

CENTRO UNIVERSITÁRIO FEI
DAVI ARAUJO DAL FABBRO

**NORMALIZAÇÃO ESPACIAL NÃO RÍGIDA DE IMAGENS ASSISTIDA POR
METAMODELOS**

São Bernardo do Campo

2018

DAVI ARAUJO DAL FABBRO

**NORMALIZAÇÃO ESPACIAL NÃO RÍGIDA DE IMAGENS ASSISTIDA POR
METAMODELOS**

Dissertação de Mestrado apresentada ao Centro Universitário da FEI para obtenção do título de Mestre em Engenharia Elétrica. Orientado pelo Prof. Dr. Carlos Eduardo Thomaz.

São Bernardo do Campo

2018

Dal Fabbro, Davi Araujo.
Normalização Espacial Não Rígida de Imagens Assistida por
Metamodelos / Davi Araujo Dal Fabbro. São Bernardo do Campo, 2018.
116 p. : il.

Dissertação - Centro Universitário FEI.
Orientador: Prof. Dr. Carlos Eduardo Thomaz.

1. Normalização espacial de imagens.. 2. Otimização assistida por
metamodelos. 3. Reconhecimento de padrões. I. Thomaz, Carlos
Eduardo, orient. II. Título.

Aluno: Davi Araujo Dal Fabbro

Matrícula: 116104-1

Título do Trabalho: Normalização espacial não rígida de imagens assistida por metamodelos.

Área de Concentração: Processamento de Sinais e Imagens

Orientador: Prof. Dr. Carlos Eduardo Thomaz

Data da realização da defesa: 09/03/2018

ORIGINAL ASSINADA

Avaliação da Banca Examinadora:

São Bernardo do Campo, 09 / 03 / 2018.

MEMBROS DA BANCA EXAMINADORA

Prof. Dr. Carlos Eduardo Thomaz

Ass.: _____

Prof. Dr. Fernando José Von Zuben

Ass.: _____

Prof. Dr. Marko Ackermann

Ass.: _____

A Banca Julgadora acima-assinada atribuiu ao aluno o seguinte resultado:

APROVADO

REPROVADO

VERSÃO FINAL DA DISSERTAÇÃO

**APROVO A VERSÃO FINAL DA DISSERTAÇÃO EM QUE
FORAM INCLUÍDAS AS RECOMENDAÇÕES DA BANCA
EXAMINADORA**

Aprovação do Coordenador do Programa de Pós-graduação

Prof. Dr. Carlos Eduardo Thomaz

À minha família e a todos que estiveram ao meu lado durante meu tempo de mestrado.

AGRADECIMENTOS

Gostaria de iniciar meus agradecimentos expressando minha gratidão à minha família. Especialmente aos meus pais, Francisco e Maria do Rosário, por todo amor e dedicação a mim, bem como o suporte necessário para concluir a redação deste trabalho; e às minhas irmãs, Rachel e Esther, pela paciência e interesse em ouvir os assuntos estudados por mim durante minha jornada de mestrado.

À minha companheira e melhor amiga Cristiane, por todas as dicas e sugestões durante a elaboração das figuras encontradas nesta dissertação, pela paciência, bondade e carinho. Certamente sem você eu teria muito mais dificuldades.

Agradeço enormemente meu orientador, Dr. Carlos Eduardo Thomaz, pela confiança depositada em mim ao conceder a autonomia necessária para condução das minhas pesquisas. Também agradeço as suas orientações, tanto pessoais como acadêmicas, durante nossas longas, além de enriquecedoras, reuniões. Sem elas, o caminho para concluir este trabalho seria infinitamente mais árduo.

Além do meu orientador, agradeço aos professores que me auxiliaram com os fundamentos deste trabalho. Ao professor Dr. Paulo Eduardo Santo, pelas aulas de Inteligência Artificial que serviram de grande motivação para a concepção do tema. Ao professor Dr. Paulo Sérgio pelas aulas de Visão Computacional, disciplina essencial para a aplicação deste trabalho. Finalmente, aos professores Dr. Fabrício Leonardi e Dr. Marko Ackermann, pelas aulas de Otimização Matemática, as quais forneceram conceitos indispensáveis para fundamentar parte da teoria encontrada ao longo desta dissertação.

Também agradeço ao Dr. Rodrigo Alves Moreto e novamente ao professor Dr. Marko Ackermann pelos comentários e sugestões. Aos meus colegas do Laboratório de Processamento de Imagens, pela companhia, grandes conversas e momentos de descontração durante o período de mestrado.

Finalizo agradecendo à CAPES e ao INCT pela bolsa de estudos que financiou todos os projetos de pesquisa realizados durante estes dois anos de caminhada.

“No man ever steps in the same river twice, for it’s not the same river and he’s not the same man.”

Heraclitus

RESUMO

Classicamente aplicados em simulações computacionais como CFD (*Computational Fluid Dynamics*) e FEM (*Finite Element Method*), metamodelos para otimização trouxeram resultados relevantes para aproximar a função objetivo no espaço de busca e resolver problemas em que o cálculo da função objetivo exige elevados recursos computacionais. Recentemente, com a evolução de novos algoritmos de aprendizado de máquina e reconhecimento de padrões, esta técnica também mostrou-se bem sucedida para acelerar o processo de ajuste dos hiperparâmetros de modelos que antes dependiam de um conhecimento profundo sobre os seus fundamentos matemáticos ou da busca exaustiva pelos parâmetros ótimos. Neste contexto, esta dissertação propõe uma metodologia para aplicação de otimização assistida por metamodelos na área de visão computacional, mais especificamente em normalização espacial não rígida de imagens. Este tópico de pesquisa tem sido pouco explorado pela literatura de otimização assistida por metamodelos, além de ser computacionalmente complexo devido às múltiplas opções de ajuste dos parâmetros. Denominada aqui de Normalização Espacial Não Rígida de Imagens Assistida por Metamodelos, a metodologia proposta descreve uma nova métrica que leva em consideração o compromisso em deformar os elementos geométricos da imagem preservando suas características estatísticas principais. Este estudo descreve também duas taxonomias, uma de otimização e outra de normalização espacial de imagens, para fundamentar teoricamente a metodologia proposta. Após os experimentos em três bases de imagens distintas, os resultados obtidos mostram que a qualidade da normalização espacial não rígida pode ser otimizada por meio da métrica proposta, e o metamodelo construído auxilia tanto no processo de otimização como na compreensão maior do comportamento da função objetivo no espaço de busca.

Palavras-chave: Normalização espacial de imagens. Otimização assistida por metamodelos. Reconhecimento de padrões.

ABSTRACT

Classically used for computational simulations such as CFD (Computational Fluid Dynamics) and FEM (Finite Element Method), surrogate-assisted optimization led to consistent findings to approximate the objective function in the search space and to deal with problems where the calculation of the objective function demands excessive computational resources. Recently, along the improvement of new machine learning and pattern recognition algorithms, this technique also proved to be successful in speeding the task of hiperparameter tuning for models which previously relied on a broad understanding about its mathematical principles or brute force search to find the optimal parameters. In this context, this work introduces a framework that applies surrogate-assisted optimization in the field of computer vision, more specifically, in nonrigid image registration. This research topic has been scarcely explored by authors in the literature of surrogate-assisted optimization, besides being computationally intricate due to the multiple options for parameter tuning. Named here as Surrogate-Assisted Nonrigid Registration (SANRR), this proposed framework describes a novel measure that takes into consideration the trade-off between deforming the geometrical elements of the image and retaining its principal statistical features. This study also defines two taxonomies, one regarding optimization and another regarding image registration, prior to establishing the theoretical concepts behind the proposed framework. After the experiments in three different image data sources, the results indicate that the performance and quality of the nonrigid registration can be optimized by the proposed measure, and the trained surrogate model assists with the optimization process, as well as a more extensive comprehension of the behavior of the objective function in the search space.

Keywords: Image Registration. Surrogate-assisted optimization. Pattern recognition.

LISTA DE ILUSTRAÇÕES

Figura 1	– Comparação do segmento de reta formado entre dois pontos a_1 e a_2 para uma função convexa (a) e para uma função não convexa (b)	26
Figura 2	– Comparação do segmento de reta formado entre dois pontos x_1 e x_2 para um conjunto convexo (a) e para um conjunto não convexo (b)	27
Figura 3	– Solução gráfica de um problema de programação linear com duas variáveis de decisão visualizada pelas curvas de contorno da função objetivo. A área delimitada pelas restrições é representada pela região sombreada e forma o conjunto de soluções candidatas, onde a solução ótima é dada pelo ponto (x_1^*, x_2^*)	27
Figura 4	– Solução gráfica de um problema de programação quadrática com duas variáveis de decisão visualizada pelas curvas de contorno da função objetivo	28
Figura 5	– Visualização em três dimensões da função $f(x_1, x_2)$ dada pela Equação (10). Vales representados pelas regiões mais escuras	32
Figura 6	– Representação gráfica de oito iterações da busca por descida do gradiente. Vales representados pelos contornos mais escuros	32
Figura 7	– Representação gráfica da condição necessária de primeira ordem para problemas de otimização com restrições. As soluções candidatas incidem sobre o segmento de reta formado pela restrição de igualdade	33
Figura 8	– Representação gráfica da condição necessária de primeira ordem para problemas de otimização com duas restrições ativas (a) e nenhuma restrição ativa (b)	35
Figura 9	– População inicial de soluções gerada aleatoriamente	37
Figura 10	– Representação gráfica do efeito do operador de recombinação entre duas soluções na convergência da solução. O operador de recombinação exemplificado é a média entre as duas soluções	39
Figura 11	– Representação gráfica do efeito do operador de mutação de uma solução na exploração de novos subespaços. O operador de mutação exemplificado é a multiplicação por uma constante	40
Figura 12	– Representação gráfica do efeito do operador de autoadaptação para aceleração da convergência por mutação em algoritmos de estratégias evolutivas	40
Figura 13	– Probabilidade proporcional à aptidão para seleção de soluções para recombinação pelo método da roleta	41
Figura 14	– Processo de evolução do algoritmo evolutivo para duas estratégias de recombinação distintas da geração 0, 1, 5 e 10, respectivamente	42
Figura 15	– As etapas de um processo de otimização por metamodelos	44
Figura 16	– Atualização do modelo para aproximação da função objetivo no espaço de busca real	45

Figura 17	– Projeto de experimentos por fatoração de cinco níveis e duas variáveis . . .	46
Figura 18	– Exemplo de 25 amostras pelo hipercubo latim (a) e amostragem ótima pelo hipercubo latim (b)	47
Figura 19	– Visualização gráfica do polinômio dado pela Equação (24)	50
Figura 20	– Visualização gráfica da função de base radial dada pela Equação (28) . . .	51
Figura 21	– Modelo clássico de um <i>perceptron</i> para n entradas	52
Figura 22	– <i>Perceptron</i> de duas entradas e seu respectivo espaço de decisão	52
Figura 23	– Rede neural artificial com dois neurônios e duas entradas e seu respectivo espaço de decisão	53
Figura 24	– Rede neural artificial com dois neurônios, duas entradas e função de ativação linear na saída e seu respectivo espaço de decisão para uma certa configuração de pesos	54
Figura 25	– Estratégia de atualização do metamodelo pelo paradigma de exploração. No gráfico de $f(x)$, a curva contínua é a função verdadeira, a curva tracejada é a função aproximadora e a área sombreada é a incerteza desta aproximação. Já $U(x)$ representa a função utilidade e a próxima amostra x_{i+1} será aquela que maximiza $U(x)$	56
Figura 26	– Estratégia de atualização do metamodelo pelo paradigma de exploração .	57
Figura 27	– Exemplo de imagem plana	60
Figura 28	– Imagem original após transformação euclidiana	61
Figura 29	– Imagem original após transformação de similaridade	62
Figura 30	– Imagem original após transformação afim	62
Figura 31	– Imagem original após transformação de perspectiva	63
Figura 32	– Imagem original após transformação não rígida	64
Figura 33	– Visualização gráfica do espaço de busca do erro quadrático médio e . As translações ótimas t_x^* e t_y^* ocorrem nos pontos -10 e -15 , respectivamente	65
Figura 34	– Visualização gráfica do espaço de busca da correlação. As translações ótimas t_x^* e t_y^* ocorrem nos pontos -10 e -15 , respectivamente	66
Figura 35	– Histograma de uma imagem em escala de cinza	68
Figura 36	– Histograma conjunto entre duas distribuições X e Y para diferentes combinações de intensidades de elementos. Os valores das entropias são: $H(X) = 3,90$, $H(Y) = 3,90$ e $H(X,Y) = 6,91$	69
Figura 37	– Histograma conjunto entre duas distribuições idênticas. A existência da diagonal no histograma indica que todos os elementos de X correspondem com todos os elementos de Y , resultando em entropias condicionais nulas, bem como máxima informação mútua. Em outras palavras $H(X Y) = H(Y X) = 0$ e $H(X) = H(Y) = H(X,Y) = I(X,Y)$	70
Figura 38	– Visualização gráfica do espaço de busca da informação mútua. As translações ótimas t_x^* e t_y^* ocorrem nos pontos -10 e -15 , respectivamente . .	70

Figura 39	– Imagem original após transformação não rígida com duas densidades de Φ	75
Figura 40	– Quinze amostras de imagens frontais de face da <i>FEI Face Database</i>	76
Figura 41	– Marcação dos 68 pontos fiduciais das quinze amostras de imagens frontais de face da <i>FEI Face Database</i>	77
Figura 42	– Imagem de referência: média das 400 imagens frontais de face da <i>FEI Face Database</i> marcadas por 68 pontos fiduciais	77
Figura 43	– Comparação entre os pontos fiduciais da imagem de referência e quinze amostras de imagens frontais de face da <i>FEI Face Database</i>	78
Figura 44	– Exemplo de imagens de entrada para normalização espacial	79
Figura 45	– Gráfico da soma das distâncias euclidianas entre os pontos fiduciais com δ na primeira resolução como variável de decisão e a deformação ótima de acordo com o objetivo 1. O espaçamento entre pontos de controle que minimiza esta função é denotado por δ^*	79
Figura 46	– Visualização das 15 autofaces que representam 76% da variância dos dados	80
Figura 47	– Gráfico da distância euclidiana entre as componentes principais com δ na primeira resolução como variável de decisão e a deformação ótima de acordo com o objetivo 2	81
Figura 48	– Comparação entre os espaços de busca do objetivo 1 e objetivo 2 sem normalização e com normalização entre 0 e 1 com δ na primeira resolução como variável de decisão	82
Figura 49	– Gráfico para a soma entre o objetivo 1 e o objetivo 2 com δ na primeira resolução como variável de decisão e a deformação ótima de acordo com o objetivo proposto	82
Figura 50	– Gráficos da soma entre o objetivo 1 e o objetivo 2 com (a) λ como variável de decisão e (b) δ_1 na primeira resolução e δ_2 na segunda resolução como variáveis de decisão	83
Figura 51	– Construção do espaço de busca pelo modelo <i>kriging</i>	84
Figura 52	– Quinze amostras de imagens frontais de face da <i>Labeled Faces in the Wild</i>	85
Figura 53	– Quinze amostras de imagens frontais de face da base da UNIFESP	86
Figura 54	– Primeira parte das 16 amostras de imagem da <i>FEI Face Database</i> em forma de mosaico, seguindo a disposição previamente definida	87
Figura 55	– Segunda parte das 16 amostras de imagem da <i>FEI Face Database</i> em forma de mosaico, seguindo a disposição previamente definida	88
Figura 56	– Comportamento da função objetivo proposta no espaço de busca modelado por <i>kriging</i> para as 16 amostras de imagem da <i>FEI Face Database</i> . O círculo representa cada amostra para a construção do espaço de busca e o mínimo da função está representado pela estrela	89
Figura 57	– Face média da <i>Fei Face Database</i> seguindo a disposição previamente definida	90

Figura 58	– Primeira parte das 16 amostras de imagem da <i>Labelled Faces in the Wild</i> em forma de mosaico, seguindo a disposição previamente definida	91
Figura 59	– Segunda parte das 16 amostras de imagem da <i>Labelled Faces in the Wild</i> em forma de mosaico, seguindo a disposição previamente definida	92
Figura 60	– Comportamento da função objetivo proposta no espaço de busca modelado por <i>kriging</i> para as 16 amostras de imagem da <i>Labelled Faces in the Wild</i> . O círculo representa cada amostra para a construção do espaço de busca e o mínimo da função está representado pela estrela	93
Figura 61	– Face média da <i>Labelled Faces in the Wild</i> seguindo a disposição previamente definida	94
Figura 62	– Primeira parte das 16 amostras de imagem da Base de Faces de Recém Nascidos da UNIFESP em forma de mosaico, seguindo a disposição previamente definida	95
Figura 63	– Segunda parte das 16 amostras de imagem da Base de Faces de Recém Nascidos da UNIFESP em forma de mosaico, seguindo a disposição previamente definida	96
Figura 64	– Comportamento da função objetivo proposta no espaço de busca modelado por <i>kriging</i> para as 16 amostras de imagem da Base de Faces de Recém Nascidos da UNIFESP. O círculo representa cada amostra para a construção do espaço de busca e o mínimo da função está representado pela estrela	97
Figura 65	– Face média da Base de Faces de Recém Nascidos da UNIFESP seguindo a disposição previamente definida	98
Figura 66	– Visualização gráfica da função de Ackley	114
Figura 67	– Visualização gráfica da função de Branin	115
Figura 68	– Visualização gráfica da função Esfera	115
Figura 69	– Visualização gráfica da função de Rastrigin	116
Figura 70	– Visualização gráfica da função de Rosenbrock	116

SUMÁRIO

1	INTRODUÇÃO	21
1.1	OBJETIVOS	22
1.2	ORGANIZAÇÃO DO TRABALHO	23
2	FUNDAMENTOS DA OTIMIZAÇÃO MATEMÁTICA	25
2.1	INTRODUÇÃO	25
2.1.1	Otimização Sem Restrições	28
2.1.2	Otimização Com Restrições	32
2.2	BUSCA POR META-HEURÍSTICAS	35
2.2.1	Algoritmos Evolutivos	36
2.3	OTIMIZAÇÃO DE FUNÇÕES CAIXA PRETA OU CUSTOSAS	43
2.3.1	Projeto de Experimentos	45
2.3.1.1	<i>O Problema da Dimensionalidade</i>	47
2.3.2	Metamodelos e Aprendizado de Máquina	47
2.3.2.1	<i>Regressão polinomial</i>	49
2.3.2.2	<i>Funções de Base Radial</i>	50
2.3.2.3	<i>Redes Neurais Artificiais</i>	51
2.3.2.4	<i>Kriging</i>	54
2.3.3	Atualização do Metamodelo	55
2.3.3.1	<i>Exploração</i>	55
2.3.3.2	<i>Exploração</i>	57
2.4	CONSIDERAÇÕES FINAIS E TAXONOMIA DE OTIMIZAÇÃO ADOTADA	58
3	NORMALIZAÇÃO ESPACIAL DE IMAGENS	59
3.1	INTRODUÇÃO	59
3.2	MODELOS DE TRANSFORMAÇÃO	59
3.2.1	Modelos de Transformação Global	60
3.2.1.1	<i>Euclidiana</i>	61
3.2.1.2	<i>Similaridade</i>	61
3.2.1.3	<i>Afim</i>	62
3.2.1.4	<i>Perspectiva</i>	63
3.2.2	Modelos de Transformação Local	63
3.3	MÉTRICAS DE SIMILARIDADE	64
3.3.1	Métodos Baseados nos Elementos da Imagem	65
3.3.1.1	<i>Minimização do Erro Quadrático Médio</i>	65
3.3.1.2	<i>Maximização da Correlação</i>	66
3.3.1.3	<i>Maximização da Informação Mútua</i>	67
3.3.2	Métodos Baseados nos Pontos Fiduciais da Imagem	70

3.4	CONSIDERAÇÕES FINAIS E TAXONOMIA DE NORMALIZAÇÃO ESPACIAL DE IMAGENS ADOTADA	71
4	NORMALIZAÇÃO ESPACIAL NÃO RÍGIDA DE IMAGENS ASSISTIDA POR METAMODELOS	73
4.1	<i>FREE-FORM DEFORMATION</i>	73
4.1.1	Implementação do <i>Free-Form-Deformation</i>	76
4.2	METODOLOGIA E FUNÇÃO OBJETIVO PROPOSTA PARA OTIMIZAÇÃO DE HIPERPARÂMETROS	78
4.2.1	Objetivo 1: soma das distâncias euclidianas entre os pontos fiduciais	79
4.2.2	Objetivo 2: distância euclidiana entre as componentes principais	80
4.2.3	Regularização e composição dos objetivos	81
4.3	OTIMIZAÇÃO ASSISTIDA POR METAMODELOS	83
5	RESULTADOS E DISCUSSÃO	85
5.1	<i>FEI FACE DATABASE</i>	87
5.2	<i>LABELLED FACES IN THE WILD</i>	91
5.3	BASE DE FACES DE RECÉM NASCIDOS DA UNIFESP	95
5.4	DISCUSSÃO DOS RESULTADOS	99
6	CONCLUSÃO	101
	REFERÊNCIAS	103
	APÊNDICE A – Funções para Teste de Algoritmos de Otimização sem Restrições	113

1 INTRODUÇÃO

Maximizar ou minimizar uma função é uma tarefa constantemente encontrada em diversas áreas da engenharia e sua aplicação depende fortemente dos recursos computacionais disponíveis e da sofisticação de algoritmos capazes de convergir com eficiência.

Inicialmente, as estratégias para otimização eram essencialmente fundamentadas em propriedades matemáticas que permitiam que a melhor solução pudesse ser encontrada (KUHN; TUCKER, 1951). Somadas às limitações computacionais da época em que estas técnicas foram pioneiramente concebidas, a aplicação cotidiana de problemas de otimização foi, por muito tempo, restrita às funções que atendessem determinadas particularidades matemáticas (DANTZIG, 1951) ou a grupos que possuíssem as máquinas com potencial de efetuar as operações e cálculos inerentes às tarefas de otimização conhecidas na época.

O investimento no aprimoramento e inovação dos algoritmos de otimização para encontrar alternativas aos problemas enfrentados pelas estratégias tradicionais, bem como o rápido crescimento da capacidade computacional, possibilitaram expandir as aplicações destas técnicas para resolver novos problemas de otimização por meio de heurísticas sofisticadas (NELDER; MEAD, 1965; HOLLAND, 1975).

Como os algoritmos de otimização são processos iterativos, é comum estas tarefas precisarem avaliar a função objetivo e suas derivadas múltiplas vezes. Em casos em que a função objetivo é custosa, ou seja, dependem da avaliação de experimentos físicos ou simulações em computadores, certos algoritmos que necessitam um elevado número de iterações tornam-se inviáveis.

O problema do custo da avaliação da função objetivo pode ser resolvido ao construir um modelo simplificado, ou metamodelo, no espaço de busca. O metamodelo permite calcular o valor esperado para regiões do espaço de busca que ainda não foram exploradas, evitando a avaliação excessiva da função objetivo verdadeira em troca de imprecisões causadas pela simplificação (QUEIPO et al., 2005).

Em princípio, esta técnica conquistou grande sucesso nas áreas que dependiam de simulações computacionais, como Dinâmica dos Fluidos Computacional (CFD¹) e Método dos Elementos Finitos (FEM²), e é conhecida na literatura por Otimização Assistida por Metamodelos (SAO³) (FORRESTER; KEANE, 2009).

No entanto, com o surgimento de novos métodos de inteligência artificial, particularmente de aprendizado de máquina, a otimização assistida por metamodelos encontrou uma nova aplicação em um campo de estudo que emerge rapidamente: a otimização dos hiperparâmetros de classificadores (BERGSTRA et al., 2011).

Visto que a escolha correta dos hiperparâmetros pode impactar significativamente o desempenho da tarefa de classificação, e o treinamento de um único modelo classificador pode

¹ *Computational Fluid Dynamics.*

² *Finite Element Method.*

³ *Surrogated-Assisted Optimization.*

já requerer um consumo considerável de recursos computacionais, o uso de técnicas que estimam o desempenho do classificador em função dos hiperparâmetros escolhidos torna-se bastante razoável. Recentemente esta proposta foi abordada por trabalhos como Eggensperger et al. (2015), Feurer, Springenberg e Hutter (2015), Rosales-Pérez et al. (2015) e Z. Wang et al. (2016).

Outra aplicação que envolve a avaliação do desempenho em função de parâmetros escolhidos empiricamente e que se beneficiaria do uso de modelos aproximados é a normalização não-rígida de imagens (RUECKERT et al., 1999). Neste caso, a função objetivo é a qualidade da normalização da imagem e as variáveis de decisão são os parâmetros que devem ser escolhidos para construir o operador de transformação.

Calcular o operador de transformação e aplicá-lo para normalizar a imagem desejada já é uma tarefa computacionalmente intensa por si só e envolve uma rotina de otimização para maximizar (ou minimizar) uma métrica de similaridade entre a imagem de referência e a imagem de interesse (KLEIN; STARING; PLUIM, 2007). Adicionalmente, dependendo dos parâmetros escolhidos, o resultado da normalização pode não ser satisfatório, sendo necessária a correção dos parâmetros e recálculo da transformação correspondente.

Auxiliar o processo de normalização espacial não-rígida de imagens por meio de modelos aproximados da relação entre desempenho da normalização e parâmetros da transformação continua um conceito ainda inexplorado na literatura, porém bastante promissor uma vez que:

- a) encontrar os parâmetros ótimos da transformação na primeira tentativa requer um grande conhecimento sobre o domínio a que as imagens pertencem, bem como o efeito que cada parâmetro tem sobre a transformação;
- b) após a construção de modelos aproximados, tem-se um modelo sintético no espaço das variáveis de decisão, que é muito conveniente para entender a fundo como as variáveis de decisão estão relacionadas e como elas interagem com a função objetivo.

1.1 OBJETIVOS

Levando-se em conta as considerações citadas acima, os objetivos principais deste trabalho são:

- a) auxiliar o processo de normalização espacial não rígido de imagens pela modelagem aproximada entre o desempenho da normalização e os parâmetros da transformação para que os parâmetros ótimos sejam encontrados na primeira tentativa;
- b) elaborar uma métrica de similaridade para avaliar o desempenho da normalização espacial não rígida de modo que esta métrica possa ser utilizada como função objetivo de um problema de otimização;

- c) construir duas taxonomias, uma de otimização e outra de normalização espacial de imagens, para fundamentar e contextualizar os princípios da metodologia proposta;
- d) desenvolver uma ferramenta computacional para normalização espacial não rígida assistida por metamodelos e disponibiliza-la em código aberto.

1.2 ORGANIZAÇÃO DO TRABALHO

Os cinco próximos capítulos deste trabalho estão divididos da seguinte forma. O capítulo 2 apresenta os fundamentos da otimização matemática e as principais abordagens para este tipo de problema e constrói uma taxonomia sobre otimização. O capítulo 3 descreve os tipos de modelos de transformação e métricas de similaridade e une estes dois conceitos em uma taxonomia sobre normalização espacial de imagens. O capítulo 4 desenvolve a metodologia proposta neste trabalho, denominada de Normalização Espacial Não Rígida de Imagens Assistida por Metamodelos, que inclui a elaboração da métrica de similaridade para avaliar o desempenho da normalização espacial não rígida. O capítulo 5 apresenta e discute os resultados dos experimentos da metodologia proposta em três bases de imagens distintas. Finalmente, o capítulo 6 encerra o trabalho com as considerações finais e trabalhos futuros.

2 FUNDAMENTOS DA OTIMIZAÇÃO MATEMÁTICA

Este capítulo propõe uma taxonomia baseada na literatura sobre os problemas de otimização matemática que são relevantes para o desenvolvimento dos próximos capítulos. Esta taxonomia apresenta apenas o formalismo matemático necessário para visualizar cada problema de otimização. A seção 2.1 introduz o problema geral de otimização e as técnicas mais conhecidas para encontrar a solução destes problemas. A seção 2.2 descreve a busca por meta-heurísticas, algoritmos evolutivos e suas aplicações em problemas de otimização global. A seção 2.3 apresenta as abordagens adotadas em problemas de otimização de funções custosas ou caixa preta, ou seja, problemas em que existe uma função custo inacessível e que precisa ser estimada por técnicas estatísticas ou de inteligência artificial. Finalmente, a seção 2.4 apresenta a taxonomia descrita neste capítulo.

2.1 INTRODUÇÃO

Otimização matemática, referida neste trabalho como apenas otimização, é a técnica que busca encontrar a solução que minimiza (ou maximiza) uma função objetivo $f(x)$ através das variáveis de decisão $x = (x_1, x_2, \dots, x_n)$, sujeita a l funções de restrição de igualdade $h_k(x)$ e m funções de restrição de desigualdade $g_j(x)$ (BOYD; VANDENBERGHE, 2004).

O projeto de algoritmos que solucionam os problemas de otimização são chamados de programação matemática e a forma padrão para escrevê-los é (VENKATARAMAN, 2009):

$$\begin{aligned}
 & \text{minimize} && f(x_1, x_2, \dots, x_n) \\
 & \text{sujeito a:} && h_k(x_1, x_2, \dots, x_n) = 0, && k = 1, 2, \dots, l \\
 & && g_j(x_1, x_2, \dots, x_n) \leq 0, && j = 1, 2, \dots, m \\
 & && x_i^{inf} \leq x_i \leq x_i^{sup} && i = 1, 2, \dots, n,
 \end{aligned} \tag{1}$$

onde x_i^{inf} é o limite inferior e x_i^{sup} o limite superior para a i -ésima variável de decisão.

Na Equação (1), a desigualdade $l \leq n$ deve ser respeitada para existir solução, ou seja, a dimensão do espaço de decisão deve ser maior ou igual ao número de restrições de igualdade. No caso em que $l = n$, a dimensão do espaço de decisão é igual ao número de restrições de igualdade, o que torna o problema em uma resolução de sistema de equações, dispensando o uso da programação matemática.

O algoritmo escolhido para solucionar um problema de otimização depende exclusivamente da natureza das funções $f(x)$, $h(x)$ e $g(x)$. Por exemplo, se essas funções forem lineares, o algoritmo *simplex* (DANTZIG, 2016) é extremamente eficiente para encontrar a solução ótima. Por outro lado, se a função objetivo $f(x)$ for quadrática e as restrições $h(x)$ e $g(x)$ forem lineares, o método dos pontos interiores demonstra-se um algoritmo de rápida convergência (NESTEROV; NEMIROVSKII, 1994). A solução por algoritmos para estes problemas é denominada de programação linear ou programação quadrática, respectivamente.

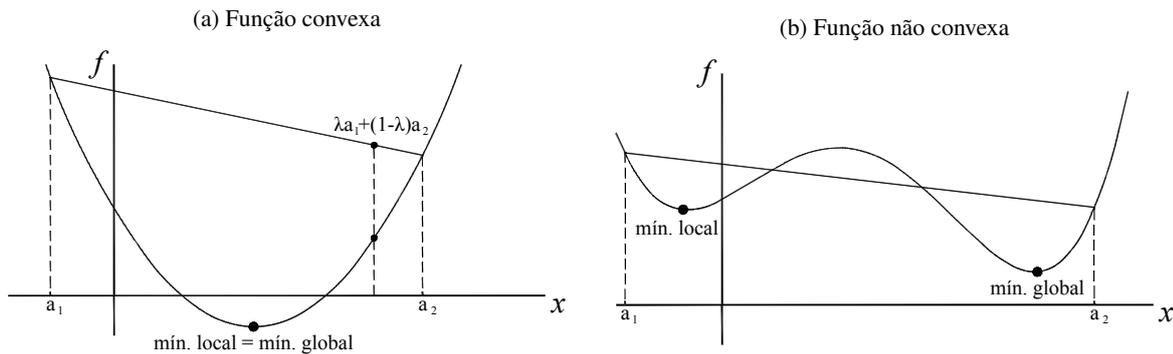
Nas situações em que a função objetivo for convexa e o espaço de soluções candidatas for composto por um conjunto convexo, o problema é classificado como otimização convexa (BOYD; VANDENBERGHE, 2004). Esta particularidade garante que qualquer mínimo local também seja o mínimo global, o que torna o processo computacional por algoritmos muito mais eficiente (BAZARAA; SHERALI; SHETTY, 2013). Analogamente, nas situações em que a função objetivo for côncava, o máximo local também será o máximo global.

Definição 1 – Função Convexa (BAZARAA; SHERALI; SHETTY, 2013). Uma função f é chamada convexa sobre o domínio \mathbb{R} se o valor da função entre quaisquer dois pontos for menor ou igual ao valor em qualquer ponto do segmento de reta formado entre eles (Figura 1). Em outras palavras, se a_1 e a_2 pertencem ao domínio da função f , então

$$f(\lambda a_1 + (1 - \lambda)a_2) \leq \lambda \cdot f(a_1) + (1 - \lambda) \cdot f(a_2) \quad (2)$$

para todo $\lambda \in [0,1]$.

Figura 1 – Comparação do segmento de reta formado entre dois pontos a_1 e a_2 para uma função convexa (a) e para uma função não convexa (b)



Fonte: Autor "adaptado de" Bazaraa; Sherali; Shetty, 2013, p. 99

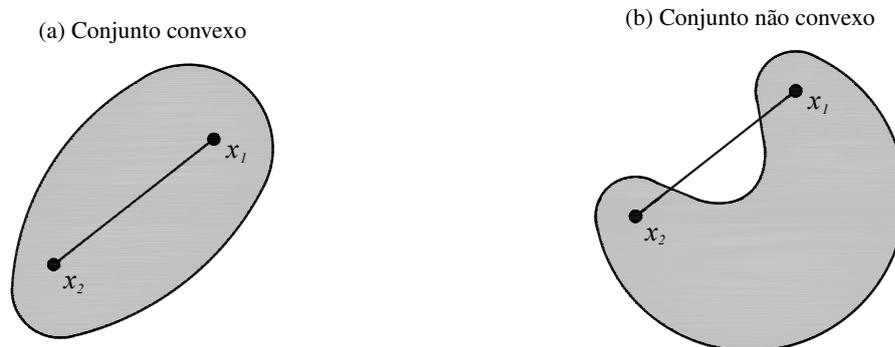
Definição 2 – Conjunto Convexo (BAZARAA; SHERALI; SHETTY, 2013). Um conjunto S é chamado convexo sobre o domínio \mathbb{R} se o segmento de reta que liga quaisquer dois pontos contidos no conjunto também pertence ao conjunto (Figura 2). Em outras palavras, se a_1 e a_2 pertencem ao conjunto S , então

$$a_n = \lambda a_1 + (1 - \lambda)a_2 \quad (3)$$

deve também pertencer ao conjunto S para todo $\lambda \in [0,1]$.

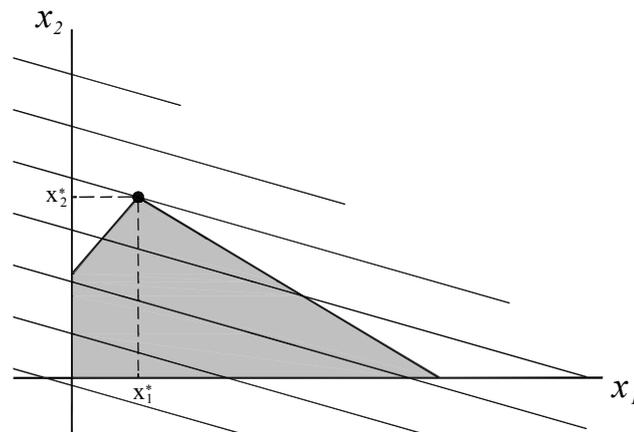
Por definição, uma função linear é convexa e côncava ao mesmo tempo, e o conjunto de soluções candidatas formado por restrições lineares sempre será convexo (Carlos A. C. COELLO; LAMONT; VAN VELDHUIZEN et al., 2007). Portanto, em problemas de programação linear, o mínimo local será o mínimo global. A Figura 3 mostra a representação gráfica da solução de um problema de programação linear.

Figura 2 – Comparação do segmento de reta formado entre dois pontos x_1 e x_2 para um conjunto convexo (a) e para um conjunto não convexo (b)



Fonte: Autor "adaptado de" Bazaraa; Sherali; Shetty, 2013, p. 41

Figura 3 – Solução gráfica de um problema de programação linear com duas variáveis de decisão visualizada pelas curvas de contorno da função objetivo. A área delimitada pelas restrições é representada pela região sombreada e forma o conjunto de soluções candidatas, onde a solução ótima é dada pelo ponto (x_1^*, x_2^*)



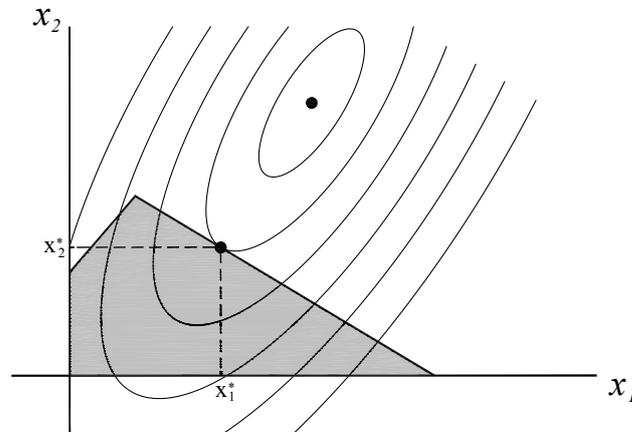
Fonte: Autor "adaptado de" Bazaraa; Sherali; Shetty, 2013, p. 132

Quanto à programação quadrática, uma função quadrática, na forma matricial $Q(x) = \frac{1}{2}x^T \mathbf{A}x + c^T x$, só será convexa se \mathbf{A} for positiva definida¹ (BURER; LETCHFORD, 2009). Satisfeita esta condição, o mínimo local será o mínimo global. A Figura 4 mostra a representação gráfica da solução de um problema de programação quadrática.

Outras particularidades em problemas de otimização incluem: funções-objetivo e restrições não lineares (programação não linear) (BAZARAA; SHERALI; SHETTY, 2013), domínio discreto ou grafos (programação inteira e programação dinâmica) (BELLMAN, 1956; DIJKSTRA, 1959; LAWLER; WOOD, 1966), múltiplos objetivos (Carlos A. C. COELLO; LAMONT; VAN VELDHUIZEN et al., 2007).

¹Uma matriz \mathbf{A} será positiva definida se todos os autovalores λ_i de \mathbf{A} forem positivos.

Figura 4 – Solução gráfica de um problema de programação quadrática com duas variáveis de decisão visualizada pelas curvas de contorno da função objetivo



Fonte: Autor "adaptado de" Bazaraa; Sherali; Shetty, 2013, p. 132

Adicionalmente, os problemas de otimização podem ser divididos em dois tipos: sem restrições e com restrições. Cada tipo possui uma série de técnicas utilizadas para direcionar o caminho de busca através de condições de otimalidade. Estas técnicas podem ser de ordem zero (não utilizam informação do gradiente), 1ª ordem (utilizam o gradiente) e 2ª ordem (utilizam a matriz hessiana) (VENKATARAMAN, 2009).

2.1.1 Otimização Sem Restrições

Em otimização sem restrições, a forma geral, descrita na Equação (1), pode ser simplificada para

$$\text{minimize } f(x_1, x_2, \dots, x_n) \quad (4)$$

Os pontos estacionários (mínimo, máximo e sela) da função ocorrem quando os hiperplanos tangentes a esses pontos forem paralelos ao hiperplano das variáveis (VENKATARAMAN, 2009). Isto quer dizer que a solução ótima, representada por $x^* = (x_1^*, x_2^*, \dots, x_n^*)$, pode ser detectada quando as variações diferenciais dx_1, dx_2, \dots, dx_n das variáveis de decisão x sobre este hiperplano não mudam a função objetivo $f(x)$. Nesta situação, afirma-se que $df = 0$. Portanto:

$$df = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 + \dots + \frac{\partial f}{\partial x_n} dx_n = 0. \quad (5)$$

Para garantir que a Equação (5) seja válida para qualquer mudança de dx_1, dx_2, \dots, dx_n , as derivadas parciais em x^* devem ser todas iguais a zero (VENKATARAMAN, 2009).

Definição 3 – Gradiente de uma Função (VENKATARAMAN, 2009). O gradiente é um vetor das derivadas parciais da função f , e em qualquer ponto representa a direção de maior

variação de f .

$$\nabla f = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \cdots \quad \frac{\partial f}{\partial x_n} \right]^T. \quad (6)$$

Por meio da Definição 3, a condição necessária de 1ª ordem para a solução ser ótima é

$$\nabla f(x^*) = 0. \quad (7)$$

Como já foi mostrado, o mínimo local de uma função convexa é também o mínimo global. Isto quer dizer que, para funções convexas, a condição necessária de 1ª ordem também será suficiente para a solução ser ótima (mínimo global).

No entanto, em funções não convexas, a Equação (7) é válida para qualquer ponto estacionário da função e não garante que a solução encontrada seja o mínimo global.

Os pontos estacionários podem ser definidos através do sinal da derivada segunda naquele ponto. Para funções com mais de uma variável, as derivadas de segunda ordem podem ser representadas pela matriz hessiana.

Definição 4 – Hessiano (VENKATARAMAN, 2009). A matriz hessiana \mathbf{H} é composta pelas derivadas de segunda ordem da função f , ou seja,

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \quad (8)$$

Calculando a matriz \mathbf{H} nos pontos estacionários encontrados pela condição necessária de 1ª ordem, podem existir três situações:

- se $H(x^*)$ for positiva definida, então x^* é um ponto de mínimo;
- se $H(x^*)$ for negativa definida, então x^* é um ponto de máximo;
- se $H(x^*)$ for indefinida, então x^* é um ponto de sela.

Finalmente, a condição suficiente de 2ª ordem para a solução ser ótima (mínimo) é

$$H(x^*) > 0. \quad (9)$$

Definidas as condições de otimalidade, algoritmos que verificam estas condições iterativamente podem ser implementados computacionalmente. Estes algoritmos podem ser de ordem zero (não utilizam informação do gradiente), 1ª ordem (utilizam o gradiente) e 2ª ordem (utilizam o hessiano). Porém, devido à complexidade computacional de calcular o hessiano, a verificação da condição suficiente de 2ª ordem geralmente é evitada em implementações computacionais.

O caso mais simples de otimização sem restrições é com uma variável de decisão. Para este tipo de problema, estima-se um chute inicial x_0 . Em seguida, calcula-se um Δx que é quanto a variável x_n será incrementada para chegar mais próxima da solução, ou seja $x_{n+1} \leftarrow x_n + \Delta x$. Se a condição de 1ª ordem $df/dx = 0$ for atendida, então o algoritmo convergiu para uma possível solução. Caso contrário, o algoritmo calcula um novo Δx e repete o procedimento até que $df/dx = 0$. O Algoritmo 1 mostra os passos básicos para otimização sem restrições de uma variável.

Algoritmo 1 – Algoritmo para otimização sem restrições de uma variável (VENKATARAMAN, 2009)

```

1 Entrada:  $x_0$ 
2 enquanto  $df/dx \neq 0$  faça
3   |   Calcular  $\Delta x$ 
4   |    $x_{n+1} \leftarrow x_n + \Delta x$ 
5 fim

```

A diferença nos diversos algoritmos de otimização sem restrições de uma variável está em como Δx é calculado. Por exemplo, no método de Newton-Raphson (1ª ordem) (ATKINSON, 2008), $\Delta x = -f'(x)^{-1} \cdot f(x)$. Outros algoritmos para otimização sem restrições de uma variável são: Método da Bisseção (1ª ordem) (ATKINSON, 2008), Método de Fibonacci (ordem zero) (BAZARAA; SHERALI; SHETTY, 2013) e Método da Seção Áurea (ordem zero) (KIEFER, 1953).

Apesar de serem restritas às aplicações no mundo real, os métodos de otimização de uma variável são fundamentais para muitos algoritmos de otimização multivariável que utilizam direção de busca (VENKATARAMAN, 2009), pois ao se definir uma direção, o problema se torna unidimensional.

Os algoritmos de otimização multivariável são inicializados por um chute inicial x_0 . Dessa vez, calcula-se um vetor de direção de busca S_n no espaço de soluções. O incremento à solução é dado por $\alpha_n S_n$, onde α_n é a amplitude do vetor S_n . Esta amplitude pode ser encontrada pela minimização ao longo da reta formada pelo vetor S_n utilizando os algoritmos de uma variável, mas também pode-se adotar uma constante c (taxa de aprendizado), como implementado no algoritmo para otimização das redes neurais artificiais (RUMELHART; HINTON; WILLIAMS, 1986).

Nos espaços de soluções de alta dimensionalidade, a verificação exata da condição $\nabla f = 0$ para a solução ótima pode levar inúmeras repetições do algoritmo. Portanto, os algoritmos para múltiplas variáveis também verificam a convergência para um ponto estacionário através da variação entre duas iterações da função objetivo $|\Delta f|$ ou da solução $\Delta x^T \Delta x$ (VENKATARAMAN, 2009). Se estas variações estiverem dentro das tolerâncias ε_1 e ε_2 especificadas, o algoritmo retorna a solução encontrada. É importante também especificar o limite de iterações N que o algoritmo realiza para que o tempo computacional seja viável em problemas

de difícil convergência. O Algoritmo 2 mostra os passos básicos para otimização sem restrições com múltiplas variáveis.

Algoritmo 2 – Algoritmo para otimização sem restrições com múltiplas variáveis (VENKATARAMAN, 2009)

```

1 Entrada:  $x_0$ 
2 para cada iteração  $n$  faça
3   Encontrar o vetor de direção de busca  $S_n$ 
4    $\alpha_n \leftarrow \min f(x_n)$  ou  $c$ 
5    $x_{n+1} \leftarrow x_n + \alpha_n S_n$ 
6   se  $\nabla f = 0$ 
7   ou  $|\Delta f| \leq \varepsilon_1$ 
8   ou  $\Delta x^T \Delta x \leq \varepsilon_2$ 
9   ou  $n + 1 \leq N$  então
10  | pare
11  | fim
12 fim

```

O maior desafio em desenvolver algoritmos de otimização multivariável está em encontrar o vetor de direção de busca S_n que converge para a solução ótima no menor número de iterações possível. Por exemplo, no caso de problemas de minimização, o algoritmo descida do gradiente (BAZARAA; SHERALI; SHETTY, 2013) adota como vetor de direção de busca o gradiente negativo da função $-\nabla f$, pois ele indica localmente a direção de maior variação negativa da função.

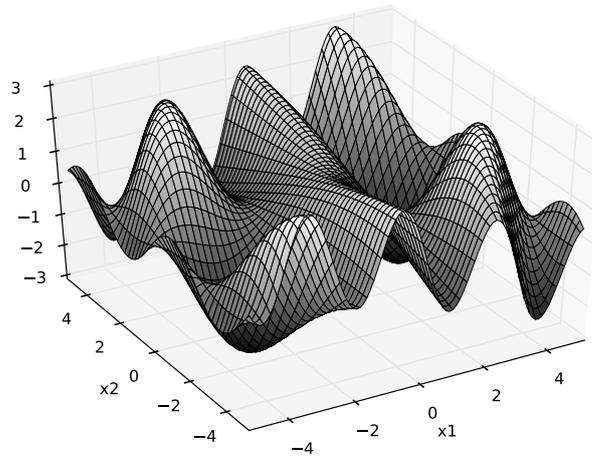
Utilizando como exemplo a função objetivo $f(x_1, x_2) = 3 \operatorname{sen}(0,5 + 0,25x_1x_2) \cos(x_1)$ (VENKATARAMAN, 2009), visualizada em três dimensões na Figura 5, o problema de otimização agora pode ser formulado como:

$$\begin{aligned} \text{minimize } & f(x_1, x_2) = 3 \cdot \operatorname{sen}(0,5 + 0,25x_1x_2) \cdot \cos(x_1) \\ & -5 \leq x_1 \leq 5; \quad -5 \leq x_2 \leq 5. \end{aligned} \quad (10)$$

Cada iteração da busca por descida do gradiente para Equação (10) pode ser visualizada na Figura 6 através do gráfico de curvas de contorno, onde a amplitude do vetor de direção de busca S_n é uma constante igual a 0,33.

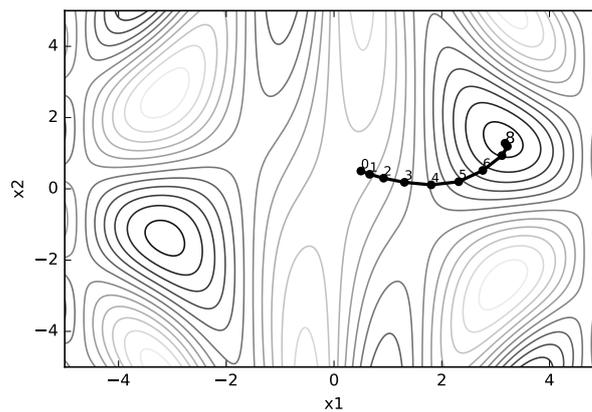
Outros algoritmos que determinam esta direção podem ser de ordem zero, como Método de Rosenbrock (ROSENBROCK, 1960), Método de Hooke-Jeeves (HOOKE; JEEVES, 1961), Método de Powell (POWELL, 1964) e Método de Nelder-Mead (NELDER; MEAD, 1965), 1ª ordem, como Gradiente Conjugado (HESTENES; STIEFEL, 1952; FLETCHER; POWELL, 1963), Método de David-Fletcher-Powell (FLETCHER; POWELL, 1963) e BFGS (Broyden-Fletcher-Goldfarb-Shanno) (BROYDEN, 1970) e de 2ª ordem como o Método Modificado de Newton (BAZARAA; SHERALI; SHETTY, 2013).

Figura 5 – Visualização em três dimensões da função $f(x_1, x_2)$ dada pela Equação (10). Vales representados pelas regiões mais escuras



Fonte: Autor

Figura 6 – Representação gráfica de oito iterações da busca por descida do gradiente. Vales representados pelos contornos mais escuros



Fonte: Autor

2.1.2 Otimização Com Restrições

Na presença de restrições no problema de otimização, a condição necessária de otimalidade $\nabla f = 0$ não pode mais garantir uma solução ótima, pois agora o conjunto de soluções candidatas estará delimitado pelas restrições e o ponto onde o gradiente da função é zero pode estar fora deste conjunto.

Considerando inicialmente apenas restrições de igualdade, o método de Lagrange transforma as restrições para um problema de otimização sem restrições através da função de Lagrange F , que é a soma da função objetivo original f com a combinação linear das restrições h , sendo seus coeficientes os multiplicadores de Lagrange λ (VENKATARAMAN, 2009).

Com o método de Lagrange, a forma geral, descrita na Equação (1), pode ser reformulada para

$$\begin{aligned} \text{minimize } F(x_1, \dots, x_n, \lambda_1, \dots, \lambda_l) &= f(x_1, \dots, x_n) + \\ &\sum_{k=1}^l \lambda_k h_k(x_1, \dots, x_n) \\ x_i^{inf} \leq x_i \leq x_i^{sup} &\quad i = 1, 2, \dots, n. \end{aligned} \quad (11)$$

Ao verificar a condição necessária de otimalidade no lagrangeano ($\nabla F = 0$), tem-se que

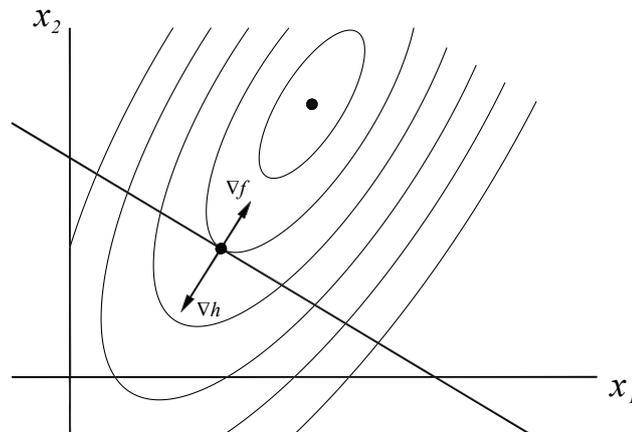
$$\begin{aligned} \frac{\partial F}{\partial x_i} &= \frac{\partial f}{\partial x_i} + \sum_{k=1}^l \lambda_k \frac{\partial h_k}{\partial x_i} = 0 & i = 1, 2, \dots, n \\ \frac{\partial F}{\partial \lambda_k} &= h_k = 0 & k = 1, 2, \dots, l. \end{aligned} \quad (12)$$

Simplificando a Equação (12), a condição necessária de 1ª ordem agora é definida por

$$\begin{aligned} \nabla F &= \nabla f + \sum_{k=1}^l \lambda_k \nabla h_k = 0 \\ h_k &= 0 \quad k = 1, 2, \dots, l. \end{aligned} \quad (13)$$

A partir da Equação (13), pode-se afirmar que a solução ótima para o problema incidirá no ponto onde o gradiente da função objetivo é igual à soma dos gradientes das restrições multiplicados por $-\lambda$. Considerando um problema com duas variáveis de decisão e uma restrição de igualdade, esta afirmação pode ser visualizada na Figura 7.

Figura 7 – Representação gráfica da condição necessária de primeira ordem para problemas de otimização com restrições. As soluções candidatas incidem sobre o segmento de reta formado pela restrição de igualdade



Fonte: Autor "adaptado de" Bazaraa; Sherali; Shetty, 2013, p. 132

Para lidar com as restrições de desigualdade, uma variável *slack* z_j^2 é introduzida no problema (VENKATARAMAN, 2009). O papel desta variável é transformar as restrições de desigualdade $g_j(x) \leq 0$ em restrições de igualdade $g_j(x) + z_j^2 = 0$ para então poder apli-

car o método de Lagrange descrito anteriormente, com os multiplicadores de Lagrange agora denominados por β .

Considerando apenas as restrições de desigualdade, a forma geral pode ser reformulada para

$$\begin{aligned} \text{minimize } F(x_1, \dots, x_n, z_1, \dots, z_m, \beta_1, \dots, \beta_m) &= f(x_1, \dots, x_n) + \\ &\sum_{j=1}^m \beta_j [g_j(x_1, \dots, x_n) + z_j^2] \quad (14) \\ x_i^{inf} \leq x_i \leq x_i^{sup} \quad &i = 1, 2, \dots, n. \end{aligned}$$

As condições necessárias de 1ª ordem ($\nabla F = 0$) são

$$\begin{aligned} \frac{\partial F}{\partial x_i} &= \frac{\partial f}{\partial x_i} + \sum_{j=1}^m \beta_j \frac{\partial g_j}{\partial x_i} = 0 \quad i = 1, 2, \dots, n \\ \frac{\partial F}{\partial z_j} &= 2\beta_j z_j = 0 \quad j = 1, 2, \dots, m \\ \frac{\partial F}{\partial \beta_j} &= g_j + z_j^2 = 0. \end{aligned} \quad (15)$$

Da Equação (15), é possível deduzir (VENKATARAMAN, 2009)

$$0 = 2\beta_j z_j = \beta_j z_j^2 = \beta_j (-g_j) = \beta_j g_j = 0. \quad (16)$$

Portanto, é possível eliminar as variáveis *slack* e reduzir as condições necessárias de 1ª ordem da Equação (15) para

$$\begin{aligned} \nabla F &= \nabla f + \sum_{j=1}^m \beta_j \nabla g_j = 0 \\ \beta_j g_j &= 0 \quad j = 1, 2, \dots, m. \end{aligned} \quad (17)$$

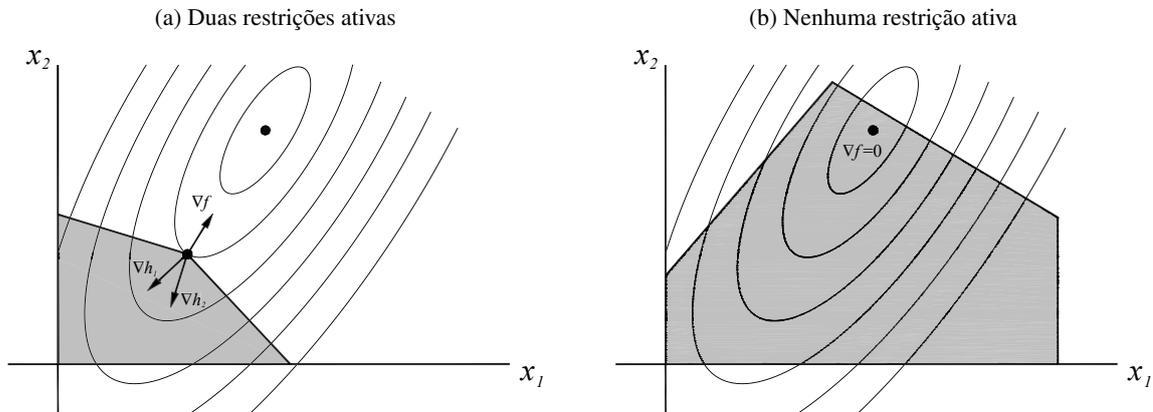
A condição $\beta_j g_j = 0$ da Equação (17) mostra que se $\beta_j \neq 0$ então $g_j = 0$, ou seja, a restrição g_j está ativa, se não, está inativa. Portanto, existem 2^m combinações possíveis de ativação das restrições de desigualdade, onde m é o número total de restrições de desigualdade do problema.

Por exemplo, para $m = 2$, as combinações possíveis de ativação das restrições de desigualdade podem ser (VENKATARAMAN, 2009):

- a) $\beta_1 = 0$ (g_1 inativa); $\beta_2 = 0$ (g_2 inativa);
- b) $\beta_1 = 0$ (g_1 inativa); $\beta_2 \neq 0$ (g_2 ativa);
- c) $\beta_1 \neq 0$ (g_1 ativa); $\beta_2 = 0$ (g_2 inativa);
- d) $\beta_1 \neq 0$ (g_1 ativa); $\beta_2 \neq 0$ (g_2 ativa).

A Figura 8 mostra duas visualizações gráficas da solução de um problema de otimização com duas variáveis de decisão. O primeiro (a) com duas restrições de desigualdade ativas e o segundo (b) com nenhuma restrição de desigualdade ativa.

Figura 8 – Representação gráfica da condição necessária de primeira ordem para problemas de otimização com duas restrições ativas (a) e nenhuma restrição ativa (b)



Fonte: Autor "adaptado de" Bazaraa; Sherali; Shetty, 2013, p. 132

Finalmente, as condições necessárias de 1ª ordem para o problema geral de otimização, conhecidas também como condições gerais de Kuhn Tucker (KUHN; TUCKER, 1951) são

$$\begin{aligned} \nabla F &= \nabla f + \sum_{k=1}^l \lambda_k \nabla h_k + \sum_{j=1}^m \beta_j \nabla g_j = 0 \\ h_k &= 0 \quad k = 1, 2, \dots, l \\ \beta_j g_j &= 0 \quad j = 1, 2, \dots, m. \end{aligned} \quad (18)$$

Um método computacional para solucionar problemas de otimização com restrições é transformar o problema original em um problema sem restrições através da adição de um termo na função objetivo que penaliza ou previne qualquer violação das restrições (FIACCO; MCCORMICK, 1990; POWELL, 1967). Em seguida, utilizam-se os algoritmos de otimização sem restrições.

Outro método é, a partir de uma solução candidata, buscar uma nova solução dentro das restrições e que seja melhor que a anterior (ZOUTENDIJK, 1960). Um algoritmo que implementa este método é o SQP (*Sequential Quadratic Programming*), o qual realiza o processo de otimização através de aproximações quadráticas do lagrangeano sujeito a aproximações lineares das restrições (BAZARAA; SHERALI; SHETTY, 2013; R. B. WILSON, 1963).

2.2 BUSCA POR META-HEURÍSTICAS

As abordagens de otimização citadas na seção 2.1 possuem uma extensa formulação matemática para garantir que, a partir de uma solução inicial, os algoritmos possam convergir para uma solução que respeite as condições de otimalidade. Para guiar o caminho de busca, os algoritmos tradicionais determinam um vetor de direção baseado no gradiente. No entanto, além de não haver nenhuma garantia de convergência para um mínimo global, o cálculo do gradiente falhará nas regiões de descontinuidades da função objetivo.

A partir disso, surgiram os métodos heurísticos, como o Método de Nelder-Mead (NELDER; MEAD, 1965), que trabalham com suposições teóricas menos estritas para determinar a direção de busca e pressupõem que, para encontrar um mínimo global, os algoritmos não devem apenas seguir as direções de busca que minimizam a função mais rapidamente, mas também estabelecer uma relação de exploração do espaço de busca, que pode ser estocástica ou baseada em informações passadas. Ao contrário de seguir sempre a direção que minimiza a função objetivo, esta exploração procura levar o processo de busca a um subespaço ainda inexplorado com a expectativa (portanto, sem garantia teórica) de que uma solução melhor que a anterior ainda possa ser encontrada.

Dentro do contexto de métodos heurísticos, um novo paradigma para a solução de problemas de otimização pode ser descrito: os métodos meta-heurísticos (GLOVER, 1986). Os métodos meta-heurísticos diferem-se dos métodos heurísticos, pois os elementos de um algoritmo meta-heurístico são descritos por uma metáfora, muitas vezes inspiradas na natureza (bioinspiradas) (YANG, 2010), mas é o projetista quem deve abstrair esta metáfora de acordo com o tipo de problema (não linear, grafo, etc), determinar os parâmetros e implementá-los. Portanto, dois algoritmos inspirados na mesma metáfora podem divergir em algumas operações por conta do leque de interpretações possíveis, dificultando a classificação destes algoritmos.

A grande maioria dos algoritmos meta-heurísticos são baseados em populações de soluções, ou seja, o processo de busca agora é dado a partir da interação entre uma população de soluções candidatas, ao contrário dos métodos que utilizam uma única solução. A grande vantagem é que a exploração de novos subespaços de soluções pode ser feita simultaneamente e com compartilhamento de informação, permitindo que a busca seja dividida em subpopulações e implementada em máquinas paralelas (BACK; HAMMEL; SCHWEFEL, 1997).

Exemplos de algoritmos baseados em populações podem ser: evolutivos (L. J. FOGEL; OWENS; WALSH, 1966; HOLLAND, 1975) (inspirado no processo de evolução e seleção natural, onde as melhores soluções da população são recombinadas para eventualmente gerar soluções melhores ainda), enxame de partículas (KENNEDY, 2011) (inspirado na interação social descentralizada entre diferentes organismos) e colônia de formigas (DORIGO; MANI-EZZO; COLORNI, 1996) (inspirado no comportamento das formigas para traçar uma rota e encontrar mantimentos).

Algoritmos meta-heurísticos também podem ser baseados em um única solução, como: têmpera simulada (KIRKPATRICK; GELATT; VECCHI et al., 1983) (inspirado nos processos metalúrgicos de tratamento térmico de sólidos) e busca tabu (GLOVER, 1986). Este último é o único algoritmo meta-heurístico amplamente difundido que não foi inspirado pela natureza.

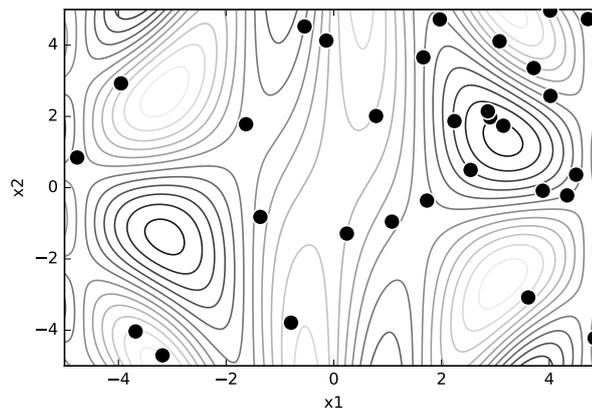
2.2.1 Algoritmos Evolutivos

A ideia de utilizar o processo de evolução como inspiração para algoritmos de otimização torna-se muito válida uma vez que, segundo a seleção natural (DARWIN, 1859), os

indivíduos mais aptos a determinado ambiente têm maiores chances de sobreviver e passar suas características para gerações futuras. Na taxonomia de otimização, isto quer dizer que os atributos das soluções mais próximas de um mínimo global devem ser mantidas e transmitidas para as outras soluções e as que estão mais longe devem ser eliminadas, de forma que, dado um número suficiente de iterações, acontece a convergência para uma única solução.

Além disso, estes métodos trabalham com um conjunto de soluções candidatas (população) e realizam diversas interações entre elas para encontrar a solução ótima. Uma das vantagens de se utilizar uma população de soluções é que o algoritmo é inicializado em diversos subespaços diferentes do espaço de busca, o que aumenta a probabilidade de que em um desses subespaços esteja o mínimo global. A Figura 9 mostra a inicialização de um algoritmo evolutivo para o problema de otimização dado pela Equação (10).

Figura 9 – População inicial de soluções gerada aleatoriamente



Fonte: Autor

Assim como na natureza, o processo de evolução das soluções não pode ser exclusivamente determinístico, pois o algoritmo também depende da exploração em busca de novos subespaços promissores. Isto acontece por meio do uso de diversos operadores matemáticos², como recombinação e mutação (BACK; HAMMEL; SCHWEFEL, 1997), que são responsáveis pela modificação (com diferentes níveis de aleatoriedade) do conjunto de soluções em cada geração do algoritmo.

O algoritmo que seguir as premissas citadas acima pode ser classificado como um algoritmo evolutivo. O Algoritmo 3 mostra o algoritmo fundamental que segue estas premissas, onde P representa a população e D os descendentes.

Os operadores que realizam a inicialização de P_0 , a variação de P_n e a seleção de D devem ser descritos por heurísticas especificadas pelo projetista. Isto implica que os algoritmos evolutivos diferem entre si para cada tipo de problema, pois são heurísticas que gerenciam outras heurísticas, daí o fato destes métodos serem meta-heurísticos. No entanto, a forma canô-

²Também podem ser chamados de operadores genéticos.

Algoritmo 3 – Algoritmo evolutivo (BACK; HAMMEL; SCHWEFEL, 1997)

```

1 Entrada:  $P_0$ 
2 para cada geração  $n$  faça
3   |  $D \leftarrow \text{variatar}(P_n)$ 
4   |  $P_{n+1} \leftarrow \text{selecionar}(D)$ 
5 fim

```

nica de um algoritmo evolutivo deve conter as seguintes características (BACK; HAMMEL; SCHWEFEL, 1997): (a) representação, (b) recombinação, (c) mutação e (d) seleção.

- a) a representação dos indivíduos pode ser uma variável ou uma estrutura de dados que possa ser implementada computacionalmente e deve ser escolhida de acordo com o tipo de problema de otimização. Por exemplo, em um problema de otimização numérica, uma representação conveniente de uma solução seria por números reais, já para otimização combinatória de um grafo, uma representação das soluções seria uma lista dos vértices. Além disso, esta representação deve ser consistente com os operadores de recombinação e mutação para evitar que soluções fora da região de soluções candidatas sejam geradas (Carlos A C. COELLO, 2002).

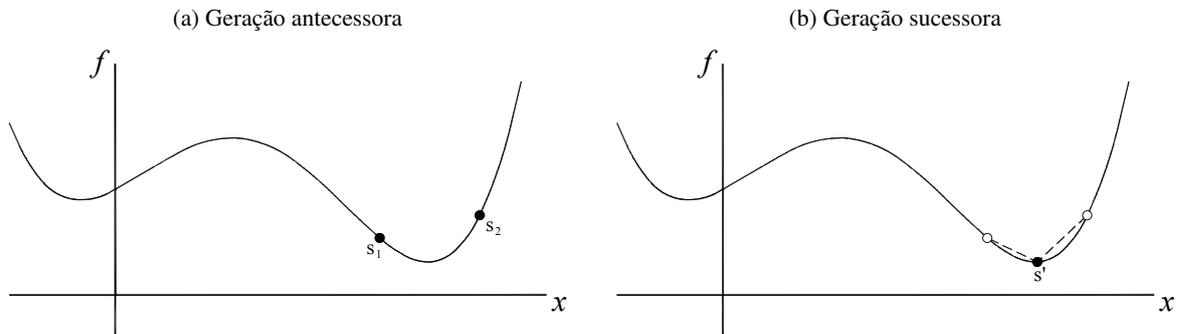
A forma de representação de indivíduos foi estudada independentemente por dois estudos pioneiros. Um deles, conhecido como estratégias evolutivas (*Evolution Strategies*) (SCHWEFEL, 1965), tratava essencialmente de problemas de funções matemáticas e sua representação era numérica. Já o outro, conhecido como algoritmos genéticos (HOLLAND, 1975), buscou inspirações na sequência de DNA e foi representada por um vetor de números binários, o que dá maior versatilidade para trabalhar no domínio discreto.

- b) a recombinação das soluções em algoritmos evolutivos é equivalente à troca de material genético na natureza e foi inicialmente um operador exclusivo dos algoritmos genéticos (BACK; HAMMEL; SCHWEFEL, 1997). É através desta operação que os atributos que tornam duas soluções promissoras são recombinados com o objetivo de potencialmente gerar uma solução melhor ainda. No entanto, se este processo acontecer com muita frequência, a população de soluções pode ficar homogênea e convergir prematuramente para um mínimo local. Por conta disso, atribui-se uma probabilidade de recombinação para controlar a taxa de convergência do algoritmo, tornando este operador estocástico.

A recombinação deve ser pensada de acordo com a representação escolhida para as soluções do problema de otimização. Por exemplo, se a representação for um número real, a recombinação pode ser a média entre duas soluções ou, para casos multidimensional, a recombinação pode ser a troca da coordenada x_n de uma solução com a coordenada x_n de outra. Já no caso em que a representação é binária, a troca de material genético pode ocorrer pela comutação entre as sequên-

cias de DNA de duas soluções. A Figura 10 mostra o efeito da recombinação entre soluções em relação à convergência do problema.

Figura 10 – Representação gráfica do efeito do operador de recombinação entre duas soluções na convergência da solução. O operador de recombinação exemplificado é a média entre as duas soluções



Fonte: Autor

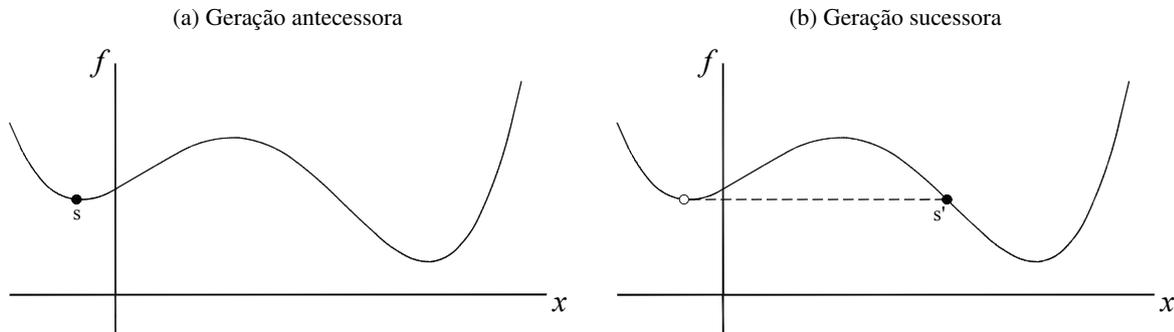
- c) a mutação é a operação que aumenta a chance de que novos subespaços de soluções sejam visitados antes que o algoritmo evolutivo convirja, e ocorre pela perturbação do material genético de cada indivíduo.

Em algoritmos genéticos, este processo deve ser estocástico e uma probabilidade de mutação é atribuída para controlar a taxa de exploração do algoritmo. A taxa de exploração é a frequência em que o algoritmo irá buscar novos subespaços em busca de soluções mais promissoras. Se esta taxa for muito alta, o algoritmo pode levar muito tempo para convergir, pois a mutação frequentemente levará a solução a um novo subespaço. Porém, uma taxa muito baixa pode dificultar o algoritmo a sair de uma condição de estagnação em um mínimo local.

Um exemplo de mutação em representações numéricas seria a multiplicação da solução por uma constante capaz de colocá-la em um novo subespaço possivelmente melhor que o anterior. Para representações binárias, a mutação ocorre pela inversão de um—ou alguns—dos bits da sequência de DNA. A Figura 11 mostra o efeito da mutação de uma solução em relação à exploração de novos subespaços. Neste cenário, ocorre a migração da solução para uma região mais promissora do espaço de busca.

Em estratégias evolutivas, o operador de mutação também é um fator estocástico, porém não fica à escolha do projetista. A mutação neste caso é uma perturbação gaussiana em cada solução da população (BACK; HAMMEL; SCHWELFEL, 1997). Para aumentar as chances de se obter uma solução melhor após a mutação, os algoritmos de estratégias evolutivas possuem o operador de autoadaptação, que busca alinhar a matriz de covariância da distribuição normal com a direção esperada de maior variação (D. B. FOGEL, 1994). A Figura 12 mostra

Figura 11 – Representação gráfica do efeito do operador de mutação de uma solução na exploração de novos subespaços. O operador de mutação exemplificado é a multiplicação por uma constante



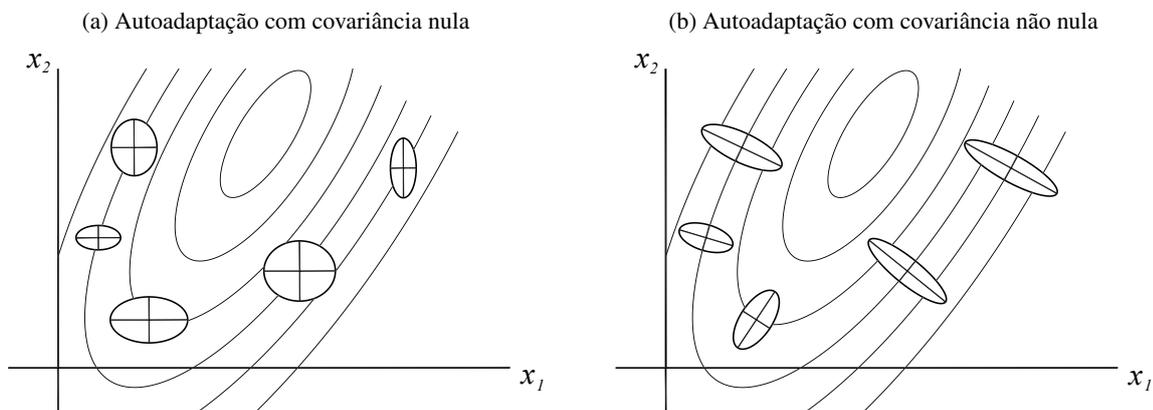
Fonte: Autor

como ocorre esta mutação em um problema de otimização numérica e compara com o efeito do operador de autoadaptação.

- d) finalmente, a seleção é a operação responsável por escolher os indivíduos mais aptos para reprodução. O primeiro passo para a seleção é determinar a função de aptidão para avaliar as soluções, que na taxonomia de otimização é o equivalente à função objetivo. Apesar de selecionar os melhores indivíduos parecer a escolha mais razoável, existem diversas estratégias de seleção.

A seleção em algoritmos genéticos considera que sempre escolher as melhores soluções para reprodução pode comprometer o processo de evolução por conta da homogenização da população de indivíduos precocemente. Portanto, um fator estocástico, que é a probabilidade de selecionar uma solução para recombinação, é acrescentado na etapa de seleção. Esta probabilidade é proporcional à aptidão de cada indivíduo, ou seja, os indivíduos mais aptos continuam com

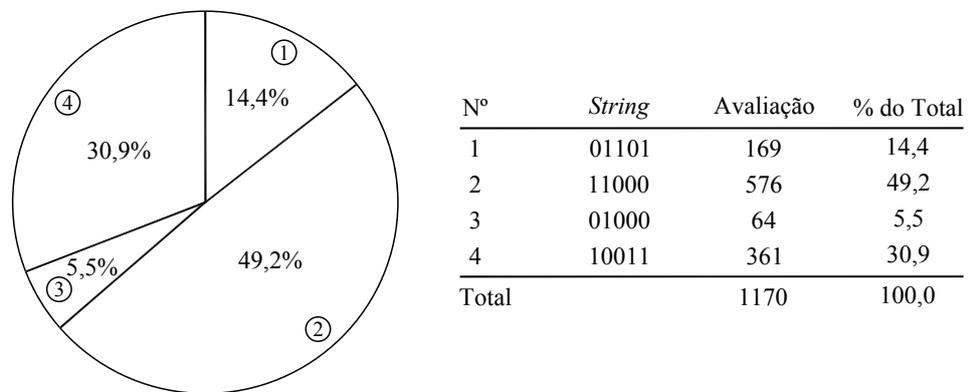
Figura 12 – Representação gráfica do efeito do operador de autoadaptação para aceleração da convergência por mutação em algoritmos de estratégias evolutivas



Fonte: Autor "adaptado de" D. B. Fogel, 1994, p. 9

maiores chances de recombinarem entre si, mas agora os indivíduos menos aptos também possuem chances de serem selecionados, o que garante a diversidade da população de soluções. A implementação desta estratégia é conhecida como método da roleta (HOLLAND, 1975) e teve forte inspiração na natureza, onde diversas espécies competem pela sobrevivência. Já nos algoritmos de estratégias evolutivas, a seleção de indivíduos é dada de forma determinística, onde as melhores soluções são selecionadas para as próximas gerações. A Figura 13 mostra a seleção pelo método da roleta.

Figura 13 – Probabilidade proporcional à aptidão para seleção de soluções para recombinação pelo método da roleta



Fonte: Autor "adaptado de" D. B. Fogel, 1994, p. 5

Tomando-se novamente como exemplo o problema de otimização da Equação (10), um algoritmo evolutivo pode ser desenvolvido para solucioná-lo. Na Figura 14, é mostrado o processo de evolução utilizando duas estratégias distintas para recombinação. Na coluna (a), a recombinação é dada pela média entre dois indivíduos, já na coluna (b) a recombinação é dada pela troca das coordenadas entre dois indivíduos. Nos dois casos a probabilidade de recombinação é 85%. Outros parâmetros do algoritmo são:

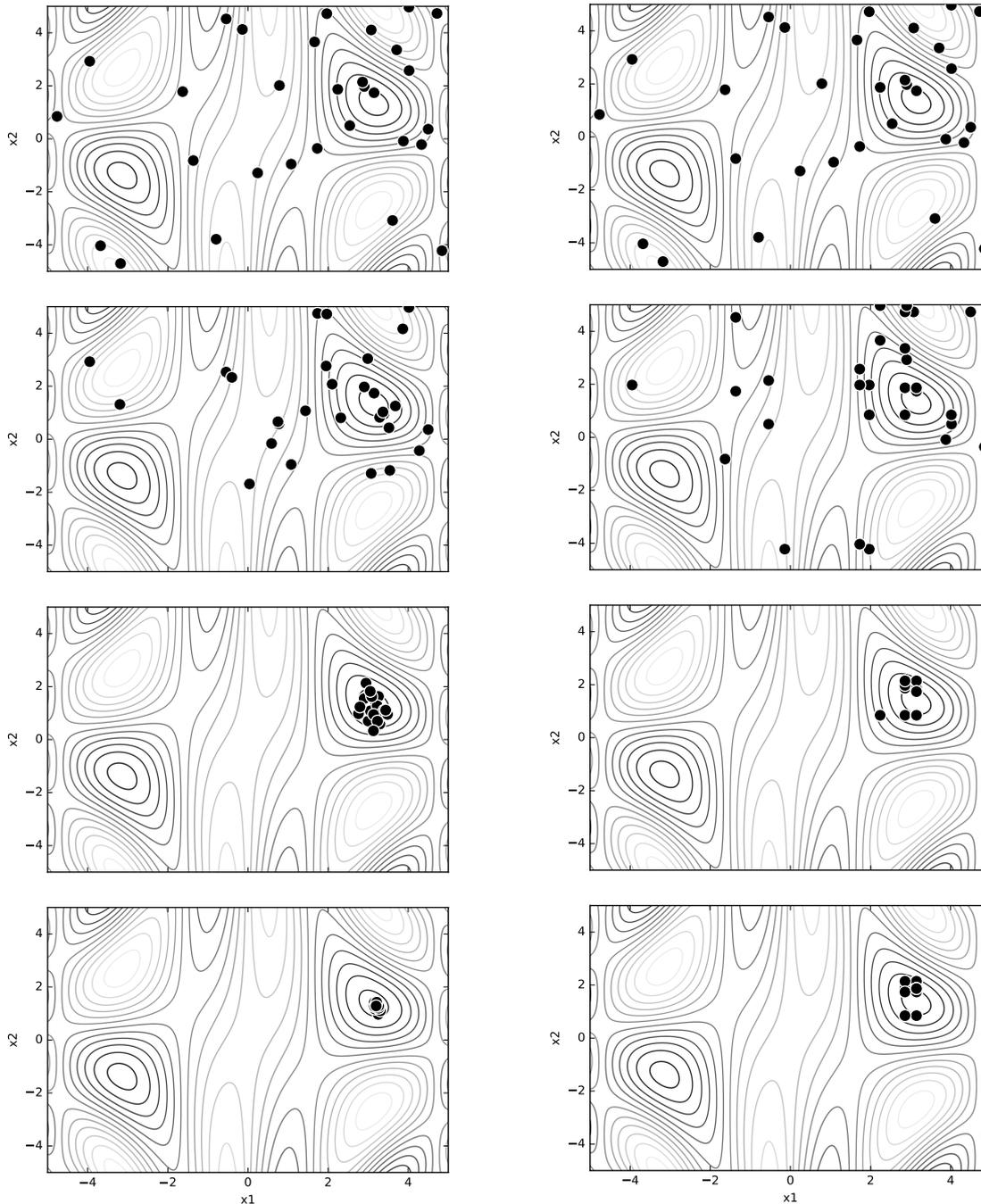
- representação: Número real entre -5 e $+5$;
- mutação: Multiplicação por um fator aleatório que segue uma distribuição uniforme no intervalo de $+0,8$ e $+1,2$. A probabilidade de haver mutação de uma solução é de 12%;
- seleção: Método da roleta, com o melhor indivíduo da população atual P_n mantido para a próxima população P_{n+1} (elitismo).

Na Figura 14, é possível visualizar o efeito que o operador de recombinação pode trazer para a convergência do algoritmo. Em ambos os casos, a população de soluções encontrou o subespaço com o mínimo global da função. Porém, a estratégia de recombinação adotada na coluna (b) da Figura 14 não foi suficiente para manter a diversidade do problema e a população convergiu antes de chegar ao mínimo da função. Neste caso, a diversidade da população nas

Figura 14 – Processo de evolução do algoritmo evolutivo para duas estratégias de recombinação distintas da geração 0, 1, 5 e 10, respectivamente

$$(a) \text{filho} = \left(\frac{\text{pai}_1[x_1] + \text{pai}_2[x_1]}{2}, \frac{\text{pai}_1[x_2] + \text{pai}_2[x_2]}{2} \right)$$

$$(b) \text{filho} = (\text{pai}_1[x_1], \text{pai}_2[x_2])$$



Fonte: Autor

próximas gerações só ocorrerá através da mutação de indivíduos, que possui uma probabilidade baixa.

Na coluna (a) da Figura 14, apesar de também não chegar ao mínimo, ainda existe diversidade de soluções na população de indivíduos. Nestas condições, o processo evolutivo ainda pode gerar soluções melhores e convergir para o mínimo global.

Além dos métodos apresentados acima, existem outros algoritmos evolutivos que são relativamente bem conhecidos na literatura, como: DE (*Differential Evolution*) (PRICE; STORN; LAMPINEN, 2006), EDA (*Estimation of Distribution Algorithm*) (LARRAÑAGA; LOZANO, 2002) e CMA-ES (*Covariance Matrix Adaptation Evolution Strategy*) (HANSEN; OSTERMEIER, 2001), este último uma extensão das estratégias evolutivas.

2.3 OTIMIZAÇÃO DE FUNÇÕES CAIXA PRETA OU CUSTOSAS

Em tarefas de otimização em que o comportamento da função objetivo é desconhecida, incerta, de difícil acesso ou computacionalmente custosa, torna-se indispensável uma nova abordagem para o problema, uma vez que os métodos citados anteriormente necessitam do gradiente da função e/ou de muitos cálculos do valor da função objetivo em diferentes pontos do espaço de busca. Na literatura, otimização de funções caixa preta ou custosas é conhecida como otimização assistida por metamodelos ou método da superfície de resposta, pois utilizam aproximações da função objetivo no espaço de busca (FORRESTER; KEANE, 2009).

Quando a função objetivo é desconhecida, ou seja, não existe uma expressão matemática que a defina, ou incerta, o desafio está em estimá-la através de técnicas estatísticas para poder iniciar o processo de busca (QUEIPO et al., 2005). A palavra estimar pode parecer inadequada no vocabulário de otimização, pois existe a possibilidade de a aproximação suavizar algumas regiões do espaço de busca, escondendo os mínimos globais verdadeiros, ou até mesmo modelando falsos mínimos. Por conta disso, existem diversos modelos e estratégias para aproximação da função objetivo (JONES, 2001).

Outra situação que requer otimização assistida por metamodelos é quando a função objetivo é de difícil acesso ou computacionalmente custosa. Nestes casos, para evitar ao máximo a avaliação da função objetivo verdadeira, é construído um modelo mais simples em cima de dados conhecidos (JIN; CHEN; SIMPSON, 2001). A diferença é que agora a função objetivo é conhecida, portanto o processo de otimização recorre à função original para ajustar o caminho de busca e evitar o problema de suavização dos mínimos.

O processo de otimização assistida por metamodelos é fundamentalmente composto por três etapas (FORRESTER; KEANE, 2009). A primeira delas, conhecida como projeto de experimentos, consiste em fazer a amostragem inicial do espaço de busca para a construção do modelo de partida. Esta amostragem deve ser feita de modo a maximizar o ganho de informação sobre o espaço que se deseja modelar com o mínimo de amostras possível, pois a função objetivo pode ser custosa (SIMPSON et al., 2001). A segunda etapa está na escolha da construção do modelo em si. Com as amostras realizadas na primeira etapa, deve-se em seguida modelar a função objetivo no espaço de busca utilizando o modelo mais apropriado para o tipo de problema (G. G. WANG; SHAN, 2007). A terceira e última etapa é a atualização do modelo. Nesta etapa, são realizadas novas amostragens para aperfeiçoar o modelo nas regiões mais pro-

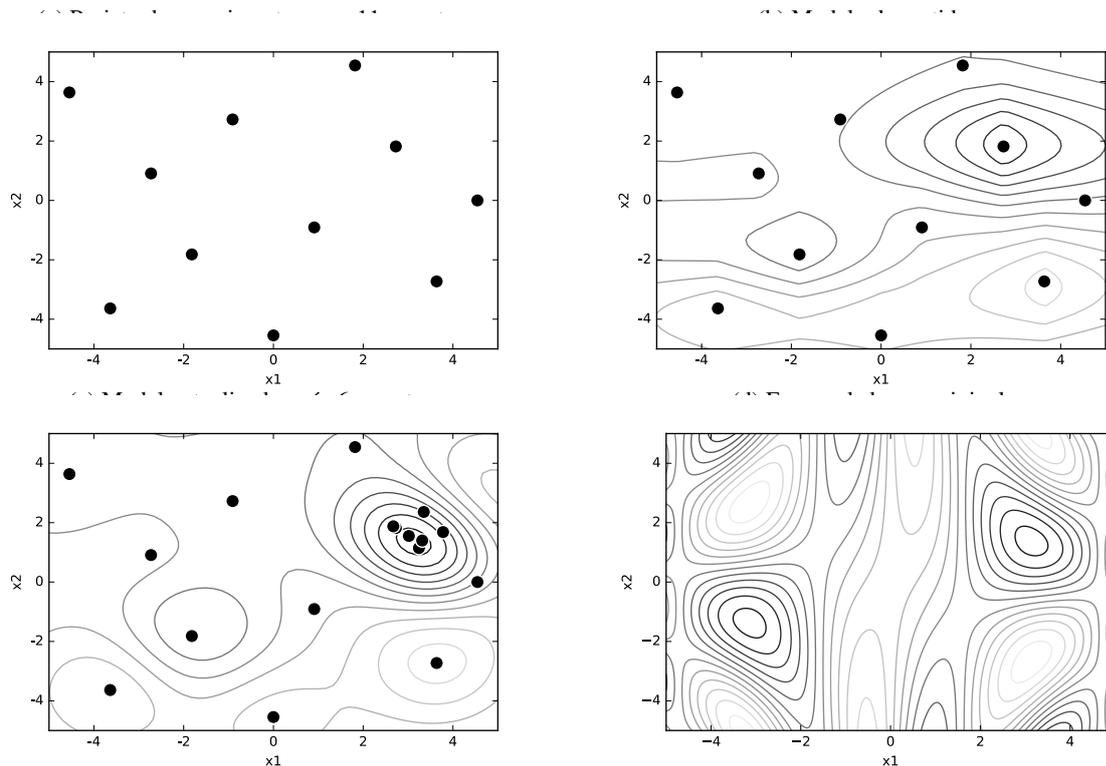
missoras e convergir para uma boa solução com o mínimo de amostras possível. Finalmente, o processo de otimização assistido por metamodelos é essencialmente dado pelo Algoritmo 4.

Algoritmo 4 – Algoritmo de otimização assistido por metamodelos.

- 1 Amostragem inicial (Projeto de experimentos)
- 2 Construir o modelo inicial
- 3 **enquanto não convergir faça**
- 4 | Amostrar os espaços com maiores chances de encontrar o mínimo global
- 5 | Atualizar o modelo com novas amostras
- 6 **fim**

Considerando que a função objetivo do problema da Equação (10) é desconhecida, as etapas do processo de otimização assistido por metamodelos podem ser visualizadas na Figura 15. Para a otimização deste problema, foi preciso apenas modelar a função objetivo no subespaço que continha o mínimo da função, o que corresponde a um modelo simplificado do espaço de busca original. Isto significa que o erro do modelo construído fora da região amostrada pode ser alto.

Figura 15 – As etapas de um processo de otimização por metamodelos



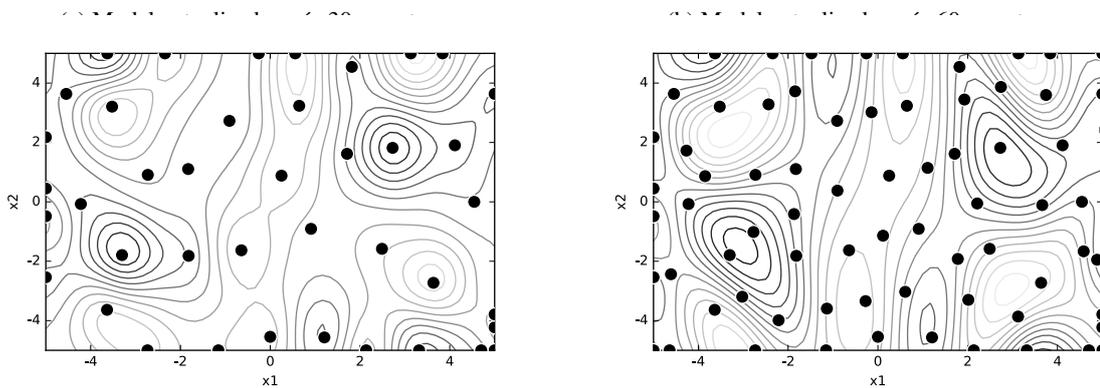
Fonte: Autor

Além de evitar o excesso de avaliações da função objetivo, a metamodelagem também pode ser utilizada para uma aproximação da função objetivo no espaço de busca real (JONES; SCHONLAU; WELCH, 1998). Neste caso, a etapa de atualização do modelo irá amostrar as regiões de maior incerteza do modelo atual. Apesar de aproximar a função objetivo no espaço de busca não ser um processo de otimização, esta aproximação pode ser utilizada como espaço de

busca para vários dos algoritmos de otimização, principalmente em problemas em que a função objetivo não é conhecida, além de ser particularmente útil quando se deseja compreender mais a fundo o comportamento do sistema em questão. Obviamente que, quanto maior a confiabilidade desejada do modelo, maior deverá ser o número de amostras.

Adotando os mesmos parâmetros da Figura 15, a Figura 16 mostra as etapas de atualização do modelo quando o critério de amostragem é de minimizar a incerteza do modelo ao invés da função objetivo.

Figura 16 – Atualização do modelo para aproximação da função objetivo no espaço de busca real



Fonte: Autor

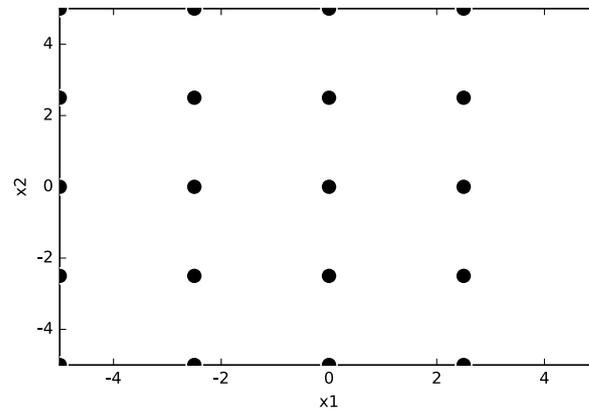
Diversos trabalhos sobre as opções de projeto de experimentos e modelagem podem ser encontrados na literatura (JONES; SCHONLAU; WELCH, 1998; JONES, 2001; SIMPSON et al., 2001; JIN; CHEN; SIMPSON, 2001; QUEIPO et al., 2005; G. G. WANG; SHAN, 2007; FORRESTER; KEANE, 2009). Uma síntese sobre as técnicas de otimização assistida por metamodelos encontradas nestes trabalhos será apresentada nas seções 2.3.1, 2.3.2 e 2.3.3.

2.3.1 Projeto de Experimentos

O principal objetivo do projeto de experimentos é escolher a disposição das amostras para descrever adequadamente o comportamento da função no espaço das variáveis de decisão. Tradicionalmente, o projeto de experimentos consiste em observar o efeito individual de cada variável de decisão. Para que isto seja possível, as amostras devem ser feitas para os n níveis possíveis das k variáveis de decisão, o que resulta em n^k amostras. Esta amostragem chama-se projeto por fatoração completa e é evidentemente inviável no domínio contínuo, onde $n \rightarrow \infty$, além de crescer exponencialmente conforme o número de variáveis de decisão aumenta. A inviabilidade do projeto por fatoração completa pode ser contornada através da limitação dos níveis de cada variável, ou pelo projeto por fatoração parcial, que utiliza n^{k-p} como número de amostras, onde p é o tamanho da partição da fatoração completa. Outros métodos similares são: projeto de Plackett-Burman, projeto por composição central e Box-Behnken. Estes métodos, também chamados de métodos clássicos, tendem a contemplar grande parte de suas amostras

na fronteira do espaço das variáveis de decisão e poucas no centro, conforme ilustra a Figura 17.

Figura 17 – Projeto de experimentos por fatoração de cinco níveis e duas variáveis



Fonte: Autor

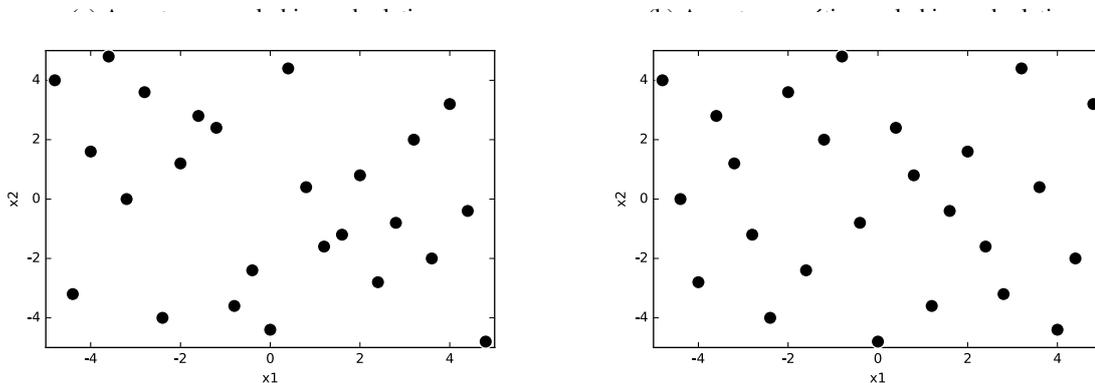
Para processos que envolvem erros determinísticos, como os computacionais, Sacks et al. (1989) mostram que os métodos clássicos para projetos de experimentos são ineficientes ou até mesmo inapropriados por serem voltados a experimentos físicos que carregam erros aleatórios. Para experimentos computacionais, o projeto de experimentos ideal tende a preencher o espaço das variáveis de decisão, ao invés das fronteiras, o que dá o nome a estas técnicas de métodos de preenchimento.

Processos de otimização e modelagem que ocorrem através da amostragem de resultados de simulações ou operadores computacionais é um exemplo de processos computacionais que envolvem erros determinísticos e, portanto, o método de preenchimento é o mais adequado para amostrar o espaço das variáveis de decisão.

Um exemplo de amostragem por preenchimento é a amostragem pelo hipercubo latim, que procura garantir que as amostras sejam alocadas em partições diferentes do espaço das variáveis, evitando amostras redundantes. No entanto, a amostragem pelo hipercubo latim não garante a uniformidade da disposição das amostras, o que pode ser uma característica indesejável para projetos de experimento. Para evitar este problema, existe a amostragem ótima por hipercubo latim, que otimiza os parâmetros de dispersão, como distância mínima ou correlação. A Figura 18 mostra a comparação da amostragem pelo hipercubo latim e a amostragem ótima pelo hipercubo latim.

Também é comum a utilização do método de Monte Carlo para simulação de experimentos, que é uma técnica de preenchimento estocástica para amostragem de experimentos computacionais. Porém, este método não é citado por trabalhos sobre otimização assistida por metamodelos.

Figura 18 – Exemplo de 25 amostras pelo hipercubo latim (a) e amostragem ótima pelo hipercubo latim (b)



Fonte: Autor

2.3.1.1 O Problema da Dimensionalidade

As figuras mostradas foram exemplificadas para um espaço com até duas variáveis de decisão para facilitar sua visualização. Muitos problemas aplicados, no entanto, são problemas de n variáveis, onde n é muito maior do que 2. O problema de se trabalhar em espaços de alta dimensionalidade é que o número de amostras necessárias para descrever um modelo com a mesma precisão cresce exponencialmente conforme o número de dimensões aumenta (*Curse of Dimensionality* (BELLMAN, 1956)). Consequentemente, uma rotina de otimização assistida por metamodelos pode ser extremamente prejudicada se o espaço de decisões for de alta dimensionalidade, uma vez que amostrar o espaço nestes problemas pode ser altamente custoso. Apesar disso, já existem trabalhos publicados na literatura que abordam o problema da dimensionalidade em otimização assistida por metamodelos (SHAN; G. G. WANG, 2010).

2.3.2 Metamodelos e Aprendizado de Máquina

O nome metamodelo é empregado quando se cria um modelo do modelo. Por exemplo: uma simulação computacional é um modelo simplificado de um sistema do mundo real. No entanto, para tarefas de otimização, que geralmente envolvem muitas iterações, o modelo de simulação pode ainda ser computacionalmente custoso, daí a necessidade de construir um novo modelo em cima da simulação.

Dado um conjunto de amostras, técnicas estatísticas ou computacionais são capazes de encontrar uma função que melhor descreve este conjunto. Posteriormente, a função encontrada pode ser utilizada para prever ou classificar dados futuros. Na literatura de estatística, encontrar uma função que melhor descreve um conjunto de dados é tradicionalmente conhecida por reconhecimento de padrões (BISHOP, 2006). Porém, com o avanço dos métodos computacionais, estas técnicas também podem ser encontradas na literatura de inteligência artificial pelo nome de aprendizado de máquina (RUSSELL; NORVIG, 2010).

Adotando o vocabulário de aprendizado de máquina, existem dois tipos principais de aprendizado que podem ser facilmente distinguidos: supervisionado e não supervisionado. Em aprendizado supervisionado, um modelo é construído a partir de dados cujas saídas da função a ser modelada são conhecidas ou podem ser calculadas, ou seja, é um aprendizado baseado em exemplos previamente conhecidos. Já no aprendizado não supervisionado, o objetivo é entender o comportamento dos dados para poder agrupá-los por similaridades. Ao contrário do aprendizado supervisionado, o aprendizado não supervisionado não utiliza exemplos da saída esperada da função a ser modelada e, por conta disso, este método não é utilizado para construir um modelo analítico a partir dos dados.

Adicionalmente, as tarefas de aprendizado supervisionado são empregadas em dois tipos de problemas: regressão e classificação. Na regressão, o objetivo é encontrar um modelo que aproxima no espaço contínuo a função que melhor descreve os dados, ou seja, a saída deste modelo deve ser um valor contínuo. Já na classificação, o objetivo é encontrar um hiperplano que melhor separa os dados no espaço discreto (classes), ou seja a saída deste modelo deve ser um valor discreto. Naturalmente, o tipo de aprendizado supervisionado deve ser escolhido de acordo com a continuidade do espaço de busca que se deseja modelar.

Há ainda um terceiro tipo de aprendizado de máquina que é o aprendizado por reforço. Este tipo de aprendizado tem sido recentemente beneficiado pelas técnicas de otimização bayesiana (BROCHU; CORA; FREITAS, 2010), que pertence à literatura de otimização assistida por metamodelos descrita neste trabalho.

A construção de um metamodelo para otimização é fundamentalmente baseada nos conceitos de aprendizado supervisionado e, conseqüentemente, também é um método estatístico. Em outras palavras, o algoritmo de otimização deve "aprender" com as amostras disponíveis a melhor função que descreve os dados para então poder iniciar o processo de busca. Ironicamente, o próprio processo de construção de um metamodelo é um processo de otimização, o que significa que um algoritmo de otimização assistido por metamodelos necessita "otimizar para otimizar". Obviamente que a função objetivo para otimizar a construção de um metamodelo é computacionalmente mais simples que a tarefa de otimização principal.

Existem inúmeros métodos de aprendizado de máquina supervisionado, inclusive a combinação de métodos diferentes (*ensembles*). Os algoritmos tradicionalmente utilizados na literatura de otimização assistida por metamodelos, que serão descritos nas próximas seções, são: regressão polinomial (George E. P BOX; K. B. WILSON, 1992), funções de base radial (BUHMANN, 2003), redes neurais artificiais (RUMELHART; HINTON; WILLIAMS, 1986) e *kri- ging* (JONES; SCHONLAU; WELCH, 1998). Outros algoritmos de aprendizado de máquina para a construção de metamodelos são: máquinas de vetores de suporte (VAPNIK, 2013) e regressão adaptativa por *splines* multivariadas (FRIEDMAN, 1991).

2.3.2.1 Regressão polinomial

Este método é tradicionalmente conhecido como o pioneiro e o mais estudado metamodelo na literatura, apesar de sua popularidade ter caído nos últimos anos devido ao surgimento de novos métodos mais modernos e mais eficientes (George E. P. BOX; DRAPER, 1987). No entanto, devido à sua facilidade de implementação, este método é o mais recomendável para problemas simples (até 10 variáveis de decisão) e com espaço de busca uniforme.

Dada uma saída y e as variáveis de decisão $x = (x_1, x_2, \dots, x_n)$ que influenciam y , a relação entre y e x é dada por

$$y = f(x) + \varepsilon, \quad (19)$$

onde ε representa o erro aleatório descrito por uma distribuição normal de média zero e desvio padrão σ . Considerando que a função $f(x)$ não é conhecida, um polinômio $g(x)$ de ordem m deve ser encontrado para aproximar $f(x)$. Portanto, o valor estimado da saída é dado por

$$\hat{y} = g(x). \quad (20)$$

A ordem m do polinômio $g(x)$ deve ser escolhida de acordo com a complexidade esperada de $f(x)$. Adotar um polinômio de ordem elevada dará maior flexibilidade ao modelo para incluir as multimodalidades de $f(x)$. Porém, a capacidade de predição do modelo fica prejudicada pois *outliers* do conjunto de dados ou até mesmo ruído podem ser modelados. Este problema é conhecido por *overfitting*, que em português significa "super especialização", e é um problema inerente a todos os modelos de aprendizado de máquina. Por outro lado, modelar por um polinômio de ordem muito baixa compromete a flexibilidade e pode resultar em um modelo inconsistente com os dados.

Dada estas considerações, escolher a ordem do polinômio não é uma tarefa simples. Uma premissa é a Navalha de Occam (RUSSELL; NORVIG, 2010): "*prefira o modelo $g(x)$ mais simples consistente com os dados*".

Para problemas de otimização assistida por metamodelos, geralmente são escolhidos polinômios de ordem baixa com $m = 1$, $m = 2$ ou $m = 3$, descritos pelas Equações (21), (22) e (23), respectivamente.

$$\hat{y} = \beta_0 + \sum_{i=1}^n \beta_i x_i, \quad (21)$$

$$\hat{y} = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{i=1}^n \sum_{j=1}^n \beta_{ij} x_i x_j, \quad (22)$$

$$\hat{y} = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{i=1}^n \sum_{j=1}^n \beta_{ij} x_i x_j + \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \beta_{ijk} x_i x_j x_k, \quad (23)$$

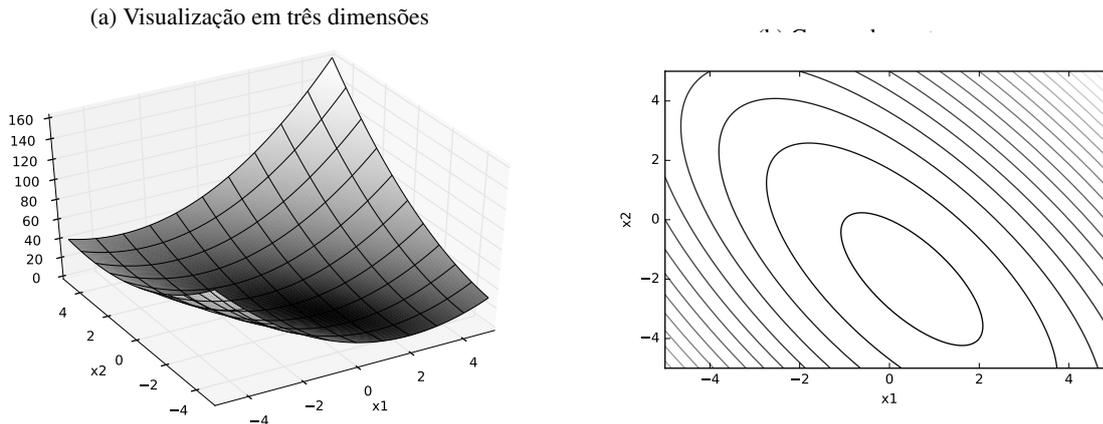
onde β são os coeficientes do modelo polinomial que devem ser ajustados para que o erro entre $f(x)$ e \hat{y} sejam mínimos.

Considerando um modelo polinomial $g(x)$ de ordem 2 e coeficientes $\beta_0 = 10$, $\beta_1 = 2$, $\beta_2 = 3$, $\beta_{11} = 2$, $\beta_{12} = 1$, $\beta_{21} = 1$ e $\beta_{22} = 1$, a função de saída correspondente é dada por

$$\hat{y} = 10 + 2x_1 + 3x_2 + 2x_1^2 + x_1x_2 + x_2x_1 + x_2^2 \quad (24)$$

e sua respectiva visualização é mostrada na Figura 19.

Figura 19 – Visualização gráfica do polinômio dado pela Equação (24)



Fonte: Autor

Em processos computacionais determinísticos, o erro entre o modelo polinomial e os dados não é aleatório. Enquanto que assumir um erro aleatório para experimentos físicos geralmente é útil, para modelos de processos computacionais o erro é inteiramente dado pela modelagem, e pode ser eliminado por interpolação. A desvantagem de usar um modelo polinomial de ordem baixa é que quase nunca será possível a interpolação. Como consequência, se a ordem do polinômio $g(x)$ escolhido for muito baixa para a complexidade de $f(x)$, o modelo será incapaz de adaptar-se conforme novos dados são amostrados.

2.3.2.2 Funções de Base Radial

A regressão por funções de base radial parte da mesma premissa que a regressão polinomial. A diferença é que agora o modelo assume que a saída y , para as variáveis de decisão $x = (x_1, x_2, \dots, x_n)$, é a soma de N funções radiais $\phi(x, \mu) = \phi(\|x - \mu\|)$ simétricas, ponderadas por w e erro ε , ou seja,

$$y = \sum_{i=1}^N w_i \phi(\|x - \mu_i\|) + \varepsilon_i. \quad (25)$$

A norma Euclidiana ($\|x - \mu\|$) é geralmente escolhida, mas outras medidas de distância podem ser adotadas.

Funções de base radiais são essencialmente caracterizadas pela redução (ou aumento) do valor da saída y conforme distancia-se (ou aproxima-se) do ponto central μ com sensibilidade

(raio) σ^2 —este último dado na forma matricial para funções multivariadas. O formato (*kernel*) da função de base radial e a sensibilidade σ^2 são parâmetros que devem ser escolhidos para o modelo. Dois exemplos de formatos de funções de base radial são a gaussiana e a quadrática inversa, mostradas pelas Equações (26) e (27), respectivamente.

$$\phi(\|x - \mu\|) = \exp\left(-\frac{\|x - \mu\|^2}{2\sigma^2}\right), \quad (26)$$

$$\phi(\|x - \mu\|) = \frac{1}{\sqrt{\|x - \mu\|^2 + \mu^2}}. \quad (27)$$

A flexibilidade das funções de base radial, isto é, a capacidade de ajustar-se a diversos tipos de funções, vem da liberdade de escolha para diferentes valores de w e μ , que devem ser encontrados de maneira similar aos coeficientes β dos modelos polinomiais.

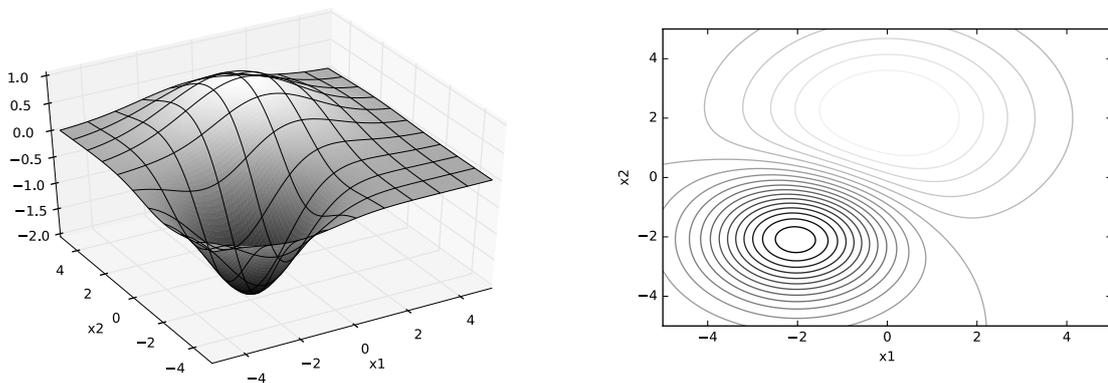
Considerando o modelo \hat{y} , a soma entre duas funções radiais $\phi_1(x, \mu_1)$ e $\phi_2(x, \mu_2)$ de *kernel* gaussiano, $w_1 = 1$, $w_2 = -2$, $\mu_1 = [0, 3]$, $\mu_2 = [-2, -2]$, $\sigma_1^2 = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$ e $\sigma_2^2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$, a função correspondente é dada por

$$\hat{y} = \exp\left(-\frac{x_1^2 + (x_2 - 3)^2}{3}\right) - 2 \cdot \exp\left(-\frac{(x_1 + 2)^2 + (x_2 + 2)^2}{2}\right) \quad (28)$$

e sua respectiva visualização é mostrada na Figura 20.

Figura 20 – Visualização gráfica da função de base radial dada pela Equação (28)

(a) Visualização em três dimensões



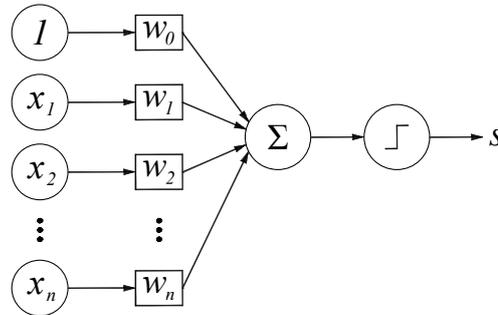
Fonte: Autor

2.3.2.3 Redes Neurais Artificiais

Inspiradas pela capacidade de processamento paralelo e aprendizado do cérebro humano, as redes neurais artificiais têm a capacidade de modelar espaços de busca extremamente complexos, não lineares e com descontinuidades por modelos de neurônios artificiais, denominados pioneiramente de *perceptrons* (MCCULLOCH; PITTS, 1943).

O modelo clássico para um único *perceptron*, ilustrado pela Figura 21, supõe que a saída s é uma função degrau da soma de n entradas x multiplicadas por n pesos w mais um coeficiente linear w_0 , conforme mostra a Equação (29).

Figura 21 – Modelo clássico de um *perceptron* para n entradas

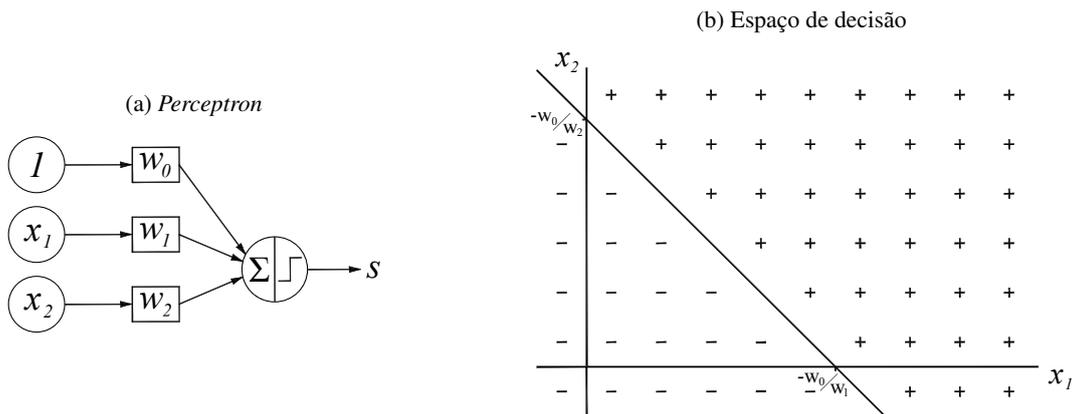


Fonte: Autor "adaptado de" McCulloch; Pitts, 1943, p. 18

$$s = \begin{cases} +1 & \text{se } \sum_{i=1}^n w_i x_i + w_0 > 0 \\ -1 & \text{c.c} \end{cases} \quad (29)$$

Para visualizar o espaço das variáveis de decisão de um *perceptron*, assume-se $n = 2$, ou seja, duas variáveis de entrada. A saída s será $+1$ se $w_1 x_1 + w_2 x_2 + w_0 > 0$ e -1 caso contrário. O espaço de decisão resultante é ilustrado pela Figura 22.

Figura 22 – *Perceptron* de duas entradas e seu respectivo espaço de decisão



Fonte: Autor

Pela Figura 22, é possível observar que um único *perceptron* não tem a capacidade de representar superfícies discriminantes que não sejam lineares. No entanto, a combinação de *perceptrons* em camadas, que formam as redes neurais artificiais, soluciona esta limitação (RUMELHART; HINTON; WILLIAMS, 1986).

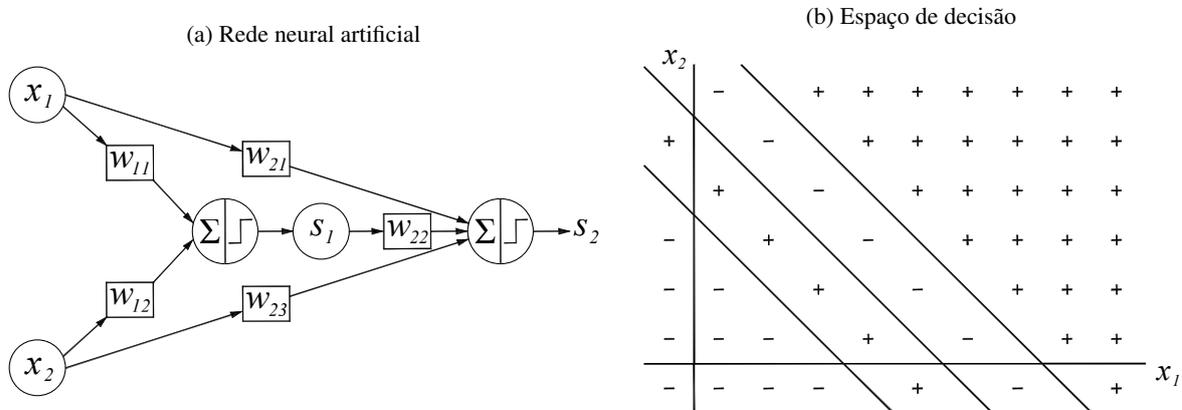
Adotando a rede neural artificial da Figura 23 com as saídas s_1 e s_2 dadas por

$$s_1 = \begin{cases} +1 & \text{se } w_{11}x_1 + w_{12}x_2 > w_{10} \\ -1 & \text{c.c} \end{cases} \quad e \quad (30)$$

$$s_2 = \begin{cases} +1 & \text{se } w_{21}x_1 + w_{22}s_1 + w_{23}x_2 > w_{20} \\ -1 & \text{c.c} \end{cases}, \quad (31)$$

onde w_{10} e w_{20} são os coeficientes lineares de cada *perceptron*, o espaço de decisão resultante ganha uma complexidade de representação maior adicionando apenas um *perceptron*.

Figura 23 – Rede neural artificial com dois neurônios e duas entradas e seu respectivo espaço de decisão



Fonte: Autor

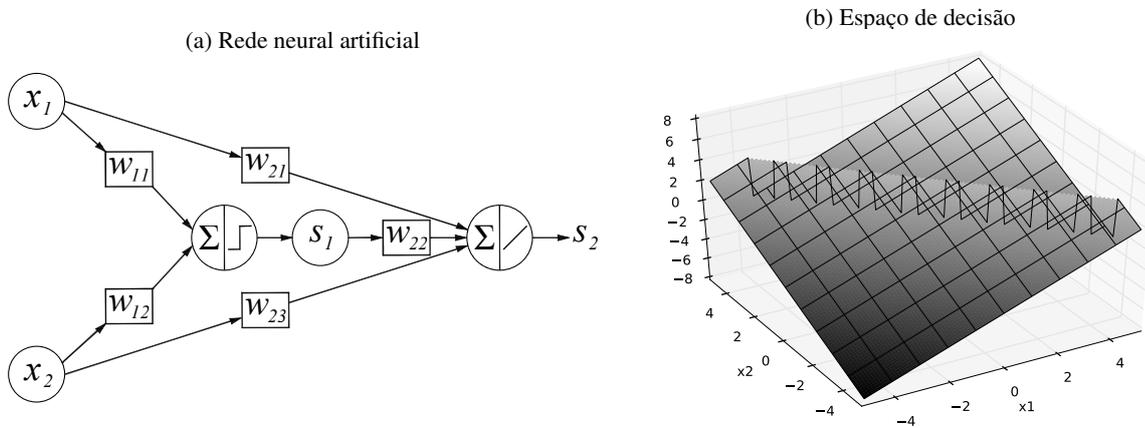
Além da função degrau, as redes neurais artificiais podem utilizar outras funções de transformação na saída, o que transforma este método em um possível modelo de regressão. O emprego de múltiplas camadas de *perceptrons*, junto com a escolha de diferentes funções de transformação na saída de cada *perceptron*, permite às redes neurais artificiais modelarem espaços com as mais diversas características contínuas como: lineares, quadráticas, exponenciais, logarítmicas, radiais, etc.

Alterando a função da saída s_2 do exemplo anterior para $s_2 = w_{21}x_1 + w_{22}s_1 + w_{23}x_2$ (função de saída linear), por exemplo, é possível montar uma função com descontinuidades, o que não seria possível pelos métodos anteriores, conforme a Figura 24.

A liberdade para escolher a quantidade de neurônios, camadas e funções de saída, dá a este modelo a flexibilidade necessária para modelagem de espaços extremamente complexos, inclusive para tratar restrições no problema, ou até mesmo se tornarem aproximadores universais (CYBENKO, 1989).

A desvantagem das redes neurais artificiais é que, após o ajuste dos pesos w , não existe um modelo analítico que explique a função de cada neurônio na saída final, dificultando ajustes manuais do modelo e o entendimento de seus componentes para reaproveitamento em outros modelos.

Figura 24 – Rede neural artificial com dois neurônios, duas entradas e função de ativação linear na saída e seu respectivo espaço de decisão para uma certa configuração de pesos



Fonte: Autor

Apesar de a literatura recomendar seu uso apenas para problemas de alta dimensionalidade, a aplicação computacional deste modelo é bastante acessível, uma vez que existem implementações das redes neurais artificiais em diversas linguagens de programação.

2.3.2.4 Kriging

Tradicionalmente uma técnica estatística de modelagem de dados geológicos, *kriging* também mostrou ser aplicável em modelagem de dados de origem computacional com erros determinísticos.

A resposta y do modelo *kriging* é uma combinação de modelos polinomiais $f(x)$ (responsável pela aproximação global do espaço de busca) acrescentada por um processo gaussiano $Z(x)$ (responsável por perturbações locais para interpolação das amostras), ou seja,

$$y = \sum_{j=1}^k \beta_j f_j(x) + Z(x). \quad (32)$$

Assume-se $Z(x)$ um processo gaussiano com média μ igual a zero e covariância

$$Cov[Z(x^{(i)}), Z(x^{(j)})] = \sigma^2 R(x^{(i)}, x^{(j)}), \quad (33)$$

onde σ^2 é a variância e $R(x^{(i)}, x^{(j)})$ a função de correlação entre duas amostras $x^{(i)}$ e $x^{(j)}$ do conjunto de dados amostrados n_s . A função de correlação $R(x^{(i)}, x^{(j)})$ fica a critério do projetista. Sacks et al. (1989) mostra algumas opções para esta função, mas frequentemente é dada por

$$R(x^{(i)}, x^{(j)}) = \exp\left(\sum_{k=1}^{n_s} \theta_k |x_k^{(j)} - x_k^{(i)}|^2\right), \quad (34)$$

onde $x_k^{(i)}$ e $x_k^{(j)}$ são a k -ésima componente de $x^{(i)}$ e $x^{(j)}$ e θ_k um parâmetro de correlação que deve ser otimizado pela estimação da máxima verossimilhança, ou seja,

$$\begin{aligned} & \text{maximize} && -\frac{n_s}{2} \cdot \ln(\hat{\sigma}^2) - \frac{1}{2} \cdot \sum_{k=1}^{n_s} \theta_k \left| x_k^{(j)} - x_k^{(i)} \right|^2 \\ & \text{sujeito a:} && \theta_k > 0, \end{aligned} \quad (35)$$

onde $\hat{\sigma}^2$ é a variância estimada do modelo global de y dada por

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^T R(x^{(i)}, x^{(j)})^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu})}{n_s}, \quad (36)$$

onde \mathbf{y} é o vetor de tamanho n_s com a saída de cada amostra, $\mathbf{1}$ um vetor de dimensões $n_s \times 1$ e $\hat{\mu}$ a estimação da média do modelo global dada por

$$\hat{\mu} = \frac{\mathbf{1}^T R(x^{(i)}, x^{(j)})^{-1} \mathbf{y}}{\mathbf{1}^T R(x^{(i)}, x^{(j)})^{-1} \mathbf{1}}. \quad (37)$$

Finalmente, a saída estimada \hat{y} do processo gaussiano pode ser calculada por

$$\hat{y} = \hat{\mu} + \mathbf{r}^T R(x^{(i)}, x^{(j)})^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}), \quad (38)$$

onde \mathbf{r} é o vetor de correlação de tamanho n_s entre um ponto desconhecido e as n_s amostras $\{x^{(1)}, x^{(2)}, \dots, x^{(n_s)}\}$, dado por

$$\mathbf{r} = \left[R(x, x^{(1)}), R(x, x^{(2)}), \dots, R(x, x^{(n_s)}) \right]. \quad (39)$$

2.3.3 Atualização do Metamodelo

Além da escolha do metamodelo mais apropriado, o sucesso de um processo de otimização assistido por metamodelos também depende da escolha certa para amostrar e atualizar o modelo. Enquanto que a amostragem busca os pontos mais promissores para encontrar o mínimo, a atualização é necessária para diminuir o erro entre o modelo e a função verdadeira, uma vez que o metamodelo é uma aproximação e seus mínimos podem não corresponder aos mínimos da função verdadeira, e vice-versa.

O ponto mais promissor do modelo a ser amostrado deve ser calculado pela maximização de uma função utilidade $U(x)$, que baseia-se em dois paradigmas para atualização do metamodelo: exploração³ e exploração.

2.3.3.1 Exploração

Considerando um metamodelo globalmente preciso, ou seja, de erro nulo ou desprezível, encontrar o mínimo global da função verdadeira equivale a encontrar o mínimo global do

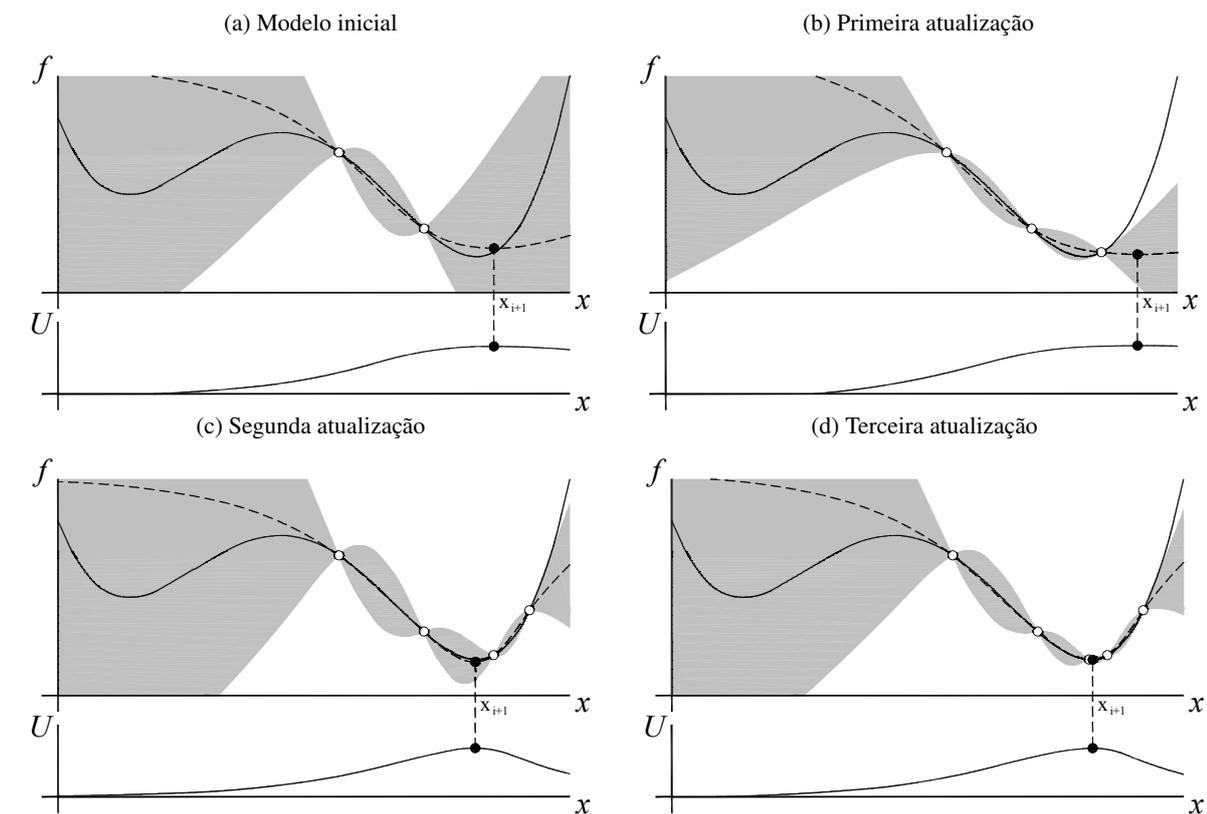
³Tradução da palavra inglesa *exploit*: tirar proveito.

metamodelo. Neste caso, a etapa de atualização deve amostrar sucessivamente os pontos que minimizam o metamodelo até convergir.

Na prática, é inviável construir um metamodelo globalmente preciso, pois seria necessário um número muito alto de amostras, principalmente em problemas de alta dimensionalidade.

Em problemas reais, em que o erro entre o metamodelo e a função verdadeira existe, a atualização por exploração do modelo busca amostrar o mínimo do metamodelo e atualizá-lo naquele ponto. Repetindo este processo, o algoritmo terá aproximado localmente a região que contém o mínimo e convergirá, conforme ilustrado na Figura 25.

Figura 25 – Estratégia de atualização do metamodelo pelo paradigma de exploração. No gráfico de $f(x)$, a curva contínua é a função verdadeira, a curva tracejada é a função aproximadora e a área sombreada é a incerteza desta aproximação. Já $U(x)$ representa a função utilidade e a próxima amostra x_{i+1} será aquela que maximiza $U(x)$



O problema em sempre atualizar o modelo no ponto de mínimo é que o processo de otimização fica susceptível a convergir para um mínimo local, pois o mínimo global do modelo pode ser, na verdade, apenas um mínimo local da função verdadeira.

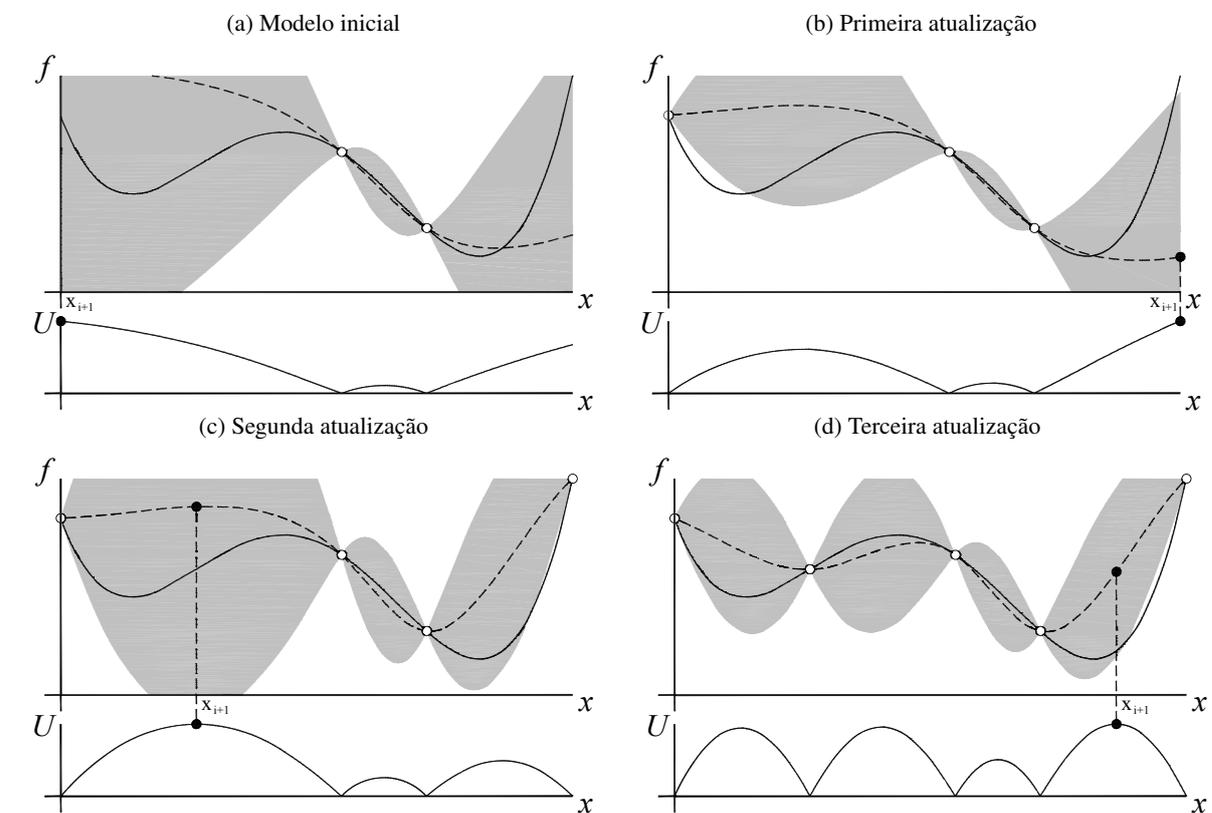
Isto traz a necessidade de também amostrar o espaço de busca nas regiões de maior incerteza (ou erro). Em outras palavras, explorar o modelo.

2.3.3.2 Exploração

A exploração do modelo é um paradigma que surge quando a informação sobre o espaço de busca é incompleta. Este paradigma também pode ser encontrado, por exemplo, em algoritmos evolutivos através do operador de mutação.

Na etapa de construção do modelo, é avaliada também a incerteza em diferentes regiões do espaço de busca. Esta incerteza aumenta conforme distancia-se dos pontos amostrados. Portanto, a exploração do modelo busca atualizá-lo no ponto de maior incerteza. Repetindo este processo, o algoritmo terá aproximado globalmente o espaço de busca, porém não terá encontrado o mínimo, conforme ilustrado na Figura 26.

Figura 26 – Estratégia de atualização do metamodelo pelo paradigma de exploração



Fonte: Autor

Como a atualização do metamodelo ocorre através da consulta da função verdadeira, que muitas vezes é custosa, o algoritmo ideal é aquele que encontra o mínimo global com um menor número de consultas da função verdadeira. Enquanto que um processo de otimização que apenas explora o modelo está sujeito a convergir a um mínimo local, um processo de otimização que apenas explora o metamodelo nunca irá convergir.

Evidentemente que o algoritmo mais indicado é aquele com equilíbrio entre exploração e exploração. Este equilíbrio é calculado por critérios de utilidade do ponto a ser amostrado, como: limite de confiança, probabilidade de melhora e melhora esperada.

Visto que a utilização destes paradigmas propõe que a confiança sobre o modelo aumenta conforme ganha-se conhecimento sobre ele através da amostragem (evidências) do espaço de busca, tem-se uma forma implícita do teorema de Bayes. A literatura muitas vezes denomina os processos de otimização que usam o equilíbrio entre exploração e exploração de otimização bayesiana, e é um tópico que ganhou mais atenção com a popularização de técnicas de inteligência artificial custosas, como o *Deep Learning* (SHAHRIARI et al., 2016). No entanto, muitos autores não fazem a distinção entre a otimização assistida por metamodelos e otimização bayesiana, e consideram apenas uma das várias formas de modelagem por *kriging*.

2.4 CONSIDERAÇÕES FINAIS E TAXONOMIA DE OTIMIZAÇÃO ADOTADA

Dentro do contexto de otimização apresentado neste capítulo, a seguinte taxonomia pode ser construída:

- a) Otimização Local
 - Convexo
 - i. Programação Linear
 - ii. Programação Quadrática
 - Não Convexo
 - i. Programação Não Linear
 - A. Ordem Zero
 - B. Primeira Ordem
 - C. Segunda Ordem
- b) Otimização Global
 - Heurísticas
 - Meta-heurísticas
 - i. Única Solução
 - ii. População
- c) Otimização Assistida por Metamodelos
 - Erro Aleatório
 - i. Amostragem por Fatoração
 - ii. Modelagem por Regressão
 - Erro Determinístico
 - i. Amostragem por Preenchimento
 - ii. Modelagem por Interpolação

Como este trabalho envolve o uso de metamodelos para encontrar os parâmetros ótimos do modelo de transformação, que é um processo computacional de erro determinístico, a metodologia que será adotada se encontra em (c) Otimização Assistida por Metamodelos–Erro Determinístico.

3 NORMALIZAÇÃO ESPACIAL DE IMAGENS

Segundo Sotiras, Davatzikos e Paragios (2013), a normalização espacial de imagens envolve três etapas principais: (i) modelo de transformação, (ii) função objetivo e (iii) algoritmo de otimização. A etapa (iii) já foi discutida no capítulo anterior, cobrindo as várias abordagens para otimização de um problema. Este capítulo faz a fundamentação teórica necessária para desenvolver as etapas (i) e (ii), que são o modelo de transformação e a função objetivo. A seção 3.1 faz uma introdução sobre os conceitos e aplicações para normalização espacial de imagens. A seção 3.2 apresenta os tipos de modelos de transformação de imagens. A seção 3.3 define as várias opções de funções-objetivo, que são métricas que determinam a similaridade entre duas imagens. Finalmente, a seção 3.4 apresenta a taxonomia descrita neste capítulo.

3.1 INTRODUÇÃO

Normalização espacial de imagens, ou registro de imagens, é o campo da visão computacional responsável por alinhar um conjunto de imagens para o mesmo sistema de coordenadas. Este alinhamento se faz necessário nos casos em que as imagens capturadas são oriundas de múltiplos tipos de sensores, posições, momentos e tamanhos (BROWN, 1992).

Entre as principais situações do mundo real em que as imagens capturadas estão sujeitas a apresentar estes tipos de problemas estão: o processamento de imagens aéreas ou de satélites (ZITOVÁ; FLUSSER, 2003), e o processamento de imagens médicas (MAINTZ; VIERGEVER, 1998).

Quando se trata da análise estatística de imagens, assim como na normalização de dados, a normalização espacial de imagens também é necessária para reduzir variações na imagem que são irrelevantes e, portanto, contaminam o conjunto de dados para futuras análises de reconhecimento de padrões e aprendizado de máquina (SONKA; HLAVAC; BOYLE, 2014).

Em termos matemáticos, o registro de imagens ocorre por meio de um operador de transformação \mathbf{T} aplicado a um sistema de coordenadas \mathbf{u} . Geralmente, para o problema de registro de imagens, o sistema de coordenadas \mathbf{u} é um sistema de coordenadas cartesianas e é representado por $\mathbf{u} = [i, j, 1]^T$ para imagens planas (duas dimensões) compostas por *pixels*, ou $\mathbf{u} = [i, j, k]^T$ para imagens volumétricas (três dimensões) compostas por *voxels*.

3.2 MODELOS DE TRANSFORMAÇÃO

De acordo com Maintz e Viergever (1998), o operador de transformação \mathbf{T} é definido global quando a transformação aplicada é a mesma para todos os elementos da imagem, e local quando cada subconjunto de elementos da imagem passa por uma transformação própria.

Geometricamente, a transformação global mantém certas propriedades da imagem que a transformação local não seria capaz de manter, como as distâncias, os ângulos, as linhas

paralelas ou, no mínimo, as linhas retas. Manter estas propriedades geométricas traz uma vantagem da transformação global em relação à local, uma vez que as características locais da imagem, que são irrelevantes para uma análise estatística, serão mantidas. Estatisticamente, como consequência, manter as características irrelevantes dos dados pode resultar na extração de variâncias e complexidades da imagem que são indesejáveis para uma futura aplicação de reconhecimento de padrões.

3.2.1 Modelos de Transformação Global

Existem quatro transformações principais globais para registro de imagens, são elas: euclidiana, similaridade, afin e perspectiva (PRINCE, 2012). Cada uma destas transformações possui um limite de liberdade permitida para alterar a geometria da imagem que será transformada. Esta limitação, por sua vez, garante que certas propriedades geométrica sejam mantidas.

Em modelos globais, o novo sistema de coordenadas da imagem transformada (móvel) u' é calculado pela multiplicação da transformação T pelo sistema de coordenadas da imagem de referência (atlas) u , ou seja

$$u' = T \cdot u. \quad (40)$$

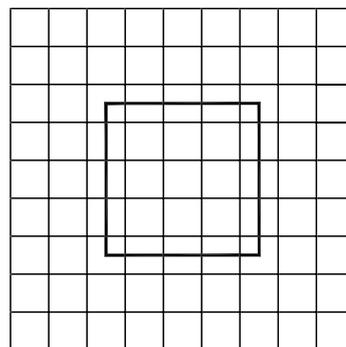
Como exemplo, a imagem de um objeto de geometria quadrada será utilizada para facilitar a visualização do efeito de cada modelo de transformação de duas dimensões. Adicionalmente, esta imagem será particionada em uma grade de subconjuntos de elementos menores por meio de uma malha de pontos de controle, onde cada nó corresponde a um ponto de controle ϕ (RUECKERT, 2001). Tanto a imagem original como a imagem particionada podem ser visualizadas pela Figura 27.

Figura 27 – Exemplo de imagem plana

(a) Imagem original



(b) Imagem original com grade de subconjuntos



Fonte: Autor "adaptado de" Rueckert, 2001, p. 283

Apesar dos modelos de transformação global não necessitarem explicitamente do particionamento da imagem por uma malha de pontos de controle, introduzir este conceito nos

exemplos a seguir facilitará a descrição do funcionamento dos modelos de transformação local, bem como mostrar a relação entre os dois tipos de operadores.

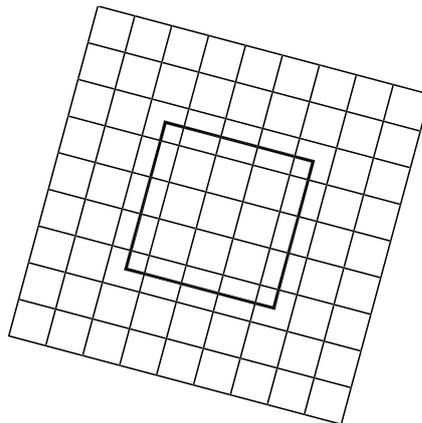
3.2.1.1 Euclidiana

A transformação euclidiana é composta pelos operadores de translação horizontal t_x , translação vertical t_y e o ângulo de rotação θ . A matriz correspondente para transformação euclidiana é

$$\mathbf{T} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & t_x \\ -\sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (41)$$

Esta transformação não preserva a orientação da imagem, mas mantém as distâncias entre os pontos de controle, como pode ser observado pela Figura 28.

Figura 28 – Imagem original após transformação euclidiana



Fonte: Autor "adaptado de" Rueckert, 2001, p. 283

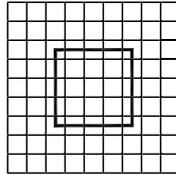
3.2.1.2 Similaridade

Acrescentar a variável escala λ à transformação euclidiana resulta na transformação de similaridade, que além da rotação e translação, permite a transformação do tamanho da imagem. Esta transformação possui quatro parâmetros (λ , θ , t_x e t_y) na forma

$$\mathbf{T} = \begin{bmatrix} \lambda \cdot \cos(\theta) & \lambda \cdot \sin(\theta) & t_x \\ -\lambda \cdot \sin(\theta) & \lambda \cdot \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (42)$$

A transformação de similaridade deixa de preservar as distâncias entre os pontos de controle, mas ainda mantém os ângulos da imagem, como mostra a Figura 29.

Figura 29 – Imagem original após transformação de similaridade



Fonte: Autor "adaptado de" Rueckert, 2001, p. 283

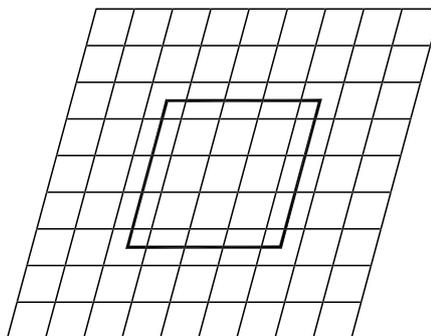
3.2.1.3 Afim

A transformação afim pode ser considerada o caso geral para a transformação de similaridade. Esta transformação adota quatro elementos da matriz (a_{11} , a_{12} , a_{21} e a_{22}) como arbitrários, além dos dois parâmetros de translação t_x e t_y , totalizando seis parâmetros possíveis. A transformação afim permite a alteração da proporção da imagem e deformações de cisalhamento, e sua matriz é dada por

$$\mathbf{T} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (43)$$

Esta transformação não mantém os ângulos da imagem, mas preserva o paralelismo entre as retas formadas pela malha de pontos de controle, conforme mostrado pela Figura 30.

Figura 30 – Imagem original após transformação afim



Fonte: Autor "adaptado de" Rueckert, 2001, p. 283

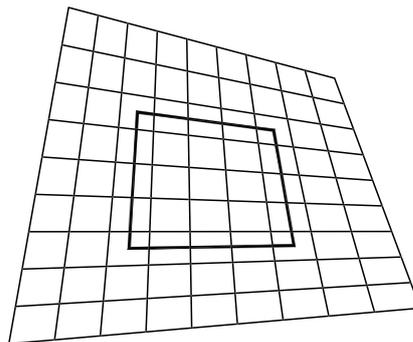
3.2.1.4 *Perspectiva*

Quando os nove parâmetros de matriz \mathbf{T} são arbitrários, a transformação resultante é chamada de transformação de perspectiva. A única propriedade geométrica mantida são as linhas retas formadas pela malha de pontos de controle. A matriz correspondente para transformação de perspectiva é

$$\mathbf{T} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}. \quad (44)$$

A imagem resultante após a transformação de perspectiva pode ser visualizada pela Figura 31.

Figura 31 – Imagem original após transformação de perspectiva



Fonte: Autor "adaptado de" Rueckert, 2001, p. 283

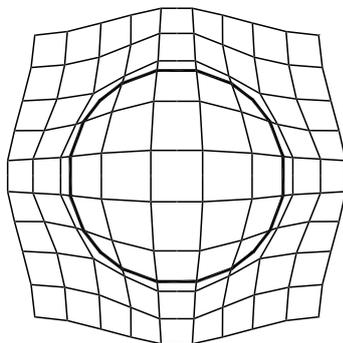
3.2.2 Modelos de Transformação Local

Os modelos de transformação local possibilitam um maior grau de flexibilidade para deformação não rígida de imagens, uma vez que cada subconjunto de elementos passa por uma transformação própria. Consequentemente, ao contrário dos modelos de transformação global, não existe uma matriz \mathbf{T} que defina uma operação de transformação local.

Com a ausência da matriz de transformação \mathbf{T} , a transformação da imagem agora é dada por um algoritmo que irá deformar cada subconjunto de elementos, definido por uma malha de pontos de controle Φ , de forma independente. A imagem resultante após a deformação independente de cada subconjunto pode ser visualizada na Figura 32.

No entanto, para garantir esta flexibilidade, é inevitável que a complexidade do modelo de transformação seja maior. Por exemplo, a transformação global com maior flexibilidade de deformação é a perspectiva, que, para imagens planas, possui nove parâmetros. Por outro lado,

Figura 32 – Imagem original após transformação não rígida



Fonte: Autor "adaptado de" Rueckert, 2001, p. 283

a deformação da mesma imagem utilizando modelos de transformação local pode chegar na ordem de milhões de parâmetros (SOTIRAS; DAVATZIKOS; PARAGIOS, 2013).

Segundo Holden (2008), os algoritmos que definem os modelos de transformação local podem ser baseados em modelos físicos, como modelos dos corpos elásticos e modelos de dinâmica dos fluidos, ou em modelos de interpolação, como funções de base radial, *Thin-plate splines* e *B-splines*. Este último popularizado pelo algoritmo *Free Form Deformation* (RUECKERT et al., 1999).

Adicionalmente, Sotiras, Davatzikos e Paragios (2013) estendem a taxonomia de Holden (2008) e apresentam os modelos baseados em conhecimento, que utilizam técnicas estatísticas para estimar a melhor deformação para a malha de controle. Estes modelos ficaram conhecidos por meio do trabalho de Cootes et al. (1995), que apresentou os *Active Shape Models*, um dos principais modelos de transformação baseados em conhecimento.

3.3 MÉTRICAS DE SIMILARIDADE

A seção anterior descreveu os principais modelos de transformação e os diversos parâmetros e algoritmos possíveis de cada um. Apenas aplicar os modelos de transformação em uma imagem não pode ser considerada uma tarefa de registro de imagens, pois os parâmetros ótimos da transformação que alinham as duas imagens ainda não são conhecidos.

Surge, então, a necessidade de executar uma tarefa de otimização que encontre os parâmetros ótimos que maximizam (ou minimizam) uma função objetivo, que deve ser uma métrica de similaridade que quantifica o alinhamento da imagem móvel com a imagem de referência.

Existem diversos métodos para definir uma função objetivo capaz de quantificar a similaridade entre duas imagens (MAINTZ; VIERGEVER, 1998). Para este trabalho, os métodos abordados são os métodos baseados nos elementos da imagem e os métodos baseados nos

pontos fiduciais da imagem. Portanto, nas seções a seguir serão apresentados exemplos das principais métricas referentes a cada um destes dois métodos.

3.3.1 Métodos Baseados nos Elementos da Imagem

As métricas de similaridade pertencentes aos métodos baseados nos elementos da imagem utilizam a informação intrínseca à imagem por meio de seus *pixels* (para imagens planas) ou *voxels* (para imagens volumétrica). Esta informação muitas vezes é a própria intensidade de cinza do elemento, ou seja, um valor de 0 (preto) a 255 (branco).

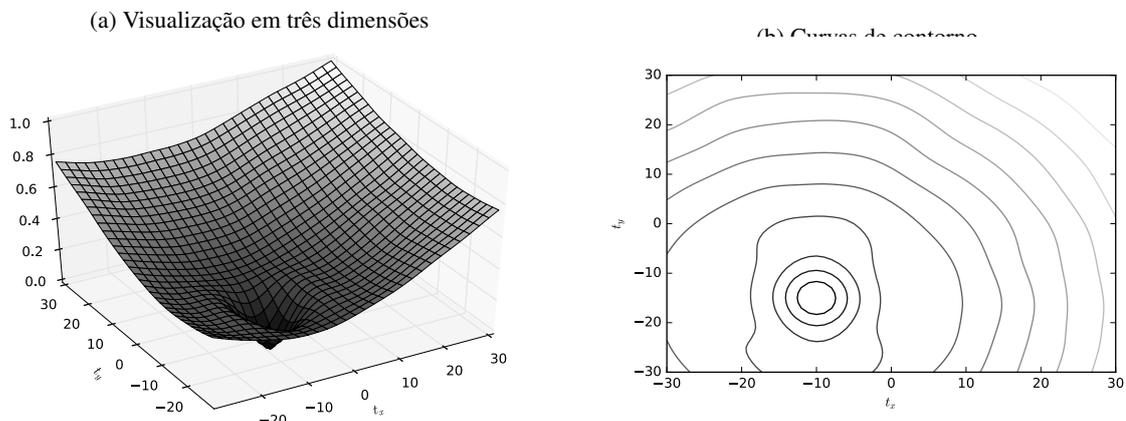
3.3.1.1 Minimização do Erro Quadrático Médio

A minimização do erro quadrático médio é uma métrica de similaridade que calcula diretamente a diferença de intensidade entre os N elementos da imagem móvel \mathbf{I}_{MOV} e os N elementos da imagem de referência \mathbf{I}_{REF} , ou seja, esta métrica assume que as duas imagens tenham a mesma quantidade de *pixels* ou *voxels*. O erro quadrático médio e pode ser calculado por

$$\frac{1}{N} \sum_{i=1}^N (I_{MOV_i} - I_{REF_i})^2. \quad (45)$$

Considerando um exemplo de problema de registro de imagens que utiliza o modelo de translação para alinhamento de uma imagem 2D e a minimização do erro quadrático médio como métrica de similaridade entre duas imagens, as variáveis de decisão são a translação na horizontal t_x e a translação na vertical t_y . O respectivo espaço de busca para este problema é mostrado pela Figura 33.

Figura 33 – Visualização gráfica do espaço de busca do erro quadrático médio e . As translações ótimas t_x^* e t_y^* ocorrem nos pontos -10 e -15 , respectivamente



Fonte: Autor

3.3.1.2 Maximização da Correlação

A correlação é uma medida estatística que quantifica a dependência entre dois sinais distintos. Como uma imagem é um sinal composto pelas intensidades de cinza de seus N elementos, é possível medir a similaridade entre a imagem móvel \mathbf{I}_{MOV} e a imagem de referência \mathbf{I}_{REF} por meio da correlação entre as duas.

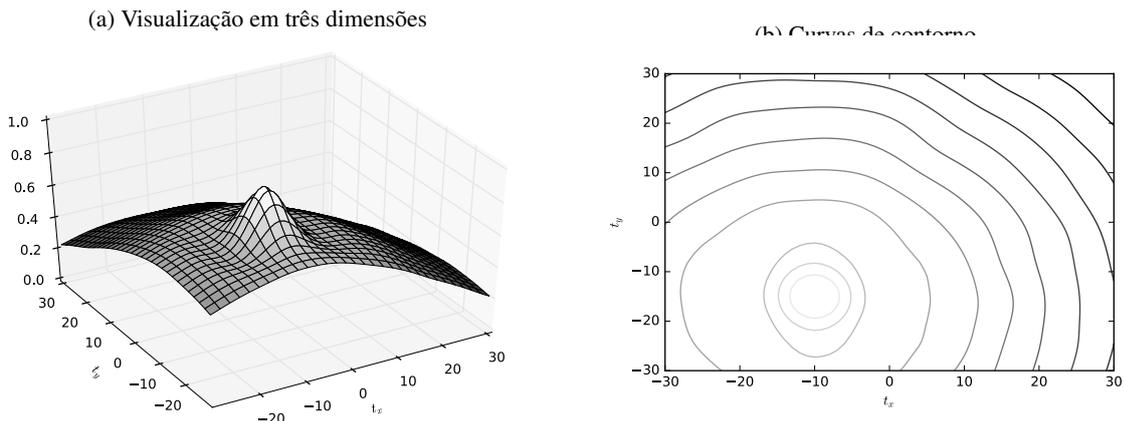
Esta métrica também pertence aos métodos estatísticos de registro de imagens, pois leva em consideração a intensidade média da imagem móvel ($\overline{\mathbf{I}_{MOV}}$) e a intensidade média da imagem de referência ($\overline{\mathbf{I}_{REF}}$).

Existem diversas formas de correlação, mas a correlação de Pearson é a medida mais conhecida para quantificar a dependência entre dois sinais (JOHNSON; WICHERN, 2014), e pode ser calculada por

$$\frac{\sum_{i=1}^N (I_{MOV_i} - \overline{\mathbf{I}_{MOV}})(I_{REF_i} - \overline{\mathbf{I}_{REF}})}{\sqrt{\sum_{i=1}^N (I_{MOV_i} - \overline{\mathbf{I}_{MOV}})^2 \sum_{i=1}^N (I_{REF_i} - \overline{\mathbf{I}_{REF}})^2}}. \quad (46)$$

Considerando o mesmo exemplo da Figura 33, mas agora utilizando a maximização da correlação como métrica de similaridade entre duas imagens, o espaço de busca pode ser visualizado pela Figura 34.

Figura 34 – Visualização gráfica do espaço de busca da correlação. As translações ótimas t_x^* e t_y^* ocorrem nos pontos -10 e -15 , respectivamente



Fonte: Autor

A maximização da correlação pode também ser utilizada para registro de imagens pela correspondência de modelos, que compara os elementos da imagem móvel com os elementos da imagem de referência com o objetivo de encontrar em qual região da imagem as duas compartilham as mesmas características, ou seja, são mais similares (LEWIS, 1995). Neste caso, adota-se a correlação cruzada, que é uma outra forma de mensurar a correlação entre dois sinais (MAINTZ; VIERGEVER, 1998).

Matematicamente, a correlação cruzada pode ser vista como a convolução entre dois sinais. No entanto, na correlação cruzada, utiliza-se o conjugado de um dos sinais, enquanto que na convolução não. Esta pequena distinção entre as duas técnicas faz com que ambas sejam idênticas caso os sinais sejam simétricos, pois o conjugado de um sinal simétrico resulta nele mesmo (GONZALEZ; WOODS, 2006; OPPENHEIM; SCHAFER, 2009).

Diferentemente da correlação de Pearson, que resulta em um coeficiente, o resultado da correlação cruzada é um sinal que indica a correspondência entre dois sinais em função do deslocamento, ou seja, a translação ótima que a imagem móvel deve ter para corresponder o máximo possível com a imagem de referência deve ser aquela que maximiza a correlação cruzada.

Para sinais discretos 2D (imagens planas) de dimensões $M \times N$, a correlação cruzada C para cada deslocamento t_x e t_y pode ser calculada por¹

$$\sum_m^{M-1} \sum_n^{N-1} \mathbf{I}_{MOV}(m,n) \cdot \mathbf{I}_{REF}(m - t_x, n - t_y). \quad (47)$$

Da Equação (47), pode-se observar que o operador de correlação cruzada utiliza o próprio modelo de translação. Isto significa que as translações ótimas t_x^* e t_y^* são encontradas diretamente pelo resultado máximo da correlação cruzada entre as duas imagens, dispensando o uso de algoritmos de otimização para encontrar as translações ótimas.

3.3.1.3 Maximização da Informação Mútua

A maximização da informação mútua é uma abordagem probabilística para registro de imagens que analisa e compara a informação contida em cada imagem por meio da entropia² marginal e entropia conjunta de Shannon (SHANNON, 2001).

Teoricamente, a entropia marginal H de uma distribuição X de n variáveis é calculada a partir da somatória do produto entre as probabilidades $p(x_i)$ da função densidade $P(X)$ e o logaritmo de suas inversas $p(x_i)^{-1}$, ou seja,

$$H(X) = \sum_{i=1}^n p(x_i) \log p(x_i)^{-1}. \quad (48)$$

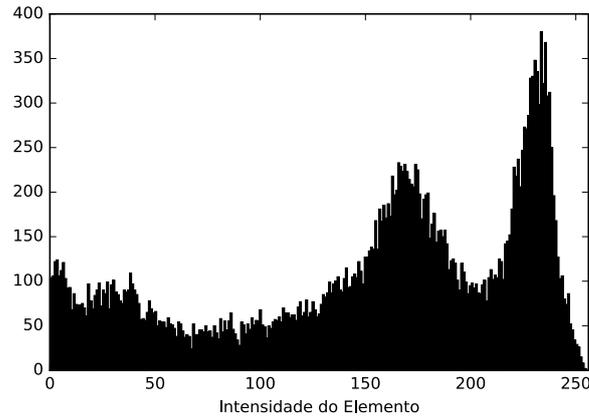
Quando se trata da entropia marginal de uma imagem, cada variável x_i da distribuição é definida pelo valor da intensidade de seus elementos, que totalizam 256 variáveis para imagens em escala de cinza. Isto significa que o termo $p(x_i)$ da Equação (48) é igual à quantidade de ocorrências de elementos de valor x_i dividida pela quantidade total de elementos da imagem.

Alternativamente, a distribuição das 256 intensidades dos elementos de uma imagem pode ser descrita por um histograma. A partir deste histograma, pode ser calculada a probabilidade de cada intensidade, e em seguida, a entropia $H(X)$ da imagem. Um exemplo de histograma de uma imagem qualquer pode ser visualizado pela Figura 35.

¹<<https://www.mathworks.com/help/signal/ref/xcorr2.html>>.

²Em teoria da informação, a entropia pode ser entendida como uma medida de incerteza.

Figura 35 – Histograma de uma imagem em escala de cinza



Fonte: Autor

Adotar separadamente a diferença de entropia marginal entre duas distribuições, mesmo que espacialmente divergentes, pode resultar em níveis de incertezas idênticas. Logo, a entropia marginal não é adequada para quantificar a similaridade entre duas imagens.

No entanto, é possível calcular a entropia conjunta entre duas ou mais distribuições para medir a incerteza associada entre elas. A entropia conjunta é calculada da mesma forma que a entropia marginal, porém utilizando a probabilidade conjunta entre as distribuições para cada variável. Considerando duas distribuições X de n variáveis e Y de m variáveis, o cálculo da entropia conjunta é dado por

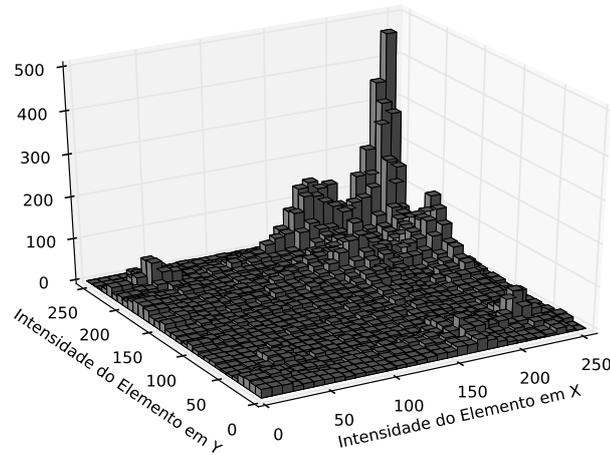
$$H(X,Y) = \sum_{j=1}^m \sum_{i=1}^n p(x_i,y_j) \log p(x_i,y_j)^{-1}. \quad (49)$$

Na Equação (49), o termo $p(x_i,y_i)$ pode ser calculado de forma similar ao termo $p(x_i)$ da Equação (48), ou seja, pela quantidade de ocorrências da combinação de elementos (x_i,y_i) dividida pela quantidade total de combinações de elementos da imagem.

Um outro método para calcular a entropia conjunta pode ser feito pelo histograma entre as duas distribuições. Neste caso, o histograma deve ser relativo à distribuição conjunta de X e Y , e portanto é representado por três dimensões para mostrar a quantidade de ocorrências para cada combinação de elementos (x_i,y_i) . Um exemplo de histograma conjunto entre duas imagens desalinhadas, mas que possuem o mesmo valor de entropia marginal, ou seja, individualmente possuem o mesmo histograma, pode ser visualizado pela Figura 36.

A Figura 36 mostra que, apesar de os histogramas das distribuições X e Y serem idênticos (mesma entropia marginal), as intensidades dos elementos das duas imagens possivelmente estão desalinhadas, pois o valor da entropia conjunta é elevado em relação às entropias marginais. Em outras palavras, o alinhamento ótimo entre as imagens existirá quando a entropia conjunta é mínima, ou seja, $H(X,Y) = H(X) = H(Y)$. Assim, o máximo desalinhamento entre as imagens existirá quando a entropia conjunta for máxima, ou seja, $H(X,Y) = H(X) + H(Y)$.

Figura 36 – Histograma conjunto entre duas distribuições X e Y para diferentes combinações de intensidades de elementos. Os valores das entropias são: $H(X) = 3,90$, $H(Y) = 3,90$ e $H(X,Y) = 6,91$



Fonte: Autor

Na maior parte dos casos, no entanto, a condição de mínima entropia conjunta não pode ser satisfeita, a menos que as duas imagens sejam idênticas. Da mesma forma que é praticamente impossível satisfazer a condição de máxima entropia conjunta, pois, em alguma região da imagem, mesmo que por acaso, as duas imagens compartilham a mesma informação.

Esta informação compartilhada é chamada de informação mútua $I(X,Y)$ e pode ser calculada em função das entropias marginais e conjuntas por (COVER; THOMAS, 2012)

$$\begin{aligned}
 I(X,Y) &= H(X) + H(Y) - H(X,Y) \\
 &= \sum_{j=1}^m \sum_{i=1}^n p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}.
 \end{aligned} \tag{50}$$

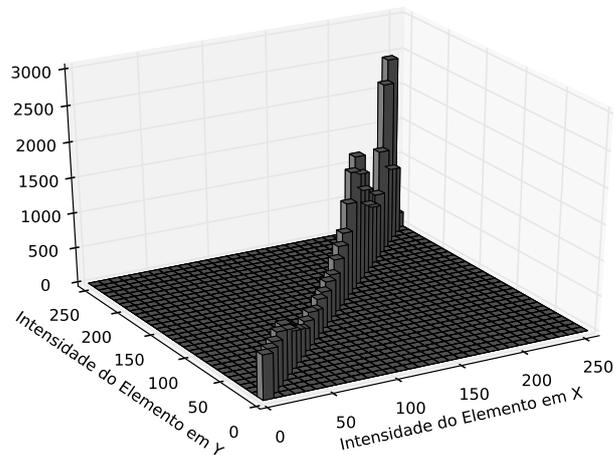
Como a informação mútua representa explicitamente a quantidade de informação compartilhada entre duas imagens, a maximização desta medida para registro de imagens é mais conveniente do que a minimização da entropia conjunta (MAES et al., 1997; VIOLA; WELLS III, 1997).

Adicionalmente, a entropia conjunta pode ser expressa em função da informação mútua e das entropias condicionais $H(X|Y)$ e $H(Y|X)$, que correspondem à incerteza de uma distribuição dada que a outra é conhecida, como

$$H(X,Y) = H(X|Y) + H(Y|X) + I(X,Y). \tag{51}$$

Pela Equação (51) pode-se concluir que a informação mútua será máxima quando as entropias condicionais forem nulas. Isto pode ser visualizado pelo histograma conjunto de duas imagens alinhadas e idênticas, conforme Figura 37.

Figura 37 – Histograma conjunto entre duas distribuições idênticas. A existência da diagonal no histograma indica que todos os elementos de X correspondem com todos os elementos de Y , resultando em entropias condicionais nulas, bem como máxima informação mútua. Em outras palavras $H(X|Y) = H(Y|X) = 0$ e $H(X) = H(Y) = H(X,Y) = I(X,Y)$

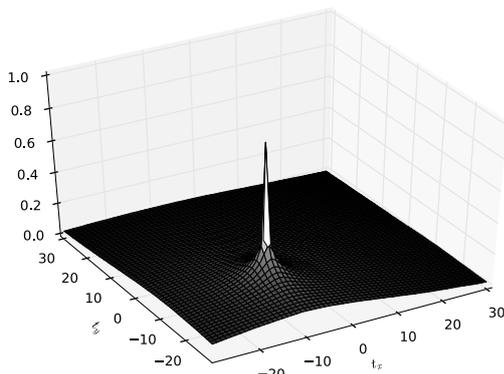


Fonte: Autor

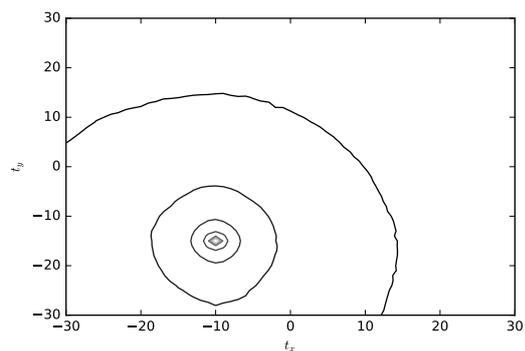
Finalmente, o espaço de busca do problema de otimização das translações entre duas imagens desalinhadas utilizando a maximização da informação mútua como métrica de similaridade pode ser visualizado pela Figura 38.

Figura 38 – Visualização gráfica do espaço de busca da informação mútua. As translações ótimas t_x^* e t_y^* ocorrem nos pontos -10 e -15 , respectivamente

(a) Visualização em três dimensões



(b) Curvas de contorno



Fonte: Autor

3.3.2 Métodos Baseados nos Pontos Fiduciais da Imagem

Uma outra estratégia para quantificar a similaridade entre duas imagens é por meio dos métodos baseados nos pontos fiduciais da imagem, que são marcadores de referência que indicam os pontos de maior interesse de uma imagem. A principal diferença é que estes métodos

não dependem diretamente da intensidade dos *pixels* ou *voxels* da imagem para calcular a semelhança. No entanto, as informações dos elementos são necessárias para encontrar os pontos fiduciais da imagem, portanto, indiretamente, esta informação é utilizada por estes métodos.

Ao contrário dos métodos da subseção anterior, que dependem de milhares de variáveis³, medir a similaridade por meio dos pontos fiduciais reduz o número de variáveis para manter apenas aquelas que são relevantes. Por conta disso, a qualidade do alinhamento entre duas imagens por meio dos pontos fiduciais acaba muitas vezes sendo superior, especialmente em imagens de face, onde o alinhamento não necessariamente deve ser feito pela intensidade do elemento (textura), e sim, pela disposição entre eles (estrutura) (ZHU; RAMANAN, 2012).

Para uma imagem plana, a correspondência entre os n pontos fiduciais da imagem móvel, descritos pelas coordenadas cartesianas (x_{MOV_i}, y_{MOV_i}) , e os n pontos fiduciais da imagem de referência (x_{REF_i}, y_{REF_i}) é calculada pela soma da distância entre eles. A transformação ótima é aquela que minimiza esta soma, ou seja, a diferença mínima entre os pontos fiduciais da imagem.

A distância mais conhecida é a distância euclidiana e a soma das distâncias euclidianas entre os pontos fiduciais das imagens pode ser expressa por

$$\sum_{i=1}^n \sqrt{(x_{MOV_i} - x_{REF_i})^2 + (y_{MOV_i} - y_{REF_i})^2}. \quad (52)$$

3.4 CONSIDERAÇÕES FINAIS E TAXONOMIA DE NORMALIZAÇÃO ESPACIAL DE IMAGENS ADOTADA

Dentro do contexto de modelos de transformação de imagens apresentado neste capítulo, a seguinte taxonomia pode ser construída:

- a) Global
 - Euclidiana
 - Similaridade
 - Afim
 - Perspectiva
- b) Local
 - Modelos Físicos
 - Modelos de Interpolação
 - Modelos Baseados em Conhecimento

Como este trabalho envolve o uso de normalização não rígida, ou seja, modelos de transformação local para deformar a imagem, a metodologia que será adotada para modelos de transformação se encontra em (a) Local–Modelos de Interpolação.

Adicionalmente, dentro do contexto de métrica de similaridade, a seguinte taxonomia pode ser construída:

³considerando que uma imagem é composta por $M \times N$ elementos.

- a) Métodos Baseados nos Elementos da Imagem
 - Minimização do Erro Quadrático Médio
 - Maximização da Correlação
 - Maximização da Informação Mútua
- b) Métodos Baseados nos Pontos Fiduciais da Imagem
 - Minimização da Distância Euclidiana

Como este trabalho propõe uma nova métrica de similaridade para avaliar o resultado após a deformação, uma combinação entre os dois métodos será utilizada.

4 NORMALIZAÇÃO ESPACIAL NÃO RÍGIDA DE IMAGENS ASSISTIDA POR METAMODELOS

Este capítulo detalha a metodologia proposta, bem como as bibliotecas utilizadas, para normalização espacial não rígida de imagens assistida por metamodelos de acordo com a mesma taxonomia apresentada nos capítulos 2 e 3. Segundo esta taxonomia, um problema de registro de imagens é definido por três principais componentes: (i) modelo de transformação, (ii) função objetivo e (iii) algoritmo de otimização. Portanto, este capítulo está dividido em três seções. A seção 4.1 fundamenta o modelo de transformação escolhido para este trabalho, o *Free-form Deformation* e sua implementação via *software* disponibilizada em código aberto publicamente. A seção 4.2 desenvolve a função objetivo proposta para quantificar a qualidade da normalização não rígida da imagem. Por último, a seção 4.3 descreve a implementação via *software*, também em código aberto, do algoritmo de otimização assistida por metamodelos. O repositório com a implementação do *framework* proposto, que também disponibiliza um tutorial e exemplo de como utilizá-lo, pode ser encontrado em <<https://github.com/ddfabbro/SANRR>>.

4.1 FREE-FORM DEFORMATION

Concebido sobre o modelo de interpolação baseado em *B-Splines* de multiníveis (LEE; WOLBERG; SHIN, 1997), o *Free-Form-Deformation* (FFD) foi pioneiramente apresentado por Rueckert et al. (1999) como um algoritmo de registro não rígido de imagens médicas.

A transformação \mathbf{T} adotada pelo FFD é composta por um modelo de transformação global \mathbf{T}_{global} para eliminar os fatores de desalinhamento globais, como translação, rotação, escala, etc, e um modelo de transformação local \mathbf{T}_{local} , para alinhar as características locais da imagem.

Rueckert et al. (1999) definem a transformação local \mathbf{T}_{local} como uma deformação baseada em *B-splines* sobre o domínio de imagens volumétricas descritas por $\Omega = \{(x, y, z) \mid 0 \leq x < X, 0 \leq y < Y, 0 \leq z < Z\}$. Considerando Φ a malha de $n_x \times n_y \times n_z$ pontos de controle, onde cada nó desta malha é denominado por $\phi(x, y, z)$ com espaçamento uniforme δ entre eles, o FFD pode ser definido matematicamente como um produto tensorial entre *B-splines* cúbicas, ou seja,

$$\mathbf{T}_{local} = \sum_{l=0}^3 \sum_{m=0}^3 \sum_{n=0}^3 B_l(u) B_m(v) B_n(w) \phi(x, y, z)_{i+l, j+m, k+n}, \quad (53)$$

onde $i = \lfloor x/n_x \rfloor - 1$, $j = \lfloor y/n_y \rfloor - 1$, $k = \lfloor z/n_z \rfloor - 1$, $u = x/n_x - \lfloor x/n_x \rfloor$, $v = y/n_y - \lfloor y/n_y \rfloor$, $w = z/n_z - \lfloor z/n_z \rfloor$, sendo $\lfloor x \rfloor$ a função piso, e B a função de base *B-splines* para o FFD, definida por

$$\begin{aligned}
B_0(x) &= (1-x)^3/6 \\
B_1(x) &= (3x^3 - 6x^2 + 4)/6 \\
B_2(x) &= (-3x^3 + 3x^2 + 3x + 1)/6 \\
B_3(x) &= x^3/6.
\end{aligned} \tag{54}$$

Apesar de a Equação (53) descrever uma transformação local baseada em *B-Splines* para imagens volumétricas, é possível simplificá-la para o problema de imagens planas (XAVIER et al., 2015), que são os objetos de estudo deste trabalho, por

$$\mathbf{T}_{local} = \sum_{l=0}^3 \sum_{m=0}^3 B_l(u)B_m(v)\phi(x,y)_{i+l,j+m}. \tag{55}$$

Os parâmetros responsáveis pelo resultado final da interpolação por *B-splines*, que consequentemente influenciam o resultado da deformação, são as posições dos pontos de controle da malha Φ , pois são eles que descrevem quais subconjuntos da imagem que devem ser deformados para minimizar, por meio de um algoritmo de otimização de 1ª ordem (baseado em gradiente), uma função objetivo \mathcal{C} descrita por

$$\mathcal{C} = -\mathcal{C}_{similaridade} + \lambda\mathcal{C}_{suavidade}, \tag{56}$$

onde $\mathcal{C}_{similaridade}$ é a informação mútua normalizada entre a imagem móvel \mathbf{I}_{MOV} e a imagem de referência \mathbf{I}_{REF} , definida por

$$\mathcal{C}_{similaridade}(\mathbf{I}_{MOV}, \mathbf{I}_{REF}) = \frac{H(\mathbf{I}_{MOV}) + H(\mathbf{I}_{REF})}{H(\mathbf{I}_{MOV}, \mathbf{I}_{REF})}, \tag{57}$$

$\mathcal{C}_{suavidade}$ é o termo de penalização que regulariza a interpolação por *B-splines* para manter a suavidade da transformação (WAHBA, 1990 apud RUECKERT et al., 1999), e λ é um fator de ponderação que define o compromisso entre $\mathcal{C}_{similaridade}$ e $\mathcal{C}_{suavidade}$.

Rueckert et al. (1999) recomenda $\lambda = 0,01$, baseado em seus experimentos em imagens de ressonância magnética de seios femininos. Porém, este é apenas um valor prático e outros valores de λ podem ser investigados para outros tipos de imagem.

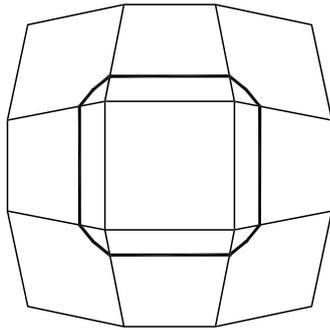
Adicionalmente, o espaçamento δ entre os pontos de controle é um parâmetro que não faz parte do algoritmo de otimização do FFD e, portanto, deve ser especificado como um valor constante antes de iniciar o registro da imagem (hiperparâmetro).

Além de garantir os graus de liberdade do FFD, δ também estabelece a quantidade de pontos de controle, pois esta distância define a resolução da malha Φ . Como qualquer ponto de controle que não esteja contido no domínio Ω da imagem é desconsiderado, o máximo valor válido de δ são os próprios limites de Ω para que pelo menos um ponto de controle incida sobre a imagem. Um valor baixo de espaçamento entre os pontos de controle resulta em uma malha Φ densa, garantindo um maior grau de liberdade de deformação ao custo de complexidade computacional. Por outro lado, um valor alto resulta em uma malha Φ menos densa e um modelo FFD com menor grau de liberdade e menor custo computacional. O efeito da densidade

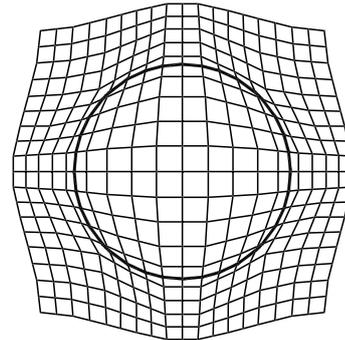
da malha Φ pode ser visualizado na Figura 39, que mostra duas deformações por FFD para dois valores distintos de espaçamento entre pontos de controle.

Figura 39 – Imagem original após transformação não rígida com duas densidades de Φ

(a) Transformação não rígida com δ alto



(b) Transformação não rígida com δ baixo



Fonte: Autor "adaptado de" Rueckert, 2001, p. 283

No entanto, determinar o grau de liberdade do modelo FFD não é apenas uma questão de complexidade computacional, uma vez que esta flexibilidade também influencia a intensidade da deformação local na imagem, e, caso esta flexibilidade seja excessiva, pode descaracterizar a imagem completamente. Isto significa que o parâmetro δ deve ser escolhido de forma a eliminar as características locais irrelevantes da imagem, mas preservando suas características principais.

Este compromisso traz uma dificuldade para a determinação, descrita como empírica por Rueckert et al. (1999), do melhor valor para o parâmetro δ que define a malha Φ ótima para executar a deformação capaz de atingir um bom compromisso.

A incerteza causada pela determinação empírica do parâmetro δ , bem como a chance de convergir a uma malha Φ sub ótima, pode ser reduzida ao adotar uma abordagem multinível para a interpolação por *B-splines*, que é a estratégia adotada pelo FFD. De forma resumida, a abordagem multinível inicia a malha Φ com um espaçamento δ alto (resolução baixa) para deformar as características locais mais rudimentares e diminui δ sucessivamente para realizar os ajustes mais finos (resolução alta).

Segundo Rueckert et al. (1999), dividir os valores de δ pela metade a cada nível de resolução é uma estratégia plausível, uma vez que os pontos de controle ϕ coincidem nas divisões sucessoras, simplificando o processo computacional. Porém, esta é uma estratégia generalizada e não há nenhuma garantia de atender a premissa de eliminar as características locais irrelevantes da imagem, mas preservando suas características principais.

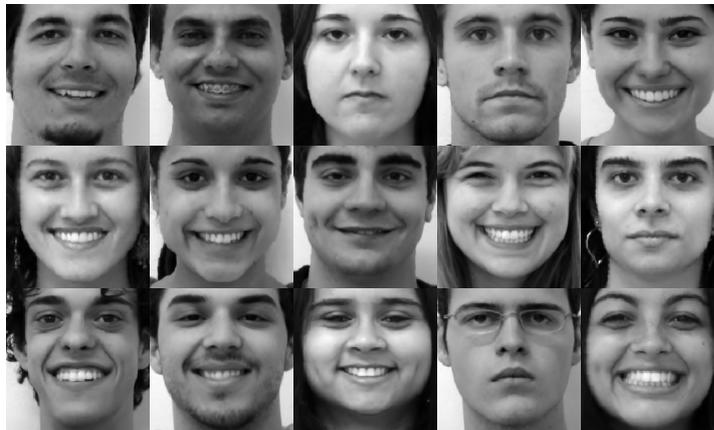
4.1.1 Implementação do *Free-Form-Deformation*

Neste trabalho, a implementação do FFD é feita pelo MIRTk (*Medical Image Registration ToolKit*¹), uma biblioteca em código aberto disponibilizada publicamente pelo grupo BioMedIA do Departamento de Computação do Imperial College London.

Apesar de não ser mencionado no algoritmo original do FFD, o MIRTk possibilita a especificação dos pontos fiduciais da imagem para destacar os elementos mais importantes durante a avaliação da função objetivo. Particularmente em normalização espacial de imagens, os pontos fiduciais são uma informação extremamente relevante devido aos níveis de detalhes inerentes à face humana.

Xavier et al. (2015) utilizaram o MIRTk para investigar os efeitos do FFD nas imagens frontais de face da *FEI Face Database*², uma base de imagens de 200×200 pixels das faces de 200 alunos, professores e funcionários da FEI. A proporção entre homens e mulheres é igual (100 homens e 100 mulher), e para cada participante existem duas imagens frontais de face, uma sorrindo e a outra não sorrindo, totalizando 400 imagens. Uma amostra da *FEI Face Database* pode ser visualizada na Figura 40.

Figura 40 – Quinze amostras de imagens frontais de face da *FEI Face Database*



Fonte: Autor

Adicionalmente, Xavier (2015) explorou a aplicação do FFD comparando duas quantidades diferentes de pontos fiduciais (46 e 284), e constatou que a qualidade da normalização não rígida foi superior para quantidades maiores de pontos.

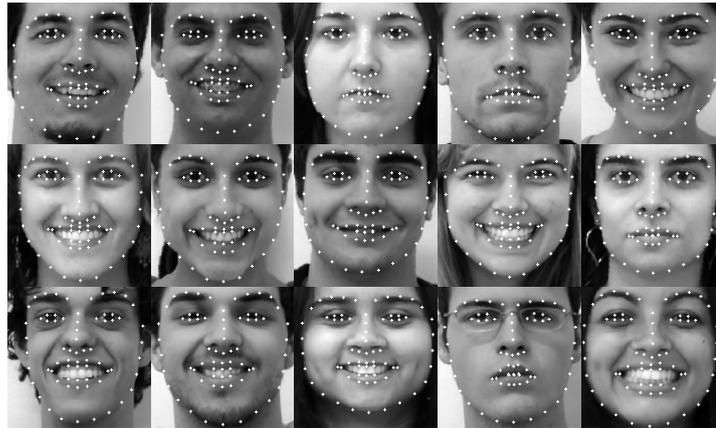
No entanto, esta descoberta foi feita adotando os valores de δ sugeridos por Rueckert et al. (1999). Por este motivo, suspeita-se que seja possível obter uma normalização não rígida de boa qualidade sem a necessidade de aumentar a quantidade de pontos fiduciais, ou seja, por meio de uma estratégia que auxilie na seleção dos valores de δ para encontrar a malha Φ otimizada para cada resolução.

¹<https://mirtk.github.io/>.

²<http://fei.edu.br/~cet/facedatabase.html>.

Para detecção dos pontos fiduciais da face, este trabalho usou um algoritmo que faz a marcação automática de 68 pontos (KAZEMI; SULLIVAN, 2014), implementado pela biblioteca de código aberto dlib³. A Figura 41 mostra a marcação dos 68 pontos das faces da Figura 40.

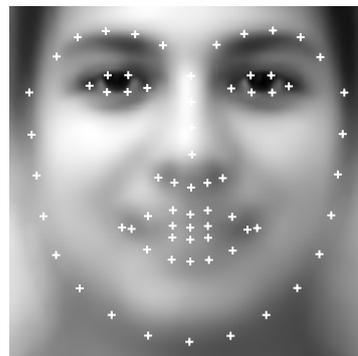
Figura 41 – Marcação dos 68 pontos fiduciais das quinze amostras de imagens frontais de face da *FEI Face Database*



Fonte: Autor

Antes de inicializar o registro por FFD, é necessário especificar qual será a imagem de referência para normalização. No caso da *FEI Face Database*, a imagem de referência, visualizada pela Figura 42, é a média aritmética das intensidades de cinza dos elementos de cada face do conjunto de dados.

Figura 42 – Imagem de referência: média das 400 imagens frontais de face da *FEI Face Database* marcadas por 68 pontos fiduciais



Fonte: Autor

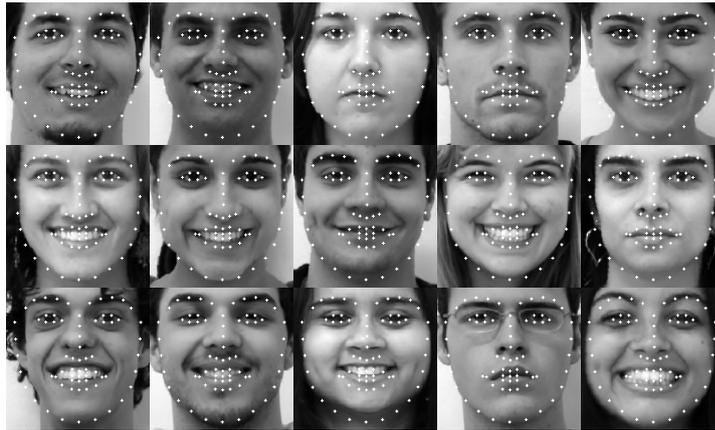
Uma forma de verificar se uma imagem de face está localmente alinhada é comparando-a com os pontos fiduciais da imagem de referência. Caso as imagens apresentarem esta divergência, o registro não rígido pode ser uma etapa importante para eliminar fatores como posição e

³<http://dlib.net/>.

tamanho do rosto, ou, no caso de reconhecimento de faces, a expressão facial, que é irrelevante para este tipo de tarefa.

Esta afirmativa é exemplificada na Figura 43, que compara os pontos fiduciais da imagem de referência com as imagens da *FEI Face Database*.

Figura 43 – Comparação entre os pontos fiduciais da imagem de referência e quinze amostras de imagens frontais de face da *FEI Face Database*



Fonte: Autor

4.2 METODOLOGIA E FUNÇÃO OBJETIVO PROPOSTA PARA OTIMIZAÇÃO DE HIPERPARÂMETROS

Como o MIRTk é uma ferramenta de linha de comando, foi preciso incluir esta biblioteca em um *script*⁴ na linguagem Python para colocá-lo dentro de uma metodologia de otimização baseada na taxonomia já discutida. Este *script* possibilita que a tarefa de registro de imagens seja executada múltiplas vezes sem a interferência de uma pessoa, permitindo a aplicação de algoritmos de otimização sobre as entradas e saídas do MIRTk.

Quanto aos hiperparâmetros do FFD, ou seja, os parâmetros que precisam ser definidos antes de iniciar a tarefa de normalização, este *script* permite otimizar os valores de espaçamento entre pontos de controle δ para cada resolução e o fator λ de ponderação que define o compromisso entre $C_{similaridade}$ e $C_{suavidade}$.

Para atender a premissa de eliminar as características locais irrelevantes da imagem, mas preservando suas características principais, a função objetivo proposta para normalização espacial não rígida de imagens assistida por metamodelos é composta por dois objetivos: (i) soma das distância euclidiana entre os pontos fiduciais e (ii) distâncias euclidianas entre as componentes principais.

⁴<https://github.com/ddfabbro/SANRR/blob/master/sanrr/register.py>.

As duas próximas seções descrevem cada objetivo tomando como exemplo o problema de normalização das imagens mostradas na Figura 44.

Figura 44 – Exemplo de imagens de entrada para normalização espacial



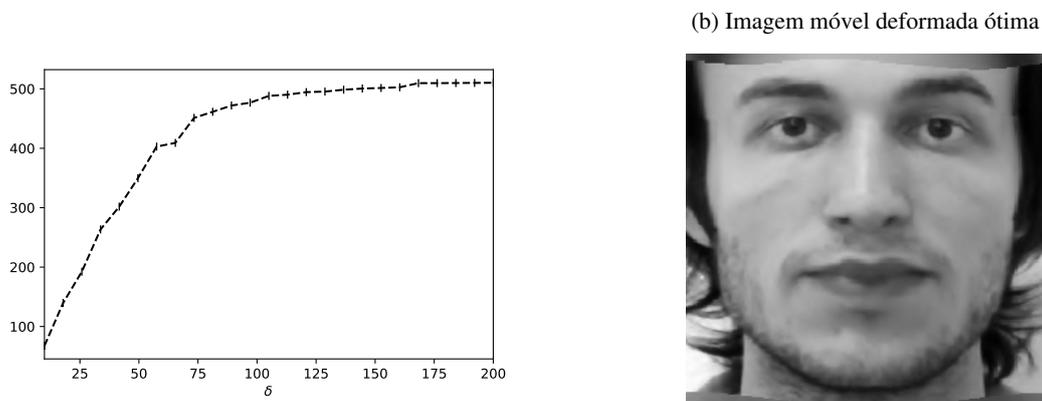
Fonte: Autor

4.2.1 Objetivo 1: soma das distâncias euclidianas entre os pontos fiduciais

O primeiro objetivo (\mathcal{J}_1) é eliminar as características locais irrelevantes da imagem. Portanto, é avaliada a soma das distâncias euclidianas entre os pontos fiduciais da imagem móvel e a imagem de referência, sendo que a soma mínima significa que todas as características locais foram eliminadas, inclusive as relevantes, descaracterizando por completo a face.

Considerando como hiperparâmetros do FFD apenas o espaçamento entre pontos de controle da primeira resolução, o gráfico para o objetivo 1, bem como a face resultante obtida pela minimização deste objetivo, podem ser visualizados na Figura 45.

Figura 45 – Gráfico da soma das distâncias euclidianas entre os pontos fiduciais com δ na primeira resolução como variável de decisão e a deformação ótima de acordo com o objetivo 1. O espaçamento entre pontos de controle que minimiza esta função é denotado por δ^*



Fonte: Autor

4.2.2 Objetivo 2: distância euclidiana entre as componentes principais

O segundo objetivo (\mathcal{J}_2) é preservar as características principais da imagem. Para isso, calcula-se a distância euclidiana entre as componentes principais da imagem móvel antes e após a deformação. Quanto menor for esta distância, mais as componentes principais foram preservadas.

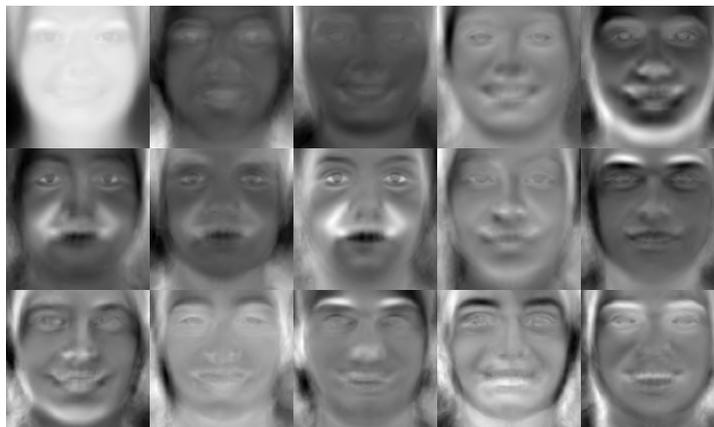
Em linhas gerais, calcular as componentes principais de uma imagem significa transformar linearmente sua base original de n dimensões para uma nova base ortogonal, onde cada eixo representa o sentido de maior variação dos dados (FUKUNAGA, 2013). Esta transformação permite, por exemplo, reduzir a excessiva dimensionalidade de imagens ao descartar os eixos que não capturam uma variância significativa nos dados, e que muitas vezes são redundantes.

Calcular as componentes principais de imagens de face foi uma técnica pioneiramente proposta por Sirovich e Kirby (1987) como uma estratégia estatística para a representação de faces em um espaço de baixa dimensionalidade e, posteriormente, como uma técnica de reconhecimento de faces por Turk e Pentland (1991). As componentes principais da face são chamadas de autofaces (*eigenfaces*), pois as componentes principais nada mais são que os autovetores da transformação.

Adicionalmente, o segundo objetivo permite escolher a variância cumulativa a ser considerada. Quanto maior for a variância cumulativa desejada, maior será o número de componentes para representá-la. No caso extremo, se a variância cumulativa for de 100%, todas as componentes serão aproveitadas e a dimensionalidade não será reduzida.

Supondo que as características principais devem acumular uma variância maior do que 75%, para a *FEI Face Database*, seriam necessárias 15 componentes principais (autofaces) para representar aproximadamente esta variação. Estas componentes, que são as características que o objetivo 2 busca preservar após a deformação, podem ser visualizadas pela Figura 46.

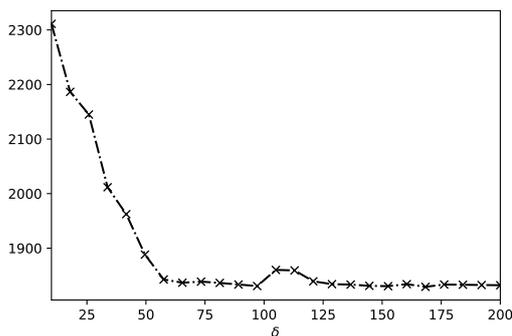
Figura 46 – Visualização das 15 autofaces que representam 76% da variância dos dados



Fonte: Autor

Aproveitando o mesmo exemplo para gerar a Figura 45, o gráfico para o objetivo 2, bem como a face resultante obtida pela minimização deste objetivo, podem ser visualizados na Figura 47.

Figura 47 – Gráfico da distância euclidiana entre as componentes principais com δ na primeira resolução como variável de decisão e a deformação ótima de acordo com o objetivo 2



Fonte: Autor

(b) Imagem móvel deformada ótima



4.2.3 Regularização e composição dos objetivos

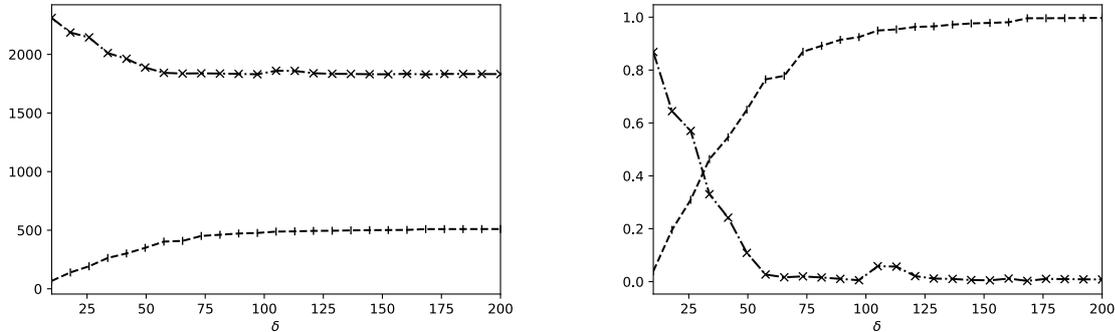
As Figura 45(b) e 47(b) indicam que a face normalizada ótima deve ser um meio termo entre as duas, pois a primeira deformou a face a ponto de tornar o indivíduo irreconhecível e a segunda não foi capaz de eliminar as características locais da imagem. Além disso, as Figuras 45(a) e 47(a) mostram que ambos os objetivos são conflitantes e devem ser combinados.

O problema é que a ordem de grandeza entre os dois objetivos é diferente e combiná-los acabaria favorecendo o objetivo com a ordem de grandeza maior.

Este problema pode ser solucionado ao normalizar cada objetivo entre 0 (mínimo) e 1 (máximo). No entanto, a normalização requer que o mínimo e o máximo de cada objetivo sejam conhecidos. Felizmente, sabe-se que a soma mínima das distâncias euclidianas entre os pontos fiduciais acontece para o menor valor de δ e a soma máxima acontece para o maior valor de δ . Similarmente, a distância euclidiana mínima entre as componentes principais da face acontece para o maior valor de δ e a máxima acontece para o menor valor de δ .

Isto é razoável pois o espaçamento entre pontos de controle é a variável que determina a flexibilidade de deformação do FFD, ou seja, quanto menor for δ , maior será a deformação e, conseqüentemente, menor será a soma das distâncias euclidianas dos pontos fiduciais e maior será a distância euclidiana das componentes principais, e vice-versa. A Figura 48 mostra os gráficos dos dois objetivos não normalizados (a) e normalizados (b).

Figura 48 – Comparação entre os espaços de busca do objetivo 1 e objetivo 2 sem normalização e com normalização entre 0 e 1 com δ na primeira resolução como variável de decisão



Fonte: Autor

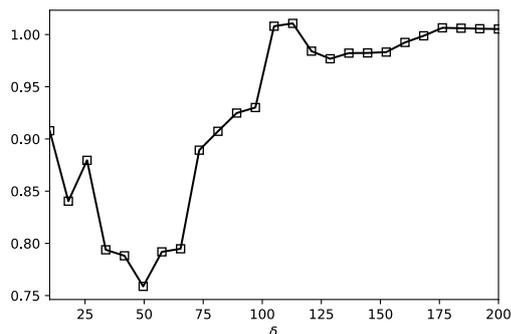
Após a normalização, os dois objetivos podem ser devidamente combinados. Por simplicidade, esta combinação é feita pela soma de ambos. Portanto, a função objetivo final para cada imagem de face k da metodologia proposta é

$$f_k(\lambda, \delta_1, \delta_2, \dots, \delta_i) = \mathcal{J}_1 + \mathcal{J}_2, \quad (58)$$

onde λ é o fator de ponderação entre os objetivos da métrica de similaridade implementada pelo FFD, δ o espaçamento entre pontos de controle, sendo i a quantidade de resoluções da malha, \mathcal{J}_1 é a soma das distâncias euclidianas entre os pontos fiduciais e \mathcal{J}_2 é a distância euclidiana entre as componentes principais.

Finalmente, o gráfico para a soma entre o objetivo 1 e o objetivo 2, matematicamente definida pela Equação (58), bem como a face resultante obtida pela minimização deste objetivo, podem ser visualizados na Figura 49.

Figura 49 – Gráfico para a soma entre o objetivo 1 e o objetivo 2 com δ na primeira resolução como variável de decisão e a deformação ótima de acordo com o objetivo proposto



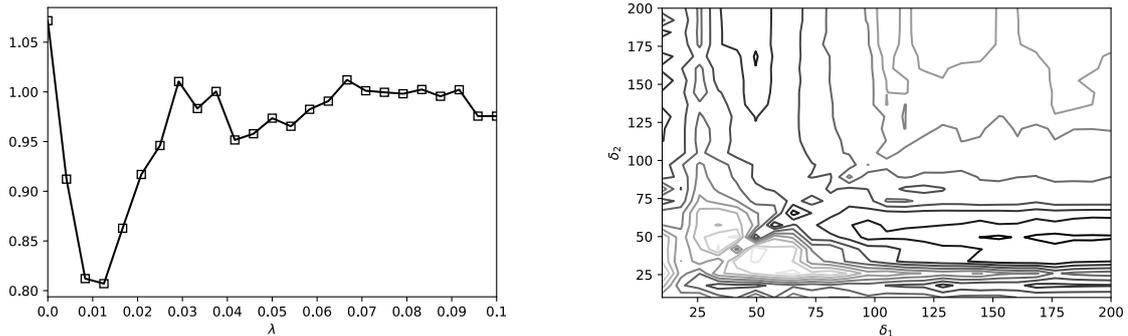
Fonte: Autor

(b) Imagem móvel deformada ótima



É interessante também visualizar a função objetivo para outras opções de variáveis de decisão. A Figura 50 mostra os gráficos da função objetivo para (a) duas resoluções com λ como variável de decisão e (b) duas resoluções com δ_1 e δ_2 como variáveis de decisão.

Figura 50 – Gráficos da soma entre o objetivo 1 e o objetivo 2 com (a) λ como variável de decisão e (b) δ_1 na primeira resolução e δ_2 na segunda resolução como variáveis de decisão



Fonte: Autor

4.3 OTIMIZAÇÃO ASSISTIDA POR METAMODELOS

Pode ser observado pelas figuras apresentadas na seção anterior que a função objetivo para o problema de otimização de hiperparâmetros do FFD possui um comportamento multi-modal e irregular. Este comportamento já é uma grande desmotivação para usar os algoritmos de otimização local.

Adicionalmente, a tarefa de registro de imagens pelo FFD pode ser extremamente custosa, e usar algoritmos evolutivos seria impraticável por exigir inúmeras avaliações da função objetivo, que são calculadas a partir da imagem resultante após a normalização.

Como este trabalho investiga os algoritmos de otimização assistida por metamodelos, a metodologia proposta irá amostrar o espaço de busca para construir um modelo aproximado. Em seguida, o modelo simplificado da função objetivo será atualizado incrementalmente, seguindo as premissas de exploração e exploração, até que um critério de parada seja satisfeito.

Começando pelo projeto de experimentos, como o registro de imagem por FFD é um processo computacional determinístico, a estratégia para amostragem foi por preenchimento ótimo por hipercubo latin. A implementação deste projeto de experimentos é feita pela biblioteca pyDOE⁵, um módulo em Python escrito em código aberto.

Em seguida, o modelo escolhido foi o *kriging*, por ser um modelo capaz de capturar as complexidades do espaço de busca da função objetivo e por já existir a implementação deste modelo em Python, o pyKriging⁶, também disponibilizado em código aberto.

⁵<<https://pythonhosted.org/pyDOE/>>.

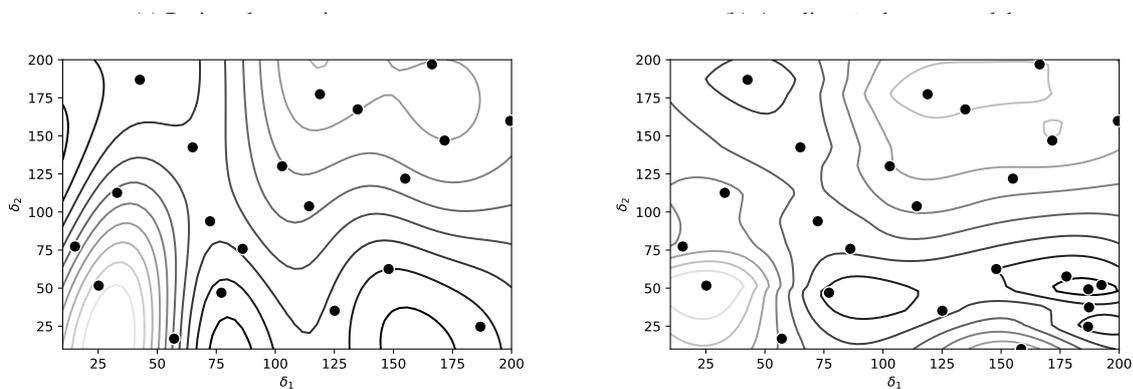
⁶<<http://pykriging.com/>>.

O último passo é a otimização e atualização do espaço de busca descrito pelo modelo seguindo as premissas de exploração e exploração. Estas premissas já fazem parte do modelo *kriging*, ou seja, o próprio modelo já calcula o erro e a saída esperada para determinar a próxima região a ser amostrada.

Portanto, os únicos parâmetros a serem definidos antes de iniciar a normalização não rígida de imagens assistida por metamodelos é a quantidade de amostras para o projeto de experimento e a quantidade de vezes que o modelo será atualizado.

Seguindo o mesmo exemplo da seção anterior, sendo as variáveis de decisão o espaçamento entre pontos de controle na primeira e segunda resolução, e adotando um projeto de experimentos com 20 amostras iniciais e 5 atualizações do modelo *kriging*, o espaço resultante para a função objetivo proposta neste trabalho pode ser visualizado pela Figura 51.

Figura 51 – Construção do espaço de busca pelo modelo *kriging*



Fonte: Autor

5 RESULTADOS E DISCUSSÃO

Os resultados deste trabalho são obtidos a partir da normalização de três tipos de bases de imagens de face, uma controlada, outra para *benchmarking* e outra do mundo real.

A base de imagens de face controlada é a *FEI Face Database*, detalhada no capítulo anterior. Esta base é considerada controlada pois todas as imagens de face são frontais e foram capturadas em um ambiente controlado.

A base de dados para *benchmarking* é a *Labeled Faces in the Wild*¹ (LFW), uma coleção de mais de 13000 imagens de pessoas famosas coletadas na Internet. Por padrão, utilizam-se as imagens da base LFW previamente alinhadas pelo algoritmo de Huang, Jain e Learned-Miller (2007). No entanto, como muitas das imagens não são frontais ou possuem uma grande variação de pose e expressão facial irrelevantes para o reconhecimento de faces, o registro não rígido destas imagens torna-se importante para reduzir estas variações. Uma amostra desta base pode ser visualizada pela Figura 52.

Figura 52 – Quinze amostras de imagens frontais de face da *Labeled Faces in the Wild*



Fonte: Autor

Observou-se que a face média da base LFW deixa de sofrer grandes alterações após 400 amostras. Também por questões de tempo computacional, a análise dos resultados para esta base será feita para 400 amostras.

Por fim, a base de dados do mundo real é a base de recém nascidos da UNIFESP, que contém 252 imagens. Esta base pertence a um projeto em parceria com a FEI, cujo objetivo é a classificação de dores em recém nascidos. Por se tratar de uma base do mundo real, as imagens não possuem nenhuma forma de alinhamento e o registro, tanto rígido, como não rígido, é necessário antes de prosseguir para o problema de classificação. Uma amostra desta base pode ser visualizada na Figura 53.

¹<http://vis-www.cs.umass.edu/lfw/>.

Figura 53 – Quinze amostras de imagens frontais de face da base da UNIFESP



Fonte: Autor

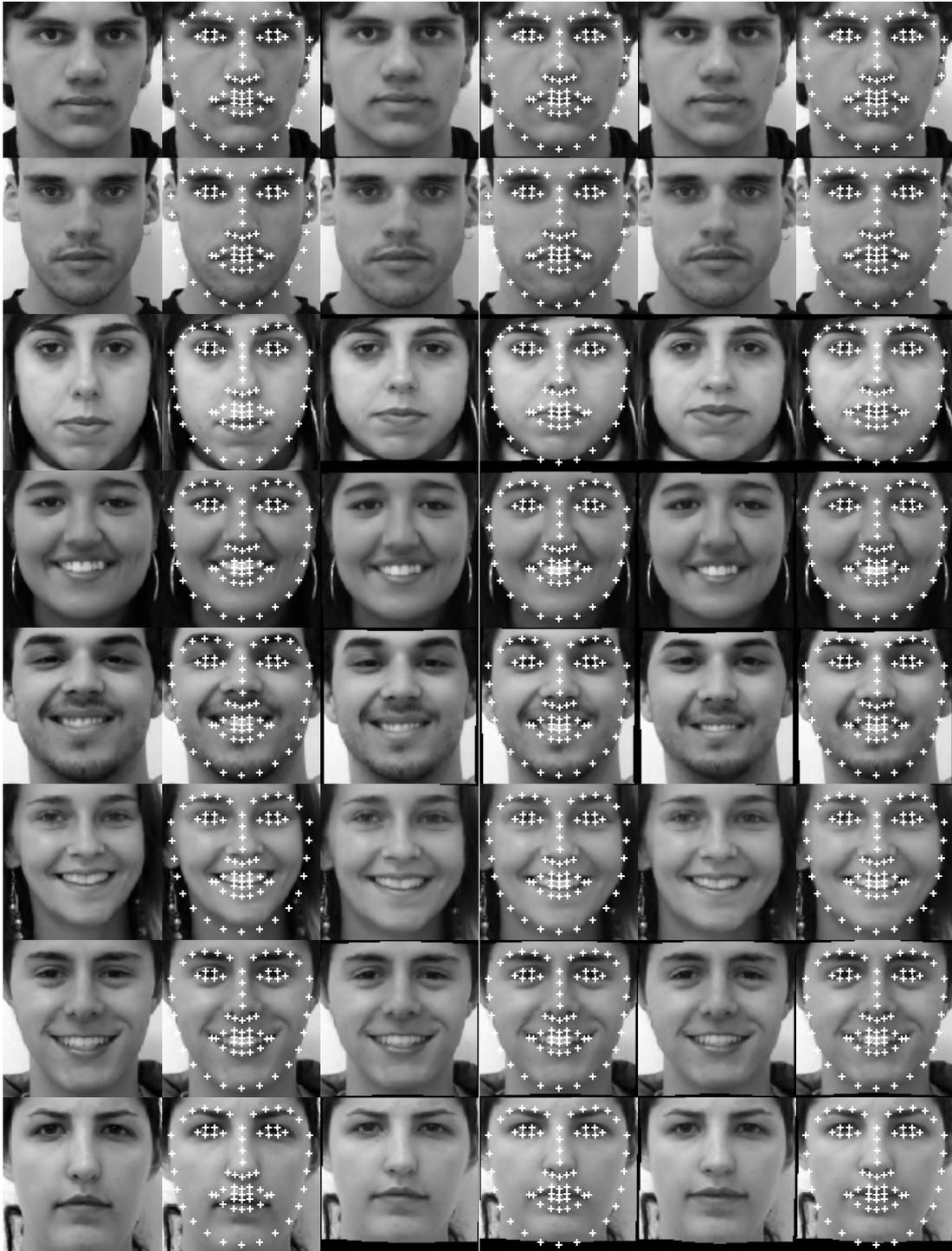
A primeira parte dos resultados é feita sobre uma amostragem de 16 imagens de cada base de dados e apresentada em um mosaico de figuras, onde as linhas são as amostras e as colunas são respectivamente: a imagem de face original, a imagem de face original com os pontos fiduciais do atlas, a imagem de face normalizada não rigidamente utilizando os valores recomendados de δ segundo Rueckert et al. (1999), a imagem de face normalizada não rigidamente utilizando os valores recomendados de δ segundo Rueckert et al. (1999) com os pontos fiduciais do atlas, a imagem de face normalizada não rigidamente utilizando os valores ótimos de δ segundo a função objetivo proposta e imagem de face normalizada não rigidamente utilizando os valores ótimos de δ segundo a função objetivo proposta com os pontos fiduciais do atlas.

Em seguida, é apresentado o comportamento da função objetivo aproximada no espaço de busca construído pelo metamodelo *kriging* durante o processo de otimização de δ das 16 amostras. Para a construção do modelo, foram utilizadas 11 amostras iniciais e 5 atualizações.

A segunda parte dos resultados apresenta o resultado da face média original, da face média normalizada não rigidamente utilizando os valores recomendados de δ e da face média normalizada não rigidamente utilizando os valores ótimos de δ , respectivamente para cada base de dados.

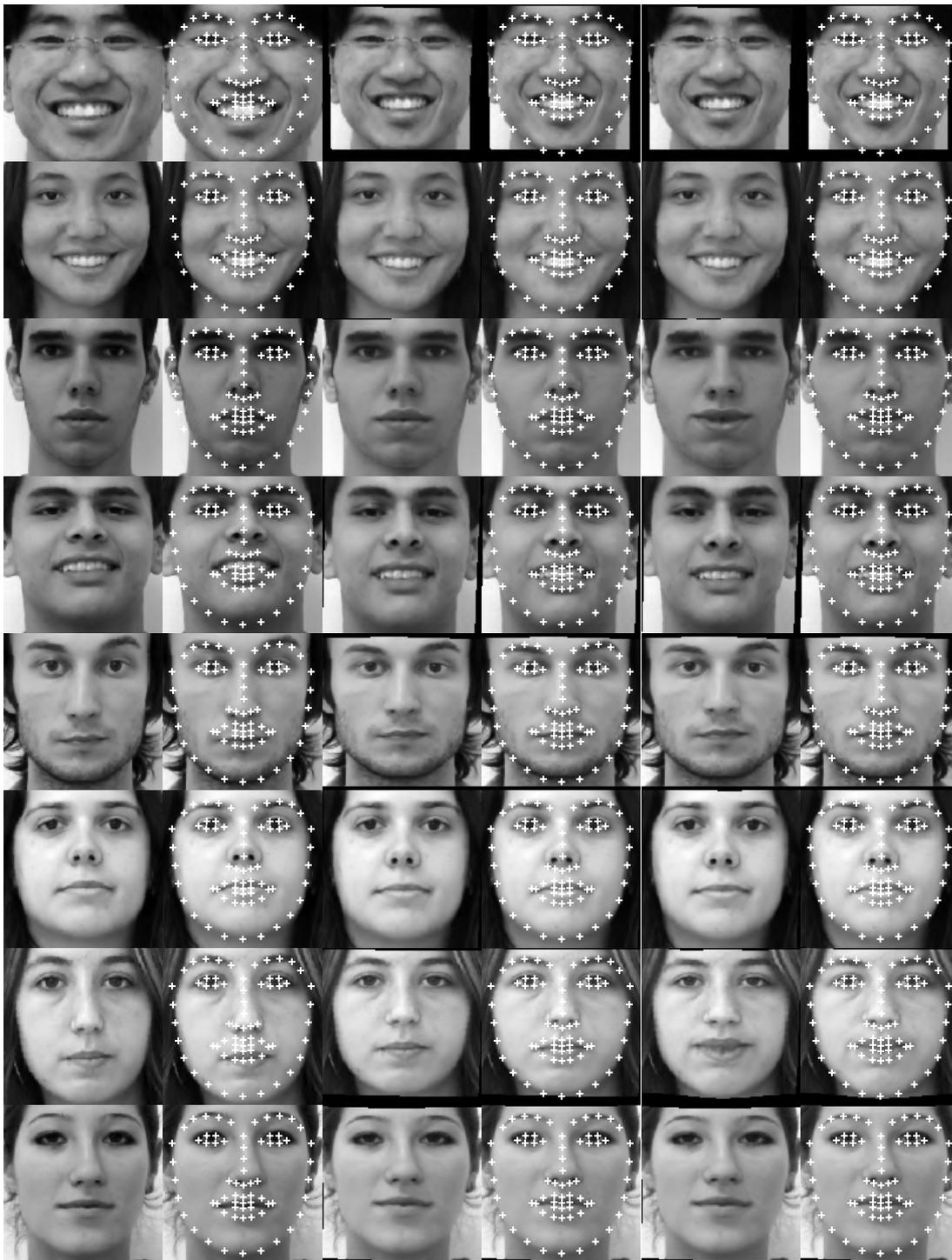
5.1 FEI FACE DATABASE

Figura 54 – Primeira parte das 16 amostras de imagem da *FEI Face Database* em forma de mosaico, seguindo a disposição previamente definida

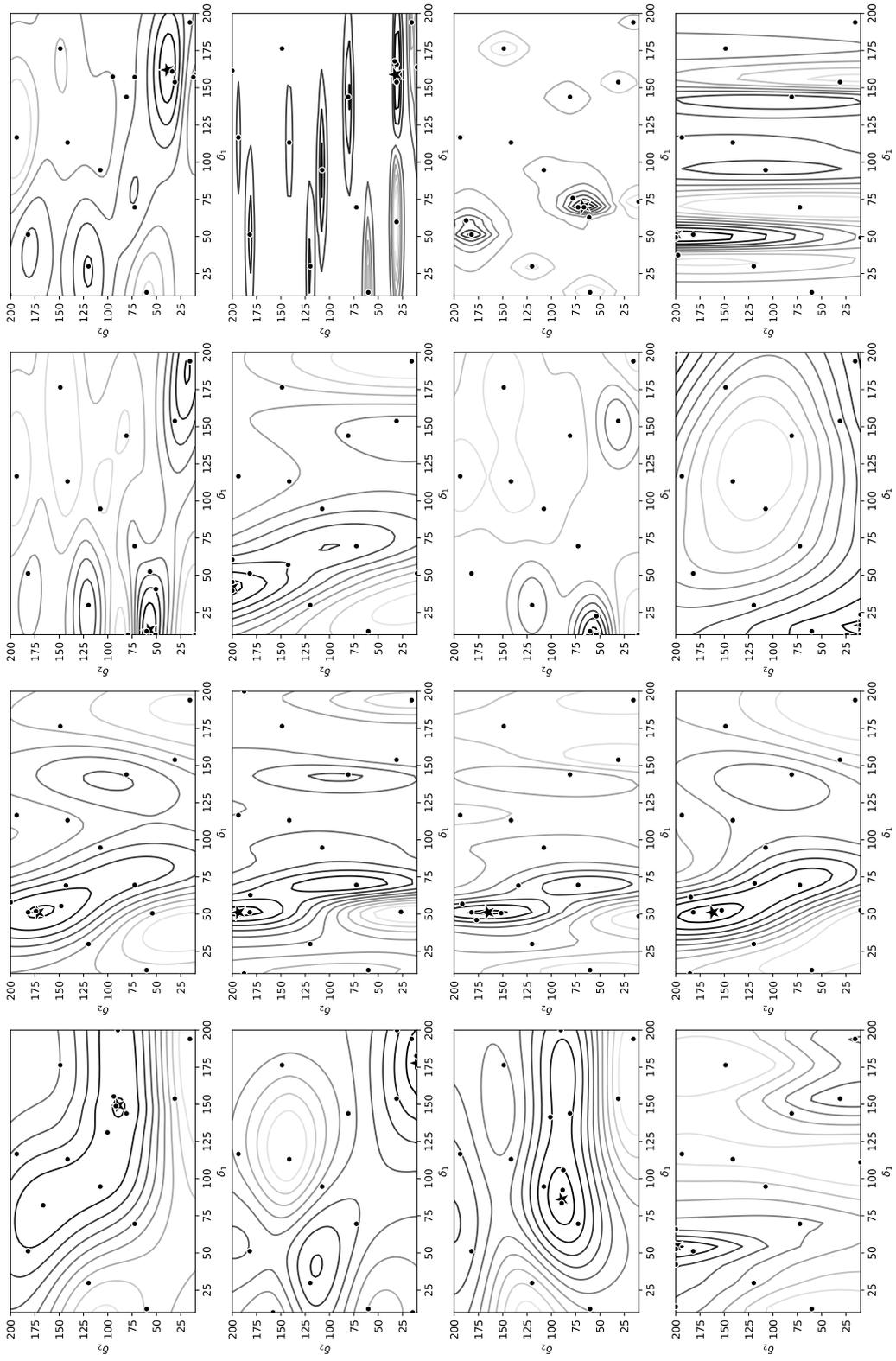


Fonte: Autor

Figura 55 – Segunda parte das 16 amostras de imagem da *FEI Face Database* em forma de mosaico, seguindo a disposição previamente definida

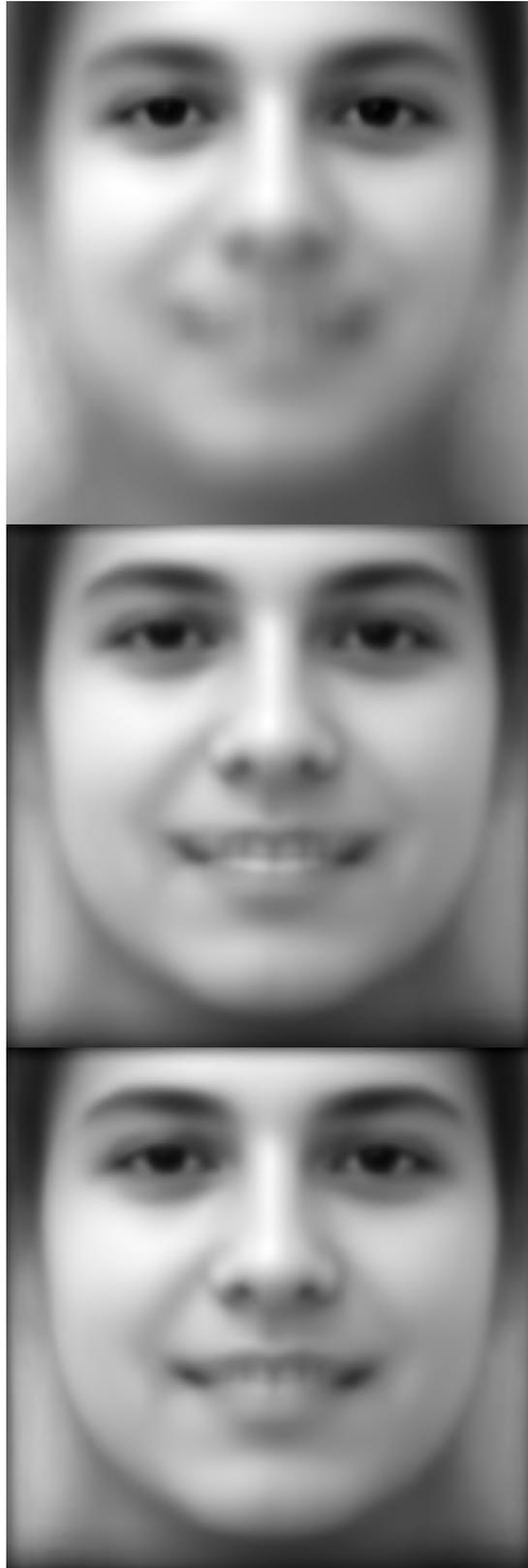


Fonte: Autor



Fonte: Autor

Figura 57 – Face média da *Fei Face Database* seguindo a disposição previamente definida



Fonte: Autor

5.2 LABELLED FACES IN THE WILD

Figura 58 – Primeira parte das 16 amostras de imagem da *Labelled Faces in the Wild* em forma de mosaico, seguindo a disposição previamente definida

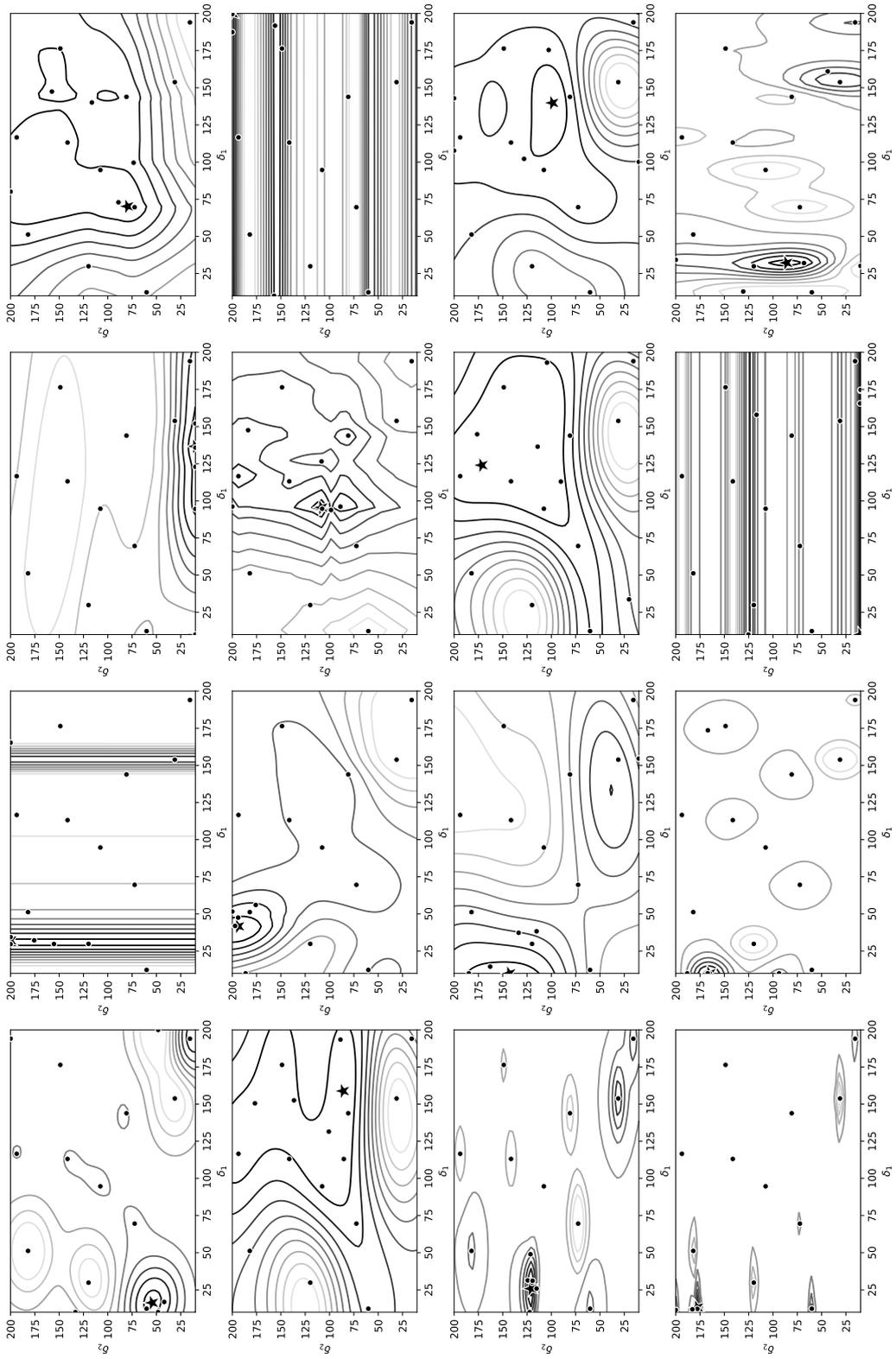


Fonte: Autor

Figura 59 – Segunda parte das 16 amostras de imagem da *Labelled Faces in the Wild* em forma de mosaico, seguindo a disposição previamente definida



Fonte: Autor



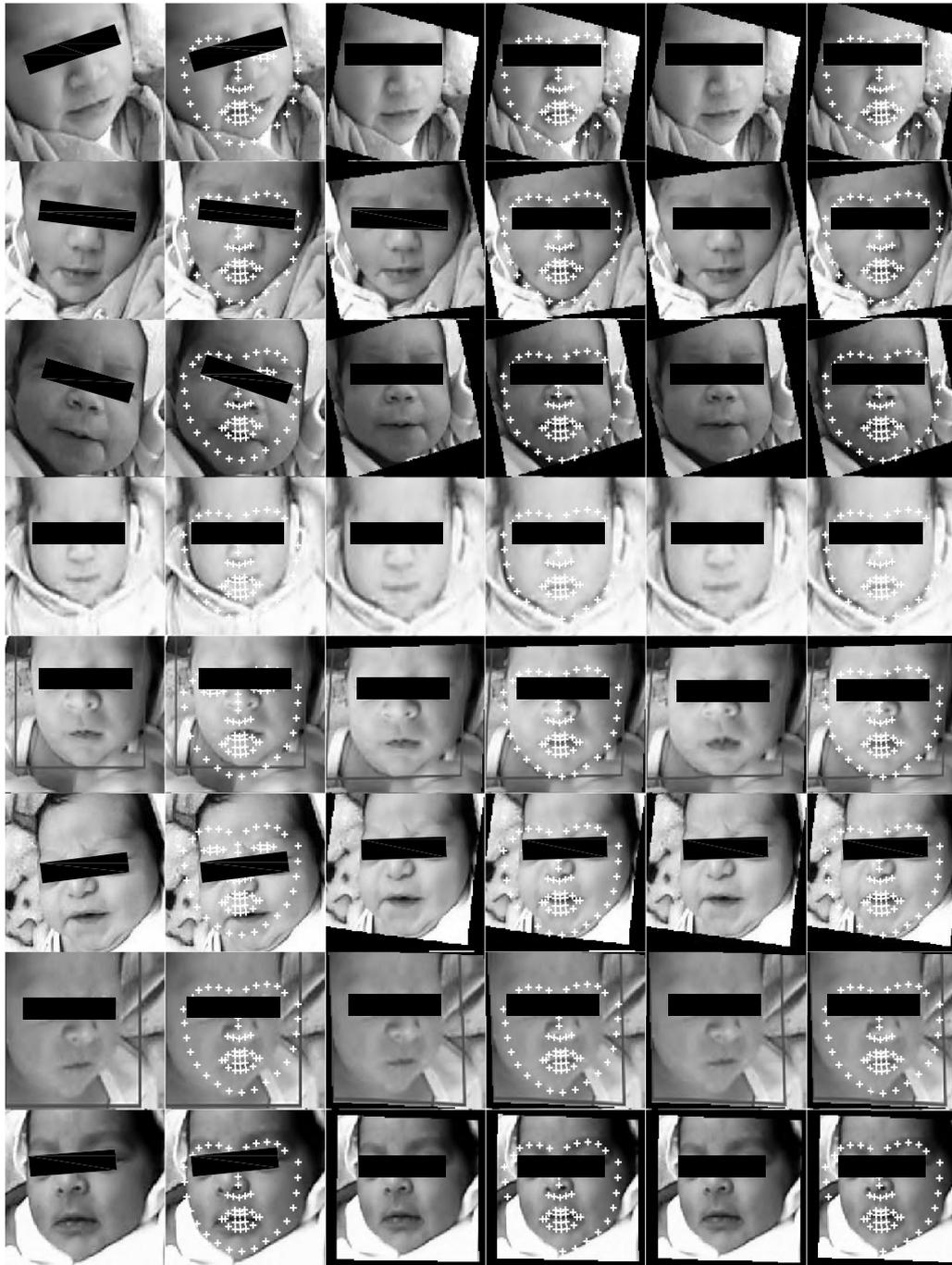
Fonte: Autor

Figura 61 – Face média da *Labelled Faces in the Wild* seguindo a disposição previamente definida



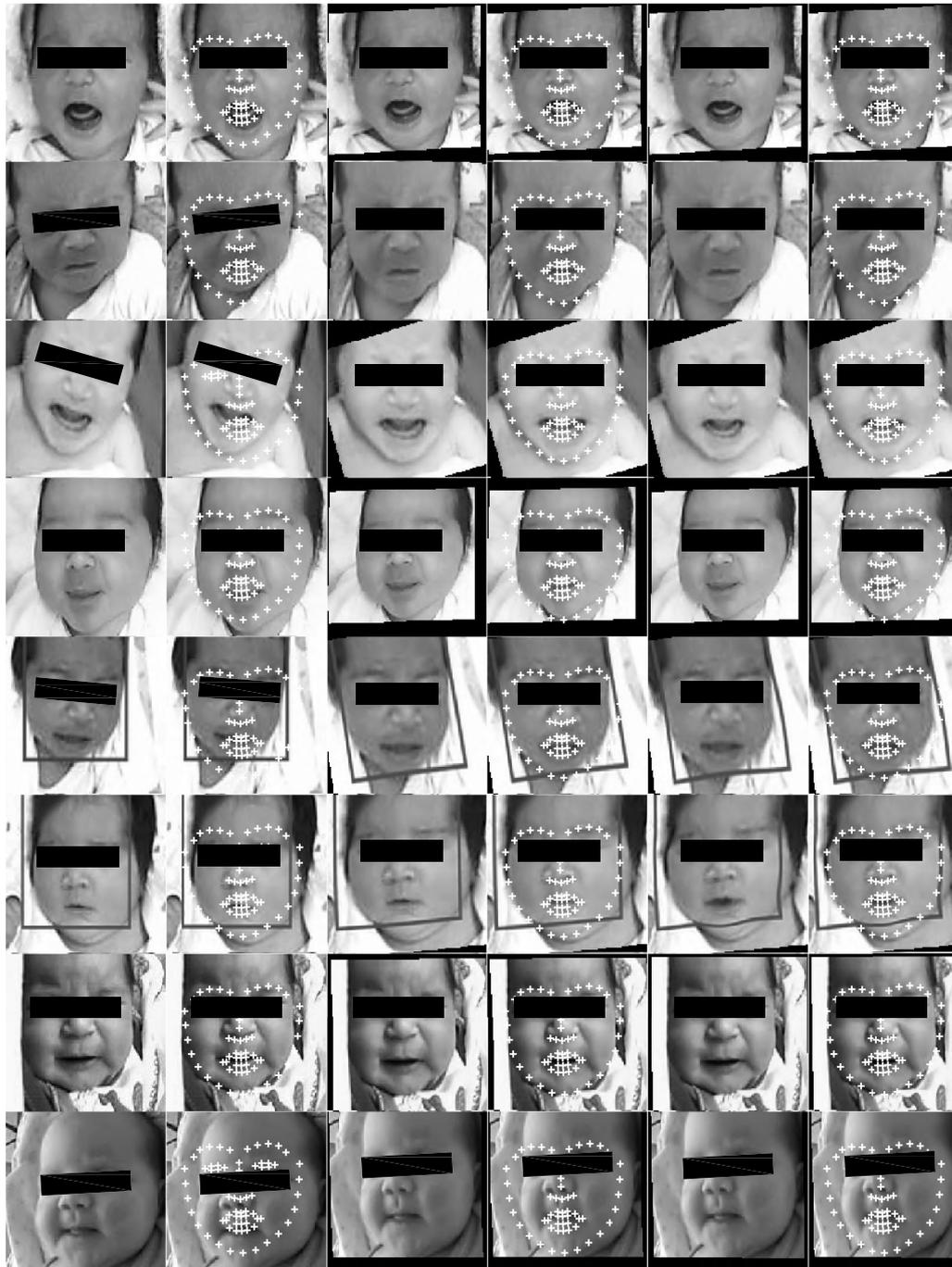
5.3 BASE DE FACES DE RECÉM NASCIDOS DA UNIFESP

Figura 62 – Primeira parte das 16 amostras de imagem da Base de Faces de Recém Nascidos da UNIFESP em forma de mosaico, seguindo a disposição previamente definida

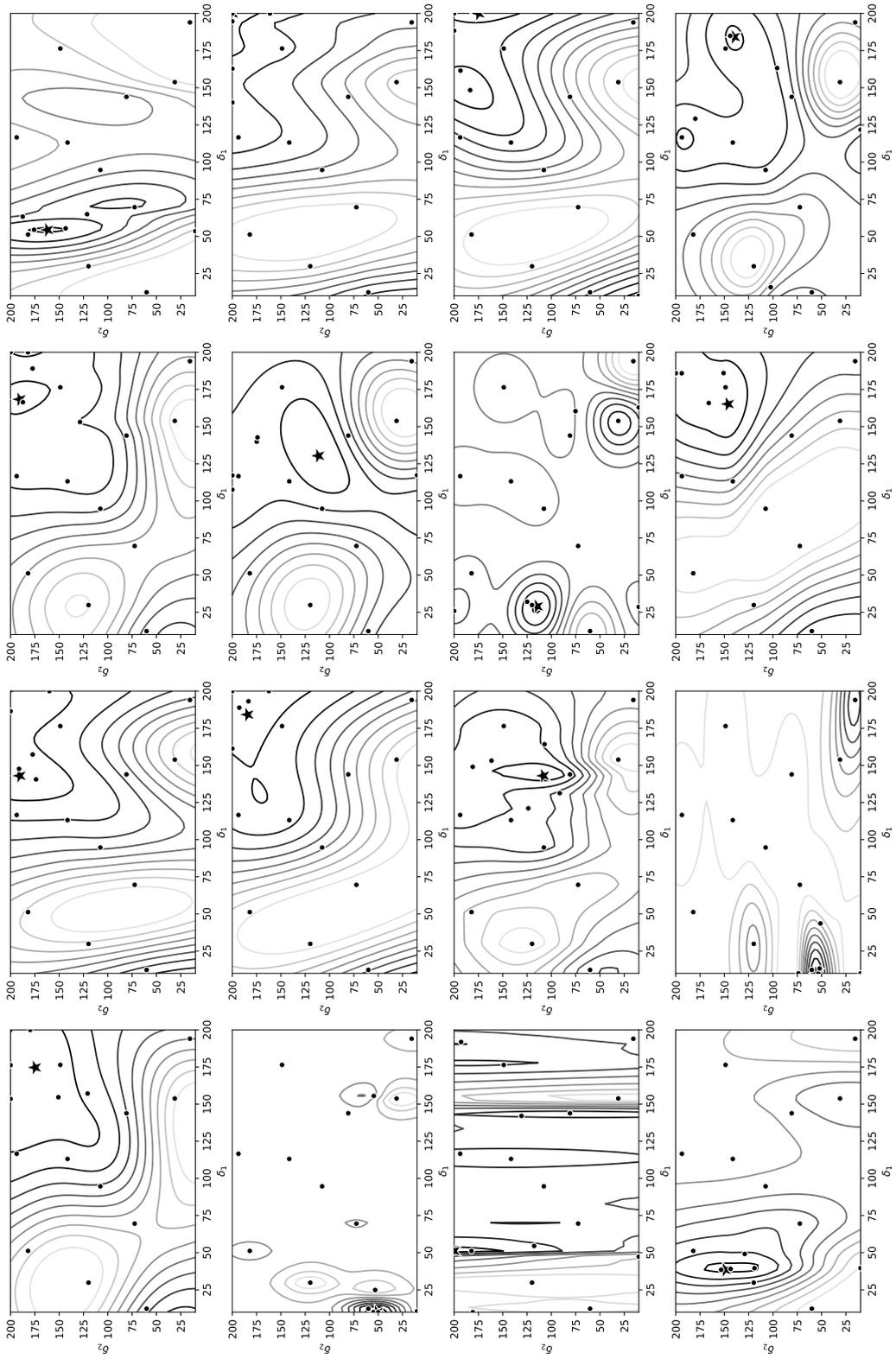


Fonte: Autor

Figura 63 – Segunda parte das 16 amostras de imagem da Base de Faces de Recém Nascidos da UNIFESP em forma de mosaico, seguindo a disposição previamente definida



Fonte: Autor



Fonte: Autor

Figura 65 – Face média da Base de Faces de Recém Nascidos da UNIFESP seguindo a disposição previamente definida



5.4 DISCUSSÃO DOS RESULTADOS

As primeiras duas colunas das Figuras 54, 55, 58, 59, 62 e 63 mostram a discrepância entre os pontos fiduciais da imagem de referência e a imagem de face original, ou seja, a necessidade pelo registo não rígido para o alinhamento das imagens. Já nas imagens normalizadas, é possível notar que os pontos fiduciais da imagem de referência estão mais coerentes com a imagem móvel, indicando que os principais elementos da face estão alinhados com o atlas.

Ao comparar o resultado individual entre o FFD com valores de δ recomendados e o FFD com valores de δ ótimos, a diferença, em muitos casos, é extremamente sutil. Esta sutileza mostra que a otimização do espaçamento entre pontos de controle atua apenas na transformação local como ajuste fino ao resultado final, que por sua vez é quase imperceptível visualmente, porém significativo sob uma perspectiva estatística das componentes principais da imagem.

As Figuras 56, 60 e 64 apresentam as aproximações por metamodelos da função objetivo proposta no espaços de busca. A partir destes resultados, é possível analisar o comportamento da transformação não rígida em função dos hiperparâmetros δ .

Foi observado durante o experimento que o hiperparâmetro λ não possui uma influência significativa na função objetivo proposta para imagens planas e, como usar duas variáveis de decisão facilita a visualização da função objetivo no espaço de busca para apresentação dos resultados, apenas δ_1 e δ_2 são analisados.

A principal observação é que as aproximações das funções compartilham pouca similaridade entre si, mesmo em casos pertencentes à mesma base de dados. Esta descoberta é importante, uma vez que caracteriza o problema de seleção do espaçamento entre pontos de controle como sendo uma tarefa complexa. No entanto, alguns espaços de busca, principalmente aqueles das imagens da base LFW (Figura 60), comportaram-se de forma extremamente ruidosa. Foi constatado que este problema é gerado durante a modelagem *Kriging*. Mais especificamente, trata-se de uma singularidade na matriz de correlação que ocorre durante a própria implementação computacional deste modelo.

Ainda nos resultados das aproximações no espaço de busca, é possível notar que quando $\delta_1 \ll \delta_2$ significa que o FFD não precisa da segunda resolução já que a imagem sofreu a deformação necessária logo na primeira resolução. Esta afirmação pode ser verificada pois um alto valor de δ elimina a capacidade do modelo ajustar as características locais, ou seja, se na segunda resolução o espaçamento entre pontos de controle for muito alto, não existem mais elementos locais para serem deformados.

As Figuras 57, 61 e 65 apresentam a última análise dos resultados, que é a média das imagens para cada base de faces. Como estes resultados acabam acumulando os efeitos dos ajustes finos causados pela otimização de δ , é possível notar um aumento da definição da média das imagens normalizadas com valores de δ otimizados quando comparada com a definição da média das imagens normalizadas com valores de δ recomendados.

Este aumento da definição foi possível por conta do maior grau de deformação permitido pela seleção otimizada dos valores de δ , ou seja, ao invés de adotar sempre o mesmo espaçamento entre pontos de controle, como faz o algoritmo original FFD, a normalização não rígida assistida por metamodelos leva os valores de δ ao limite do grau de deformação aceitável, que é aquela que atende ao máximo a premissa de eliminar as características locais irrelevantes da imagem, mas preservando suas características principais.

Em relação ao custo computacional, a normalização não rígida de imagens assistida por metamodelos apresenta uma grande desvantagem quando comparada ao FFD. Esta desvantagem acontece pois a metodologia proposta acrescenta uma etapa adicional, que é a construção de um metamodelo. Por outro lado, como a transformação por FFD não adota uma função objetivo para avaliar a qualidade da normalização, esta avaliação acaba sendo realizada pela análise visual de uma pessoa, que é uma tarefa extremamente custosa quando comparada com a ordem de grandeza do tempo computacional, principalmente nos casos em que encontrar uma boa deformação leva inúmeras tentativas.

6 CONCLUSÃO

Este estudo desenvolveu uma metodologia para otimização assistida por metamodelos dos hiperparâmetros do *Free-Form Deformation* para imagens de faces 2D. Para o desenvolvimento desta metodologia, o conteúdo deste trabalho foi apresentado em duas partes.

A primeira parte, de caráter mais teórico, classificou os problemas de otimização em três grandes categorias: (i) otimização local, (ii) otimização por meta-heurísticas e (iii) otimização de funções caixa preta por metamodelos. Com esta classificação, este trabalho construiu uma nova taxonomia que une os conceitos de otimização matemática e aprendizado de máquina.

A segunda parte, de teor mais prático, fundamentou as diversas técnicas de modelos de transformação de imagens utilizando a contextualização de otimização matemática descrita na parte anterior. Consequentemente, esta parte foi desenvolvida com maior ênfase na definição das funções-objetivo e visualização das aproximações obtidas nos respectivos espaços de busca destes modelos.

A partir da assimilação destas duas partes, este trabalho identificou a oportunidade de aplicar os conceitos teóricos em um problema do mundo real: o registro não rígido de imagens. Esta aplicação, denominada de normalização espacial não rígida de imagens assistida por metamodelos, aproveitou a terminologia explorada na fundamentação teórica para conceber uma função objetivo e algoritmo de otimização implementados em uma ferramenta disponível em código aberto.

Os resultados mostram que o registro não rígido de imagens pode se beneficiar ao optar por uma escolha inteligente de seus hiperparâmetros. Como o maior compromisso de deformar uma imagem é eliminar seus elementos irrelevantes, uma abordagem que leva em consideração suas componentes principais mostra-se como uma forma de eliminar as incertezas da escolha de tais parâmetros.

Quanto aos metamodelos, observa-se que sua grande vantagem está na maior probabilidade de encontrar a região do mínimo global, quando comparada aos métodos de otimização local, e na menor quantidade de avaliação da função objetivo para encontrar o mínimo global, quando comparada às meta-heurísticas. Em outras palavras, a otimização assistida por metamodelos aumenta as chances de encontrar a região do mínimo global com poucas avaliações da função objetivo.

Por se tratar de um problema ainda pouco explorado, ainda há considerações com grande potencial a serem melhor investigadas. Como trabalhos futuros, ressaltam-se as seguintes considerações:

- a) investigar o comportamento da normalização espacial de imagens com outros modelos além do *kriging*;
- b) adotar uma função objetivo que leve em consideração a base de imagens como um todo, ao invés de uma função para cada imagem;

- c) estudar a função objetivo proposta a partir de outros algoritmos de otimização, em especial os algoritmos evolutivos assistidos por metamodelos;
- d) otimizar a questão da implementação computacional do algoritmo utilizado neste trabalho, especialmente na parte de treinamento e atualização do metamodelo, que são as etapas com maior tempo de processamento;
- e) explorar outros modelos de transformação não rígida que não sejam o FFD;
- f) analisar a premissa da função objetivo proposta em outros domínios que não sejam de imagens de faces. Por exemplo, imagens médicas, que é o domínio em que o FFD foi inicialmente concebido;
- g) apesar da diferença entre os resultados do registro não rígido com valores de δ recomendados e do registro não rígido com valores de δ ótimos serem visualmente pouco perceptíveis, é importante comparar a eficácia entre ambos os resultados por meio de uma métrica estatística complementar à média, como a variância, curtose e assimetria da distribuição, ou pelo ganho de precisão em um problema real reconhecimento de imagens.

Recentemente, Huang et al. (2012) mostrou que o alinhamento de imagens pode ser alternativamente realizado por *Deep Learning*. No entanto, apesar desta técnica ser uma boa alternativa para os métodos tradicionais, suas aplicações, em sua maioria, estão restritas às deformações globais.

Tendo em vista o grande sucesso da aplicação multidisciplinar do *Deep Learning*, tanto na área científica, como na indústria, o uso de *Deep Learning* para normalização espacial não rígida de imagens é uma oportunidade para trabalhos futuros bastante promissora, porém ainda pouco explorada.

REFERÊNCIAS

ACKLEY, David H. **A connectionist machine for genetic hillclimbing**. Boston, MA: Springer, 1987.

ATKINSON, Kendall E. **An introduction to numerical analysis**. 2nd. [S.l.]: Wiley India Pvt. Limited, 2008.

BACK, Thomas. **Evolutionary algorithms in theory and practice**: Evolution strategies, evolutionary programming, genetic algorithms. New York, NY: Oxford University Press, 1996.

BACK, Thomas; HAMMEL, Ulrich; SCHWEFEL, Hans-Paul. Evolutionary computation: Comments on the history and current state. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 1, n. 1, p. 3–17, 1997. Disponível em: <<https://doi.org/10.1109/4235.585888>>. Acesso em: 15 mar. 2018.

BAZARAA, Mokhtar S.; SHERALI, Hanif D.; SHETTY, Chitharanjan M. **Nonlinear programming**: Theory and algorithms. Hoboken, NJ: John Wiley & Sons, 2013.

BELLMAN, Richard. Dynamic programming and Lagrange multipliers. **Proceedings of the National Academy of Sciences**, National Academy of Sciences, v. 42, n. 10, p. 767–769, 1956. Disponível em: <<http://www.pnas.org/content/42/10/767>>. Acesso em: 15 mar. 2018.

BERGSTRA, James S. et al. Algorithms for hyper-parameter optimization. In: **ADVANCES in Neural Information Processing Systems 24**. Granada, Spain: Curran Associates, Inc., 2011. p. 2546–2554. Disponível em: <<http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf>>. Acesso em: 15 mar. 2018.

BISHOP, Christopher M. **Pattern recognition and mMachine learning**. New York, NY: Springer, 2006.

BOX, George E. P.; DRAPER, Norman R. **Empirical model-building and response surfaces**. Hoboken, NJ: John Wiley & Sons, 1987. v. 424.

BOX, George E. P.; WILSON, Kenneth B. On the experimental attainment of optimum conditions. In: **BREAKTHROUGHS in Statistics: Methodology and Distribution**. New York, NY: Springer, 1992. p. 270–310. Disponível em: <https://doi.org/10.1007/978-1-4612-4380-9_23>. Acesso em: 15 mar. 2018.

BOYD, Stephen; VANDENBERGHE, Lieven. **Convex optimization**. Cambridge, UK: Cambridge University Press, 2004.

BRANIN, Franklin H. Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. **IBM Journal of Research and Development**, International Business Machines Corporation, v. 16, n. 5, p. 504–522, 1972. Disponível em: <<https://doi.org/10.1147/rd.165.0504>>. Acesso em: 15 mar. 2018.

BROCHU, Eric; CORA, Vlad M.; FREITAS, Nando de. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. **arXiv preprint arXiv:1012.2599**, abs/1012.2599, 2010. Disponível em: <<http://arxiv.org/abs/1012.2599>>. Acesso em: 15 mar. 2018.

BROWN, Lisa G. A survey of image registration techniques. **ACM Computing Surveys**, Association for Computing Machinery, New York, NY, v. 24, n. 4, p. 325–376, dez. 1992. Disponível em: <<http://doi.acm.org/10.1145/146370.146374>>. Acesso em: 15 mar. 2018.

BROYDEN, Charles G. The convergence of a class of double-rank minimization algorithms 1. general considerations. **IMA Journal of Applied Mathematics**, Oxford University Press, v. 6, n. 1, p. 76–90, 1970. Disponível em: <<https://doi.org/10.1093/imamat/6.1.76>>. Acesso em: 15 mar. 2018.

BUHMANN, Martin D. **Radial basis functions: Theory and implementations**. Cambridge, UK: Cambridge University Press, 2003. v. 12.

BURER, Samuel; LETCHFORD, Adam N. On nonconvex quadratic programming with box constraints. **Journal on Optimization**, Society for Industrial e Applied Mathematics, v. 20, n. 2, p. 1073–1089, 2009. Disponível em: <<https://doi.org/10.1137/080729529>>. Acesso em: 15 mar. 2018.

COELLO, Carlos A C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 191, n. 11, p. 1245–1287, 2002. Disponível em: <[https://doi.org/10.1016/S0045-7825\(01\)00323-1](https://doi.org/10.1016/S0045-7825(01)00323-1)>. Acesso em: 15 mar. 2018.

COELLO, Carlos A. C.; LAMONT, Gary B.; VAN VELDHUIZEN, David A. et al. **Evolutionary algorithms for solving multi-objective problems**. New York, NY: Springer, 2007. v. 5.

COOTES, Timothy F. et al. Active shape models: Their training and application. **Computer Vision and Image Understanding**, v. 61, n. 1, p. 38–59, 1995. Disponível em: <<https://doi.org/10.1006/cviu.1995.1004>>. Acesso em: 15 mar. 2018.

COVER, Thomas M.; THOMAS, Joy A. **Elements of information theory**. Hoboken, NJ: John Wiley & Sons, 2012.

CYBENKO, George. Approximation by superpositions of a sigmoidal function. **Mathematics of Control, Signals and Systems**, Springer, v. 2, n. 4, p. 303–314, dez. 1989. Disponível em: <<https://doi.org/10.1007/BF02551274>>. Acesso em: 15 mar. 2018.

DANTZIG, George B. **Linear programming and extensions**. Princeton, NJ: Princeton University Press, 2016.

_____. **Maximization of a linear function of variables subject to linear inequalities, in activity analysis of production and allocation**. Stanford, CA: Stanford University Press, 1951.

DARWIN, Charles. **On the origin of species by means of natural selection**. London, UK: John Murray, 1859. Disponível em: <http://darwin-online.org.uk/EditorialIntroductions/Chancellor_vanWyhe_Origin1st.html>. Acesso em: 15 mar. 2018.

DIJKSTRA, Edsger W. A note on two problems in connexion with graphs. **Numerische Mathematik**, Springer, v. 1, n. 1, p. 269–271, 1959. Disponível em: <<https://doi.org/10.1007/BF01386390>>. Acesso em: 15 mar. 2018.

DORIGO, Marco; MANIEZZO, Vittorio; COLORNI, Alberto. Ant system: Optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, IEEE, v. 26, n. 1, p. 29–41, 1996. Disponível em: <<https://doi.org/10.1109/3477.484436>>. Acesso em: 15 mar. 2018.

EGGENSPERGER, Katharina et al. Efficient benchmarking of hyperparameter optimizers via surrogates. In: **PROCEEDINGS of the Twenty-Ninth AAAI Conference on Artificial Intelligence**. Austin, TX: AAAI Press, 2015. p. 1114–1120. Disponível em: <<http://doi.acm.org/10.1145/2887007.2887162>>. Acesso em: 15 mar. 2018.

FEURER, Matthias; SPRINGENBERG, Jost T.; HUTTER, Frank. Initializing bayesian hyperparameter optimization via meta-learning. In: **PROCEEDINGS of the Twenty-Ninth AAAI Conference on Artificial Intelligence**. Austin, TX: AAAI Press, 2015. p. 1128–1135. Disponível em: <<http://dl.acm.org/citation.cfm?id=2887007.2887164>>. Acesso em: 15 mar. 2018.

FIACCO, Anthony V.; MCCORMICK, Garth P. **Nonlinear programming: Sequential unconstrained minimization techniques**. Philadelphia, PA: Society for Industrial e Applied Mathematics, 1990.

FLETCHER, Roger; POWELL, Michael J. D. A rapidly convergent descent method for minimization. **The Computer Journal**, Oxford University Press, v. 6, n. 2, p. 163–168, 1963. Disponível em: <<https://doi.org/10.1093/comjnl/6.2.163>>. Acesso em: 15 mar. 2018.

FOGEL, David B. An introduction to simulated evolutionary optimization. **IEEE Transactions on Neural Networks**, IEEE, v. 5, n. 1, p. 3–14, 1994. Disponível em: <<https://doi.org/10.1109/72.265956>>. Acesso em: 15 mar. 2018.

FOGEL, Lawrence J.; OWENS, Alvin J.; WALSH, Michael J. **Artificial intelligence through simulated evolution**. New York, NY: John Wiley & Sons, Inc., 1966.

FORRESTER, Alexander I. J.; KEANE, Andy J. Recent advances in surrogate-based optimization. **Progress in Aerospace Sciences**, Elsevier, v. 45, n. 1, p. 50–79, 2009. Disponível em: <<https://doi.org/10.1016/j.paerosci.2008.11.001>>. Acesso em: 15 mar. 2018.

FRIEDMAN, Jerome H. Multivariate Adaptive Regression Splines. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 19, n. 1, p. 1–67, 1991. Disponível em: <<http://www.jstor.org/stable/2241837>>. Acesso em: 15 mar. 2018.

FUKUNAGA, Keinosuke. **Introduction to statistical pattern recognition**. San Diego, CA: Academic Press, 2013.

GLOVER, Fred. Future paths for integer programming and links to artificial intelligence. **Computers & Operations Research**, Elsevier, v. 13, n. 5, p. 533–549, 1986. Disponível em: <[https://doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/10.1016/0305-0548(86)90048-1)>. Acesso em: 15 mar. 2018.

GONZALEZ, Rafael C.; WOODS, Richard E. **Digital Image Processing**. 3rd. Upper Saddle River, NJ: Prentice Hall, 2006.

HANSEN, Nikolaus; OSTERMEIER, Andreas. Completely derandomized self-adaptation in evolution strategies. **Evolutionary Computation**, MIT Press, v. 9, n. 2, p. 159–195, 2001.

HESTENES, Magnus R.; STIEFEL, Eduard. Methods of conjugate gradients for solving linear systems. **Journal of Research**, National Bureau of Standards, v. 49, n. 6, p. 409–436, 1952. Disponível em: <<http://dx.doi.org/10.6028/jres.049.044>>. Acesso em: 15 mar. 2018.

HOLDEN, Mark. A review of geometric transformations for nonrigid body registration. **IEEE Transactions on Medical Imaging**, v. 27, n. 1, p. 111–128, jan. 2008. Disponível em: <<https://doi.org/10.1109/TMI.2007.904691>>. Acesso em: 15 mar. 2018.

HOLLAND, John H. **Adaptation in natural and artificial systems: An introductory analysis with application to biology, control, and artificial intelligence**. Ann Arbor, MI: University of Michigan Press, 1975.

HOOKE, Robert; JEEVES, T. A. "Direct Search" Solution of Numerical and Statistical Problems. **Journal of the ACM**, Association for Computing Machinery, New York, NY, v. 8, n. 2,

p. 212–229, abr. 1961. Disponível em: <<http://doi.acm.org/10.1145/321062.321069>>. Acesso em: 15 mar. 2018.

HUANG, Gary B.; JAIN, Vidit; LEARNED-MILLER, Erik G. Unsupervised joint alignment of complex images. In: 2007 IEEE 11th International Conference on Computer Vision. Rio de Janeiro, RJ: IEEE, out. 2007. p. 1–8. Disponível em: <<https://doi.org/10.1109/ICCV.2007.4408858>>. Acesso em: 15 mar. 2018.

HUANG, Gary B. et al. Learning to align from scratch. In: ADVANCES in Neural Information Processing Systems 25. Lake Tahoe, NV: Curran Associates, Inc., 2012. p. 764–772. Disponível em: <<http://papers.nips.cc/paper/4769-learning-to-align-from-scratch.pdf>>. Acesso em: 15 mar. 2018.

JAMIL, Momin; YANG, Xin-She. A literature survey of benchmark functions for global optimisation problems. **International Journal of Mathematical Modelling and Numerical Optimisation**, Inderscience Publishers Ltd, v. 4, n. 2, p. 150–194, 2013. Disponível em: <<https://doi.org/10.1504/IJMMNO.2013.055204>>. Acesso em: 15 mar. 2018.

JIN, Ruichen; CHEN, Wei; SIMPSON, Timothy W. Comparative studies of metamodelling techniques under multiple modelling criteria. **Structural and Multidisciplinary Optimization**, Springer, v. 23, n. 1, p. 1–13, 2001. Disponível em: <<https://doi.org/10.1007/s00158-001-0160-4>>. Acesso em: 15 mar. 2018.

JOHNSON, Richard A.; WICHERN, Dean. **Multivariate Analysis**. Hoboken, NJ: John Wiley & Sons, 2014. v. 4.

JONES, Donald R. A taxonomy of global optimization methods based on response surfaces. **Journal of Global Optimization**, Springer, v. 21, n. 4, p. 345–383, 2001. Disponível em: <<https://doi.org/10.1023/A:1012771025575>>. Acesso em: 15 mar. 2018.

JONES, Donald R.; SCHONLAU, Matthias; WELCH, William J. Efficient global optimization of expensive black-box functions. **Journal of Global Optimization**, Springer, v. 13, n. 4, p. 455–492, 1998. Disponível em: <<https://doi.org/10.1023/A:1008306431147>>. Acesso em: 15 mar. 2018.

KAZEMI, Vahid; SULLIVAN, Josephine. One millisecond face alignment with an ensemble of regression trees. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH: IEEE, jun. 2014. p. 1867–1874. Disponível em: <<https://doi.org/10.1109/CVPR.2014.241>>. Acesso em: 15 mar. 2018.

KENNEDY, James. Particle swarm optimization. In: *ENCYCLOPEDIA of Machine Learning*. Boston, MA: Springer, 2011. p. 760–766. Disponível em: <https://doi.org/10.1007/978-0-387-30164-8_630>. Acesso em: 15 mar. 2018.

KIEFER, Jack. Sequential minimax search for a maximum. **Proceedings of the American Mathematical Society**, v. 4, n. 3, p. 502–506, 1953. Disponível em: <<https://doi.org/10.1090/S0002-9939-1953-0055639-3>>. Acesso em: 15 mar. 2018.

KIRKPATRICK, Scott; GELATT, C. Daniel; VECCHI, Mario P. et al. Optimization by simulated annealing. **Science**, American Association for the Advancement of Science, v. 220, n. 4598, p. 671–680, 1983. Disponível em: <<https://doi.org/10.1126/science.220.4598.671>>. Acesso em: 15 mar. 2018.

KLEIN, Stefan; STARING, Marius; PLUIM, Josien P. W. Evaluation of optimization methods for nonrigid medical image registration using mutual information and B-splines. **IEEE Transactions on Image Processing**, v. 16, n. 12, p. 2879–2890, dez. 2007. Disponível em: <<https://doi.org/10.1109/TIP.2007.909412>>. Acesso em: 15 mar. 2018.

KUHN, Harold W.; TUCKER, Albert W. Nonlinear Programming. In: *PROCEEDINGS of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, CA: University of California Press, 1951. p. 481–492. Disponível em: <<http://projecteuclid.org/euclid.bsm/1200500249>>. Acesso em: 15 mar. 2018.

LARRAÑAGA, Pedro; LOZANO, Jose A. **Estimation of distribution algorithms: A new tool for evolutionary computation**. Norwell, MA: Kluwer Academic Publishers, 2002.

LAWLER, Eugene L.; WOOD, David E. Branch-and-bound methods: A survey. **Operations Research, INFORMS**, v. 14, n. 4, p. 699–719, 1966. Disponível em: <<https://doi.org/10.1287/opre.14.4.699>>. Acesso em: 15 mar. 2018.

LEE, Seungyong; WOLBERG, George; SHIN, Sung Y. Scattered data interpolation with multilevel B-splines. **IEEE Transactions on Visualization and Computer Graphics**, v. 3, n. 3, p. 228–244, jul. 1997. Disponível em: <<https://doi.org/10.1109/2945.620490>>. Acesso em: 15 mar. 2018.

LEWIS, John P. Fast template matching. In: *VISION interface*. [S.l.: s.n.], 1995. v. 95, p. 15–19. Disponível em: <<https://pdfs.semanticscholar.org/57d7/97a1389ed6211ef39e203eecabcd0d7e37e5.pdf>>. Acesso em: 15 mar. 2018.

MAES, Frederik et al. Multimodality image registration by maximization of mutual information. **IEEE Transactions on Medical Imaging**, v. 16, n. 2, p. 187–198, abr. 1997. Disponível em: <<https://doi.org/10.1109/42.563664>>. Acesso em: 15 mar. 2018.

MAINTZ, J. B. Antoine; VIERGEVER, Max A. A survey of medical image registration. **Medical Image Analysis**, Elsevier, v. 2, n. 1, p. 1–36, 1998. Disponível em: <[https://doi.org/10.1016/S1361-8415\(01\)80026-8](https://doi.org/10.1016/S1361-8415(01)80026-8)>. Acesso em: 15 mar. 2018.

MCCULLOCH, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943. Disponível em: <<https://doi.org/10.1007/BF02478259>>. Acesso em: 15 mar. 2018.

NELDER, John A.; MEAD, Roger. A simplex method for function minimization. **The Computer Journal**, Oxford University Press, v. 8, n. 4, p. 308–313, 1965. Disponível em: <<https://doi.org/10.1093/comjnl/7.4.308>>. Acesso em: 15 mar. 2018.

NESTEROV, Yurii; NEMIROVSKII, Arkadii. **Interior-point polynomial algorithms in convex programming**. Philadelphia, PA: Society for Industrial e Applied Mathematics, 1994.

OPPENHEIM, Alan V.; SCHAFER, Ronald W. **Discrete-Time Signal Processing**. 3rd. Upper Saddle River, NJ: Prentice Hall, 2009.

POWELL, Michael J. D. **A method for non-linear constraints in minimization problems**. [S.l.]: United Kingdom Atomic Energy Authority, 1967.

_____. An efficient method for finding the minimum of a function of several variables without calculating derivatives. **The Computer Journal**, Oxford University Press, v. 7, n. 2, p. 155–162, 1964. Disponível em: <<https://doi.org/10.1093/comjnl/7.2.155>>. Acesso em: 15 mar. 2018.

PRICE, Kenneth; STORN, Rainer M.; LAMPINEN, Jouni A. **Differential evolution: A practical approach to global optimization**. Berlin: Springer, 2006.

PRINCE, Simon J. D. **Computer vision: Models, learning, and inference**. New York, NY: Cambridge University Press, 2012.

QUEIPO, Nestor V. et al. Surrogate-based analysis and optimization. **Progress in Aerospace Sciences**, Elsevier, v. 41, n. 1, p. 1–28, 2005. Disponível em: <<https://doi.org/10.1016/j.paerosci.2005.02.001>>. Acesso em: 15 mar. 2018.

RASTRIGIN, L. A. About convergence of random search method in extremal control of multi-parameter systems. **Automation and Remote Control**, Plenum Publ Corp Consultants Bureau, v. 24, n. 11, p. 1467–1473, 1963. Disponível em: <<http://mi.mathnet.ru/eng/at12312>>. Acesso em: 15 mar. 2018.

ROSALES-PÉREZ, Alejandro et al. Surrogate-assisted multi-objective model selection for support vector machines. **Neurocomputing**, v. 150, p. 163–172, 2015. Disponível em: <<https://doi.org/10.1016/j.neucom.2014.08.075>>. Acesso em: 15 mar. 2018.

ROSENBROCK, Howard H. An automatic method for finding the greatest or least value of a function. **The Computer Journal**, Oxford University Press, v. 3, n. 3, p. 175–184, 1960. Disponível em: <<https://doi.org/10.1093/comjnl/3.3.175>>. Acesso em: 15 mar. 2018.

RUECKERT, Daniel. Nonrigid registration: Concepts, algorithms, and applications. In: **MEDICAL Image Registration**. Boca Raton, FL: CRC Press, 2001. p. 281–301.

RUECKERT, Daniel et al. Nonrigid registration using free-form deformations: Application to breast MR images. **IEEE Transactions on Medical Imaging**, IEEE, v. 18, n. 8, p. 712–721, 1999. Disponível em: <<https://doi.org/10.1109/42.796284>>. Acesso em: 15 mar. 2018.

RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **Nature**, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986. Disponível em: <<https://doi.org/10.1038/323533a0>>. Acesso em: 15 mar. 2018.

RUSSELL, Stuart J.; NORVIG, Peter. **Artificial intelligence: A modern approach**. 3rd. Upper Saddle River, NJ: Prentice Hall, 2010.

SACKS, Jerome et al. Design and analysis of computer experiments. **Statistical Science**, Institute of Mathematical Statistics, v. 4, n. 4, p. 409–423, 1989. Disponível em: <<http://www.jstor.org/stable/2245858>>. Acesso em: 15 mar. 2018.

SCHWEFEL, Hans-Paul. **Kybernetische evolution als strategie der experimentellen forschung in der strömungstechnik**. 1965. Diss. (Mestrado) – Technical University of Berlin, Berlin, Germany.

SHAHRIARI, Bobak et al. Taking the human out of the loop: A review of bayesian optimization. **Proceedings of the IEEE**, IEEE, v. 104, n. 1, p. 148–175, 2016. Disponível em: <<https://doi.org/10.1109/JPROC.2015.2494218>>. Acesso em: 15 mar. 2018.

SHAN, Songqing; WANG, G. Gary. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. **Structural and Multidisciplinary Optimization**, Springer, v. 41, n. 2, p. 219–241, 2010. Disponível em: <<https://doi.org/10.1007/s00158-009-0420-2>>. Acesso em: 15 mar. 2018.

SHANNON, Claude E. A Mathematical Theory of Communication. **ACM SIGMOBILE Mobile Computing and Communications Review**, Association for Computing Machinery, New

York, NY, v. 5, n. 1, p. 3–55, jan. 2001. Disponível em: <<http://doi.org/10.1145/584091.584093>>. Acesso em: 15 mar. 2018.

SIMPSON, Timothy W. et al. Metamodels for computer-based engineering design: Survey and recommendations. **Engineering with Computers**, Springer, v. 17, n. 2, p. 129–150, 2001. Disponível em: <<https://doi.org/10.1007/PL00007198>>. Acesso em: 15 mar. 2018.

SIROVICH, Lawrence; KIRBY, Michael. Low-dimensional procedure for the characterization of human faces. **Journal of the Optical Society of America**, OSA, v. 4, n. 3, p. 519–524, mar. 1987. Disponível em: <<https://doi.org/10.1364/JOSAA.4.000519>>. Acesso em: 15 mar. 2018.

SONKA, Milan; HLAVAC, Vaclav; BOYLE, Roger. **Image processing, analysis, and machine vision**. 4th. Stamford, CT: Cengage Learning, 2014.

SOTIRAS, Aristeidis; DAVATZIKOS, Christos; PARAGIOS, Nikos. Deformable medical image registration: A survey. **IEEE Transactions on Medical Imaging**, v. 32, n. 7, p. 1153–1190, 2013. Disponível em: <<https://doi.org/10.1109/TMI.2013.2265603>>. Acesso em: 15 mar. 2018.

SURJANOVIC, S.; BINGHAM, D. **Virtual library of simulation experiments**: Test functions and datasets. Disponível em: <<http://www.sfu.ca/~ssurjano/optimization.html>>. Acesso em: 15 mar. 2018.

TURK, Matthew A.; PENTLAND, Alex P. Face recognition using eigenfaces. In: PROCEEDINGS. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Maui, HI: IEEE, jun. 1991. p. 586–591. Disponível em: <<https://doi.org/10.1109/CVPR.1991.139758>>. Acesso em: 15 mar. 2018.

VAPNIK, Vladimir. **The nature of statistical learning theory**. New York, NY: Springer, 2013.

VENKATARAMAN, Panchapakesan. **Applied optimization with MATLAB programming**. Hoboken, NJ: John Wiley & Sons, 2009.

VIOLA, Paul; WELLS III, William M. Alignment by maximization of mutual information. **International Journal of Computer Vision**, v. 24, n. 2, p. 137–154, set. 1997. Disponível em: <<https://doi.org/10.1023/A:1007958904918>>. Acesso em: 15 mar. 2018.

WAHBA, Grace. **Spline models for observational data**. Philadelphia, PA: Society for Industrial e Applied Mathematics, 1990.

WANG, G. Gary; SHAN, Songqing. Review of metamodeling techniques in support of engineering design optimization. **Journal of Mechanical Design**, American Society of Mechanical Engineers, v. 129, n. 4, p. 370–380, 2007. Disponível em: <<https://doi.org/10.1115/1.2429697>>. Acesso em: 15 mar. 2018.

WANG, Ziyu et al. Bayesian optimization in a billion dimensions via random embeddings. **Journal of Artificial Intelligence Research**, AI Access Foundation, v. 55, n. 1, p. 361–387, jan. 2016. Disponível em: <<https://doi.org/10.1613/jair.4806>>. Acesso em: 15 mar. 2018.

WILSON, Robert B. **A simplicial algorithm for concave programming**. 1963. Tese (Doutorado) – Graduate School of Business Administration, Harvard University, Cambridge, MA.

XAVIER, Igor R. R. **Análise multivariada de imagens frontais de face normalizadas espacialmente**. Jun. 2015. Diss. (Mestrado) – Centro Universitário da FEI, São Bernardo do Campo. Disponível em: <http://fei.edu.br/~cet/dissertacao_IgorXavier_2015.pdf>. Acesso em: 15 mar. 2018.

XAVIER, Igor R. R. et al. A photo-realistic generator of most expressive and discriminant changes in 2D face images. In: 2015 Sixth International Conference on Emerging Security Technologies (EST). Braunschweig, Germany: IEEE, set. 2015. p. 80–85. Disponível em: <<https://doi.org/10.1109/EST.2015.17>>. Acesso em: 15 mar. 2018.

YANG, Xin-She. **Nature-inspired metaheuristic algorithms**. 2nd. Cambridge, UK: Luniver Press, 2010.

ZHU, Xiangxin; RAMANAN, Deva. Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. Providence, RI: IEEE, jun. 2012. p. 2879–2886. Disponível em: <<https://doi.org/10.1109/CVPR.2012.6248014>>. Acesso em: 15 mar. 2018.

ZITOVÁ, Barbara; FLUSSER, Jan. Image registration methods: A survey. **Image and Vision Computing**, Elsevier, v. 21, n. 11, p. 977–1000, 2003. Disponível em: <[https://doi.org/10.1016/S0262-8856\(03\)00137-9](https://doi.org/10.1016/S0262-8856(03)00137-9)>. Acesso em: 15 mar. 2018.

ZOUTENDIJK, Guus. **Methods of feasible directions: A study in linear and non-linear programming**. [S.l.]: Elsevier, 1960.

**APÊNDICE A – FUNÇÕES PARA TESTE DE ALGORITMOS DE OTIMIZAÇÃO SEM
RESTRICÇÕES**

O desempenho dos diversos algoritmos de otimização que são desenvolvidos na literatura são testados e comparados em funções padrões de teste. Estas funções possuem características que dificultam ou até impossibilitam a convergência dos algoritmos mais tradicionais, requerendo técnicas mais sofisticadas de otimização. Entre algumas das características que podem dificultar a convergência de um algoritmo de otimização estão: múltiplos mínimos locais (multimodalidade), descontinuidade e assimetria (JAMIL; YANG, 2013).

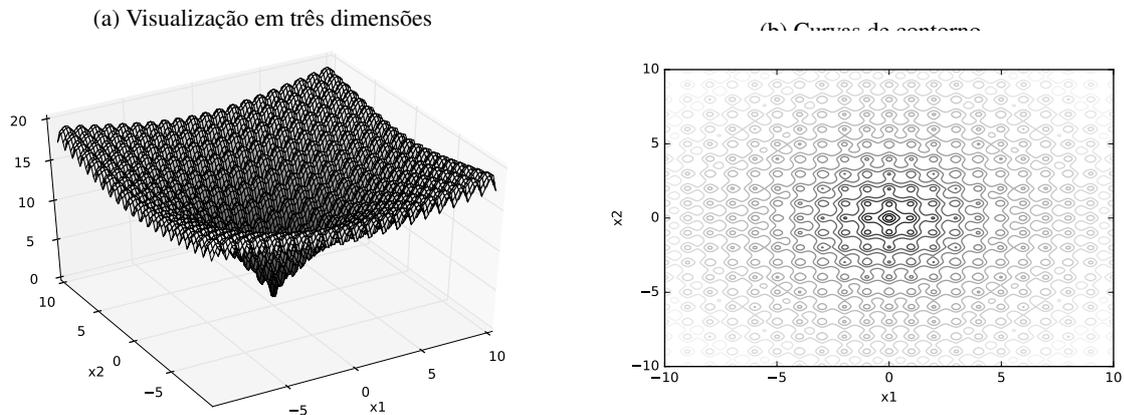
Neste trabalho, são apresentadas as funções mais utilizadas na literatura relacionada. Uma lista das funções de teste pode ser encontrada em trabalhos específicos (SURJANOVIC; BINGHAM, 2018; JAMIL; YANG, 2013).

a) função de Ackley (ACKLEY, 1987; BACK, 1996)

A Função de Ackley é multimodal e é caracterizada por uma planície conforme afasta-se do mínimo, o que dificulta a convergência de algoritmos que utilizam informação do gradiente (SURJANOVIC; BINGHAM, 2018). O mínimo global ocorre em $x^* = (0,0, \dots, 0)$, $f(x^*) = 0$.

$$f(x_1, x_2, \dots, x_n) = -20 \cdot \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e. \quad (59)$$

Figura 66 – Visualização gráfica da função de Ackley



Fonte: Autor

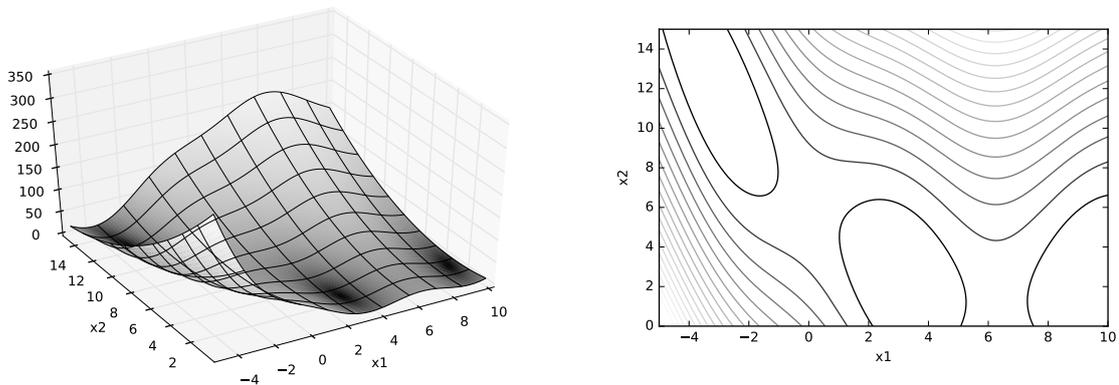
b) função de Branin (BRANIN, 1972)

A Função de Branin ficou conhecida por testar algoritmos de otimização através da modelagem *kriging* (JONES; SCHONLAU; WELCH, 1998). Os três mínimos globais ocorrem em $x_1^* = (-\pi; 12,275)$, $x_2^* = (\pi; 2,275)$, $x_3^* = (3\pi; 2,425)$, $f(x^*) = 0,3978873$ (JAMIL; YANG, 2013).

$$f(x_1, x_2, \dots, x_n) = \left(x_2 - \frac{5,1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi} \cdot \cos(x_1)\right) + 10. \quad (60)$$

Figura 67 – Visualização gráfica da função de Branin

(a) Visualização em três dimensões



Fonte: Autor

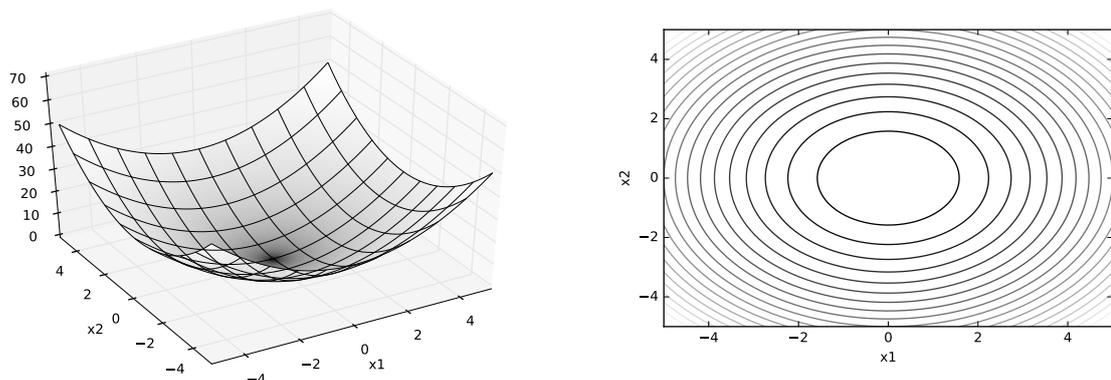
c) função Esfera

A Função Esfera, é uma função quadrática, unimodal, contínua e convexa (SURJANOVIC; BINGHAM, 2018). É comum encontrar variantes desta função na literatura para fundamentar alguns conceitos básicos de algoritmos existentes, bem como propostos (PRICE; STORN; LAMPINEN, 2006). O mínimo global ocorre em $x^* = (0,0, \dots, 0)$, $f(x^*) = 0$.

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n x_i^2. \quad (61)$$

Figura 68 – Visualização gráfica da função Esfera

(a) Visualização em três dimensões



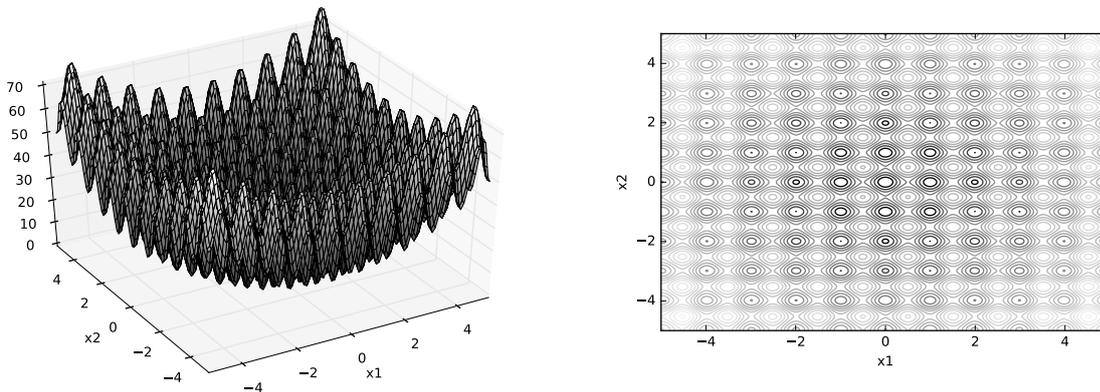
Fonte: Autor

- d) função de Rastrigin (RASTRIGIN, 1963; SURJANOVIC; BINGHAM, 2018)
 A Função de Rastrigin é multimodal, porém com a localização dos mínimos uniformemente distribuídos (SURJANOVIC; BINGHAM, 2018). O mínimo global ocorre em $x^* = (0,0, \dots, 0)$, $f(x^*) = 0$.

$$f(x_1, x_2, \dots, x_n) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cdot \cos(2\pi x_i)). \quad (62)$$

Figura 69 – Visualização gráfica da função de Rastrigin

(a) Visualização em três dimensões



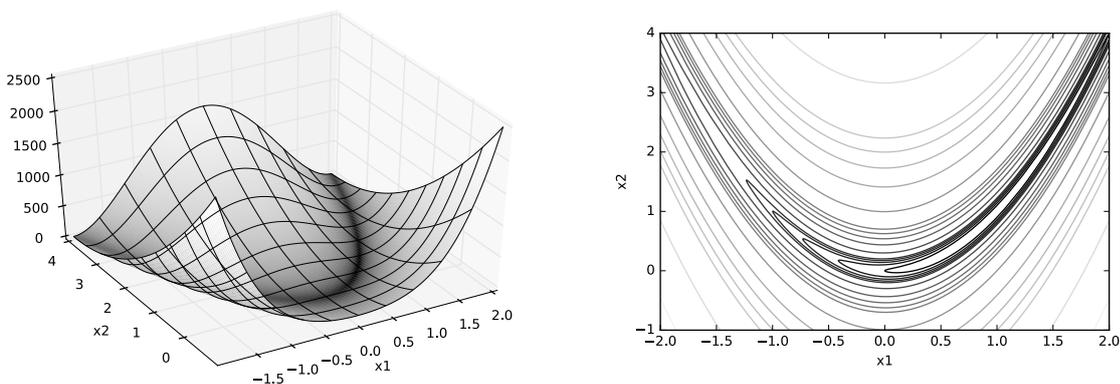
Fonte: Autor

- e) função de Rosenbrock (ROSENBROCK, 1960)
 A Função de Rosenbrock, apesar de ser unimodal, apresenta grande dificuldade de convergência devido à presença de uma vale em formato de "banana". Esta geometria faz com que muitos dos algoritmos baseados em gradiente apresentem um comportamento de busca em "zigzag", o que aumenta o número de iterações até a convergência (SURJANOVIC; BINGHAM, 2018). O mínimo global ocorre em $x^* = (1,1, \dots, 1)$, $f(x^*) = 0$.

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2). \quad (63)$$

Figura 70 – Visualização gráfica da função de Rosenbrock

(a) Visualização em três dimensões



Fonte: Autor