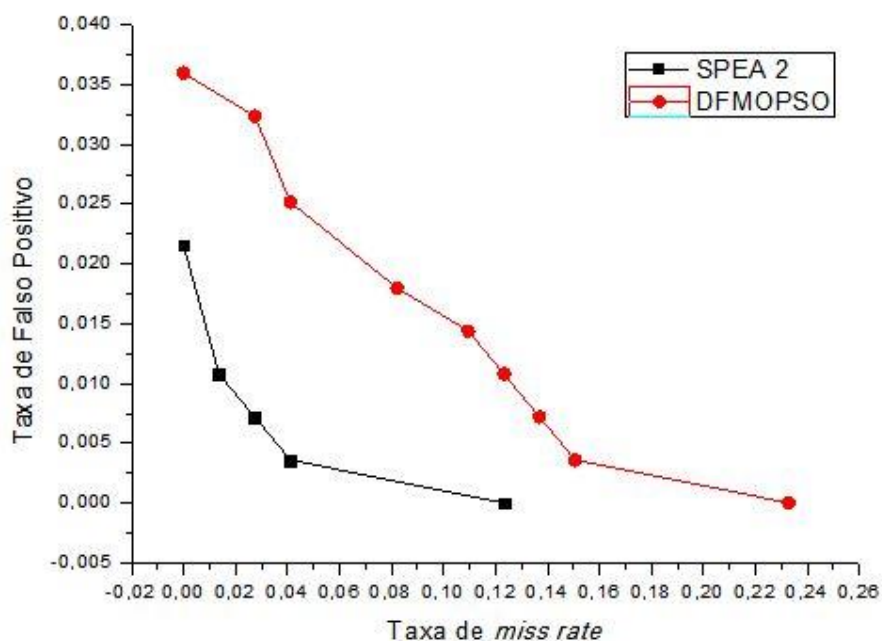


Figura 5.6: Comparativo entre as duas técnicas implementadas analisando os dois objetivos conflitantes utilizando 10 classificadores.

O gráfico esquerdo da Figura 5.7 demonstra um comparativo entre as taxas de *miss rate* das duas técnicas implementadas neste trabalho. O SPEA 2 apresenta menor taxa de *miss rate*, havendo uma diferença significativa em relação ao DFMOPSO.

5.2 Análise dos Resultados

O gráfico direito da Figura 5.7 demonstra que o DFMOPSO apresenta uma maior taxa de falsos positivos, sendo o SPEA 2 o que apresenta melhores resultados. Observando o gráfico, nota-se que o DFMOPSO tem uma disparada em relação ao SPEA 2, mostrando que o algoritmo evolucionário apresenta melhor resultado para o treinamento com 10 classificadores.

Ao analisar a Figura 5.8, resultado oriundo do trabalho proposto por Silva [38], nota-se novamente que o GA apresenta melhores resultados em relação às outras métricas de otimização objetiva. O SPEA 2 apresenta menor taxa de *miss rate* e apresenta diferença significativa em relação ao GA. Como também o SPEA 2 representa menor taxa de falso positivo em relação ao GA.

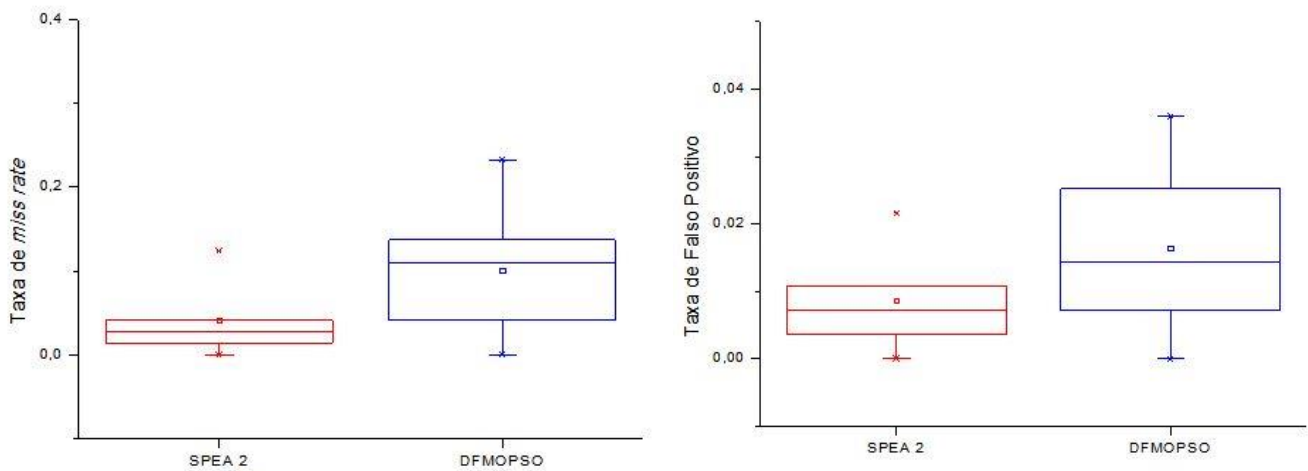


Figura 5.7: Comparativo entre as técnicas, utilizando 10 classificadores: O gráfico esquerdo representa a taxa de *miss rate* e o direito a taxa de falso positivo.

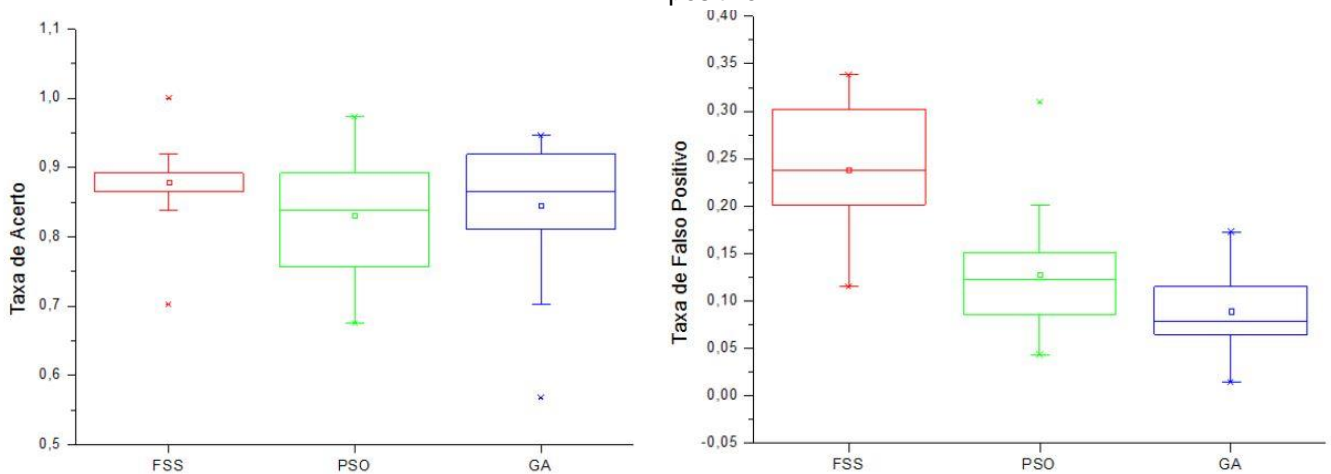


Figura 5.8: Resultados das métricas utilizadas por Silva [38], utilizando 10 classificadores.

5.2.3 Experimento utilizando 20 Classificadores Fracos

A Figura 5.9 mostra uma comparação entre as Frentes de Pareto das técnicas do SPEA 2 (do inglês, *Strength Pareto Evolutionary Algorithm 2*) e do DFMOPSO (do inglês, *Diversity Factor Multi-Objective Particle Swarm Optimization*), sendo o SPEA 2 o que apresenta melhor resultado.

Observa-se na Figura 5.10 que o SPEA 2 apresenta uma menor taxa de *miss rate* em relação ao DFMOPSO, como pode ser visto no gráfico esquerdo. As duas técnicas apresentam valores próximos.

No gráfico direito da Figura 5.10, podemos observar que o SPEA 2 apresenta menor taxa de falso positivo, havendo uma pequena diferença entre ele e o DFMOPSO.

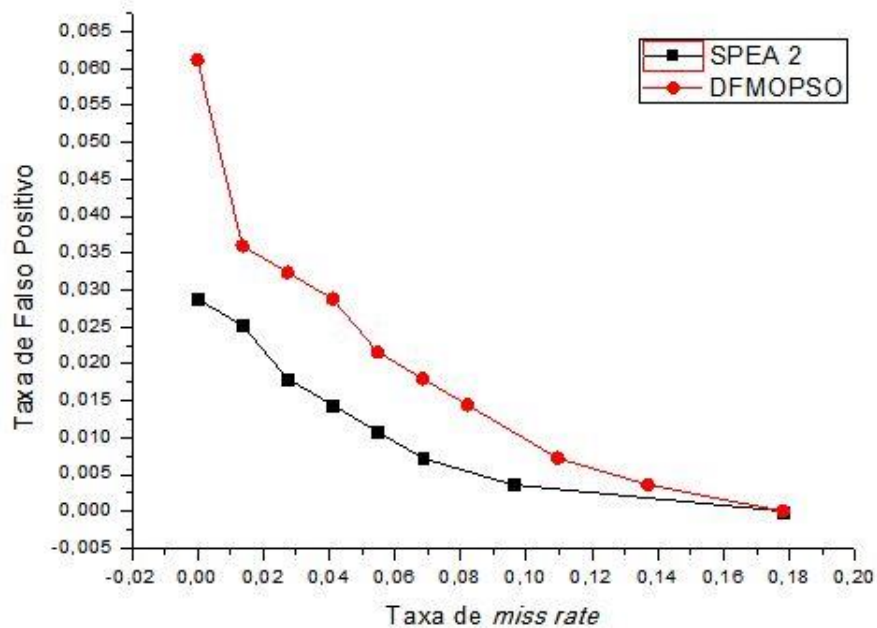


Figura 5.9: Comparativo entre as duas técnicas implementadas analisando os dois objetivos conflitantes utilizando 20 classificadores.

5.2 Análise dos Resultados

Ao analisar os resultados obtidos no trabalho proposto por Silva [38] em relação ao experimento utilizando 20 classificadores, Figura 5.11, podemos observar que o SPEA 2 apresenta melhores resultados em relação à taxa de *miss rate* se comparado com o PSO (*Particle Swarm Optimization*), cujo resultado foi melhor que as outras técnicas no trabalho de Silva [38] utilizando 20 classificadores.

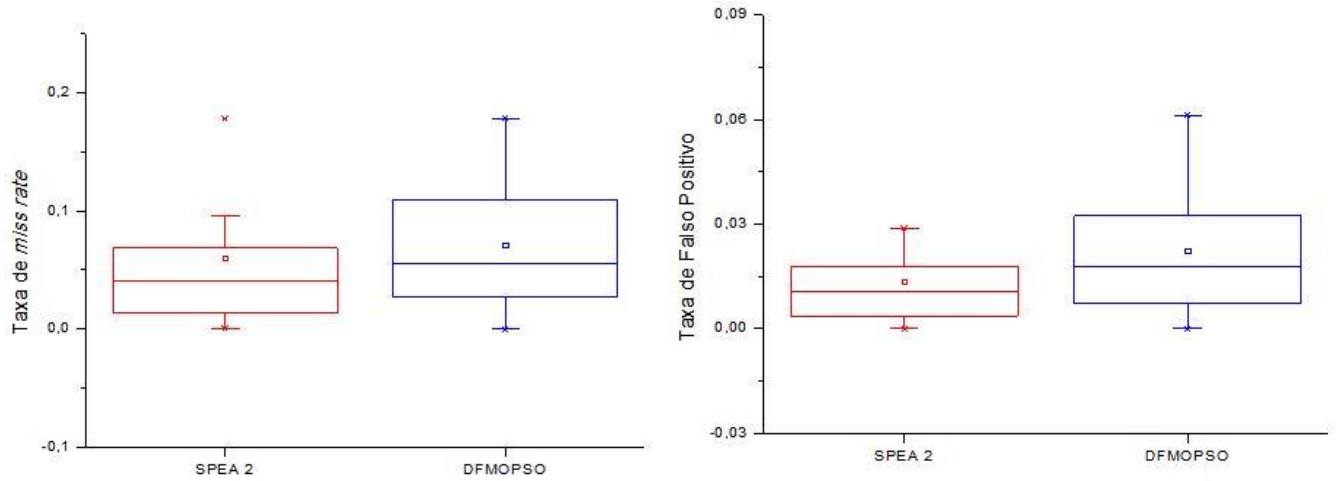


Figura 5.10: Comparativo entre as técnicas, utilizando 20 classificadores: O gráfico esquerdo representa a taxa de *miss rate* e o direito a taxa de falso positivo.

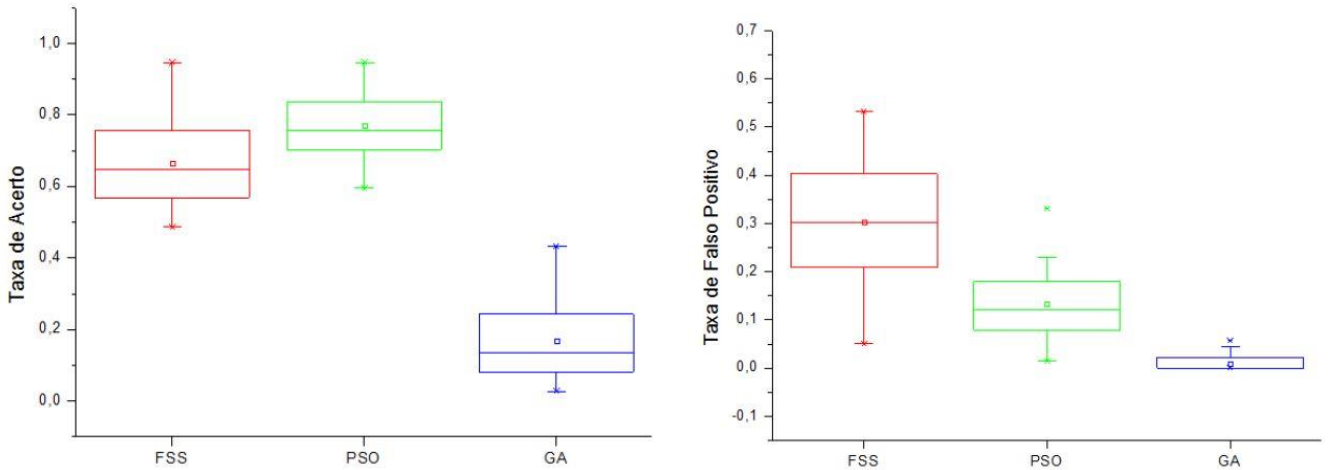


Figura 5.11: Resultados das métricas utilizadas por Silva [38], utilizando 20 classificadores.

5.2.4 Experimento Geral

Após análises dos resultados feita por cada quantidade de classificadores individualmente, a Figura 5.12 apresenta a comparação entre as Frentes de Paretos do SPEA 2 (do inglês, *Strength Pareto Evolutionary Algorithm 2*) em relação a 5,10 e 20 classificadores. Nota-se que o SPEA 2 obtém bons resultados para o treinamento do *AdaBoost* com 10 classificadores.

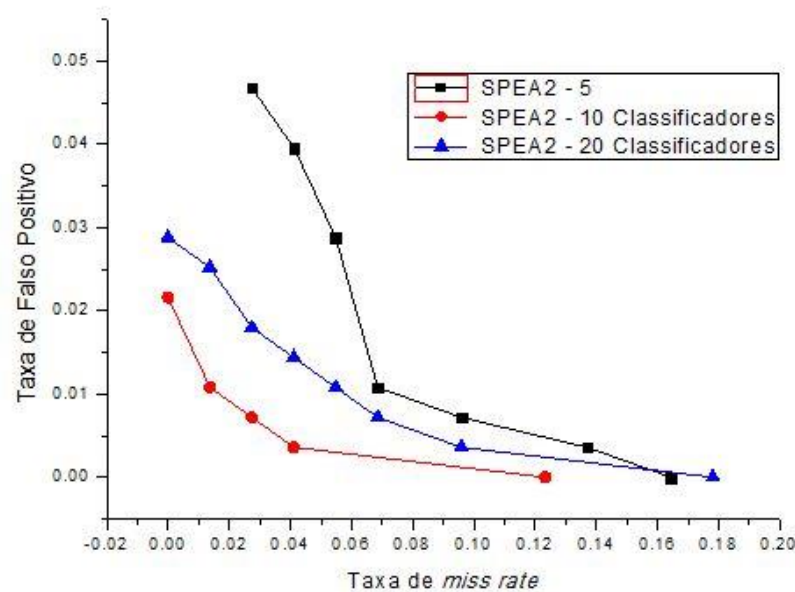


Figura 5.12: Comparação entre as Frentes de Paretos para 5,10 e 20 classificadores no SPEA 2.

Podemos observar na Figura 5.13 a comparação entre as Frentes de Paretos do DFMOPSO (do inglês, *Diversity Factor Multi-Objective Particle Swarm Optimization*). Observa-se que o DFMOPSO apresenta melhores resultados para o treinamento utilizando 5 classificadores.

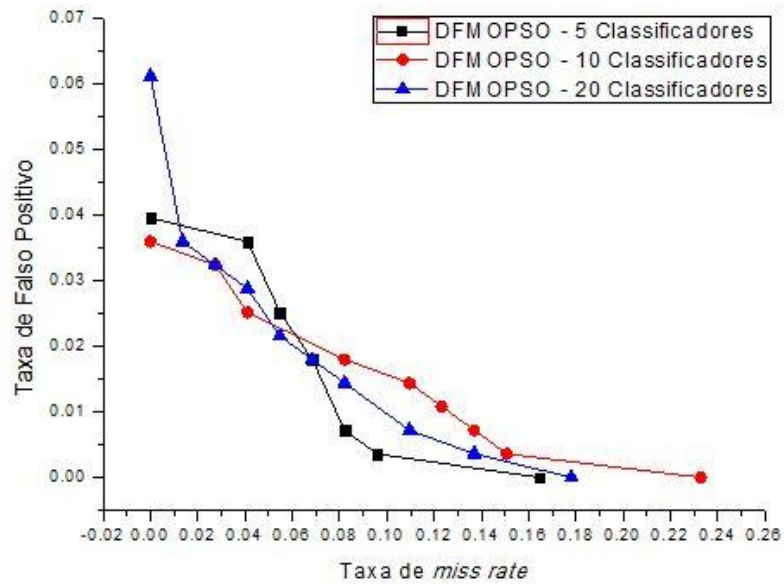


Figura 5.13: Comparação entre as Frentes de Paretos para 5,10 e 20 classificadores no DFMOPSO.

6 Conclusões e Trabalhos Futuros

“A melhor maneira de prever o futuro é criá-lo”

– Peter Drucker

Neste trabalho de conclusão de curso foi proposto o uso de técnicas de otimização de problemas multiobjetivos para otimizar o treinamento de classificadores *AdaBoost* com o intuito de auxiliar na identificação e contagem automática de células sanguíneas como, mais especificamente neste trabalho, as plaquetas.

6.1 Conclusões e Contribuições

Os dois algoritmos estudados e implementados neste trabalho obtiveram resultados muito próximos uns dos outros. O SPEA 2 e o DFMOPSO são algoritmos multiobjetivos e fazem uso de um Arquivo Externo para armazenar as soluções não-dominadas.

No treinamento de classificadores *AdaBoost* utilizando 5 classificadores, os resultados de ambas as técnicas foram muito parecidos, mas o DFMOPSO obteve melhores resultados. Quando comparado com a métrica do GA, obteve resultados parecidos em relação à taxa de *miss rate* enquanto que em relação à taxa de falso positivo o GA apresentou melhores resultados. O SPEA 2 obteve melhores resultados para o treinamento com 10 classificadores, tanto comparando com o DFMOPSO quanto com o GA, cujo desempenho foi superior em relação às outras técnicas aplicadas por Silva [38]. Observando a Figura 5.12 nota-se que o SPEA 2 convergiu, mas apresentou convergência lenta em relação ao treinamento com 20 classificadores. O DFMOPSO apresentou uma ótima convergência, principalmente em relação ao treinamento com 5 classificadores. Para os experimentos com 20 classificadores, o SPEA 2 obteve melhores resultado, inclusive em relação ao PSO,

cuja métrica apresentou melhores resultados no trabalho de Silva [38]. Os resultados deste trabalho apresentaram ótimas contribuições para a detecção e contagem automática de plaquetas por terem gerados resultados com uma baixa taxa de *miss rate* e de falso positivo.

6.2 Trabalhos Futuros

Como trabalhos futuros a serem desenvolvidos, menciona-se primeiramente o uso da mesma tecnologia que foi apresentada neste trabalho, com o uso de metaheurísticas para treinamento de classificadores *AdaBoost*, para detecção das demais células sanguíneas humanas, os eritrócitos (glóbulos vermelhos ou hemácias) e os leucócitos (glóbulos brancos). Assim como nas plaquetas, a contagem automática das hemácias e dos leucócitos torna-se custosa e pode apresentar alterações tanto qualitativamente quanto quantitativamente, fazendo com que doenças específicas de cada célula possa passar despercebida.

Além disso, também pode ser proposto a utilização de vários classificadores como um comitê de classificadores. A ideia de combinar diferentes métodos para melhorar o desempenho do sistema surge da percepção de que nenhum classificador pode ser considerado totalmente satisfatório em uma atividade em particular. O uso de um comitê de classificadores pode melhorar a efetividade do processo de reconhecimento como um todo, uma vez que oferecer informações complementares sobre padrões a serem classificados. Conseguindo uma melhora na detecção de objetos, a detecção e contagem automática de elementos figurados do sangue consequentemente obterá uma melhora também.

Referências Bibliográficas

- [1] SANGUE. In: **Wikipédia: a enciclopédia livre**. Disponível em: <pt.wikipedia.org/wiki/Sangue> Acesso em: 25 abr 2014.
- [2] VIVAS, WANESSA L. P. **Manual Prático de Hematologia**. 33 f.
- [3] COMAR, S. R.; DANCHURA, H. S. M.; SILVA, P. H. **Contagem de plaquetas: avaliação de metodologias manuais e aplicação na rotina laboratorial**. Revista Brasileira de Hematologia e Hemoterapia, v. 31, n. 6, p. 431-436, 2009. ISSN 1516-8484. ALEKSANDER, I. e MORTON, H. **An Introduction to Neural Computing**. International Thomson Computer Press, 1995. 490 p.
- [4] ARTHUR, L. et al. **Comparação entre a Contagem de Plaquetas pelos Métodos Manual e Automatizado**. 2007.
- [5] GROTTTO, Helena Z. W. **O Hemograma: importância para a interpretação da biópsia**. Revista Brasileira de Hematologia e Hemoterapia, v. 31, n. 3, 2009. ISSN 1516-8484.
- [6] FARIAS, M. G.; BO, S. D. **Importância clínica e laboratorial do volume plaquetário médio**. Jornal Brasileiro de Patologia e Medicina Laboratorial, v. 46, n. 04, 2010. ISSN 1676-2444.
- [7] BAIN, B. J. **Células Sanguíneas: um guia prático**. Ed. 4. Porto Alegre. Artmed, 2006.
- [8] FAILACE, Renato et al. **Hemograma: Manual de Interpretação**. Ed. 5. Porto Alegre: ARTMED, 2009. 424p. ISBN 978-85-363-2081-6.
- [9] SANTOS, M. E.; GALVÃO, T.; OLIVEIRA, A. L. M. **Tamanho de Plaquetas e Doença Vascular**. Laboratório de Hemostasia/HEMOPE. 2008.
- [10] SILVA, T. C. **Biologia 3**. Sistema de Ensino POLIEDRO. 1. Ed. São Paulo, Editora Poliedro. ISBN 978-85-98293-04-2.

- [11] CÂMARA, Brunno. **Conhecendo a Câmara de Neubauer**. In: Biomedicina Padrão. Disponível em: <<http://www.biomedicinapadrao.com/2013/10/conhecendo-camara-de-neubauer.html>> Acesso em: 03 mai 2014.
- [12] VIOLA, P.; JONES, M. J. **Robust Real-Time Face Detection**. *International Journal of Computer Vision*, Springer, v. 57, n. 2, p. 137-154, 2004. ISSN 09205691.
- [13] VIOLA, P.; JONES, M. J. **Rapid Object Detection using a Boosted Cascade of Simple Features**. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, no. C, p. I-511-I-518.
- [14] FREUND, Y.; SCHAPIRE, R. E.; AVENUE, P. **A Short Introduction to Boosting**. *Society, Citeseer*, v. 14, n. 5, p. 771-780, 1999. ISSN 09128085.
- [15] HAAR-LIKE FEATURES. **Wikipedia: The Free Encyclopedia**. Disponível em: <en.wikipedia.org/wiki/Haar-like_features> Acesso em 20 mai 2014.
- [16] PAPAGEORGIOU, C. P.; OREN, M.; POGGIO, T. **A General Framework for Object Detection**. *Sixth International Conference on Computer Vision IEEE, Narosa Publishing House*, v.6, n. January, p. 555-562, 1998.
- [17] CARBONETTO, P. **Robust object detection using boosted learning**. *Department of Computer Science, University of British Columbia, Vancouver*, 2002.
- [18] APRENDIZADO DE MÁQUINA. In: **Wikipédia: a enciclopédia livre**. Disponível em: <pt.wikipedia.org/wiki/Aprendizado_de_máquina> Acesso em: 25 mai 2014.
- [19] OLIVEIRA, Eduardo. **Captura de expressões faciais para identificação de emoções básicas em humanos**. 2008. 88 f. Monografia para obtenção do grau de Bacharel em Sistemas de Informação, Universidade do Vale do Rio dos Sinos.
- [20] EDGEWORTH, F. Y. **Mathematical Psychics**. Na *Essay on the Application of Mathematics to the Moral Sciences*.

- [21] PARETO, V. **Cours d'économie politique, volume i and ii**. F. Rouge, Lausame.
- [22] GABRIEL, P. H. R.; DELBEM, A. C. B. **Fundamentos de Algoritmos Evolutivos**. P. 33.
- [23] ZITZLER, E.; LAUMANNNS, M.; BLEULER, S. **A Tutorial on Evolutionary Multiobjective Optimization**. Swiss Federal Institute of Technology Zurich. Computer Engineering and Networks Laboratory (TIK), Gloriastrasse, Switzerland.
- [24] ZITZLER, E.; LAUMANNNS M.; THIELE, L. **SPEA2: Improving the Strength Pareto Evolutionary Algorithm**. Computer Engineering and Networks Laboratory (TIK). 2001.
- [25] COELLO, C.; LAMONT, G.; VELDHUIZEN, D. V. **Evolutionary Algorithms for Solving Multi-Objective Problems**. Nova York: Springer-Verlag New York Inc, 2007.
- [26] KENNEDY, J.; EBERHART, R. **Particle Swarm Optimization**. In: *Proceedings of IEEE International Conference on Neural Networks*. Perth, Aust: IEEE, 1995. p. 1945-1948.
- [27] COELLO, C.; LECHUGA, M. **MOPSO: A proposal for multiple objective particle swarm optimization**. In: IEEE. *Proceedings of the Congress on Evolutionary Computation (CEC 2002)*. Honolulu, HI, EUA, 2002. v. 2, p. 1051-1056.
- [28] AZUMA, Regina M. **Otimização multiobjetivo em problema de estoque e roteamento gerenciados pelo fornecedor**. 2011. 121f. Dissertação de Mestrado do curso Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.
- [29] SANTANA, R.; PONTES, M.; BASTOS-FILHO, C. **A multiple objective particle swarm optimization approach using crowding distance and roulette wheel**. In: IEEE. *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*. Pisa, Itália, 2009. p. 237–242.

- [30] SILVA, Dennis R. C. **Otimizador Multiobjetivo baseado em inteligência de enxames utilizando fator de diversidade**. 2012. 70f. Trabalho de Conclusão de Curso de Engenharia da Computação, UPE, Pernambuco.
- [31] SILVA, Dennis R. C.; BASTOS-FILHO, Carmelo J. A. **A Multi-Objective Particle Swarm Optimizer Based on Diversity**. INTELLI. *The Second International Conference on Intelligent Systems and Applications*, 2013. Venice, Italy. p. 109 – 114.
- [32] SILVA, Dennis R. C.; BASTOS-FILHO, Carmelo J. A. **A Speed-Constrained Multi-Objective Particle Swarm Optimizer Guided by a Diversity Metrics**. In: *The Latin American Congress on Computational Intelligence*. San Carlos de Bariloche, Argentina, 2014. p. 1-6.
- [33] CLERC, M. **The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization**. In: *Evolutionary Computation*, 1999. CEC 99. *Proceedings of the 1999 Congress on*. 1999. v. 3. Disponível em: <<http://dx.doi.org/10.1109/CEC.1999.785513>>.
- [34] CLERC, M.; KENNEDY, J. **The Particle Swarm – explosion, stability, and convergence in a multidimensional complex space**. *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, 2002. Disponível em: <<http://dx.doi.org/10.1109/4235.985692>>.
- [35] OPENCV. In: **Wikipédia: a enciclopédia livre**. Disponível em: <pt.wikipedia.org/wiki/OpenCV> Acesso em: 20 jun 2014.
- [36] NASCIMENTO, Débora N. O. **Classificação AdaBoost para detecção e contagem automática de plaquetas**. 2011. 55f. Trabalho de Conclusão de Curso de Engenharia da Computação, UPE, Pernambuco.
- [37] American Society of Hematology. **American Society of Hematology Image Bank**. 2011. Disponível em: <<http://imagebank.hematology.org/>>.
- [38] SILVA, Willamos A. S. **Comparing Meta-heuristics for AdaBoost Training Applied to Platelets Detection**. In: *The Latin American Congress on Computational Intelligence*. San Carlos de Bariloche, Argentina, 2014. p. 13-18.