



ANDRÉ DE AGUIAR BRAGA

**UTILIZAÇÃO DE META-HEURÍSTICAS PARA A SELEÇÃO
AUTOMÁTICA DE PARÂMETROS DO ALGORITMO
K-SEGMENTOS**

LAVRAS – MG

2019

ANDRÉ DE AGUIAR BRAGA

**UTILIZAÇÃO DE META-HEURÍSTICAS PARA A SELEÇÃO AUTOMÁTICA DE
PARÂMETROS DO ALGORITMO K-SEGMENTOS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós Graduação em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

Prof. DSc. Danton Diego Ferreira

Orientador

Prof. DSc. Bruno Henrique Groenner Barbosa

Coorientador

LAVRAS – MG

2019

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Braga, André de Aguiar.

Utilização de Meta-heurísticas para a Seleção Automática de
Parâmetros do Algoritmo k-Segmentos / André de Aguiar Braga. -
2019.

94 p. : il.

Orientador(a): Danton Diego Ferreira.

Coorientador(a): Bruno Henrique Groenner Barbosa.

Dissertação (mestrado acadêmico) - Universidade Federal de
Lavras, 2019.

Bibliografia.

1. Curvas Principais. 2. k-Segmentos. 3. Meta-Heurísticas. I.
Ferreira, Danton Diego. II. Barbosa, Bruno Henrique Groenner. III.
Título.

ANDRÉ DE AGUIAR BRAGA

**UTILIZAÇÃO DE META-HEURÍSTICAS PARA A SELEÇÃO AUTOMÁTICA DE
PARÂMETROS DO ALGORITMO K-SEGMENTOS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós Graduação em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

APROVADA em 03 de Julho de 2019.

DSc. Danton Diego Ferreira	UFLA
DSc. Bruno Henrique Groenner Barbosa	UFLA
DSc. Giovanni Bernardes Vitor	UNIFEI



Prof. DSc. Danton Diego Ferreira
Orientador

Prof. DSc. Bruno Henrique Groenner Barbosa
Coorientador

**LAVRAS – MG
2019**

AGRADECIMENTOS

Meus sinceros agradecimentos a todos e todas que me apoiaram durante minha pesquisa, em especial aos meus pais e amigos mais próximos, cujos nomes não cito para que não seja injusto.

Agradeço aos meus orientadores Danton e Bruno pelo constante apoio e disposição aos membros e suplentes da banca, Giovani, Wilian e Leandro, que gentilmente se propuseram a contribuir com o trabalho.

Meus agradecimentos à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (Fapemig) pelo apoio financeiro e ao Programa de Pós-Graduação em Engenharia de Sistemas e Automação pela oportunidade.

RESUMO

Sendo uma generalização não linear de análise de componentes principais, a técnica de curvas principais é uma ferramenta robusta para análise e classificação de dados. Em reconhecimento de padrões, um dos algoritmos mais populares para construção de Curvas Principais é o algoritmo k-segmentos. Esse algoritmo apresenta bons resultados e ótima aplicabilidade por sua convergência garantida e robustez. Contudo, sua utilização e desempenho dependem de parâmetros definidos pelo usuário. Este trabalho apresenta uma técnica de seleção automática da quantidade e comprimento dos segmentos do algoritmo k-segmentos com a utilização de diferentes meta-heurísticas mono-objetivo e multiobjetivo, sobretudo TLBO (Teaching-Learning-Based Optimization). Diferentes aplicações do método proposto são estudadas, como representação, classificação supervisionada e classificação não supervisionada de dados, para as quais são utilizadas como funções-custo equações que levam em consideração o comprimento da curva e a distância dos eventos aos segmentos onde eles se projetam, além da minimização do erro de classificação das bases de validação para a classificação supervisionada de dados. Testes experimentais feitos com bases de dados sintéticos bidimensionais e reais, em sua maioria multidimensionais, retirados de repositórios conhecidos, são apresentados para demonstrar a eficiência do método proposto. Para representação, a qualidade dos resultados é observada visualmente, enquanto para classificação supervisionada e não supervisionada de dados, são feitas comparações com os métodos k-NN e k-means, respectivamente. Para a classificação supervisionada observa-se que ambos os métodos comparados apresentam resultados similares, destacando a superioridade observada no método proposto para a base de dados de maior dimensão. Para clusterização de dados, observa-se que o método proposto alcança resultados superiores ao método comparado para a grande maioria das bases de dados, dependendo da função-custo utilizada, sendo observada ainda a importância da otimização multiobjetivo para tal fim.

Palavras-chave: Curvas Principais. k-Segmentos. Meta-Heurísticas. Otimização baseada em ensino-aprendizagem.

ABSTRACT

Being a nonlinear generalization of principal component analysis, the principal curves technique is a robust tool for data analysis and classification. In pattern recognition one of the most popular algorithms to build Principal Curves is the k-segments algorithm. This algorithm presents good results and excellent applicability due to its guaranteed convergence and robustness. However, its use and performance depend on user-defined parameters. This work presents an automatic selection technique of the quantity and length of segments of the k-segment algorithm with the use of different mono-objective and multiobjective meta-heuristics, especially TLBO (Teaching-Learning-Based Optimization). Different applications of the proposed method are studied, as representation, supervised classification and unsupervised classification of data, for which are used as cost functions equations that take into account the length of the curve and the distance of the events to the segments where they are projected, in addition to minimizing the classification error of the validation bases for the supervised classification of data. Experimental tests made with two-dimensional synthetic and real, mostly multidimensional, databases, taken from known repositories, are presented to demonstrate the efficiency of the proposed method. For representation the quality of results is observed visually, while for supervised and unsupervised classification of data, comparisons are made with the k-NN and k-means methods, respectively. For the supervised classification it is observed that both methods have similar results, highlighting the superiority observed in the proposed method for the database with the largest dimension. For data clustering, it is observed that the proposed method achieves superior results than the comparative method for most databases, depending on the cost-function used, being also observed the importance of multiobjective optimization for this purpose.

Keywords: Principal Curves. k-Segment. Meta-Heuristics. Teaching-learning-based optimization.

LISTA DE FIGURAS

Figura 2.1 – Curva principal para um conjunto de dados de duas dimensões	15
Figura 2.2 – Projeção dos dados na curva principal	16
Figura 2.3 – Auto-consistência de uma curva principal	17
Figura 2.4 – Sequência da construção de uma curva principal pelo algoritmo de k-segmentos	19
Figura 2.5 – Fluxograma simplificado do algoritmo k-segmentos	21
Figura 2.6 – Fluxograma de um algoritmo bioinspirado	23
Figura 2.7 – Fluxograma do algoritmo TLBO	24
Figura 2.8 – Fluxograma do algoritmo ITLBO	27
Figura 2.9 – Fluxograma do algoritmo NSGA-II	31
Figura 3.1 – Fluxograma do funcionamento do método proposto	34
Figura 3.2 – Fluxograma do Processo de Otimização com a utilização de TLBO	35
Figura 3.3 – Conjuntos de dados utilizados	37
Figura 3.4 – Curvas geradas com o valor do comprimento do primeiro segmento fixado em $3\sigma/2$, construídas a partir da minimização da Equação (3.2)	38
Figura 3.5 – Curvas geradas com o valor do comprimento do primeiro segmento fixado em $3\sigma/2$, construídas a partir da minimização da Equação (3.3)	40
Figura 3.6 – Curvas geradas com o quantidade e comprimento dos segmentos como pa- râmetros otimizados, construídas a partir da minimização da Equação (3.2)	42
Figura 3.7 – Curvas geradas com o quantidade e comprimento dos segmentos como pa- râmetros otimizados, construídas a partir da minimização da Equação (3.3)	43
Figura 3.8 – Convergência do algoritmo TLBO nas primeiras 15 iterações.	45
Figura 3.9 – Curvas geradas a partir da minimização da Equação (3.4).	46
Figura 4.1 – Curvas geradas a partir de parâmetros otimizados com a utilização das di- ferentes meta-heurísticas, para contorno de dados em espiral.	50
Figura 4.2 – Convergência das meta-heurísticas de acordo com a quantidade de chama- das das funções de avaliação para o contorno em espiral.	51
Figura 4.3 – Curvas geradas a partir de parâmetros otimizados com a utilização das di- ferentes meta-heurísticas, para contorno de dados em rabisco.	53
Figura 4.4 – Convergência das meta-heurísticas de acordo com a quantidade de chama- das das funções de avaliação para o contorno em rabisco.	54
Figura 4.5 – Fronteira de Pareto para dados em espiral.	57

Figura 4.6 – Curvas geradas a partir de parâmetros pertencentes ao conjunto Pareto e da otimização TLBO para dados em espiral.	58
Figura 4.7 – Fronteira de Pareto para dados em rabisco.	60
Figura 4.8 – Curvas geradas a partir de parâmetros pertencentes ao conjunto Pareto e da otimização TLBO para dados em rabisco.	61
Figura 5.1 – Fluxograma da criação do Modelo de Classificação Supervisionada por meio da minimização do comprimento da curva e distância aos eventos como função-custo	67
Figura 5.2 – Processo de classificação supervisionada de dados com utilização de CP	68
Figura 5.3 – Fluxograma da criação do Modelo de Classificação Supervisionada por meio da minimização do erro de classificação como função custo	71
Figura 6.1 – Funcionamento do algoritmo de clusterização de dados com a utilização do algoritmo k-segmentos.	75
Figura 6.2 – Fluxograma da criação do Modelo de Clusterização com seleção automática dos parâmetros do algoritmo k-segmentos	76
Figura 6.3 – Fronteira de Pareto para base de dados Wine.	83
Figura 1 – Curvas geradas a partir da minimização da Equação (3.4) com a utilização do algoritmo k-segmentos suave.	92

LISTA DE TABELAS

Tabela 4.1 – Valores da função-custo para dados em espiral	50
Tabela 4.2 – Valores da função-custo para dados em rabisco	54
Tabela 4.3 – Valores das funções-custo de diferentes pontos para dados em espiral . . .	57
Tabela 4.4 – Valores das funções-custo de diferentes pontos para dados em rabisco . . .	60
Tabela 5.1 – Bases de dados utilizadas	65
Tabela 5.2 – Desempenho da classificação supervisionada, em %, para as bases de teste	68
Tabela 5.3 – Comparação de desempenho da classificação supervisionada, em %, para diferentes funções-custo	72
Tabela 6.1 – Desempenho da classificação não supervisionada, em %, para as bases de teste com validação cruzada	78
Tabela 6.2 – Desempenho da classificação não supervisionada, em %, para as bases de dados completas	80
Tabela 6.3 – Quantidade e comprimento dos segmentos obtidos durante os testes de clas- sificação não supervisionada	81

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivos	13
1.2	Organização do trabalho	14
2	REFERENCIAL TEÓRICO	15
2.1	Curvas Principais	15
2.1.1	Definição de curvas principais	15
2.1.2	Estudos a respeito de curvas principais	17
2.1.3	Algoritmo k-segmentos	19
2.2	Algoritmos Bioinspirados	21
2.3	<i>Teaching-learning-based optimization</i>	23
2.4	<i>Improved Teaching-learning-based optimization</i>	26
2.5	<i>Grey Wolf Optimization</i>	26
2.6	<i>Chaotic Grey Wolf Optimization</i>	29
2.7	<i>Non-dominated Sorting Genetic Algorithm II</i>	30
3	OTIMIZAÇÃO DE PARÂMETROS DO ALGORITMO K-SEGMENTOS PARA REPRESENTAÇÃO COM UTILIZAÇÃO DE TLBO	32
3.1	Método de seleção automática de parâmetros	32
3.2	Base de dados	36
3.2.1	Contorno de dados em espiral	36
3.2.2	Contorno de dados em rabisco	36
3.3	Resultados com comprimento do segmento fixo	37
3.4	Resultados com comprimento do segmento como parâmetro otimizado	41
3.5	Uma discussão sobre o coeficiente de suavidade λ	46
4	OTIMIZAÇÃO DE PARÂMETROS DO ALGORITMO K-SEGMENTOS COM UTILIZAÇÃO DE ITLBO, GWO, CGWO E NSGA-II	48
4.1	Utilização de meta-heurísticas mono-objetivo	48
4.1.1	Análise das meta-heurísticas mono-objetivo para contorno em forma de espiral	49
4.1.2	Análise das meta-heurísticas mono-objetivo para contorno em forma de rabisco	52
4.2	Utilização de meta-heurísticas multiobjetivo	55
4.2.1	Análise da meta-heurística multiobjetivo para contorno em forma de espiral	56
4.2.2	Análise da meta-heurística multiobjetivo para contorno em forma de rabisco	59

5	CLASSIFICAÇÃO SUPERVISIONADA COM A UTILIZAÇÃO DO ALGORITMO K-SEGMENTOS COM PARÂMETROS INICIAIS OTIMIZADOS	64
5.1	Banco de dados	64
5.1.1	Pré-processamento dos dados	65
5.2	Minimização do comprimento da curva e distância aos eventos como função-custo	66
5.3	Minimização do erro de classificação como função-custo	70
6	CLUSTERIZAÇÃO COM A UTILIZAÇÃO DO ALGORITMO K-SEGMENTOS COM PARÂMETROS INICIAIS OTIMIZADOS	74
6.1	Resultados da classificação não supervisionada nos testes com validação cruzada	77
6.2	Resultados da classificação não supervisionada nos testes com bases de dados completas	79
7	CONSIDERAÇÕES FINAIS E CONCLUSÃO	85
	REFERÊNCIAS	87
	APENDICE A – OTIMIZAÇÃO DOS PARÂMETROS DO ALGORITMO K-SEGMENTOS SUAVE	91
	APENDICE B – ARTIGO PUBLICADO EM ANAIS DE CONGRESSO	94

1 INTRODUÇÃO

Análise e processamento de dados são áreas de grande importância para o desenvolvimento científico e tecnológico e vêm ganhando cada vez mais espaço em diversos setores da sociedade. Em rápidas pesquisas por *sites* de notícias é possível ver o alcance da utilização de análise de dados para inúmeras aplicações. Redes sociais, mecanismos de buscas e sites de venda *online* traçam perfis de usuários baseados nas buscas realizadas e nas páginas visitadas. Pelo histórico das postagens e *links* acessados é possível determinar qual o direcionamento político, gosto musical e produtos pelos quais se interessa certo indivíduo. Dessa maneira é possível associar propagandas e sugestões específicas a cada usuário. Políticos se utilizam de tais ferramentas para conseguir votos por meio de campanhas direcionadas a cada perfil de eleitor.

A utilização de ferramentas de análise e processamento de dados também tem ganhado espaço em setores de grande importância para a sociedade, como a medicina e a indústria. Na medicina, tais ferramentas podem auxiliar médicos a diagnosticar doenças de maneira mais eficiente e ágil. As aplicações são difundidas em diversas áreas, desde a análise e identificação de produtos tóxicos por meio de sensores ao reconhecimento facial e de caracteres por meio de câmeras. Tais aplicações vêm se desenvolvendo e se tornando mais eficazes devido à utilização de diferentes técnicas de reconhecimento de padrões e aprendizado de máquina.

Na indústria, a aplicabilidade dessas ferramentas se estende desde a utilização de algoritmos para controle de qualidade dos produtos à otimização da cadeia de suprimentos e planejamento e controle de estoque. Com análise e processamento de dados é possível monitorar o funcionamento de grandes plantas industriais de modo a garantir a segurança de seus processos e auxiliar a tomada de decisões de maneira rápida e eficaz.

Com a utilização de ferramentas computacionais de análise de dados é possível obter informações de grande relevância. Devido à crescente aplicação dessas técnicas em diversos ramos, essa área de pesquisa tem cada vez mais espaço e importância. Dentre as ferramentas de classificação e análise de dados pode-se destacar o método de Curvas Principais (CP) devido à sua boa capacidade de representação de dados de alta dimensão por meio de uma representação compacta unidimensional.

O conceito de curvas principais (CP) foi apresentado por Hastie e Stuetzle (1989) como uma generalização não linear da técnica de análise de componentes principais. Definidas como curvas unidimensionais suaves que passam através do centro de um conjunto multidimensional

gerando uma representação unidimensional e não linear dos dados, as CP mostram-se uma ferramenta arrojada de reconhecimento de padrões.

Diversas aplicações desse algoritmo, para fins de reconhecimento de padrões, podem ser encontradas na literatura, como em (FERREIRA et al., 2013) e (FERREIRA et al., 2014), onde o mesmo é utilizado para detecção e classificação de distúrbios, assim como em (FERREIRA et al., 2015), onde é proposto um índice de desvio de qualidade de energia com a utilização de CP. Moraes e Ferreira (2016) propõem uma técnica de clusterização de dados (separação em grupos de dados que possuem características comuns) baseada em CP. Chang e Ghosh (1998) utilizam técnicas baseadas em CP e em *Self-organizing maps* (SOM) para realizar classificação supervisionada de dados. Há ainda aplicações em sonares, para classificação de navios (FERNANDEZ, 2005; FERNANDES; Souza Filho; ALVES, 2015; OLIVEIRA; FILHO, 2017), na detecção de descargas parciais em equipamentos de potência (FAIER, 2006) e na identificação de blocos de gelo por imagem de satélites (BANFIELD; RAFTERY, 1992).

A possibilidade de aplicação dos diversos métodos de extração de CP encontrados na literatura se estende desde a extração de características para reconhecimento de caracteres manuscritos (BAI; ZHU, 2012) e impressões digitais (MA et al., 2009) ao diagnóstico de doenças de retina (YOU et al., 2011) e estimativa de valores faltantes dos dados de DNA (SHI; LUO, 2010).

O algoritmo de extração de CP k-segmentos é um algoritmo que extrai a CP de forma incremental, tendo o número de segmentos aumentado a cada iteração. Devido à sua robustez, baixa suscetibilidade a mínimos locais e convergência garantida, é muito utilizado na extração de curvas. Este algoritmo necessita da determinação de certos parâmetros pelo usuário, tais como a quantidade máxima de segmentos e o coeficiente de suavidade da curva. Por outro lado, valores numéricos que impactam na qualidade de representatividade dos dados, tais como o comprimento dos segmentos que compõem a curva, tem valores fixados. A determinação de tais valores e dos parâmetros iniciais depende de testes empíricos, uma vez que o conhecimento prévio dos dados por parte dos usuários não é suficiente para realizar essa definição. Ademais, o desempenho de métodos de reconhecimento de padrões baseados na extração de curvas principais pelo algoritmo k-segmentos está diretamente relacionado à escolha dos parâmetros de entrada.

É importante destacar que ao propor o algoritmo k-segmentos, Verbeek, Vlassis e Kröse (2002) não otimizaram a determinação de nenhum parâmetro e Wang e Lee (2006) realizam a

otimização da quantidade de segmentos e coeficiente de suavidade com a utilização de cálculos estatísticos. Por outro lado, Biau e Fischer (2012) propõem a otimização da quantidade de segmentos e comprimento da curva para o algoritmo de linhas poligonais (PLA), proposto por Kegl et al. (2000). Tal algoritmo, assim como o algoritmo k-segmentos, tem como função a construção de CP, mas segundo Verbeek, Vlassis e Kröse (2001) apresenta problemas na construção de curvas em que o contorno de dados intercepta a si mesmo, o que não ocorre com o algoritmo k-segmentos. Ao propor tal método Biau e Fischer (2012) cita que a otimização dos parâmetros da curva são importantes na construção de CP viáveis para as diversas aplicações em que são utilizadas.

1.1 Objetivos

Este trabalho tem por objetivo propor um método de determinação automática da quantidade e comprimento dos segmentos do algoritmo de extração de curvas principais k-segmentos, com a utilização de meta-heurísticas, em especial a meta-heurística *Teaching-Learning Based Optimization*.

De maneira específica objetiva-se:

- Elaborar e testar um método que determine automaticamente o comprimento e número de segmentos.
- Analisar os resultados obtidos pelo método desenvolvido. O algoritmo proposto deve alcançar resultados satisfatórios na construção da CP, com a utilização de bases de dados utilizadas na literatura, sem a utilização de cálculos estatísticos complexos. Para fins de representação, a avaliação da qualidade do método que se propõem é feita pela análise visual dos dados.
- Comparar os resultados obtidos pelo método com a utilização de diferentes meta-heurísticas e realizar o estudo da viabilidade da função-custo utilizada.
- Utilizar o método proposto para classificação supervisionada e não supervisionada, de modo que testes da viabilidade para tais aplicações são realizados com a utilização de bancos de dados reais e simulados. Para a classificação supervisionada tem-se como objetivo, ainda, testar a minimização do erro das bases de validação como função-custo do método de otimização.

- Como objetivo secundário tem-se a utilização do método proposto na otimização dos parâmetros do algoritmo k-segmentos suave, proposto por Verbeek, Vlassis e Kröse (2001), sem que haja a necessidade de alterações significativas na técnica de otimização aqui proposta.

1.2 Organização do trabalho

O restante do trabalho é organizado da seguinte maneira: O Capítulo 2, página 15, traz a revisão bibliográfica acerca dos temas tratados neste documento. No Capítulo 3, página 32, é apresentado o método proposto para otimização dos parâmetros iniciais do algoritmo k-segmentos e as discussões a respeito dos resultados encontrados. No Capítulo 4, página 48, são realizados testes com diferentes meta-heurísticas de modo a observar a eficácia do método proposto. No Capítulo 5, página 64, são apresentados testes a respeito da utilização da técnica de otimização proposta em classificação supervisionada. Já no Capítulo 6, página 74, são apresentados resultados e discussões a respeito de sua utilização na classificação não supervisionada com a utilização de Curvas Principais. Por fim, no Capítulo 7, página 85, são apresentadas as conclusões a respeito do trabalho desenvolvido. Destaca-se, também, que no Apêndice A, página 92, são apresentados os resultados da otimização dos parâmetros iniciais do algoritmo k-segmentos suave, não utilizado nas aplicações estudadas neste trabalho, mas cuja otimização mostra-se possível pelas técnicas nele desenvolvidas.

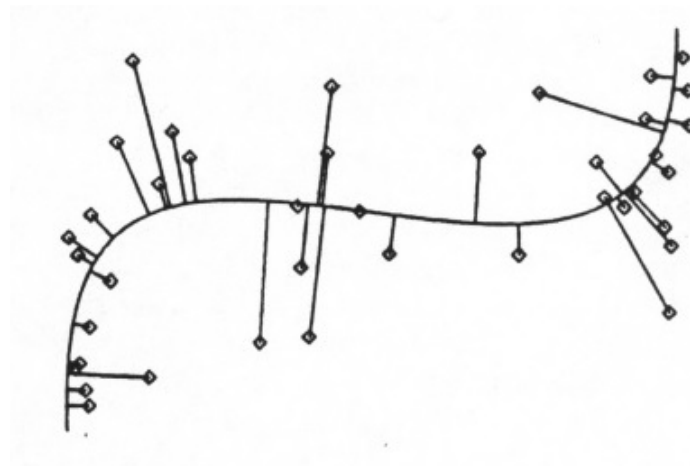
2 REFERENCIAL TEÓRICO

2.1 Curvas Principais

2.1.1 Definição de curvas principais

Computacionalmente, a extração das CP de um conjunto de dados se dá pela definição do primeiro componente principal do conjunto como ponto de partida para construção da curva, que é alterada sucessivamente até a convergência do algoritmo de obtenção da CP, cuja representação é exemplificada na Figura 2.1.

Figura 2.1 – Curva principal para um conjunto de dados de duas dimensões



Fonte: Hastie e Stuetzle (1989)

A definição matemática das CP é feita a partir do conceito de auto-consistência, definido de acordo com o índice de projeção de um ponto \mathbf{x}_i em uma curva \mathbf{f} . Uma curva unidimensional, num espaço de d dimensões, \mathfrak{R}^d , é um vetor $\mathbf{f}(t)$ de d funções contínuas de uma única variável t , ou seja:

$$\mathbf{f}(t) = \{f_1(t), f_2(t), \dots, f_d(t)\} \quad (2.1)$$

Essas funções são denominadas funções de coordenada e o parâmetro t dá ordenamento ao longo da curva.

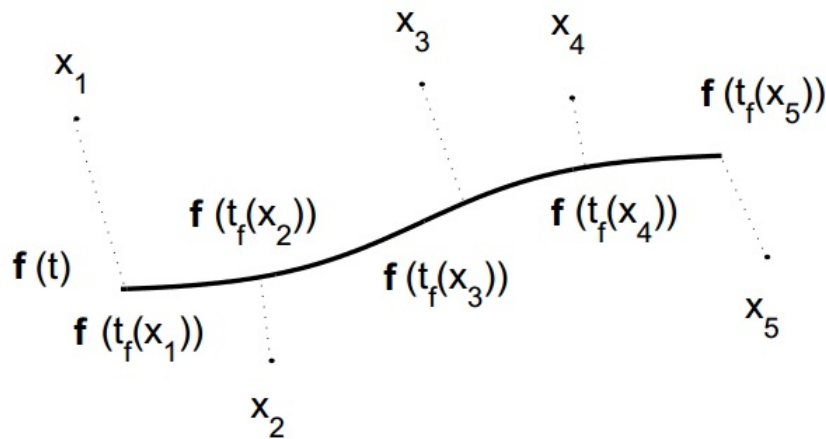
Seja \mathbf{x} um vetor aleatório em \mathfrak{R}^d , de densidade de probabilidade h , e momento de segunda ordem finito. Seja \mathbf{f} uma curva suave que não intercepta a si própria, ou seja, $t_1 \neq t_2 \Rightarrow f(t_1) \neq f(t_2)$, parametrizada no intervalo fechado $I \subseteq \mathfrak{R}$. O índice de projeção $t_f : \mathfrak{R}^d \rightarrow \mathfrak{R}$ é definido como:

$$t_f(\mathbf{x}) = \sup_t \{t : \|\mathbf{x} - \mathbf{f}(t)\| = \inf_{\mu} \|\mathbf{x} - \mathbf{f}(\mu)\|\} \quad (2.2)$$

sendo μ uma variável auxiliar definida em \mathfrak{X} .

O índice de projeção $t_f(\mathbf{x})$ é o valor de t para o qual a CP $\mathbf{f}(t)$ está mais próxima de \mathbf{x} . A existência de mais de um valor possível implica na escolha do maior deles. O índice de projeção relativo a uma curva principal é ilustrado na Figura 2.2. Nela, são apresentados cinco eventos $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_5$, que projetam na curva, nos pontos $\mathbf{f}(t_f(\mathbf{x}_1)), \mathbf{f}(t_f(\mathbf{x}_2)), \dots, \mathbf{f}(t_f(\mathbf{x}_5))$, respectivamente. É possível observar que o parâmetro t relativo ao ponto da curva mais próximo ao evento \mathbf{x}_i é dado por $t = t_f(\mathbf{x}_i)$, correspondente, então, ao índice de projeção desse ponto sobre a curva.

Figura 2.2 – Projeção dos dados na curva principal



Fonte: Fernandez (2005)

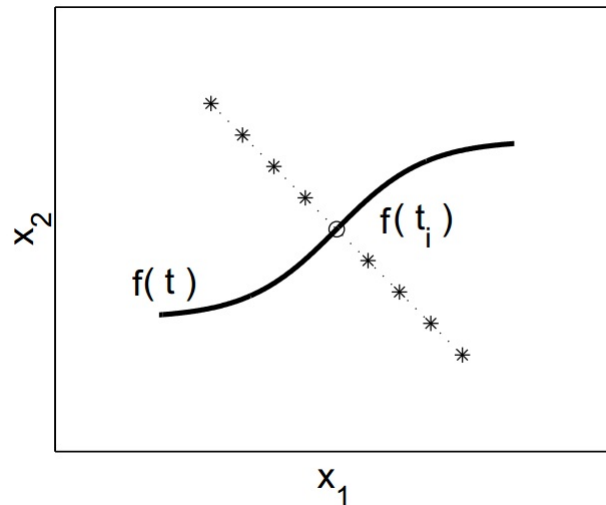
A partir da definição do índice de projeção de um ponto em uma curva pode-se definir as CP.

Definição 2.1.1. Uma curva é auto-consistente ou uma curva principal se, para todos os valores do parâmetro t , seus pontos forem a média dos dados que nela projetam ortogonalmente, ou seja

$$\mathbf{f}(t) = \mathbf{E}[\mathbf{x} | t_f(\mathbf{x}) = t], \quad \forall t \quad (2.3)$$

A auto-consistência decorre da propriedade dos pontos que compõem uma CP consistirem na média dos dados que nela projetam. Na Figura 2.3, onde é representada uma CP de duas dimensões, é possível observar oito eventos cuja projeção determinam o ponto $f(t_i)$ da curva principal. Dessa forma, o ponto $f(t_i)$ é a média dos oito eventos nele projetados.

Figura 2.3 – Auto-consistência de uma curva principal



Fonte: Fernandez (2005)

De forma geral, uma CP $f(t)$, definida em \mathfrak{R}^d , possui as seguintes características:

- É uma curva suave, ou seja, infinitamente diferenciável;
- Não intercepta a si própria, ou seja, $t_1 \neq t_2 \Rightarrow f(t_1) \neq f(t_2)$;
- Possui comprimento finito dentro de um espaço finito em \mathfrak{R}^d ;
- É auto-consistente.

2.1.2 Estudos a respeito de curvas principais

O primeiro algoritmo de obtenção de CP foi apresentado por Hastie e Stuetzle (1989) e sofreu diversas modificações com o passar dos anos. A construção de curvas principais passa pela determinação de parâmetros, tais como número de segmentos, comprimento e curvatura. Em (BIAU; FISCHER, 2012) se encontra uma proposta de determinação de tais parâmetros através da minimização do risco empírico, para o algoritmo de linhas poligonais (PLA), proposto por Kegl et al. (2000). Desde sua apresentação o conceito de curvas principais foi redefinido e estendido por alguns autores. Uma extensão de sua definição para as variedades de Riemann é retratada por Hauberg (2016) e Ozertem e Erdogmus (2011) apresenta uma redefinição de curvas principais através do gradiente e da hessiana da estimativa da densidade de probabilidade, levando a uma visão geométrica de curvas principais e superfícies além de unificar esse campo de pesquisa ao estudo de variedades de dimensionalidades intrínsecas.

Na literatura a respeito do tema são encontradas diferentes estratégias de implementação das CP com diferentes paradigmas de programação aliados a outras técnicas de reconhecimento de padrões como em (WANG et al., 2008) onde se encontra um algoritmo de aprendizado de CP baseado em redes neurais artificiais, que transforma aprendizado não supervisionado em aprendizado supervisionado com a projeção do conjunto de dados em linhas poligonais, com a utilização de algoritmos de CP já existentes.

Um novo método de aprendizado de CP baseado em Análise de componentes principais (PCA) e *Back-propagation network* (BP), que também transforma aprendizado não supervisionado em aprendizado supervisionado, é encontrado em (WANG et al., 2007). Cleju, Franti e Wu (2005) propõe a utilização de estrutura de árvores para obtenção das características principais dos dados para aumentar o desempenho dos métodos baseados em CP.

Zhang et al. (2011) propuseram uma estratégia de partição não linear dos dados, feita em estrutura de árvore de CP que é implementada de maneira iterativa. A modelagem das curvas também pode ser realizada através da distribuição de probabilidade dos dados de uma CP como mistura de métodos não lineares, como feito em (DELICADO; SMREKAR, 2007), ou baseado em um esquema de subdivisões, de modo que as curvas são obtidas de maneira iterativa, como proposto por Zhang e Luo (2005).

Há ainda estudos que propõem a hibridização de CP com outras técnicas, como é feito por Zhang, Miao e Pedrycz (2013) e Zhang et al. (2014) que apresentam técnicas de clusterização dos dados com a utilização de Fuzzy C-Means gerando, assim, curvas principais granulares, mais robustas à grandes números de dados de entrada. São introduzidos, também, índices para avaliar a fuzzycidade das saídas das curvas principais granulares e mostradas aplicações práticas das mesmas. Já Ghassabeh e Rudzicz (2015) estuda a situação onde o conjunto de dados a ser analisado não é inteiramente conhecido previamente e é proposto um algoritmo de treinamento para superar tal limitação. Por sua vez, Zhang, Chen e Kruger (2008) apresenta um algoritmo de restrição adaptativa que pode ser utilizado na presença de distribuições desiguais e escassas, assim como *outliers*, dados que não pertencem às classes previamente classificadas.

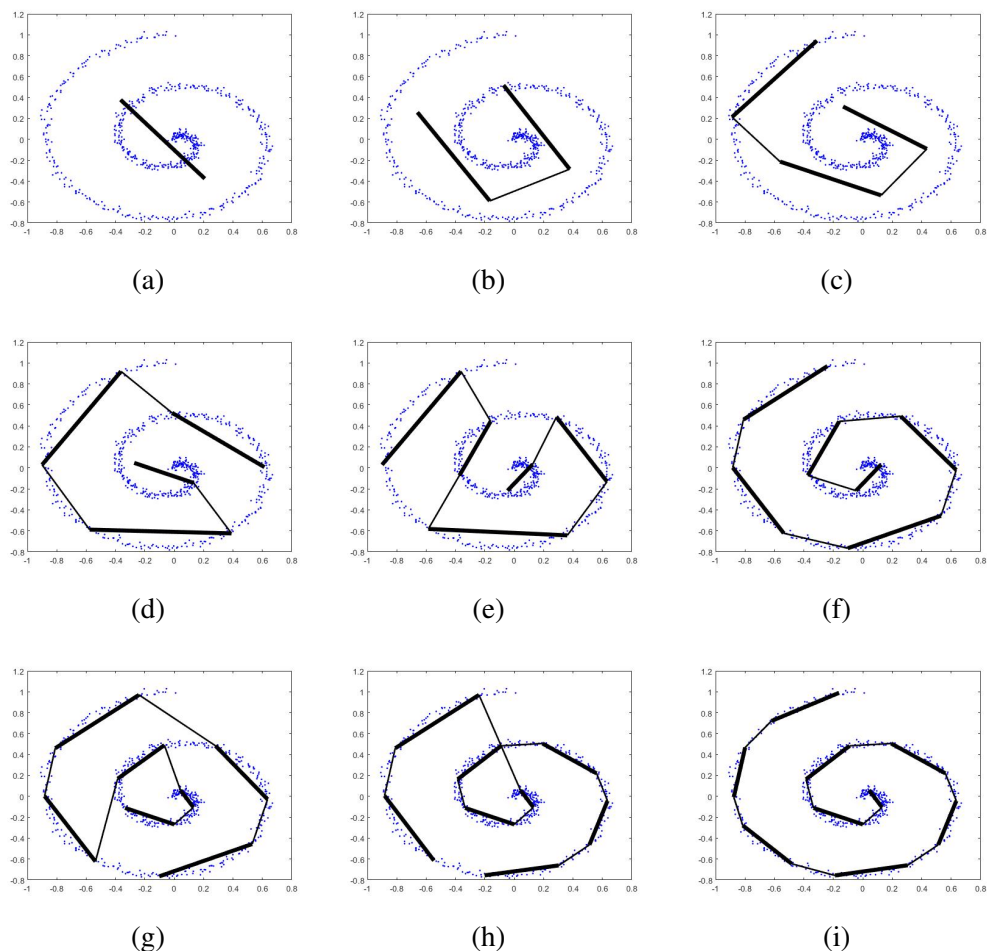
O algoritmo de extração de curvas k-segmentos, proposto por Verbeek, Vlassis e Kröse (2002), se destaca devido à sua robustez, baixa suscetibilidade a mínimos locais, convergência garantida e capacidade de lidar com dados que interceptam a si mesmo, quando comparado com outros algoritmos, tal como o PLA, proposto por Kegl et al. (2000). Devido às suas caracte-

rísticas é amplamente utilizado e é apresentado na Seção 2.1.3, uma vez que a otimização dos parâmetros iniciais deste algoritmo é o foco principal deste trabalho.

2.1.3 Algoritmo k-segmentos

A robustez desse algoritmo, sua convergência garantida e sua baixa susceptibilidade a mínimos locais faz com que esse algoritmo se destaque em relação aos demais (VERBEEK; VLASSIS; KRÖSE, 2002). A extração das CP a partir deste algoritmo é feita de maneira incremental, iniciando-se com apenas um segmento, sendo o número de segmentos aumentado progressivamente, com a otimização da curva a cada inserção.

Figura 2.4 – Sequência da construção de uma curva principal pelo algoritmo de k-segmentos



Fonte: Acervo do Autor

Na Figura 2.4 pode-se visualizar a sequência de construção de uma CP para um conjunto de dados com distribuição arbitrária, tendo como parâmetros $k = 9$, $f = 0,6$ e $\lambda = 1$, de modo que esses parâmetros representam, respectivamente, quantidade e comprimento dos segmentos e coeficiente de suavidade, que tem como função determinar a ligação entre os diferentes

segmentos. Observa-se, em (a) o segmento de reta, cuja direção coincide com a primeira componente principal do conjunto de dados. Em (b) há dois segmentos ligados por um segmento de junção e nas imagens posteriores observa-se a construção da curva que representa o conjunto de dados. Nessa figura as linhas grossas representam os segmentos obtidos pelo algoritmo de extração da curva, enquanto as linhas finas são as junções que unem os segmentos.

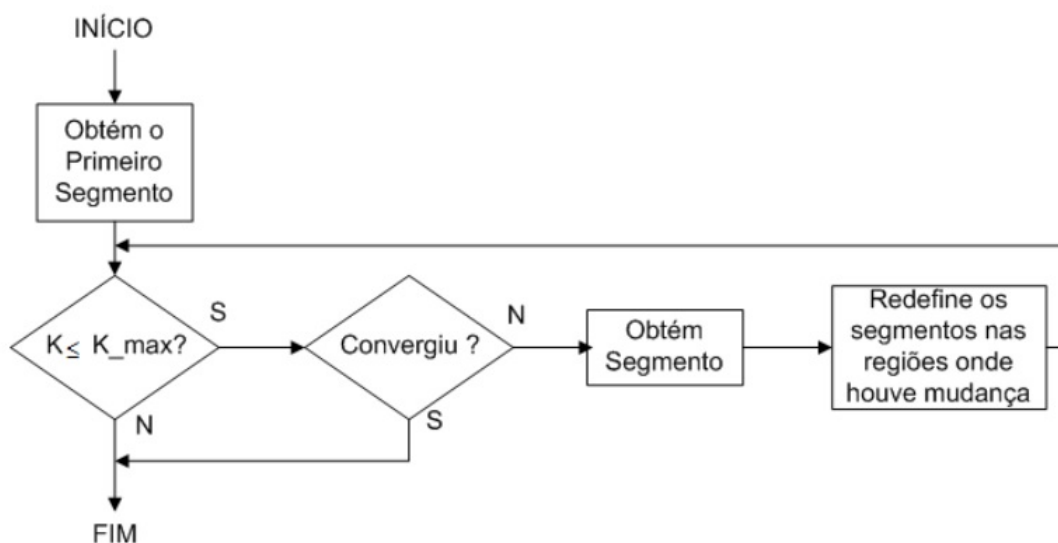
O algoritmo k -segmentos possui complexidade computacional $O(n^2)$, em que n é o número de eventos do conjunto de dados utilizado na construção da curva. Seu funcionamento pode ser dividido em três passos, que são descritos a seguir:

- **Passo 0- Inserção do primeiro segmento:** A inserção do primeiro segmento é feita considerando-se todos os eventos do conjunto de dados. Define-se um centro correspondente ao valor médio dos dados e a partir dele obtém-se o primeiro segmento, na direção do primeiro componente principal, com comprimento de $3/2$ do desvio padrão associado a este componente.
- **Passo 1- Inserção de novo segmento:** Para os segmentos seguintes define-se um ponto a ser tomado como centro a partir de agrupamento k -means. Baseando-se nas regiões de Voronoi, compostas por eventos que estão mais próximos do centro da região do que de um dos segmentos já existentes, é feita a escolha dos eventos que pertencerão ao novo agrupamento. Os segmentos existentes são recalculados uma vez que os eventos a eles pertencentes não são mais os mesmos. Une-se, então, os segmentos existentes por uma linha reta (não-suave).
- **Passo 2- Verificação dos critérios de parada:** É feita a verificação se houve a convergência do algoritmo ou se o número máximo de segmentos, definido pelo usuário, K_{max} , foi atingido, de modo que a convergência do algoritmo ocorre quando o maior agrupamento possível tem menos de três segmentos. Caso nenhum dos critérios de parada do algoritmo for atingido volta-se para o Passo 1.

Na Figura 3.1 é apresentado um fluxograma que ilustra o funcionamento do algoritmo, onde K representa o número atual de segmentos e K_{max} o número máximo de segmentos definido pelo usuário.

Após o término da execução do algoritmo, é sugerida uma determinada quantidade de segmentos para a curva principal baseando-se na variação do comprimento total da curva após

Figura 2.5 – Fluxograma simplificado do algoritmo k-segmentos



Fonte: Fernandez (2005)

a inserção de um segmento. Uma vez que a seleção das regiões de Voronoi¹ é feita em ordem decrescente, a cada iteração o segmento inserido é menor que o anterior até que o novo segmento inserido não represente uma grande modificação na curva como um todo, chegando assim ao número de segmentos sugeridos pelo algoritmo. O algoritmo *k-means*, utilizado na construção da CP possibilita o agrupamento dos dados em regiões de Voronoi apesar de não produzir curvas que, de certa maneira, incorporam o conceito de auto-consistência. Apesar disso, dois aspectos fazem da curva obtida pelo algoritmo k-segmentos uma CP:

- os segmentos com direção da primeira componente principal são auto-consistentes;
- se a região é uma região de Voronoi, então os k pontos pertencentes a ela são auto-consistentes.

2.2 Algoritmos Bioinspirados

Computação bioinspirada é um ramo de pesquisa que se baseia na natureza e em seus padrões para a elaboração de métodos de resolução de problemas. A lista de tecnologias utilizadas no cotidiano que foram desenvolvidas por meio de inspiração na natureza é extensa e varia

¹ Um diagrama de Voronoi é um particionamento de um plano em regiões com base na distância até pontos específicos do plano, chamados sementes. Para cada semente existe uma região correspondente que consiste em todos os pontos mais próximos daquela semente do que de qualquer outra. Essas regiões são chamadas de regiões de Voronoi.

desde objetos simples, tais como o velcro, inspirado em plantas, à objetos complexos como os aviões e submarinos, baseados nas aves e peixes, respectivamente. No ramo da computação pode-se destacar as Redes Neurais Artificiais, inspiradas nos neurônios humanos e a Lógica Nebulosa que se baseia na capacidade linguística dos homens.

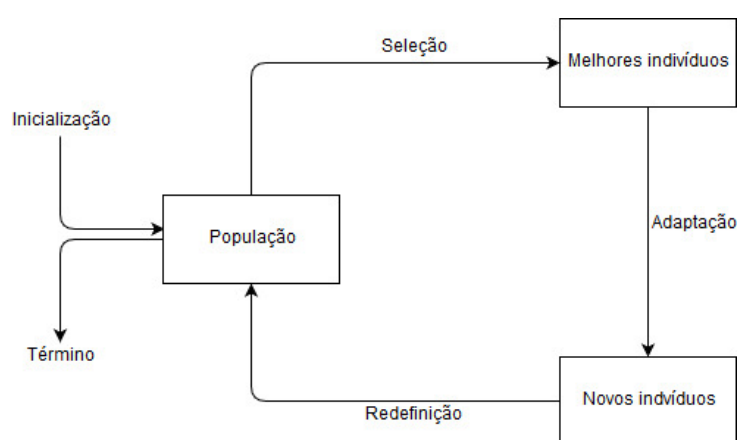
A inspiração na natureza também ocorre em diversos métodos heurísticos de otimização. Segundo Rao e Patel (2013) o desenvolvimento de tais técnicas ocorre devido à necessidade de se encontrar ótimos globais em problemas não-diferenciáveis, para quais métodos baseados em gradiente não podem ser utilizados. Tais algoritmos podem ser classificados em diversos grupos dependendo dos critérios considerados. De acordo com a natureza do fenômeno simulado a separação pode ser feita em dois grupos: Algoritmos Evolucionários, tais como Algoritmos Genéticos e de Evolução Diferencial, e Algoritmos baseadas em Inteligência de Populações, tais como Algoritmo de Colônia de Formigas e Otimização de Enxame de Partículas (PSO).

Há, ainda, algoritmos baseados em sistemas imunológicos e em diversas outras inspirações. A utilização desses algoritmos pode apresentar resultados ainda melhores por meio de sua combinação, onde as melhores características de cada um é mantida. Os algoritmos bioinspirados apresentam diversas variantes, mas, de maneira geral, seu funcionamento pode ser explicado pelos seguintes passos:

- **Passo 0- Inicialização e definição da população:** O algoritmo é inicializado com a definição da população inicial, definida de maneira aleatória. Dá-se o nome de população aos valores candidatos a serem solução do problema definido.
- **Passo 1- Seleção dos melhores indivíduos:** Após a avaliação da qualidade dos indivíduos da população, são escolhidos aqueles que se mostram melhores candidatos a solucionar o problema. Os piores candidatos não são completamente eliminados, uma vez que a sua eliminação poderia levar à uma convergência precoce do algoritmo a “melhores resultados” locais impossibilitando a obtenção de “melhores resultados” globais, pois nesse caso a população perderia um fator importante de sua evolução, a diversidade.
- **Passo 2- Adaptação dos indivíduos:** Nessa etapa ocorre a troca de informação entre os indivíduos da população ou alterações no próprio indivíduo. Tal processo geralmente ocorre por meio da interação entre os melhores membros da população, selecionados no passo anterior, e os demais.

- **Passo 3- Redefinição da população:** Os indivíduos gerados no passo anterior são incorporados à população, redefinindo-a.
- **Passo 4- Verificação dos critérios de parada:** É feita a verificação se a população é solução para o problema, de acordo com os critérios de qualidade das respostas definidos pelo usuário, ou se o número máximo de iterações foi atingido. Caso nenhum dos dois critérios de parada foi alcançado volta-se para o Passo 1.

Figura 2.6 – Fluxograma de um algoritmo bioinspirado



Fonte: Adaptado de (EIBEN; SMITH, 2003)

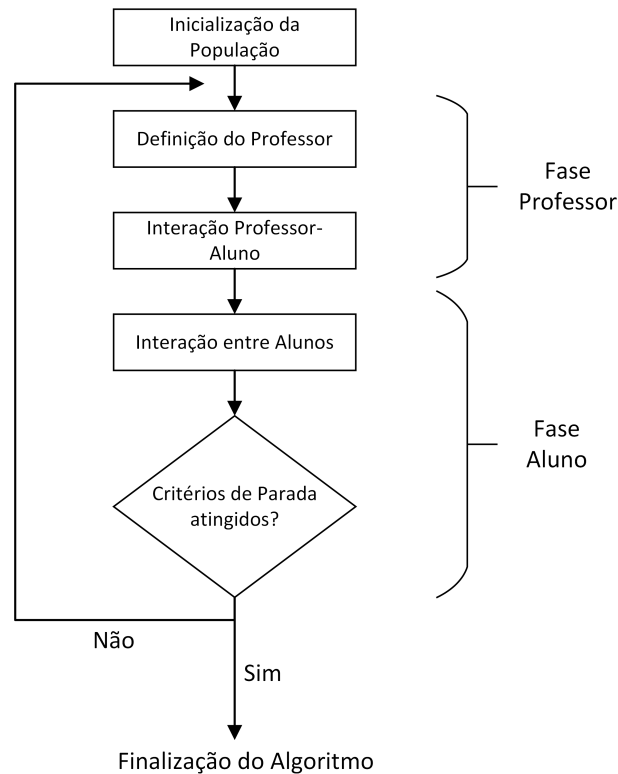
Destaca-se que o algoritmo explicado por meio dos passos e representado na Figura 2.6 é um modelo geral de algoritmo bioinspirado, cuja real implementação pode apresentar diversas modificações de modo a atender especificidades de diversos problemas e devido à inspiração utilizada em sua concepção. Um exemplo de quão variadas são as inspirações encontradas para o desenvolvimento de meta-heurísticas é o algoritmo *Teaching-learning-based optimization*, que se baseia nas interações que ocorrem dentro de uma sala de aula, apresentado na Seção 2.3.

2.3 *Teaching-learning-based optimization*

Teaching-learning-based optimization (TLBO) é uma meta-heurística baseada em populações proposta em (RAO; SAVSANI; VAKHARIA, 2011). Seu funcionamento se inspira nas interações sociais existentes dentro de uma sala de aula. É dividido em duas fases, como apresentadas a seguir e visualizado no fluxograma da Figura 2.7:

- **Fase Professor:** a cada iteração identifica-se na população o indivíduo que apresenta melhor aptidão para resolver o problema em questão. A este indivíduo é atribuída a

Figura 2.7 – Fluxograma do algoritmo TLBO



Fonte: Acervo do Autor

função de professor. Os outros indivíduos “aprendem” com o professor se movendo em sua direção, por meio de fatores de aprendizado calculados de maneira aleatória.

- **Fase Aluno:** nesta fase os “alunos” interagem entre si de maneira a aumentar seu “conhecimento”. O fator de interação entre os indivíduos é calculado de maneira aleatória. Esta interação faz com que os indivíduos da população se aproximem das soluções ótimas. Verifica-se, então, se os critérios de parada foram atingidos para a realização de mais um laço iterativo ou finalização do algoritmo.

A escolha desse método de otimização foi feita devido à sua simplicidade computacional e bom desempenho para solução de problemas complexos, uma vez que apresenta fácil implementação e requer poucos parâmetros de ajuste pelo usuário, tais como o tamanho da população e número de iterações (GARCIA et al., 2017). Seu funcionamento é apresentado com mais detalhes no pseudocódigo 1. Estas características fazem com que esse método seja efetivo para a otimização da escolha dos parâmetros iniciais do algoritmo de k-segmentos proposta neste trabalho.

Algorithm 1 Teaching-Learning-based Optimization

Require: $nDimensões \leftarrow$ número de dimensões do problema a ser endereçado

Require: $funçãoCusto \leftarrow$ função que calcula o custo, que direcionará os indivíduos ao longo das iterações

Require: $numeroDeIndividuos \leftarrow$ número de indivíduos da população a ser inicializada

```

1: inicializa população pop aleatoriamente e calcula os custos de cada indivíduo
2: MelhorSolução  $\leftarrow null$ 
3: for  $i \leftarrow 1 : nIteracoes$  do
4:    $media \leftarrow$  media das posições dos indivíduos;
5:   seleciona teacher() {melhor indivíduo}
6:
7:   {início da fase teacher}
8:   for  $j \leftarrow 1 : numeroDeIndividuos$  do
9:     cria novo indivíduo novoIndividuo
10:     $TF \leftarrow round(1 + rand(0, 1))$  {TF = fator de aprendizagem (teaching factor)}
11:     $novoIndividuo.Posicao \leftarrow pop(j).Posicao + rand() * (teacher.Posicao - (TF * media))$ 
12:     $novoIndividuo.Custo = funçãoCusto(novoIndividuo.Posicao)$ 
13:
14:    if  $novoIndividuo.Custo < pop(j).Custo$  then
15:       $pop(j) = novoIndividuo;$ 
16:      if  $pop(j).Custo < MelhorSolução.Custo$  then
17:         $MelhorSolução = pop(j)$ 
18:      end if
19:    end if
20:  end for
21:  {fim da fase teacher}
22:
23:  {início da fase learner}
24:  for  $j \leftarrow 1 : numeroDeIndividuos$  do
25:     $k \leftarrow$  número entre 1 e  $numeroDeIndividuos$  diferente de  $j$ 
26:    if  $pop(k).Custo > pop(j).Custo$  then
27:       $passo \leftarrow pop(j).Posicao - pop(k).Posicao$ 
28:    else
29:       $passo \leftarrow pop(k).Posicao - pop(j).Posicao$ 
30:    end if
31:
32:    cria novo indivíduo novoIndividuo
33:     $novoIndividuo.Posicao \leftarrow pop(j).Posicao + rand() * passo$ 
34:     $novoIndividuo.Custo = funçãoCusto(novoIndividuo.Posicao)$ 
35:
36:    if  $novoIndividuo.Custo < pop(j).Custo$  then
37:       $pop(j) = novoIndividuo;$ 
38:      if  $pop(j).Custo < MelhorSolução.Custo$  then
39:         $MelhorSolução = pop(j)$ 
40:      end if
41:    end if
42:  end for
43:  {fim da fase learner}
44: end for

```

2.4 *Improved Teaching-learning-based optimization*

Proposto por Yu, Wang e Wang (2016), o algoritmo *Improved Teaching-learning-based optimization* (ITLBO), é uma melhoria do algoritmo TLBO, ao qual são adicionados fases de *feedback*, mutação e cruzamento do algoritmo de Evolução Diferencial² (DE), além de mecanismos de perturbação caótica, com o objetivo de aumentar a performance do algoritmo original.

Além das fases Professor e Aluno, que fazem parte do algoritmo original, este algoritmo apresenta as seguintes fases:

- **Feedback:** Esta fase tem por objetivo aumentar a capacidade de aprendizado dos alunos. Isso é feito através da escolha aleatória de um aluno que apresenta resultados ruins e de sua interação direta com o professor. Esta etapa faz com que haja diminuição da área de busca, o que gera aumento da acurácia e velocidade de otimização.
- **Mutação e Cruzamento:** A perda de diversidade da população nas fases anteriores pode causar convergência prematura do algoritmo, ou seja, convergência para um mínimo local. De modo a mitigar esses problemas são realizados mutação e cruzamento entre indivíduos, com o objetivo de se aumentar a diversificação da população.
- **Perturbação Caótica:** O mecanismo de Perturbação Caótica é utilizado para que se escape dos ótimos locais e que se aumente a precisão do algoritmo TLBO. Segundo Yu, Wang e Wang (2016) a aleatoriedade e ergodicidade do caos são usadas para prevenir que a busca encontre um ótimo local e supere as deficiências de um algoritmo de otimização tradicional.

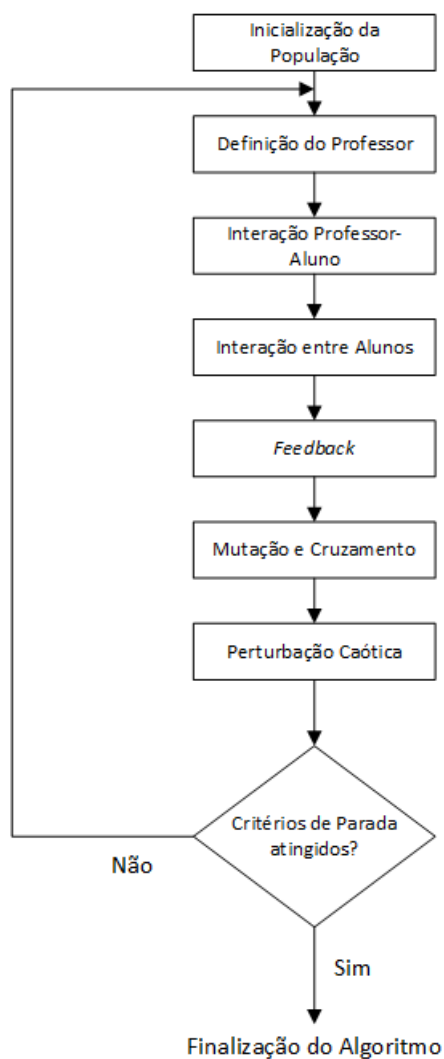
Dessa maneira, o funcionamento do algoritmo ITLBO ocorre de acordo com o fluxograma apresentado na Figura 2.8.

2.5 *Grey Wolf Optimization*

A exemplo de diversos algoritmos bioinspirados, o *Grey Wolf Optimizer* (GWO) é inspirado no comportamento de populações animais, nesse caso específico, em matilhas de lobos-cinzentos, *Canis lupus*.

² Proposto por Storn e Price (1997), é um algoritmo baseado em populações, que utiliza informações de direção e distância entre seus indivíduos, além de conceitos de mutação e cruzamento para realizar a otimização de problemas.

Figura 2.8 – Fluxograma do algoritmo ITLBO



Fonte: Acervo do Autor

Proposto por Mirjalili, Mirjalili e Lewis (2014) esta meta-heurística é baseada na forma em que a matilha de lobos se organiza durante a caça, em que os animais se dividem hierarquicamente em quatro grupos distintos:

- Alfa (α): É o líder da matilha, sendo seguido pelos outros lobos.
- Beta (β): São subordinados do líder e o ajudam nas decisões e a garantir o cumprimento delas perante os outros animais do grupo.
- Delta (δ): Parte da matilha em que se encontram os caçadores, sentinelas, velhos e outras categorias.

- Omega (ω): São os últimos na hierarquia, sendo em alguns casos responsáveis por cuidar das crias. Apesar de seu baixo prestígio hierárquico são necessários para a boa convivência da matilha.

Seu ritual de caça pode ser dividido em três etapas distintas:

1. Rastreamento, perseguição e aproximação da presa;
2. Perseguição da presa, até que ela pare de se mover;
3. Ataque à presa.

São nessas etapas e na divisão hierárquica da matilha que se inspira o algoritmo de otimização GWO. Para isso são utilizados modelos matemáticos, apresentados em (MIRJALILI; MIRJALILI; LEWIS, 2014), para simular o comportamento dos mesmos durante sua busca por alimentos, simulando os processos de busca e circundação da presa, da caça e de ataque à presa.

Algorithm 2 Grey Wolf Optimization

Require: nDimensões \leftarrow número de dimensões do problema a ser endereçado

Require: funçãoCusto \leftarrow função que calcula o custo, que direcionará os indivíduos ao longo das iterações

Require: numeroDeIndividuos \leftarrow número de indivíduos da população a ser inicializada

- 1: inicializa população de lobos cinzentos X_i ($i = 1, 2, \dots, n$)
 - 2: inicializa α, A e C
 - 3: calcula os custos de cada lobo
 - 4: $X_\alpha =$ melhor lobo
 - 5: $X_\beta =$ segundo melhor lobo
 - 6: $X_\delta =$ terceiro melhor lobo
 - 7: **while** ($t < nIteracoes$) **do**
 - 8: **for** agentes de busca **do**
 - 9: atualizar posição do lobo atual pela equação $\vec{x}(t+1) = \frac{\vec{x}_1 + \vec{x}_2 + \vec{x}_3}{3}$
 - 10: **end for**
 - 11: atualiza α, A e C
 - 12: calcula os custos de cada lobo
 - 13: atualiza X_α, X_β e X_δ
 - 14: $t = t + 1$
 - 15: **end while**
 - 16: retorna X_α
-

Por meio desse paralelo à caça, a busca por soluções ideais aos problemas tratados pelo algoritmo é realizada. O funcionamento dessa meta-heurística é apresentado no Pseudocódigo

2. Segundo (MIRJALILI; MIRJALILI; LEWIS, 2014) esse modelo de otimização apresenta

alto desempenho na busca por soluções em espaços de buscas desconhecidos além de mostrar-se robusto ao evitar ótimos locais.

Devido às características dessa meta-heurística, a mesma é utilizada para testar a viabilidade do método proposto neste documento. A exemplo do TLBO, que é incrementado pelo ITLBO, o GWO também apresenta um algoritmo que o sucede, apresentado na Seção 2.6, e também utilizado na verificação da qualidade da busca otimizada pelos parâmetros iniciais do algoritmo k-segmentos, apresentado neste documento.

2.6 Chaotic Grey Wolf Optimization

De modo a acelerar a velocidade de convergência global, Kohli e Arora (2018) propõem o algoritmo *Chaotic Grey Wolf Optimizer* (CGWO), cujo funcionamento pode ser visto no Pseudocódigo 3. Segundo estes autores, a meta-heurística GWO, em certos casos, falha ao encontrar o mínimo global, uma vez que se utiliza de mecanismos de aleatoriedade na busca das soluções.

Algorithm 3 Chaotic Grey Wolf Optimization

Require: nDimensões \leftarrow número de dimensões do problema a ser endereçado

Require: funçãoCusto \leftarrow função que calcula o custo, que direcionará os indivíduos ao longo das iterações

Require: numeroDeIndividuos \leftarrow número de indivíduos da população a ser inicializada

- 1: inicializa o contador de gerações t e a população de lobos cinzentos X_i ($i = 1, 2, \dots, n$)
 - 2: inicializa aleatoriamente o mapa caótico x_0
 - 3: inicializa α, A e C
 - 4: calcula os custos de cada lobo
 - 5: $X_\alpha =$ melhor lobo
 - 6: $X_\beta =$ segundo melhor lobo
 - 7: $X_\delta =$ terceiro melhor lobo
 - 8: **while** ($t < nIteracoes$) **do**
 - 9: Ordena a população de lobos cinzentos de acordo com o custo
 - 10: Atualiza o número caótico utilizando a equação do mapa caótico
 - 11: **for** agentes de busca **do**
 - 12: atualizar posição do lobo atual pela equação $\vec{x}(t+1) = \frac{\vec{x}_1 + \vec{x}_2 + \vec{x}_3}{3}$
 - 13: **end for**
 - 14: atualiza α, A e C
 - 15: calcula os custos de cada lobo
 - 16: atualiza X_α, X_β e X_δ
 - 17: substitui o pior lobo pelo melhor lobo
 - 18: $t = t + 1$
 - 19: **end while**
 - 20: retorna X_α
-

O algoritmo proposto por Kohli e Arora (2018) utiliza mapas caóticos³ unidimensionais com o objetivo de acelerar a convergência do GWO e fazer com que o mínimo global seja encontrado.

Segundo Kohli e Arora (2018) o caos é um método determinístico encontrado em sistemas dinâmicos não-lineares, não-periódico, não-convergente e limitado que faz com que sistemas caóticos possam ser considerados como aleatoriedades, introduzidos em sistemas de otimização por meio de mapas caóticos

A inserção desses mecanismos faz com que a busca por presas, simulada pelo GWO, seja realizada de maneira mais ampla, evitando que o algoritmo finalize sua busca em um ponto de mínimo local, fazendo com que a meta-heurística CGWO apresente bom desempenho em problemas reais de alta complexidade, o que justifica sua utilização nos testes realizados no presente trabalho.

2.7 *Non-dominated Sorting Genetic Algorithm II*

Muitos problemas de otimização possuem múltiplos objetivos. Para tais problemas, não há uma única solução ótima, mas um conjunto delas. A fronteira ou conjunto de Pareto é o conjunto das soluções ótimas para um problema de otimização multiobjetivo. De modo geral, uma solução é dita Pareto-ótima se ela não é dominada por nenhuma outra solução possível, ou seja, a melhoria de um dos objetivos da solução prejudica os outros objetivos a serem alcançados.

O *Non-dominated Sorting Genetic Algorithm II* (NSGA-II), proposto por Deb et al. (2002), é um das mais populares meta-heurísticas multiobjetivo, devido à sua qualidade na determinação do conjunto de Pareto em problemas de otimização com múltiplos objetivos.

Baseado em algoritmos genéticos⁴, implementa o conceito de Dominância, ou seja, classifica a população em castas, de acordo com o grau de dominância, sendo os indivíduos pertencentes à primeira casta considerados melhores soluções do que os demais.

Por meio do conceito de Dominância, é agregado o conceito de Elitismo, com a qual a população é classificada em diferentes categorias de acordo com a qualidade da solução de cada indivíduo. Após essa classificação, a população passa pelo Operador de Diversidade, que a ordena de acordo com a distância de cada indivíduo em relação aos pontos vizinhos. Com

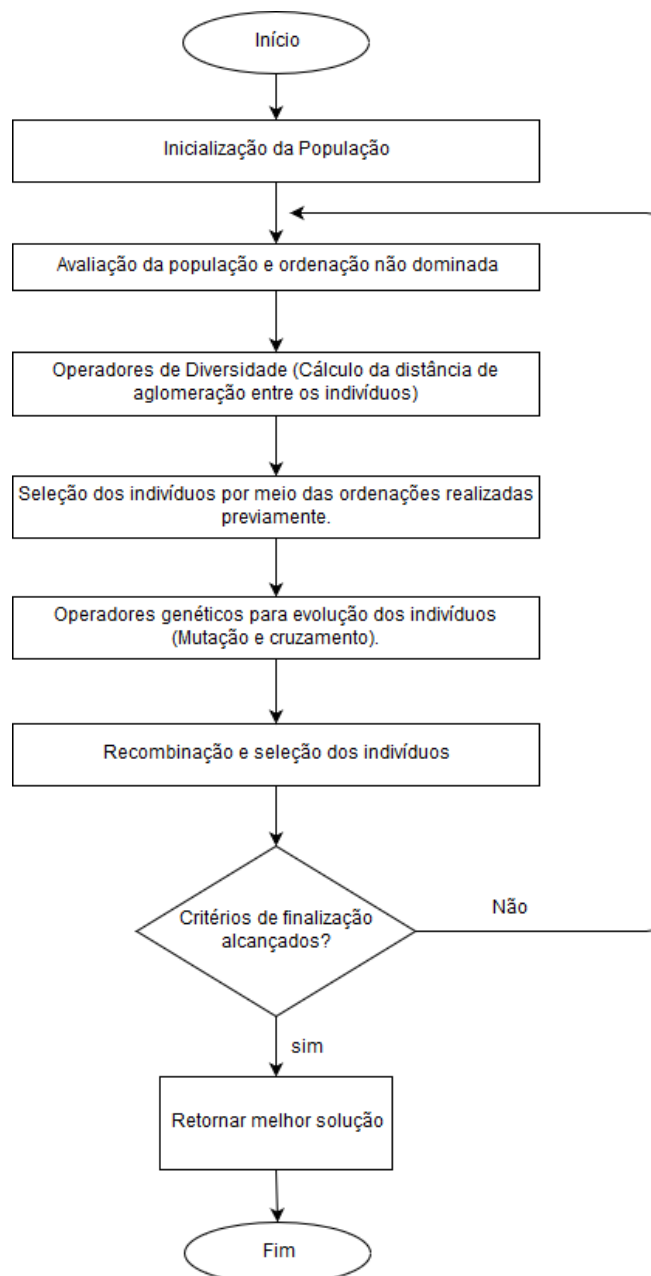
³ Mapa caótico é um mapa (função de evolução) que exhibe algum tipo de comportamento caótico.

⁴ Algoritmos genéticos são uma classe particular de algoritmos bioinspirados que utilizam técnicas inspiradas na Teoria da Evolução das Espécies de Charles Darwin, como o conceito de hereditariedade, mutação, seleção natural e recombinação.

esse processo evita-se que as soluções se aglomerem em um só ponto, garantindo a diversidade das soluções, sendo possível, dessa maneira, a determinação do conjunto de Pareto do problema tratado.

Segundo Yusoff, Ngadiman e Zain (2011), o NSGA-II pode ser, de maneira geral, dividido em 6 diferentes etapas: Inicialização da População, Ordenação não dominada, Operador de Diversidade, Seleção, Operadores Genéticos, Recombinação e Seleção. O funcionamento desse algoritmo, de acordo com essas etapas, é apresentado de maneira sucinta no fluxograma da Figura 2.9.

Figura 2.9 – Fluxograma do algoritmo NSGA-II



Fonte: Acervo do Autor

3 OTIMIZAÇÃO DE PARÂMETROS DO ALGORITMO K-SEGMENTOS PARA REPRESENTAÇÃO COM UTILIZAÇÃO DE TLBO

Neste capítulo é apresentado o estudo feito a respeito da viabilidade de se utilizar meta-heurísticas para otimização dos parâmetros iniciais do algoritmo k-segmentos. Para isso é utilizada a meta-heurística TLBO, de modo que seja possível a otimização dos parâmetros. Como proposta, é necessária a determinação de uma função-custo, cuja minimização garanta a construção de CP que represente os dados de maneira eficaz. Por esse motivo, o objetivo específico deste capítulo é a definição dessa função-custo e a apresentação dos resultados alcançados com sua utilização.

3.1 Método de seleção automática de parâmetros

A efetividade da classificação de dados pelo algoritmo k-segmentos depende da escolha correta dos parâmetros do algoritmo, tais como o comprimento e a quantidade de segmentos que compõem a curva, além da suavidade da ligação entre os segmentos, que segundo Wang e Lee (2006) tem efeitos visíveis, principalmente, em conjuntos de dados onde há interseções.

A primeira parte da pesquisa foi determinada pela busca por uma função-custo que possa ser utilizada na otimização dos parâmetros do algoritmo de construção de curvas, comprimento e quantidade de segmentos, destacando que os resultados obtidos pela otimização estão diretamente ligados à correta definição da função objetivo.

Nesta etapa inicial foi utilizado o algoritmo k-segmentos não suave proposto por Verbeek, Vlassis e Kröse (2002), de modo que o método de otimização proposto é também aplicado ao algoritmo k-segmentos suave, proposto por Verbeek, Vlassis e Kröse (2001). Os resultados da otimização para este último algoritmo são apresentados no Apêndice A.

Dessa maneira, para essa etapa foi utilizada a meta-heurística TLBO, para a qual se definiu como 50 o tamanho da população, inicializada aleatoriamente dentro de uma faixa de valores específica. Para as aplicações desse capítulo, as faixas de valores para a determinação da quantidade de segmentos é de (1; 40) e para comprimento dos segmentos, (0,5; 2). É importante observar que a escolha desses intervalos se deu pelo conhecimento das bases de dados utilizadas de modo a acelerar a determinação dos parâmetros ótimos. Caso não se tenha conhecimento dos dados a serem representados, basta escolher intervalos maiores para que o algoritmo encontre automaticamente esses parâmetros. Isso faz com que haja aumento do custo computacional, porém, sem prejuízo à qualidade de representação da curva construída.

A escolha do tamanho da população foi feita de maneira a garantir-lhe uma alta variedade. Por sua vez os indivíduos, que representam os parâmetros a serem otimizados, passam pelas fases “professor” e “aluno” por 30 iterações para garantir a convergência do algoritmo, destacando que o resultado ótimo pode ser alcançado em um número menor de iterações.

Definiu-se como objetivo do algoritmo de otimização a minimização da equação (VERBEEK; VLASSIS; KRÖSE, 2002):

$$(1 - p)n \log l + p \sum_{i=1}^k \sum_{x \in V_i} d(s_i, \mathbf{x})^2 / (2\sigma^2), \quad (3.1)$$

em que n é o número de eventos, l é o comprimento total da curva, $d(s_i, x)$ a distância de cada evento à curva, σ^2 a variância dos dados ao longo dos segmentos, V_i as regiões de Voronoi às quais pertencem os eventos e p a ponderação, cuja determinação constituiu esta etapa da pesquisa.

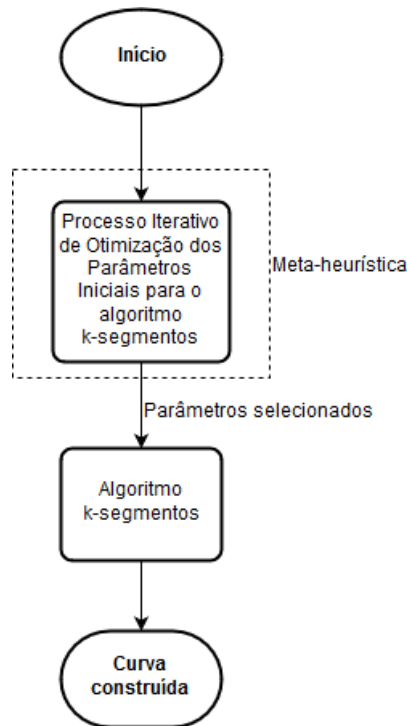
Segundo Verbeek, Vlassis e Kröse (2002), há uma variedade de critérios de parada para o algoritmo k-segmentos e, na prática, o mínimo global da Equação (3.1), com $p = 0,5$, é atingido quando há boa representatividade dos dados na curva gerada. Tal afirmação levou à escolha da Equação (3.1) como primeira função-custo a ser utilizada no presente trabalho. Apesar de Verbeek, Vlassis e Kröse (2002) ter utilizado $p = 0,5$ para todos os problemas, observou-se, durante a realização deste trabalho, a necessidade de alterar o peso dos dois termos dessa equação de modo a representar melhor os dados, como é discutido ao longo deste capítulo.

De modo geral a construção de CP pelo método proposto ocorre como apresentado na Figura 3.1.

O Processo de otimização se dá com a utilização de meta-heurísticas, destacando o fato que a avaliação da população do algoritmo de otimização de acordo com a função-custo é realizado por meio do algoritmo k-segmentos, que, durante o processo de otimização é executado, tendo como parâmetros iniciais os membros da população do algoritmo de otimização. O fluxograma que mostra o processo iterativo de otimização com a utilização da meta-heurística TLBO pode ser visto na Figura 3.2.

Durante este processo diversas curvas são construídas, para as quais são computados comprimento total e distância dos eventos à curva, de modo a avaliar os membros da população de acordo com a função-custo, tarefa componente das meta-heurísticas. Dessa maneira as interações entre indivíduos e demais processos do algoritmo de otimização ocorrem para garantir a

Figura 3.1 – Fluxograma do funcionamento do método proposto



Fonte: Acervo do Autor

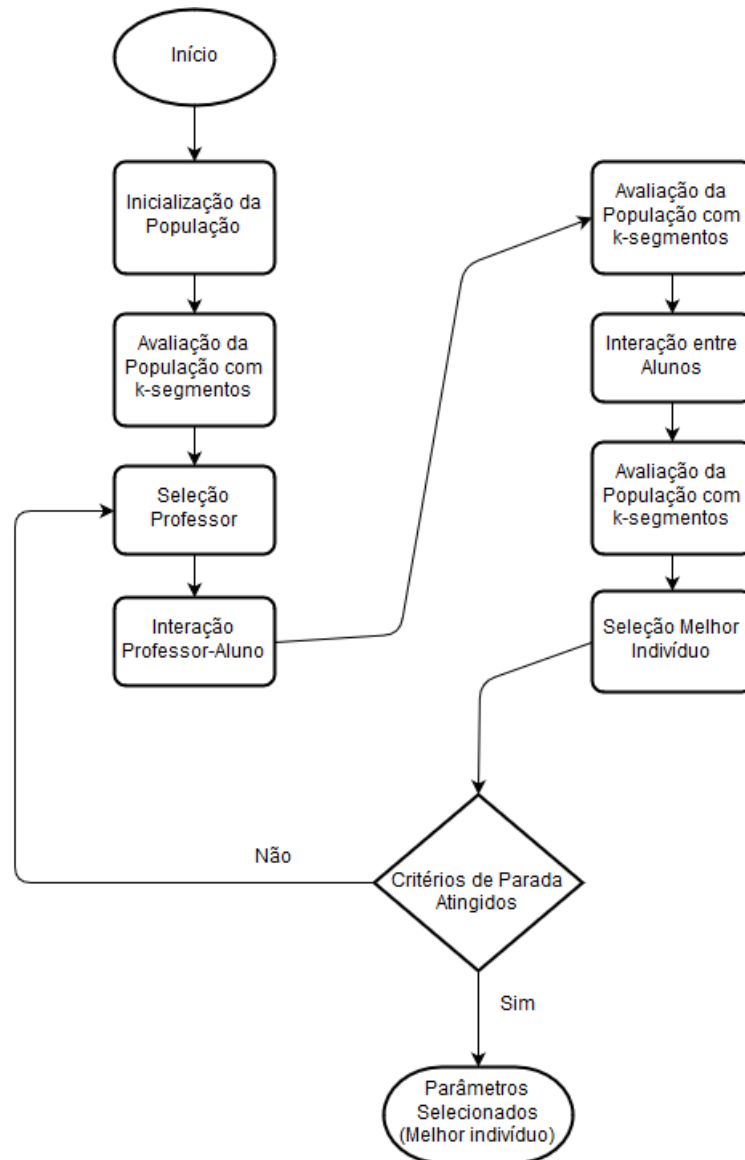
busca pela solução ideal, ou seja, os parâmetros iniciais necessários para a construção da curva que melhor representa os dados.

A definição da função-custo com a qual é possível realizar a otimização dos parâmetros foi realizada em duas diferentes etapas, tais como apresentadas a seguir:

1. Nesta etapa foi realizada a otimização da quantidade de segmentos, k , de modo que o comprimento dos segmentos foi mantido fixado em $3\sigma/2$, que segundo Verbeek, Vlassis e Kröse (2002) apresentou os melhores resultados para os testes realizados.
2. Nesta etapa realizou-se a otimização da quantidade e comprimento dos segmentos. Dessa forma, os membros da população são constituídos de pares (k, f) , onde $f\sigma$ é o comprimento dos segmentos, sendo σ^2 a variância dos dados ao longo dos segmentos, calculada pelo algoritmo, e k o número de segmentos.

A inicialização da população na fase inicial das meta-heurísticas foi feita de modo que cada parâmetro seja inicializado dentro de um intervalo de busca específico, ou seja, com restrições do problema de otimização. Isso é feito uma vez que a quantidade de segmentos deve ser um número inteiro maior ou igual a 1, enquanto o comprimento dos mesmos é dado em função da variância pelo parâmetro f , que por sua vez, possui um intervalo de variação de bom

Figura 3.2 – Fluxograma do Processo de Otimização com a utilização de TLBO



Fonte: Acervo do Autor

desempenho, observado experimentalmente, entre 0,5 e 2. A restrição do número de segmentos é utilizada de modo a garantir a correta construção da curva pelo algoritmo, uma vez que o mesmo necessita que haja pelo menos 1 segmento. Por outro lado, as restrições do comprimento dos segmentos são determinadas pelo conhecimento dos dados a serem representados, de maneira que intervalos diferentes podem ser utilizados de acordo com a aplicação e base de dados utilizada.

A discussão a respeito dessas duas etapas, 1 e 2, se encontram nas Seções 3.3 e 3.4, respectivamente.

3.2 Base de dados

De modo a verificar a capacidade de representação do método proposto foram utilizados dois diferentes padrões de dados. Tais conjuntos de dados artificiais formam uma espiral e um rabisco, que intercepta a si mesmo, ambos com ruído gaussiano. Os mesmos conjuntos de dados foram utilizados em (VERBEEK; VLASSIS; KRÖSE, 2002) e (WANG; LEE, 2006) para mostrar a efetividade dos respectivos algoritmos para construção de curvas principais.

3.2.1 Contorno de dados em espiral

A primeira base de dados utilizada para essa fase da pesquisa é o contorno de dados em espiral, como visto na Figura 3.3(a), em que cada um dos n pontos é formado por pares (x_1, x_2) , tal que

$$\begin{aligned}x_1 &= a_n \sin(x_n) + dr_1 \\x_2 &= a_n \cos(x_n) + dr_2\end{aligned}$$

em que $x_n \in X$, de modo que $X = \{x_n \in \mathbb{R} : -3\pi \leq x_n \leq 3\pi\}$ e a_n assume valores crescentes e igualmente espaçados entre si no intervalo $[0, 1]$. Por sua vez r_1 e r_2 são valores aleatórios gerados com distribuição normal e utilizou-se $d = 0,02$.

O número de pontos n é igual a 943, sendo esse o primeiro número inteiro maior que o resultado da multiplicação de 100π pelo número de voltas da espiral, 3, no caso específico utilizado.

3.2.2 Contorno de dados em rabisco

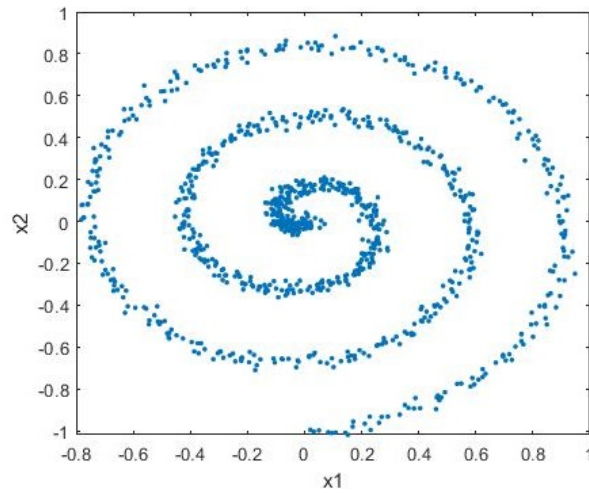
A segunda base de dados utilizada para essa fase da pesquisa é o contorno de dados em forma de rabisco, como apresentado na Figura 3.3(b), em que cada um dos pontos é formado por pares (x_1, x_2) tal que

$$\begin{aligned}x_1 &= \sqrt{t} \times (0, 1 + \sin(4\pi t + 0, 4)) + 0,01r \\x_2 &= 1, 1 + \cos(3\pi t + 0, 1) + 0, 1r\end{aligned}$$

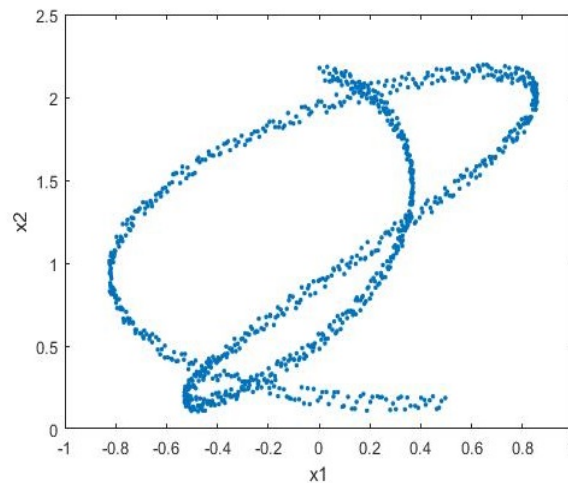
em que t pertence ao intervalo $[0, 1]$ e r é um valor aleatório com distribuição normal.

De modo a se obter uma quantidade razoável de pontos para esse conjunto de dados, foram tomados 1000 diferentes valores de t , igualmente espaçados entre si no intervalo a que pertence essa variável.

Figura 3.3 – Conjuntos de dados utilizados



(a) Contorno em forma de espiral



(b) Contorno em forma de rabisco

Fonte: Acervo do Autor

3.3 Resultados com comprimento do segmento fixo

Com o valor de comprimento dos segmentos fixado em $3\sigma/2$ foram realizados testes a respeito da capacidade do algoritmo de otimização em encontrar o número ideal de segmentos que compõem a CP para o conjunto de dados, além da viabilidade da utilização de meta-heurísticas para o fim proposto.

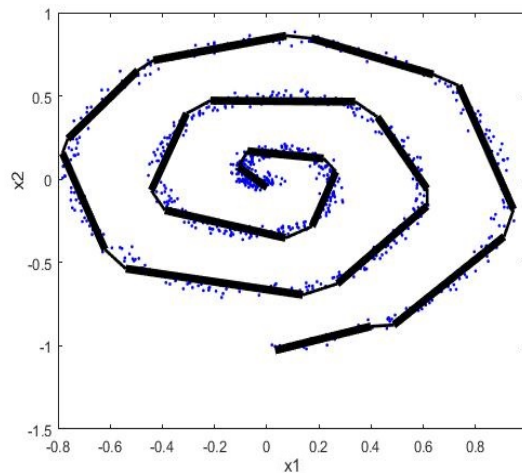
Primeiramente utiliza-se como função-custo do algoritmo de otimização TLBO a Equação (3.1), com ponderação $p = 0,5$. Essa ponderação gera a Equação:

$$0,5n \log l + 0,5 \sum_{i=1}^k \sum_{x \in V_i} d(s_i, \mathbf{x})^2 / (2\sigma^2). \quad (3.2)$$

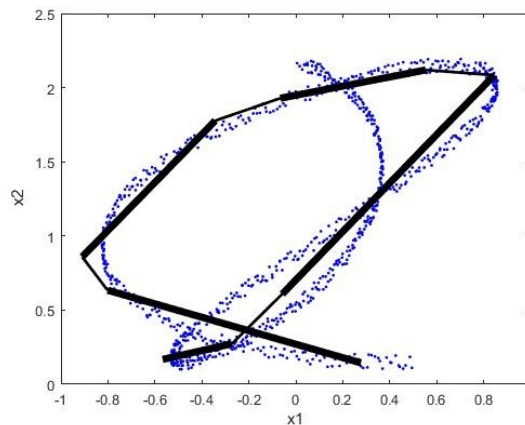
Segundo Verbeek, Vlassis e Kröse (2002), ao se alcançar mínimos locais da equação com essa ponderação, em que ambos termos da equação apresentam pesos iguais, é possível construir uma CP que representa de maneira adequada o contorno dos dados.

Com a utilização da Equação (3.2) como função-custo, obtém-se os resultados mostrados nas Figuras 3.4(a) e 3.4(b).

Figura 3.4 – Curvas geradas com o valor do comprimento do primeiro segmento fixado em $3\sigma/2$, construídas a partir da minimização da Equação (3.2)



(a) $k = 16; f = 1,5$



(b) $k = 5; f = 1,5$

Fonte: Acervo do Autor

É possível observar que a representação do contorno do conjunto de dados não é realizada de maneira eficaz pelas curvas geradas, sobretudo os dados visualizados em 3.4(b) onde se percebe de maneira clara que a quantidade de segmentos obtida pela otimização é insuficiente para gerar uma curva que percorre todo o conjunto de dados, como se espera. Na Figura 3.4(a) é possível perceber que a curva gerada não apresenta grande suavidade em seu contorno,

o que ocorre devido ao baixo número de segmentos propostos pelo TLBO. Observa-se, assim, a necessidade de se aumentar o número de segmentos para que haja representação adequada dos contornos.

De modo a se obter melhor representatividade dos dados por parte das CP gerada com parâmetros otimizados por meta-heurísticas é necessária uma análise mais detalhada dos termos da função-custo e o impacto nela causados pela variação dos parâmetro do algoritmo k-segmentos.

O primeiro termo dessa função, $n \log l$, é relativo ao comprimento total da curva, soma dos comprimentos dos segmentos e das linhas que os ligam. Esse termo assume valores maiores à medida que se aumenta o comprimento da curva e pode assumir valores negativos caso o comprimento total da curva seja suficientemente pequeno. É possível observar também que quanto maior o número de eventos do conjunto de dados, maiores os valores absolutos atingidos por esse termo, uma vez que o número de eventos é multiplicado pelo logaritmo do comprimento da curva.

O segundo termo, $\sum_{i=1}^k \sum_{x \in V_i} d(s_i, \mathbf{x})^2 / (2\sigma^2)$, é proporcional ao somatório quadrático das distâncias de cada evento aos segmentos da curva e inversamente proporcional à variância dos dados em relação aos segmentos que compõem a curva. Dessa forma, quanto maior a proximidade da curva com o contorno real dos dados menores tendem a ser os valores assumidos por esse termo.

Ao se analisar a equação quanto à variação da quantidade de segmentos é possível observar que seu aumento gera curvas de maior comprimento, aumentando, conseqüentemente, os valores relativos ao primeiro termo da equação. Por outro lado, uma maior quantidade de segmentos faz com que o somatório das distâncias dos eventos à curva diminua. Dessa maneira é possível atribuir pesos diferentes aos termos dessa equação de modo a obter curvas mais compactas e simples ou maiores e mais suaves.

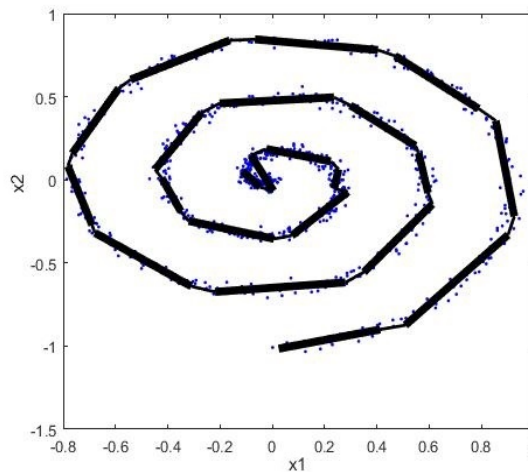
Com essa análise percebe-se que a ponderação igual dos termos ligados ao aumento do comprimento da curva e ao distanciamento dos pontos aos segmentos mostra-se ineficaz para representação de dados com alto número de eventos ou com grande dispersão, uma vez que esse tipo de conjunto de dados geralmente requer um alto número de segmentos para que a CP gerada pelo algoritmo k-segmentos seja capaz de representar de maneira eficaz seu contorno.

A partir dessa observação verificou-se a necessidade de utilizar uma ponderação diferente da utilizada por (VERBEEK; VLASSIS; KRÖSE, 2002) para que o impacto do aumento de número de segmentos no custo total da função objetivo seja menor.

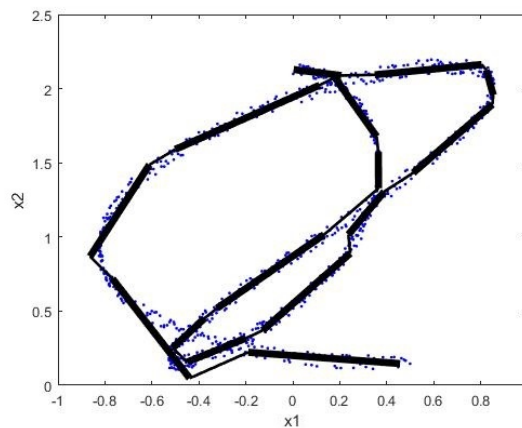
Com a realização de testes heurísticos observou-se que a utilização de valor de ponderação $p = 0,95$ mostra-se promissora para a construção de curvas que representem de maneira melhor o contorno dos dados, resultando na seguinte equação, a ser utilizada como função objetivo do algoritmo de otimização:

$$0,05n \log l + 0,95 \sum_{i=1}^k \sum_{x \in V_i} d(s_i, \mathbf{x})^2 / (2\sigma^2). \quad (3.3)$$

Figura 3.5 – Curvas geradas com o valor do comprimento do primeiro segmento fixado em $3\sigma/2$, construídas a partir da minimização da Equação (3.3)



(a) $k = 22; f = 1,5$



(b) $k = 15; f = 1,5$

Fonte: Acervo do Autor

As Figuras 3.5(a) e 3.5(b) apresentam os resultados obtidos com a utilização da função objetivo apresentada na Equação (3.3).

O aumento do número de segmentos devido à variação da função-custo é visível, o que, por sua vez faz com que a curva siga de maneira mais suave o contorno dos mesmos, evitando o subdimensionamento visualizado quando se utiliza a função objetivo proposta em (VERBEEK; VLASSIS; KRÖSE, 2002).

A baixa suavidade das curvas e as falhas encontradas na representação dos contornos visíveis, sobretudo na Figura 3.5(b), ocorrem devido ao fato de se ter utilizado valores fixos para os comprimentos dos segmentos. Desse modo, na Seção 3.4 são apresentados os resultados obtidos com a utilização do comprimento dos segmentos como parâmetro otimizado, dando prosseguimento aos testes heurísticos realizados para a determinação do valor da ponderação p , que tem como objetivo a definição da melhor função custo a ser utilizada pelo método de seleção automática de parâmetros iniciais proposto

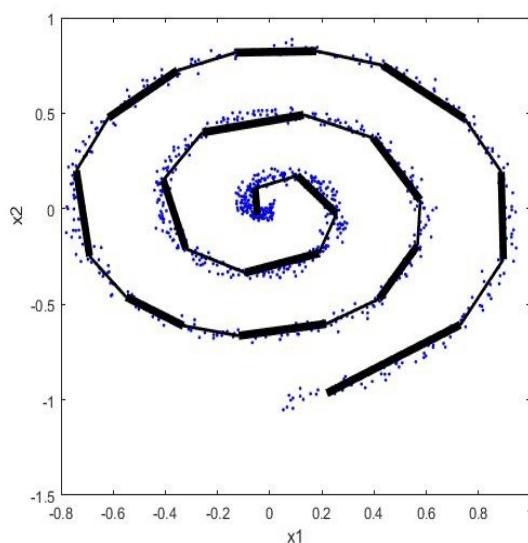
3.4 Resultados com comprimento do segmento como parâmetro otimizado

Apesar da função objetivo proposta por Verbeek, Vlassis e Kröse (2002), aqui apresentada como Equação (3.2), ter se mostrado inviável para representação por subdimensionar a curva, como apresentado na Seção 3.3, a primeira parte desta etapa é utilizar esta mesma função-custo de modo a verificar o comportamento da curva construída quando o comprimento dos segmentos é, assim como a quantidade deles, um parâmetro a ser otimizado. Observa-se nas Figuras 3.6(a) e 3.6(b) os resultados obtidos.

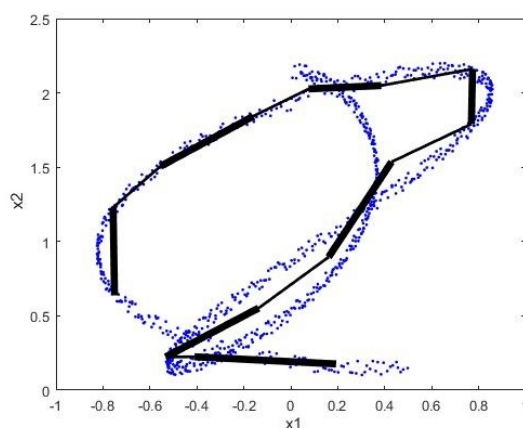
Para o conjunto de dados de 3.6(a) observa-se que, com a otimização do comprimento dos segmentos, a curva gerada é mais suave quando comparada com a curva gerada com comprimento fixo. Dessa maneira, a representação do contorno é feita de maneira satisfatória, apesar de ser perceptível que a suavidade da curva pode ser melhorada com uma diferente ponderação da função-custo. Já em 3.6(b), verifica-se que o subdimensionamento da quantidade de segmentos da curva faz com que a representação dos dados seja visualmente pior do que a obtida na Figura 3.4(b), da Seção 3.3, onde o comprimento dos segmentos foi mantido fixo.

Para ambos os casos a alteração do valor de p da função objetivo, ou seja, variação da ponderação entre os dois termos da equação, é capaz de solucionar tal deficiência, como apresentado previamente.

Figura 3.6 – Curvas geradas com o quantidade e comprimento dos segmentos como parâmetros otimizados, construídas a partir da minimização da Equação (3.2)



(a) $k = 15$; $f = 1,016797$



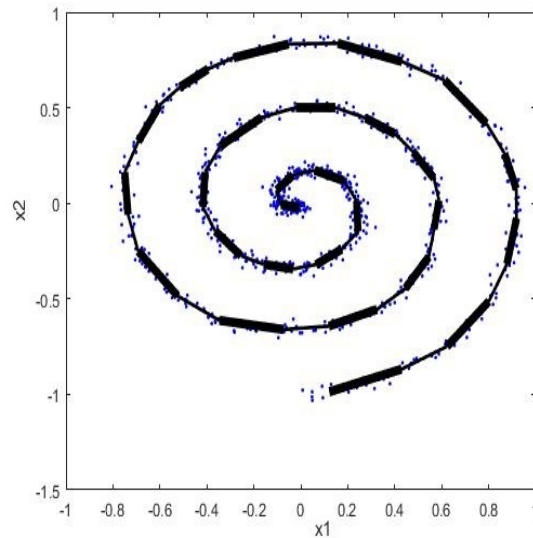
(b) $k = 7$; $f = 1,072855$

Fonte: Acervo do Autor

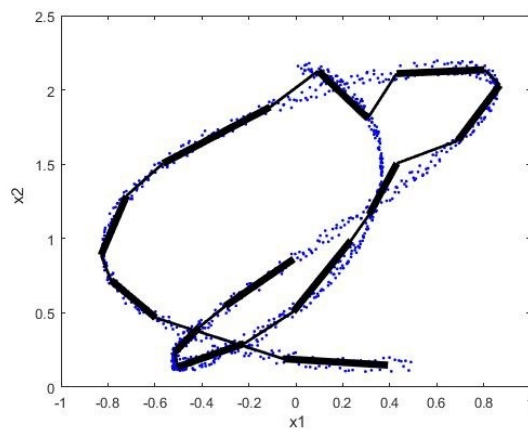
Em relação aos resultados obtidos com a otimização do comprimento dos segmentos, é importante destacar o fato que, para ambos os conjuntos de dados, o comprimento dos segmentos obtidos pela minimização da função custo é um pouco maior que σ , desvio padrão dos dados em relação ao segmento aos quais se projetam. Tal valor mostra-se divergente do valor $3\sigma/2$, recomendado por Verbeek, Vlassis e Kröse (2002), por ser o valor que retorna melhores resultados para os teste realizados.

Com a utilização da Equação (3.3), em que o aumento do número de segmentos é penalizada de modo mais brando, os problemas citados de representação são minimizados, como pode ser visto nas Figuras 3.7(a) e 3.7(b).

Figura 3.7 – Curvas geradas com o quantidade e comprimento dos segmentos como parâmetros otimizados, construídas a partir da minimização da Equação (3.3)



(a) $k = 27$; $f = 1,005536$



(b) $k = 12$; $f = 1,154600$

Fonte: Acervo do Autor

Para o conjunto de dados em espiral, a suavidade da curva gerada aumentou consideravelmente com o aumento do número de segmentos, gerado pela utilização desta função objetivo, como visto na Figura 3.7(a). Observa-se que, assim como ocorreu anteriormente, o valor de f encontrado é menor que $3/2$.

O comprimento do segmento tem conexão direta com a quantidade de segmentos da curva quando ambos são tomados como pares de parâmetros a serem otimizados, de acordo com a função-custo aqui utilizada. Dessa forma, a medida em que a quantidade de segmentos aumenta o comprimento ideal deles tende a diminuir, de modo a garantir a minimização da distância dos eventos à curva. Tal dualidade também garante a suavidade das curvas geradas, uma

vez que o aumento simultâneo da quantidade e do comprimento dos segmentos geraria curvas superdimensionadas, que, ao mesmo tempo não seguem o contorno dos dados e apresentam segmentos distantes dos eventos que neles projetam.

Por esse motivo, a manutenção do comprimento em $3\sigma/2$ mostra-se incoerente à medida que se aumenta o valor de k , uma vez que a suavidade da CP gerada é influenciada pela ponderação entre comprimento e quantidade de segmentos, sendo penalizada quando não há equilíbrio entre tais valores.

É importante notar que quanto maior o número de segmentos gerados maior tende a ser o comprimento total da curva. Por esse motivo a variação do primeiro termo da função objetivo utilizada impacta na variação do número de segmentos que compõem a CP.

Para o segundo conjunto de dados, visualizados na Figura 3.7(b), a utilização da função objetivo com $p = 0,95$ fez com que os dados fossem representados de maneira mais eficiente, quando comparado com os resultados anteriores, uma vez que ocorre um aumento considerável do número de segmentos. Por outro lado, o aumento de segmentos ainda não é grande o bastante para que a CP acompanhe os contornos do conjunto de dados e, por esse motivo, não é suficiente para que a representação possa, de fato, ser considerada satisfatória.

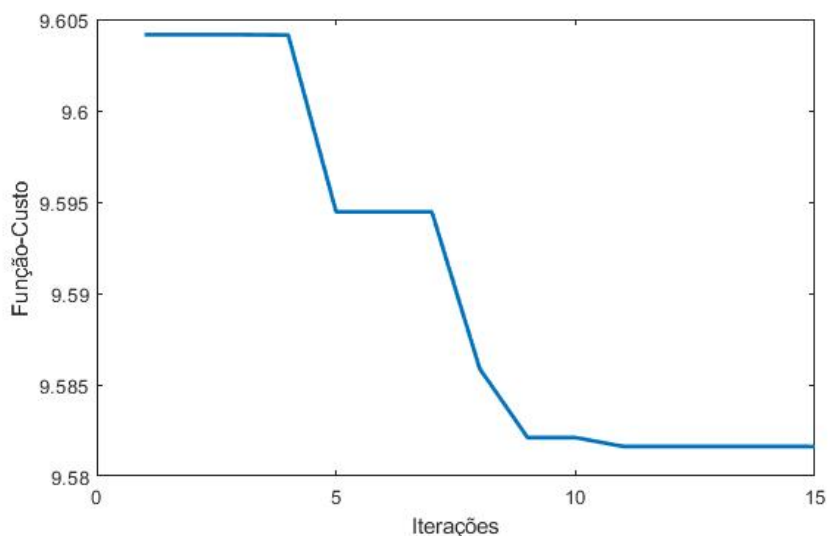
Seguindo a premissa que, para a função objetivo, o aumento da ponderação dada ao termo que trata da distância dos eventos à curva gera melhores resultados, realizou-se testes heurísticos de modo a definir uma equação cujos parâmetros que a minimizam sejam capazes de representar satisfatoriamente o contorno de ambos os conjuntos de dados. Tal resultado foi alcançado com a utilização da seguinte equação:

$$0,01 n \log l + 0,99 \sum_{i=1}^k \sum_{x \in V_i} d(s_i, \mathbf{x})^2 / (2\sigma^2). \quad (3.4)$$

De modo a possibilitar a observação da convergência do método proposto, é apresentado o gráfico da Figura 3.8.

Nesse gráfico se visualiza, para o contorno de dados em espiral, a variação dos valores assumidos pela função-custo, representada pela Equação 3.4, ao longo das iterações do algoritmo TLBO. Esse valor corresponde ao obtido pelo melhor indivíduo de cada iteração. É possível observar que no caso apresentado a função-custo atinge seu valor mínimo antes das 30 iterações utilizadas no modelo, o que mostra uma rápida convergência do algoritmo para encontrar uma solução satisfatória, para os casos apresentados.

Figura 3.8 – Convergência do algoritmo TLBO nas primeiras 15 iterações.



Fonte: Acervo do Autor

Os resultados encontrados com a utilização da Equação (3.4) podem ser visualizados nas Figuras 3.9(a) e 3.9(b), onde pode-se perceber que, por fim, resultados satisfatórios são alcançados para ambos os conjuntos de dados, em que os contornos são seguidos de maneira correta e as curvas geradas apresentam suavidade necessária em seus traços.

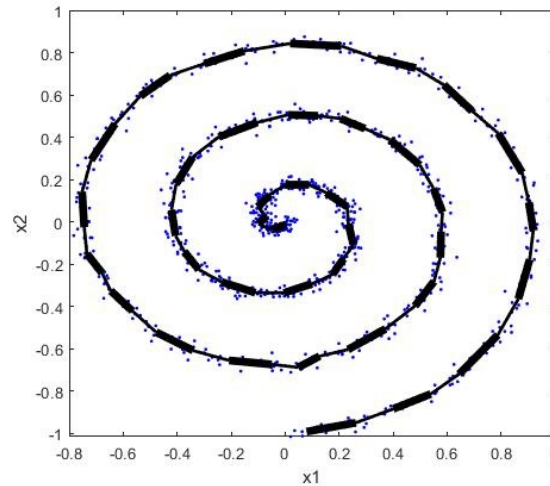
Ao se analisar a Equação (3.4) é possível notar a discrepância entre os pesos dados a cada um dos termos, o que pode levar a conclusões precipitadas a respeito da importância do primeiro deles, referente ao comprimento da curva.

A eliminação desse termo, por outro lado, pode acarretar no sobre-ajuste, *overfitting*, do modelo que representa o conjunto de dados. Isso ocorre uma vez que a não penalização do aumento excessivo do número de segmentos pode gerar uma CP em que pequeno grupo de eventos gere um segmento. Nesse caso, a distância de cada ponto à curva seria nulo fazendo com que o mínimo global da função objetivo, agora representada somente pelo segundo termo, também o fosse, mesmo com o sobredimensionamento da curva gerada.

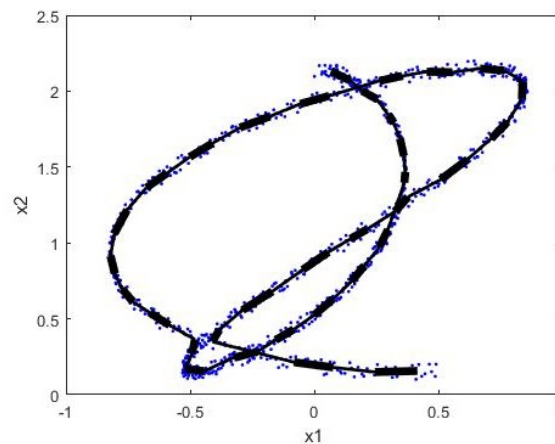
A manutenção de ambos os termos da equação se mostra necessária por esse motivo, mesmo que a ponderação de ambos seja desigual e, mesmo com diferentes ponderações desta, caso se mostre necessário em alguma aplicação específica.

Ao se observar, de maneira mais detalhada a Figura 3.9(b), verifica-se a ligação equivocada de alguns segmentos nas proximidades dos pontos onde há cruzamento dos dados. Esse fato ocorre devido ao coeficiente de suavidade, λ , ser mantido fixo. Na Seção 3.5 é apresentada

Figura 3.9 – Curvas geradas a partir da minimização da Equação (3.4).



(a) $k = 38$; $f = 0,946718$



(b) $k = 33$; $f = 0,846994$

Fonte: Acervo do Autor

uma breve discussão a respeito desse parâmetro e o motivo de mantê-lo fixo na inicialização do algoritmo k-segmentos.

3.5 Uma discussão sobre o coeficiente de suavidade λ

Segundo Wang e Lee (2006), a correta ligação entre os segmentos que compõem a curva é determinada pelo parâmetro λ , e o efeito de sua variação é visível, principalmente em conjuntos de dados onde há interseções, como o dos dados que apresentam contorno em forma de rabisco. Nos testes apresentados nas seções anteriores o valor de λ foi fixado em 1 para todos os conjuntos de dados.

É de grande importância observar que a variação de λ não influencia de maneira significativa os valores numéricos assumidos pela função objetivo. Assim, para que o parâmetro λ seja considerado uma variável a ser escolhida automaticamente pelo algoritmo, seria necessário redefinir a função-custo do algoritmo de otimização. Segundo Verbeek, Vlassis e Kröse (2001), para o algoritmo k-segmentos suave, valores de λ dentre 1/2 e 2 apresentam bons resultados. Para o algoritmo k-segmentos não suave, observou-se o mesmo durante a realização de testes experimentais. Ademais, ao se observar a Figura 3.9(b), é possível perceber que a ligação equivocada encontrada não traz prejuízos à representação do contorno de dados.

Uma vez que tal parâmetro não influencia de maneira significativa a representação do contorno de dados, observa-se que não há necessidade de realizar sua otimização, pois isso geraria custo computacional desnecessário, sobretudo para as aplicações propostas nos Capítulos 5 e 6, onde o método proposto é aplicado à classificação e clusterização de dados, respectivamente. Assim, o método de otimização proposto, juntamente com a função-custo aqui utilizada mostram-se soluções interessantes para a definição automatizada dos parâmetros iniciais do algoritmo k-segmentos, sem que haja necessidade de realização de cálculos computacionais complexos, como feitos por Wang e Lee (2006) e comentados por Lee e Wang (2006). No Capítulo 4 outras meta-heurísticas são aplicadas ao método proposto, de modo a observar sua eficácia na definição dos parâmetros iniciais do algoritmo k-segmentos.

4 OTIMIZAÇÃO DE PARÂMETROS DO ALGORITMO K-SEGMENTOS COM UTILIZAÇÃO DE ITLBO, GWO, CGWO E NSGA-II

4.1 Utilização de meta-heurísticas mono-objetivo

Uma vez definida a função-custo e observada a viabilidade da definição dos parâmetros iniciais do algoritmo k-segmentos com a utilização de algoritmos evolucionários, deve-se observar a robustez do método proposto com a utilização de outras meta-heurísticas, tais como ITLBO, desenvolvido a partir do TLBO, que se utiliza de mecanismos de perturbação caótica, além de operações de mutação e cruzamento para refinar sua capacidade de otimização. Utiliza-se, ainda, o algoritmo GWO, que se inspira no ritual de caça dos lobos cinzentos em sua busca por resultados ótimos e o CGWO, desenvolvido a partir do GWO por meio da utilização de mapas caóticos a fim de aumentar a capacidade de convergência para o mínimo global.

Neste capítulo os resultados obtidos na utilização dessas meta-heurísticas são comparados. Para as quatro meta-heurísticas, o tamanho da população foi definido como 40. A quantidade de iterações foi definida de modo a garantir que a quantidade de chamadas da função de avaliação fosse igual para todos os casos. Desse modo para a meta-heurística TLBO foram definidas 30 iterações, para a ITLBO, 12 iterações, enquanto para GWO e CGWO, 50 iterações cada.

Com a definição desses parâmetros de entrada, as meta-heurísticas, apresentaram aproximadamente o mesmo número de chamadas da função-custo, cerca de 1300. É importante observar que para que esse número seja igual, a quantidade de iterações de cada meta-heurística deve ser diferente. Isso ocorre devido aos passos de cada algoritmo de otimização, destacando o algoritmo ITLBO que apresenta alta quantidade de chamadas da função custo por iteração, o que se deve às fases de *feedback*, mutação, cruzamento e perturbação caótica, novas em relação ao TLBO tradicional.

A comparação e análise dos resultados obtidas por outros algoritmos de otimização bioinspirados mostra-se importante, uma vez que tais métodos apresentam mecanismos mais modernos de busca por soluções ótimas. Outro fato a se destacar é que tal comparação é importante na definição da meta-heurística que retorna melhores resultados de otimização para o método proposto por esse trabalho, uma vez que, segundo o *no free lunch theorem* (NFL) para problemas de otimização, apresentado por Wolpert e Macready (1997), não há meta-heurística mais adequada para resolver todos os problemas de otimização.

A seguir são apresentados os resultados obtidos pelas diferentes meta-heurísticas. As bases de dados utilizadas são as mesmas do Capítulo 3 devido à sua ampla utilização em trabalhos de objetivos semelhantes. A comparação entre os algoritmos também é feita pela avaliação do valor médio assumido pela função-custo para os algoritmos de otimização utilizados. Dessa forma, a avaliação e determinação de qual meta-heurística deve ser utilizada é feita de maneira mais adequada.

A avaliação dos resultados é apresentada nas Subseções 4.1.1 e 4.1.2, apresentando a análise das diferentes meta-heurísticas, para os contornos em forma de espiral e rabisco, respectivamente.

4.1.1 Análise das meta-heurísticas mono-objetivo para contorno em forma de espiral

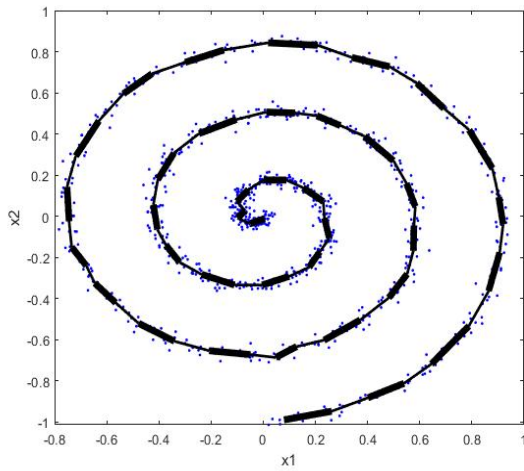
Para a realização das comparações entre as meta-heurísticas, foram realizados 10 testes com cada. As CP geradas em um desses testes, com os parâmetros iniciais definidos pelas meta-heurísticas TLBO, ITLBO, GWO e CGWO são apresentadas nas Figuras 4.1(a), 4.1(b), 4.1(c) e 4.1(d), respectivamente.

Como pode ser observado na imagem todas as curvas geradas representam de maneira satisfatória o contorno de dados utilizado. Observa-se que a quantidade de segmentos determinada pelos algoritmos são iguais a 38, com exceção dos resultados obtidos pelo CGWO, em que o número de segmentos é igual a 39. O número maior de segmentos faz com que a CP gerada seja levemente menos suave, apresentando serrilhamento pouco maior que os observados nas outras curvas.

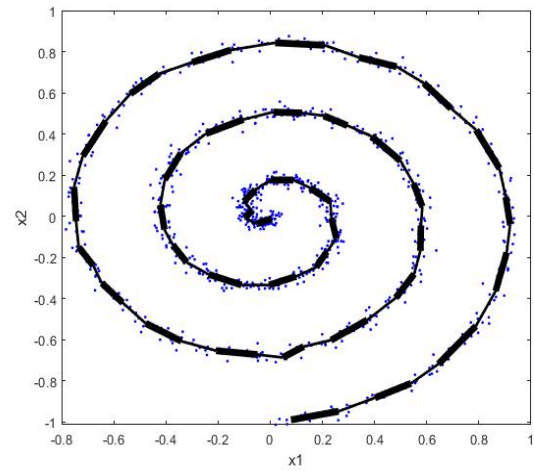
De modo geral, os resultados obtidos são satisfatórios, destacando o fato que para a utilização do método proposto como algoritmo de classificação, tais como os aqueles apresentados nos Capítulos 5 e 6, essa leve irregularidade na geração da curva é irrelevante, uma vez que, para aplicações desse tipo a visualização não é necessária e mesmo impossível, já que a maioria das bases de dados apresenta mais que duas dimensões.

O comprimento dos segmentos encontrados também são semelhantes em todos os casos, com pequenas variações, imperceptíveis à observação. Isso faz com que as CP geradas sejam, de certa forma, visualmente iguais. A similaridade dos resultados visuais obtidos faz com que se observe que qualquer uma das meta-heurísticas analisadas atinja a otimização almejada pelo método proposto.

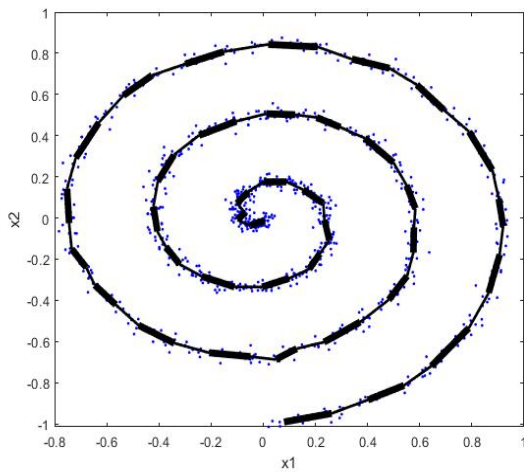
Figura 4.1 – Curvas geradas a partir de parâmetros otimizados com a utilização das diferentes meta-heurísticas, para contorno de dados em espiral.



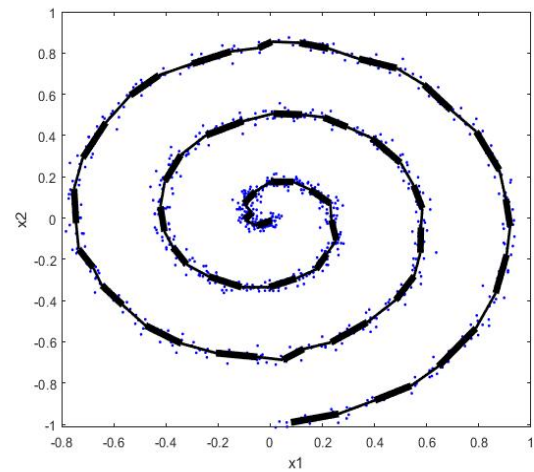
(a) TLBO: $k = 38$; $f = 0,946718$



(b) ITLBO: $k = 38$; $f = 0,946721$



(c) GWO: $k = 38$; $f = 0,946561$



(d) CGWO: $k = 39$; $f = 0,946651$

Fonte: Acervo do Autor

Na Tabela 4.1, é possível observar os valores assumidos pela função-custo para cada um dos métodos de otimização obtidos.

Tabela 4.1 – Valores da função-custo para dados em espiral

Meta-heurística	Função-custo
TLBO	$9,6036 \pm 0,0001$
ITLBO	$9,6036 \pm 0,0000$
GWO	$9,6037 \pm 0,0001$
CGWO	$9,6043 \pm 0,0005$

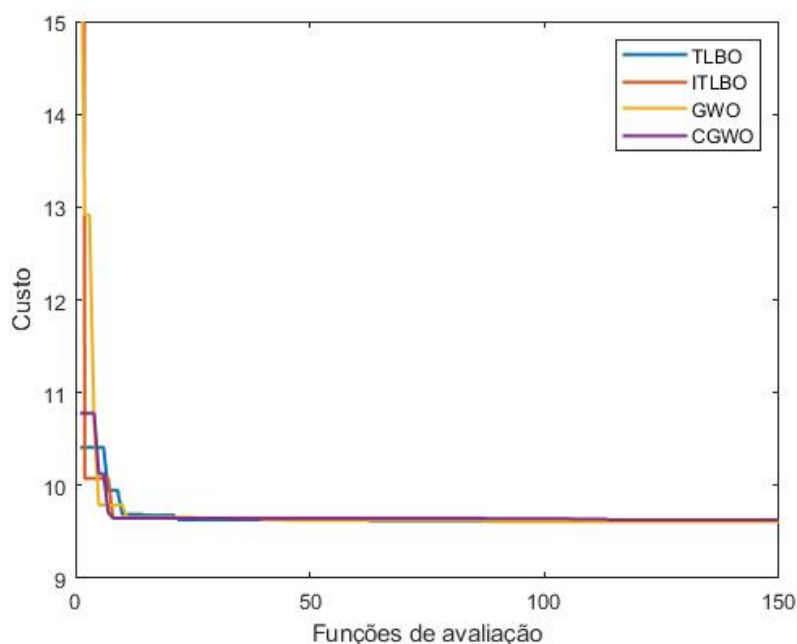
Uma vez que a quantidade e comprimento dos segmentos determinada pelas meta-heurísticas é similar, os valores assumidos pela função-custo para tais parâmetros são simi-

lares, como pode ser observado Tabela 4.1. Observa-se que a otimização realizada pelo CGWO apresenta uma média maior, além de maior dispersão em seus resultados.

As diferenças observadas, porém, são mínimas, de modo que não é possível descartar, ou tampouco eleger, a utilização de nenhuma das meta-heurísticas testadas devido aos valores alcançados pela função-custo após a otimização.

Dessa maneira, resta avaliar a convergência das meta-heurísticas, o que pode ser visto na Figura 4.2, onde se pode verificar a otimização da função-custo.

Figura 4.2 – Convergência das meta-heurísticas de acordo com a quantidade de chamadas das funções de avaliação para o contorno em espiral.



Fonte: Acervo do Autor

Como pode ser observado, a convergência das meta-heurísticas observadas é semelhante, uma vez que quando atingem pouco mais que 100 chamadas da função de avaliação, todas apresentam valores muito próximos umas das outras e por sua vez, próximos dos valores ótimos encontrados ao final das iterações. É importante destacar que o custo computacional do método proposto é atrelado à quantidade de chamadas da função-custo, já que o seu cálculo é realizado juntamente com a construção de CP, com os parâmetros da população, pelo algoritmo k-segmentos.

Dessa maneira, observa-se que, para o contorno de dados aqui utilizado, para todas as meta-heurísticas, o tamanho da população e quantidade de iterações pode ser diminuída, de modo a garantir maior rapidez na otimização dos parâmetros iniciais do algoritmo k-segmentos.

Uma vez que os resultados obtidos por todas as técnicas verificadas, são semelhantes, como apresentado para os dados aqui utilizados, a escolha de qualquer uma das meta-heurísticas analisadas se mostraria adequada. Na Seção 4.1.2 é feita uma análise semelhante para o contorno de dados em forma de rabisco.

4.1.2 Análise das meta-heurísticas mono-objetivo para contorno em forma de rabisco

De maneira análoga à Subseção 4.1.1, é feita a análise, para contorno de dados em forma de rabisco, das CP geradas com os parâmetros iniciais definidos pelas meta-heurísticas TLBO, ITLBO, GWO e CGWO, apresentadas nas Figuras 4.1(a), 4.1(b), 4.1(c) e 4.1(d), respectivamente.

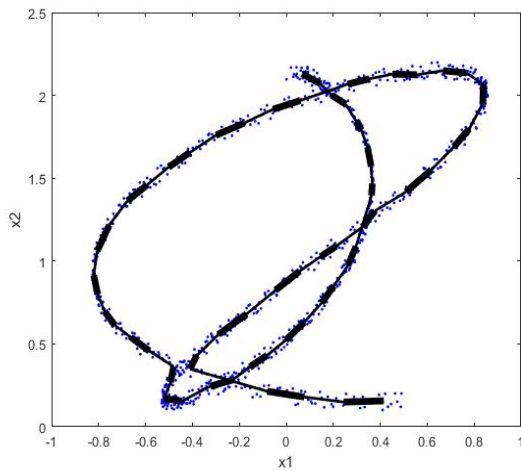
Assim como observado para o contorno de dados em espiral, todas as meta-heurísticas são capazes de representar os dados de forma satisfatória, mas, diferentemente dos resultados da Subseção 4.1.2, a quantidade e comprimento de segmentos definidos durante a otimização apresentam diferenças significativas em alguns testes, como o observado na Figura 4.3.

Se por um lado as meta-heurísticas TLBO e ITLBO definem quantidade de segmentos 32 e 33, respectivamente, com comprimento de cerca de 0,84 para ambos os casos, as meta-heurísticas GWO e CGWO apresentam quantidade de segmentos igual a 40, com comprimento igual a aproximadamente 0,53. Observando-se as Figuras 4.3(c) e 4.3(d), é possível perceber que, para esses dois últimos modelos de otimização, a ligação entre os segmentos ocorre de maneira correta em todos os pontos de intersecção do conjunto de dados.

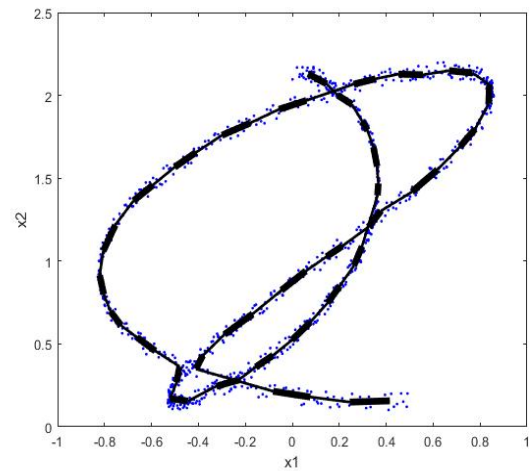
A diferença entre os valores definidos durante os testes aqui realizados demonstra a dinâmica existente entre quantidade e comprimento dos segmentos, de modo que o aumento da quantidade de segmentos exige a diminuição gradativa do comprimento dos mesmos, para que a representação seja feita de maneira adequada. Tal fato mostra a importância de se otimizar ambos os parâmetros concomitantemente, o que faz com que a fixação do valor do comprimento, como proposta por Verbeek, Vlassis e Kröse (2002) e mantida durante a otimização proposta por Wang e Lee (2006), mostre-se inadequada.

A diminuição do tamanho dos segmentos determinados pelo GWO e CGWO faz com que as linhas que ligam os segmentos sejam maiores, como pode ser visto na Figura 4.3. De modo geral, isso não causa prejuízos à CP gerada e demonstra a importância que comprimento e quantidade de segmentos sejam otimizados paralelamente, como citado anteriormente.

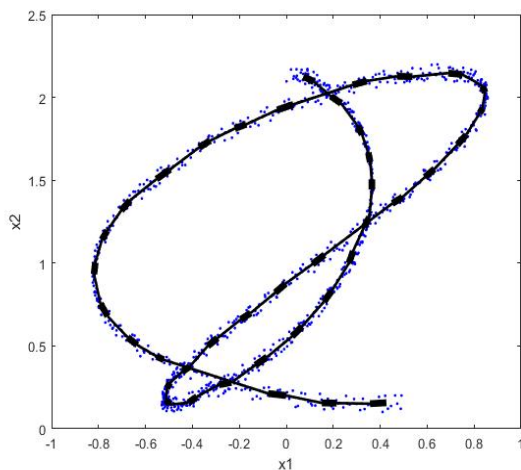
Figura 4.3 – Curvas geradas a partir de parâmetros otimizados com a utilização das diferentes meta-heurísticas, para contorno de dados em rabisco.



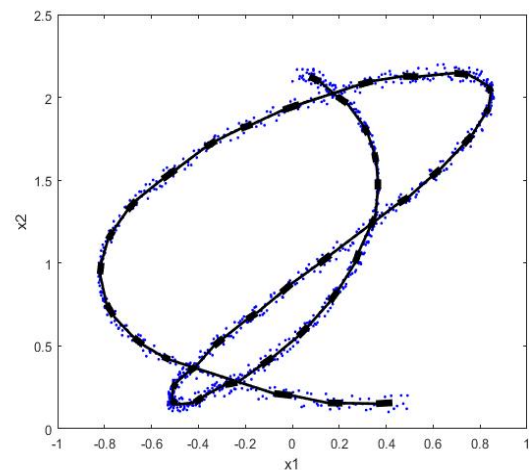
(a) TLBO: $k = 33$; $f = 0,846994$



(b) ITLBO: $k = 32$; $f = 0,846261$



(c) GWO: $k = 40$; $f = 0,534260$



(d) CGWO: $k = 40$; $f = 0,534374$

Fonte: Acervo do Autor

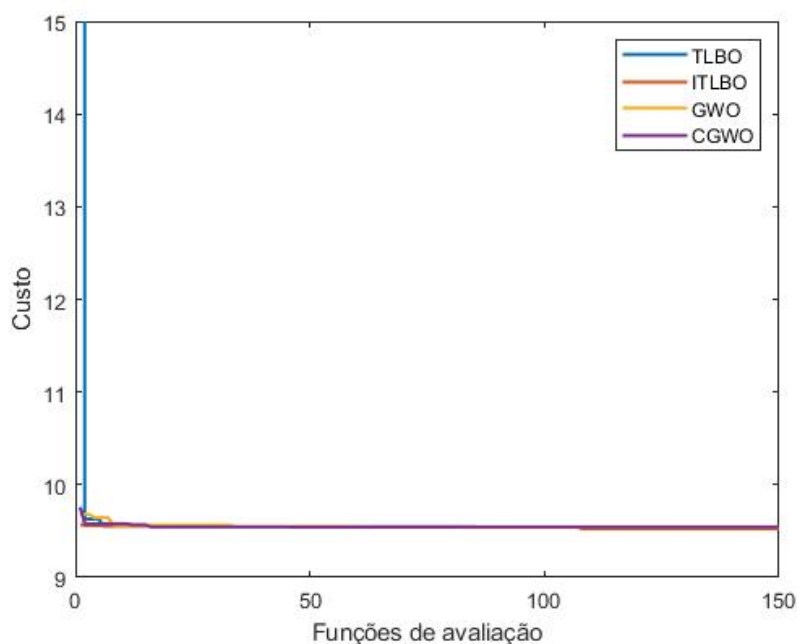
Por esse motivo, não é possível eleger visualmente dentre as meta-heurísticas analisadas, a que gera CP mais adequadas, uma vez que as curvas determinadas pelos parâmetros otimizados por todas elas mostram-se capazes de representar os dados de maneira eficiente. Na Tabela 4.2, é possível observar a média e desvio padrão dos valores assumidos pela função-custo para cada um dos métodos de otimização obtidos, durante os testes realizados.

Como pode ser observado na Tabela 4.2, a média dos valores da função-custo apresentados por GWO e CGWO são maiores, mas estatisticamente os resultados alcançados por todas as meta-heurísticas são similares. Na Figura 4.4 é possível observar a convergência dos métodos analisados em um dos testes realizados.

Tabela 4.2 – Valores da função-custo para dados em rabisco

Meta-heurística	Função-custo
TLBO	$9,5067 \pm 0,0066$
ITLBO	$9,5068 \pm 0,0083$
GWO	$9,5148 \pm 0,0092$
CGWO	$9,5103 \pm 0,0115$

Figura 4.4 – Convergência das meta-heurísticas de acordo com a quantidade de chamadas das funções de avaliação para o contorno em rabisco.



Fonte: Acervo do Autor

Assim como observado na Seção 4.1.1 a convergência dos métodos analisados é rápida e, com pouco mais de 100 chamadas da função de avaliação os valores assumidos pela função-custo mostram-se próximos, ou mesmo iguais, aos obtidos no fim das iterações. Isso demonstra a alta capacidade do método proposto na otimização da quantidade e comprimento dos segmentos do algoritmo k-segmentos, com a utilização das meta-heurísticas utilizadas nesta seção.

De acordo com os resultados apresentados, mostra-se adequada a manutenção do método proposto inicialmente, com a utilização do TLBO, sendo possível, dependendo da aplicação, modelar o algoritmo de otimização com quantidade de membros da população e número de iterações menores, de modo a diminuir o custo computacional da otimização dos parâmetros do algoritmo k-segmentos.

Na Seção 4.2 é apresentada a discussão a respeito da utilização de uma meta-heurística multiobjetivo com a qual é possível estudar de maneira mais aprofundada os termos da função-

custo utilizada na otimização dos parâmetros do algoritmo k-segmentos, proposta por esse trabalho.

4.2 Utilização de meta-heurísticas multiobjetivo

Uma vez determinada a meta-heurística a ser utilizada na otimização dos parâmetros é necessário verificar se os valores encontrados por ela representam uma das soluções ótimas na minimização simultânea de ambos os termos da equação utilizada. Para isso é utilizada a meta-heurística multiobjetivo NSGA-II, devido à sua consolidação na literatura e às suas características, apresentadas na Seção 2.7, de modo a encontrar a fronteira Pareto para o problema em questão. Ao minimizar duas funções conflitantes paralelamente, a meta-heurística NSGA-II apresenta o conjunto das soluções ótimas na escolha dos parâmetros do algoritmo k-segmentos.

As funções-custo utilizadas neste processo são apresentadas a seguir:

$$f1 = n \log l \quad (4.1)$$

$$f2 = \sum_{i=1}^k \sum_{x \in V_i} d(s_i, \mathbf{x})^2 / (2\sigma^2) \quad (4.2)$$

Como pode ser observado, cada uma delas é um termo da Equação 3.1, sem as parcelas relacionadas à ponderação dos termos. Dessa maneira, quanto à construção da CP com a utilização de parâmetros que minimizam ambas, tem-se que a Equação 4.1 é diretamente proporcional ao comprimento da curva gerada, enquanto a Equação 4.2 se relaciona com a distância dos eventos à curva.

É importante observar que a minimização do comprimento da curva causa o aumento da distância dos eventos à ela. Por outro lado, o aumento da quantidade de segmentos gerada pela minimização dos eventos à curva faz com que seu comprimento aumente. A existência deste conflito, entre os termos da função-custo, faz com que a análise do método proposto com a utilização da meta-heurística NSGA-II se torne necessária, sobretudo para avaliar as soluções ótimas, uma vez que ponderações diferentes daquelas discutidas no Capítulo 3 podem ser utilizadas na obtenção de CP, a depender do objetivo de sua construção. Por esse motivo, são utilizadas as Equações 4.1 e 4.2 como as funções-custo a serem minimizadas pelo NSGA-II.

Definiu-se, para ambas as bases de dados, espiral e rabisco, o tamanho da população do algoritmo de otimização igual a 150, de modo a garantir diversidade de indivíduos e boa visualização da fronteira Pareto. A otimização foi realizada por 100 iterações, com o intuito

de garantir a obtenção de resultados ótimos. A busca por parâmetros ideais se concentrou nos intervalos (0,5; 2) para comprimento dos segmentos e (1; 40) para quantidade de segmentos. A escolha desses intervalos ocorreu pela observação das curvas geradas em resultados anteriores. É importante destacar que sua utilização advém do conhecimento especialista das bases de dados utilizadas. Os intervalos onde se encontram resultados ótimos variam para cada base de dados de acordo com a quantidade de eventos e de sua distribuição. Caso não se tenha conhecimento a respeito da base, pode-se utilizar intervalos maiores, de modo que o algoritmo possa encontrar automaticamente os parâmetros ótimos, sem prejuízos à qualidade da curva construída, apesar de possível aumento do custo computacional do método.

Assim como na Seção 4.1 a discussão dos resultados obtidos com a utilização da otimização multiobjetivo será dividida em duas subseções. Na Subseção 4.2.1 são apresentados os resultados para a base de dados em espiral. Na Subseção 4.2.2 são apresentados os resultados para a base de dados em forma de rabisco e, ao final da mesma, são apresentadas as conclusões gerais a respeito da utilização da meta-heurística NSGA-II.

4.2.1 Análise da meta-heurística multiobjetivo para contorno em forma de espiral

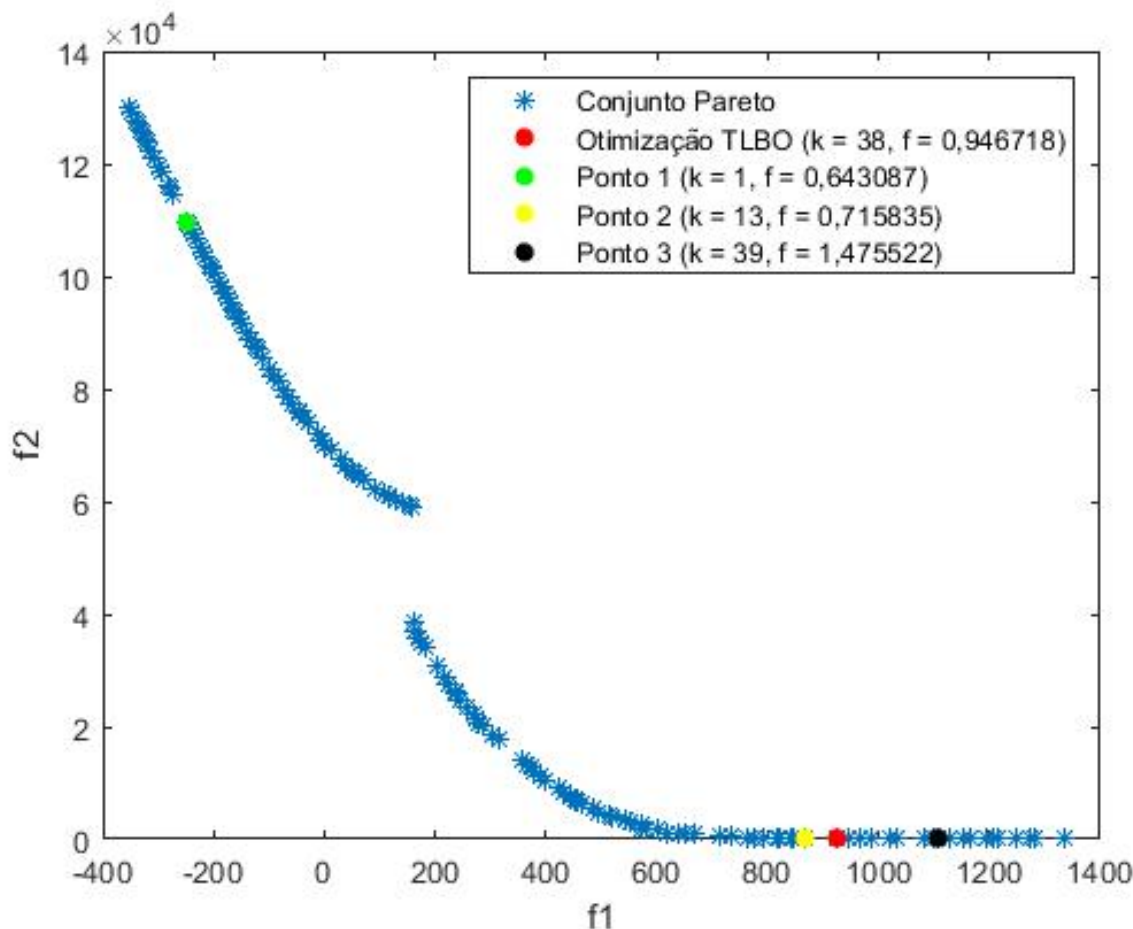
Com a utilização das Equações 4.1 e 4.2, utilizou-se a meta-heurística NSGA-II para encontrar o conjunto Pareto dos pares de parâmetro (k, f) . Os resultados obtidos para os dados em forma de espiral podem ser observados na Figura 4.5.

Na Figura 4.5 é possível observar os valores assumidos pelas funções-custo para diversos pares de parâmetros utilizados. Observa-se que a função objetivo relativa ao logaritmo do comprimento dos segmentos assume valores negativos, resultado esperado para comprimentos pequenos. Destaca-se, também, a possibilidade de se comprovar visualmente que quanto maior o comprimento assumido pela curva, menor o somatório da distância dos eventos aos segmentos.

De modo a analisar a otimização realizada pelo algoritmo mono-objetivo apresentado na Seção 4.1, foram destacados no gráfico 4 pontos pertencentes ao conjunto Pareto, escolhidos de modo a observar a alta variedade de soluções, sendo um deles, correspondente ao par obtido pela otimização mono-objetivo da Seção 4.1.1.

É possível observar que os parâmetros determinados pelo TLBO se encontram muito próximos à fronteira de Pareto, sendo possível concluir a boa capacidade de otimização dos parâmetros pelo método mono-objetivo proposto. Para uma análise mais detalhada é necessário

Figura 4.5 – Fronteira de Pareto para dados em espiral.



Fonte: Acervo do Autor

observar os valores assumidos pelas funções-custo para os 4 pontos destacados na Figura 4.5. Estes valores podem ser vistos na Tabela 4.3.

Tabela 4.3 – Valores das funções-custo de diferentes pontos para dados em espiral

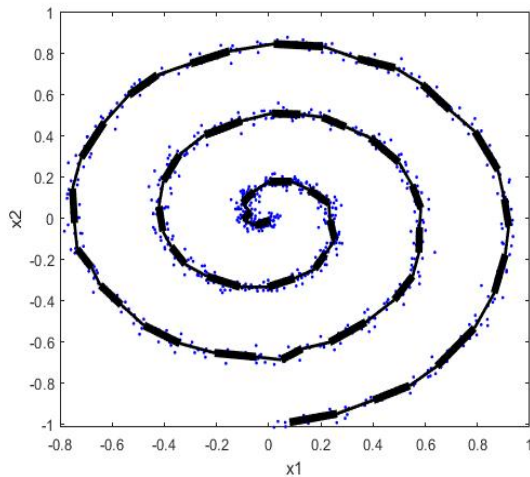
	Função-objetivo 1	Função-objetivo 2
Otimização TLBO	923,219	0,375
Ponto 1	-248,145	$1,098 \times 10^5$
Ponto 2	867,255	99,218
Ponto 3	$1,109 \times 10^3$	0,188

Na Tabela 4.3 é possível observar que para a otimização realizada pelo TLBO, a função-custo 2 apresenta valor baixo, como esperado, devido à ponderação de 99% dada à minimização do somatório das distâncias dos eventos à curva.

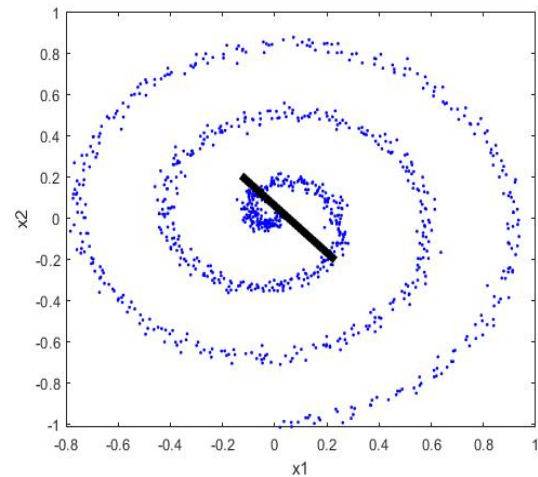
Em comparação aos pontos do conjunto Pareto escolhidos para análise, é possível verificar que os valores da otimização TLBO se encontram entre os Pontos 2 e 3. É possível observar a proximidade dos valores da otimização pelo método proposto e os que são apresentados pelo

Ponto 3 . Isso faz com que seja necessária a análise e comparação das CP construídas com os parâmetros para os quais se obtém tais valores para as funções-objetivo. Essas curvas podem ser visualizadas na Figura 4.6.

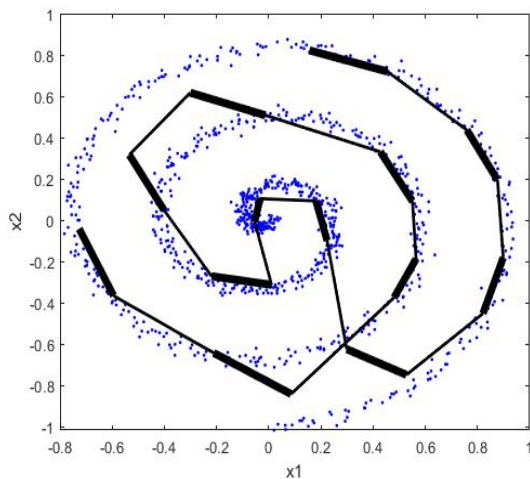
Figura 4.6 – Curvas geradas a partir de parâmetros pertencentes ao conjunto Pareto e da otimização TLBO para dados em espiral.



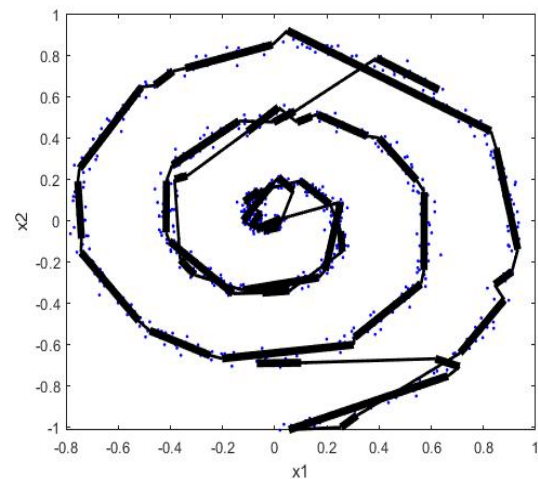
(a) Otimização TLBO: $k = 38$; $f = 0,946718$



(b) Ponto 1: $k = 1$; $f = 0,643087$



(c) Ponto 2: $k = 13$; $f = 0,715835$



(d) Ponto 3: $k = 39$; $f = 1,475522$

Fonte: Acervo do Autor

A análise das curvas observadas na Figura 4.6 é feita separadamente para cada CP. A curva da Figura 4.6(a) já foi analisada no Capítulo 3 e na Subseção 4.1.1 deste capítulo.

Na Figura 4.6(b) é apresentada a CP com os parâmetros do Ponto 1. Para esse ponto a Função-custo 2 apresenta valor elevado. Isso ocorre porque essa solução foi obtida tendo como foco principal de otimização, a minimização da Função-custo 1, para a qual se tem valores negativos, uma vez que o comprimento do único segmento da curva é pequeno. Visualmente

observa-se sua incapacidade de representar o conjunto de dados, o que mostra a importância de se minimizar a distância dos eventos aos segmentos para que se obtenha uma representação adequada.

Mais próxima da otimização realizada pelo TLBO, a curva da Figura 4.6(c) acompanha de maneira incompleta o contorno dos dados, uma vez que a quantidade de segmentos é insuficiente para tal tarefa. A ligação entre os segmentos também é feita de tal modo que a curva intercepta a si mesma, o que é inadequado para o contorno de dados em espiral.

Já na Figura 4.6(d) é possível observar o aumento da quantidade e sobredimensionamento do tamanho dos segmentos o que gera a conexão equivocada entre eles. Isso ocorre de modo que a distância dos eventos à curva sejam próximos de 0. Tal fato demonstra a importância da utilização dos dois termos da função-custo, para otimização mono-objetivo, assim como é feito na otimização multiobjetivo. Dessa maneira, mesmo que a ponderação encontrada para a otimização da curva seja desigual, a não utilização de ambos os termos pode gerar o *overfitting* dos dados, como discutido no Capítulo 3, durante a busca por uma função-custo adequada ao modelo proposto.

Por meio da observação das curvas é possível perceber que a escolha da ponderação entre as Equações 4.1 e 4.2 afeta significativamente a CP gerada. De modo a garantir uma análise mais completa é necessário observar os resultados para o contorno de dados em forma de rabisco, apresentado na Subseção 4.2.2.

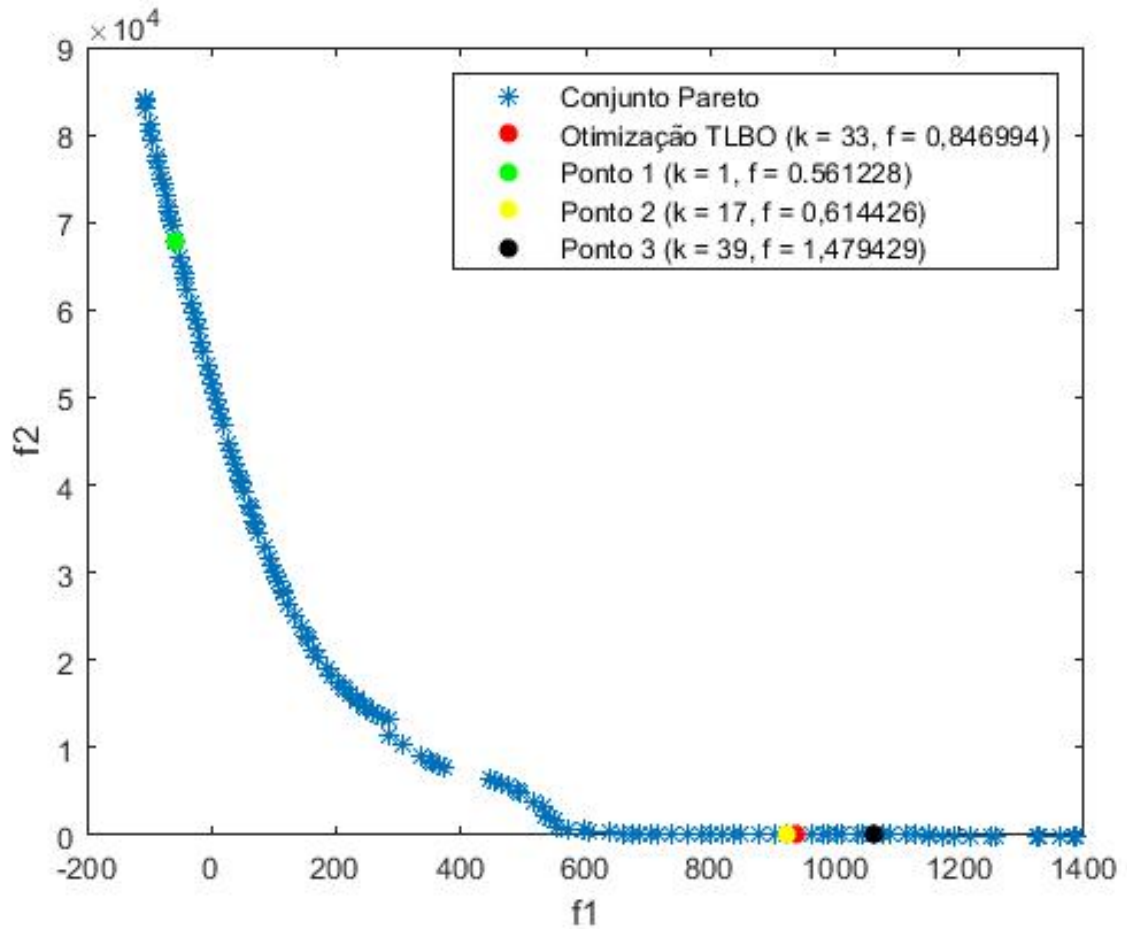
4.2.2 Análise da meta-heurística multiobjetivo para contorno em forma de rabisco

O conjunto Pareto da otimização multiobjetivo para os dados em forma de rabisco podem ser observados na Figura 4.7.

A análise realizada para o contorno de dados em forma de rabisco é feita de maneira análoga à realizada para o contorno de dados em espiral, realizada na Subseção 4.2.1. A escolha dos pontos vistos na Figura 4.7 foi feita utilizando os mesmos critérios utilizados previamente. Os valores assumidos pelas funções-custo para os 4 pontos nela destacados podem ser vistos na Tabela 4.4.

Pela análise da Figura 4.7 e da Tabela 4.4 é possível observar que os parâmetros encontrados pela otimização do TLBO se encontra próximo a fronteira de Pareto, mostrando que o par (k, f) por ele determinado possa ser considerado ótimo.

Figura 4.7 – Fronteira de Pareto para dados em rabisco.



Fonte: Acervo do Autor

Tabela 4.4 – Valores das funções-custo de diferentes pontos para dados em rabisco

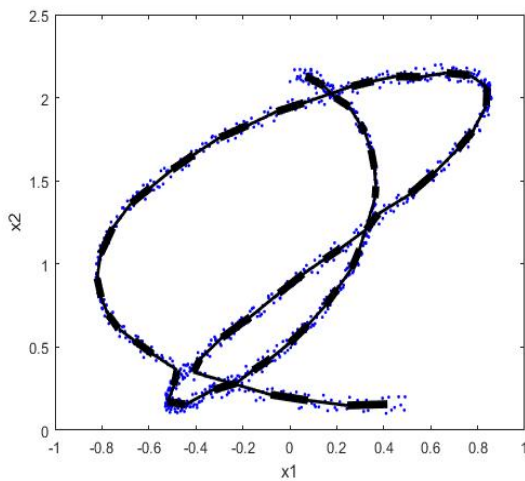
	Função-objetivo 1	Função-objetivo 2
Otimização TLBO	937,070	0,135
Ponto 1	-57,480	$6,795 \times 10^4$
Ponto 2	924,094	1,049
Ponto 3	$1,064 \times 10^3$	0,038

As CP correspondentes a cada um dos pontos podem ser visualizadas na Figura 4.8 e são analisadas separadamente.

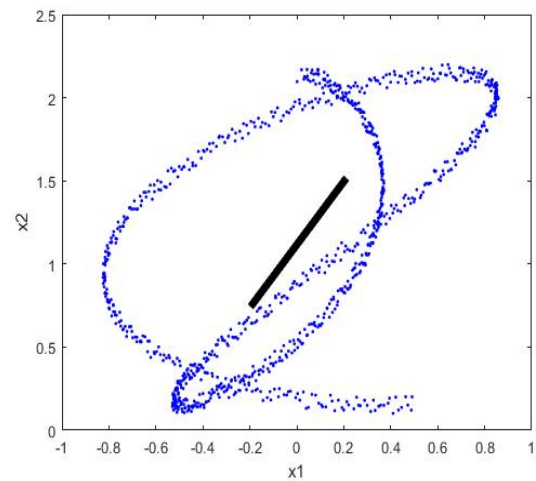
A curva observada na Figura 4.8(a) já foi analisada no Capítulo 3 e na Subseção 4.1.2 deste capítulo.

A Figura 4.8(b) apresenta a curva com os parâmetros que têm valores da função-objetivo representados pelo Ponto 1 da Figura 4.7. Para esse ponto a distância dos eventos ao único segmento que compõem a curva é alta, e como pode ser observado, esse segmento não intercepta nenhum dos eventos do conjunto de dados. Por outro lado a Função-custo 1 apresenta baixo

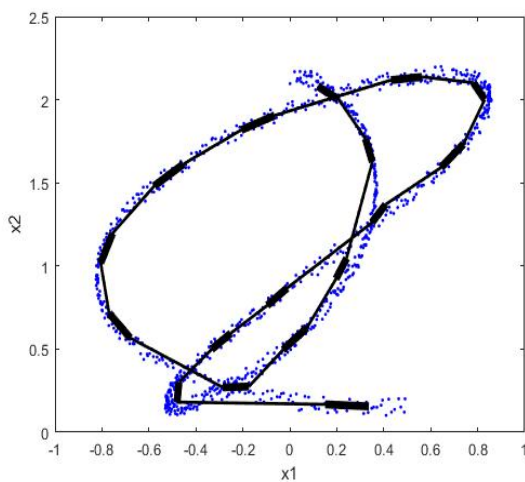
Figura 4.8 – Curvas geradas a partir de parâmetros pertencentes ao conjunto Pareto e da otimização TLBO para dados em rabisco.



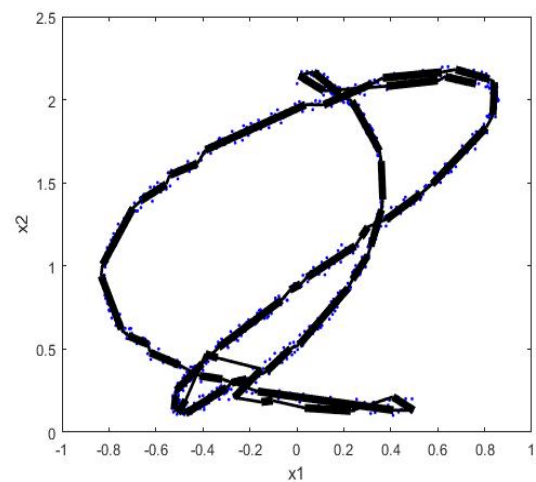
(a) Otimização TLBO: $k = 33$; $f = 0,846994$



(b) Ponto 1: $k = 1$; $f = 0.561228$



(c) Ponto 2: $k = 17$; $f = 0,614426$



(d) Ponto 3: $k = 39$; $f = 1,479429$

Fonte: Acervo do Autor

valor, já que o segmento da curva possui baixo comprimento. A análise da curva gerada reforça a conclusão da necessidade de se minimizar a distância dos eventos aos segmentos para que se obtenha uma representação adequada.

A curva correspondente ao Ponto 2 é apresentada na Figura 4.8(c) e nela se observa uma representação mais adequada do contorno de dados. Apesar da quantidade e comprimento dos segmentos serem diferentes dos obtidos pela otimização do TLBO é possível observar a proximidade de ambos os pontos no conjunto Pareto, o que demonstra que quanto maior a proximidade da ponderação proposta no Capítulo 3 melhores resultados visuais se obtém. É importante destacar que, implicitamente, os pontos pertencentes à fronteira de Pareto represen-

tam diferentes ponderações entre as funções custo, de modo que quanto mais à esquerda do gráfico maior peso se dá a diminuição da distância dos eventos à CP e quanto mais à direita maior peso se dá à minimização do comprimento da curva.

Já na Figura 4.8(d) é possível observar, assim como para o contorno de dados em espiral, o sobredimensionamento da curva, causado pelo aumento da quantidade e comprimento dos segmentos que a compõem, o que reforça a importância de se minimizar, mesmo que com menor afinco, o comprimento total da CP.

Com os resultados apresentados neste Capítulo é possível chegar a conclusão que o método de determinação automática dos parâmetros do algoritmo k-segmentos com utilização da meta-heurística TLBO é adequado à proposta apresentada. É importante ressaltar que a ponderação utilizada nesse método apresentou bons resultados para as finalidades utilizadas. Tal fato, porém, não indica que tal ponderação é a melhor para outras aplicações, ou mesmo diferentes contornos de dados.

Pelas figuras apresentadas nesta seção, sobretudo a Figura 4.7 é possível observar o quão sensível é o método mono-objetivo em relação a ponderação dos termos da função-custo. Nessa figura é possível verificar a proximidade do Ponto 2 à Otimização TLBO, na fronteira de Pareto. O Ponto 2, apresenta ponderação de 94% dada à minimização do somatório das distâncias dos eventos à curva e de 6% dada ao comprimento total da curva, enquanto a Otimização TLBO, destacada na figura, apresenta ponderações de 99% e 1% desses termos. Isso demonstra que a variação da ponderação entre os termos pode impactar no resultado das diversas aplicações do método aqui estudado.

É importante destacar que a ponderação utilizada na Seção 4.1 é apenas umas das múltiplas soluções da otimização multiobjetivo apresentada nesta seção e que as demais soluções pertencentes ao conjunto Pareto não devem ser descartadas, uma vez que podem mostrar-se adequados para outros objetivos.

Uma vez que a ponderação entre as funções-custo apresentadas pelas Equações 4.1 e 4.2 aqui tomada mostrou-se apropriada aos objetivos tratados neste capítulo, decidiu-se pela sua utilização nas aplicações apresentadas nos Capítulos 5. Já no Capítulo 6, outras ponderações são utilizadas de modo a garantir melhores resultados. Por outro lado, entende-se que pequenas variações podem garantir resultados tão bons quanto os apresentados aqui, para fins de representação ou mesmo de classificação.

Destaca-se, ainda, que a efetividade do método de seleção automática dos parâmetros iniciais do algoritmo de k-segmentos não suave aqui proposto estende-se também ao algoritmo de k-segmentos suave, sem que haja necessidade de qualquer mudança. Os resultados obtidos para este segundo algoritmo de extração de CP é apresentado no Apêndice A, entendendo-se que apesar de não ter sido efetivamente utilizado nesta pesquisa, a possibilidade da otimização de seu parâmetros não se mostra menos importante que a otimização do algoritmo k-segmentos não suave, amplamente utilizado neste trabalho.

5 CLASSIFICAÇÃO SUPERVISIONADA COM A UTILIZAÇÃO DO ALGORITMO K-SEGMENTOS COM PARÂMETROS INICIAIS OTIMIZADOS

As Curvas Principais foram utilizadas para a classificação de dados pela primeira vez por Chang e Ghosh (1998). Na literatura podem ser encontradas diversas outras aplicações com objetivos semelhantes, tais como os métodos de classificação de navios propostos por Fernandes, Souza Filho e Alves (2015) e Oliveira e Filho (2017), em que o algoritmo k-segmentos é utilizado.

Neste capítulo o método de seleção automática dos parâmetros do algoritmo k-segmentos é utilizada para a construção das curvas que representam as diferentes classes de cada um dos bancos de dados, de modo a se verificar a viabilidade de tal técnica na classificação supervisionada de dados.

O restante do capítulo é dividido da seguinte forma: na Seção 5.1 são apresentados os bancos de dados utilizados para a realização dos testes, bem como as etapas de pré-processamento necessárias para sua utilização. Na Seção 5.2 se discute a classificação supervisionada dos dados com a utilização do método proposto, em que a curva é construída pela minimização de seu comprimento total e das distâncias dos eventos à curva. Já na Seção 5.3, são apresentados os resultados do método que utiliza a minimização do erro de classificação das bases de validação como função-custo do algoritmo de otimização.

5.1 Banco de dados

Para testar a utilização do método proposto na classificação supervisionada de dados foram utilizadas 10 bases de dados, reais e simuladas. As bases de dados reais foram obtidas no *UCI Machine Learning Repository* (DUA; GRAFF, 2017). As bases de dados simuladas utilizadas neste trabalho são as mesmas usadas por Moraes (2015), de modo que a criação das bases simuladas foi realizada de acordo com as definições apresentadas por Theodoridis e Koutroumbas (2009).

Na Tabela 5.1 são encontradas informações detalhadas a respeito das bases de dados utilizadas.

Tabela 5.1 – Bases de dados utilizadas

Base de Dados UCI	Amostras	Classes	Variáveis
Diabetes	768	2	8
Half Rings	373	2	2
Iris	150	3	4
Libras Movement	360	15	90
Seeds	210	3	7
Spiral	206	2	2
Wine	178	3	13
Base de Dados Simuladas	Amostras	Classes	Variáveis
Compact	1000	2	2
Elongated	1000	2	2
Spherical	1000	2	2

5.1.1 Pré-processamento dos dados

A fase de pré-processamento de dados é de grande importância em projetos de aprendizado de máquina, uma vez que o correto funcionamento dos diversos modelos de classificação, clusterização e regressão dependem da organização e clareza dos dados por eles utilizados.

Para o método de classificação apresentado neste capítulo e o método de clusterização apresentado no Capítulo 6, o pré-processamento dos dados mostra-se importante para que a construção das curvas seja feita de maneira ideal, sem que ocorram erros computacionais durante as iterações do algoritmo k-segmentos.

Durante a realização de testes, observou-se que o algoritmo apresenta erros quando se utiliza bancos de dados que tem dados duplicados. Esses erros ocorrem durante a construção da CP por meio do algoritmo k-segmentos. Para evitar a ocorrência de tais erros deve-se excluir dados duplicados, de modo que essa eliminação tem pouco impacto na otimização dos parâmetros do algoritmo k-segmentos, tampouco na construção dos segmentos, uma vez que o efeito a ambos os termos da função-custo utilizada para essa otimização é baixo e não prejudica a ponderação entre ambos.

É importante observar que essa exclusão não causa prejuízos aos métodos de classificação e clusterização, uma vez que eventos duplicados são informação redundante e não contribuem para o aprendizado dos algoritmos de reconhecimento de padrões.

Assim como diversas técnicas de aprendizado de máquina, o método aqui proposto necessita que os dados estejam normalizados para o correto funcionamento do algoritmo. No caso do método aqui proposto, isso ocorre porque é necessário que os diversos parâmetros estejam na mesma faixa de valores, uma vez que a existência de parâmetros em faixas diferentes uns

dos outros fazem com que os segmentos não sejam construídos adequadamente pelo algoritmo, que podem apresentar problemas em criar as coordenadas de seus vértices e suas conexões.

Desse modo, os valores dos diversos parâmetros das bases de dados aqui utilizadas foram normalizados de acordo com a Equação:

$$x_{norm} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (5.1)$$

em que que x_{norm} é o valor normalizado da instância x de um dos parâmetros da base de dados a ser normalizada, x_{\min} e x_{\max} são os valores mínimos e máximos assumidos, respectivamente, pelo parâmetro em questão.

Com essa normalização, todos os parâmetros das bases de dados assumem valores entre 0 e 1, o que garante o funcionamento adequado do algoritmo de construção de CP que se utiliza. Para algumas bases de dados há necessidade de se aplicar outras técnicas de pré-processamento, tal como imputação de dados faltantes, tratamento de dados categóricos, seleção de variáveis e outras técnicas. Para as bases de dados aqui utilizadas, porém, não se mostrou necessária a aplicação de nenhum desses métodos de pré-processamento, estando, assim, os dados prontos para a utilização.

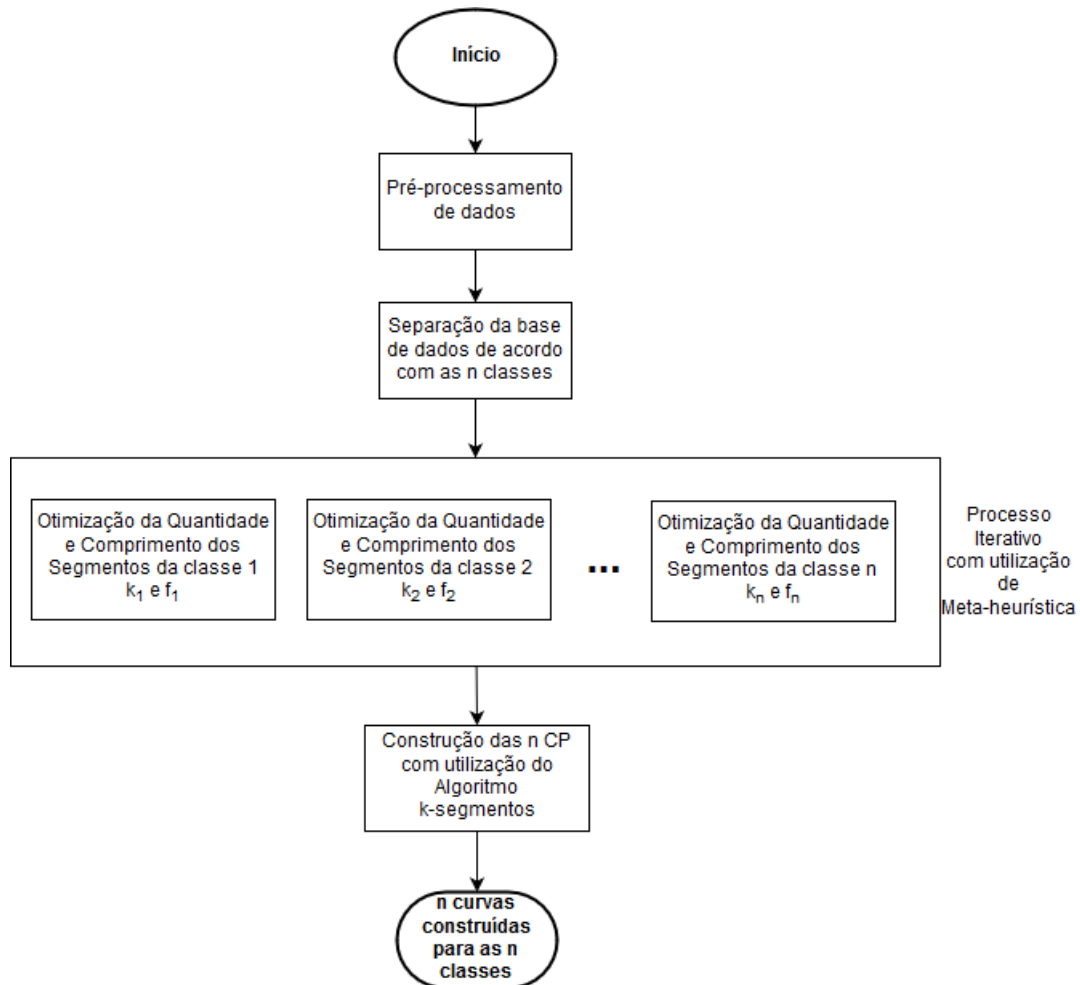
5.2 Minimização do comprimento da curva e distância aos eventos como função-custo

A criação do modelo utilizado na classificação supervisionada de dados utilizando o algoritmo k-segmentos é feita como visto no fluxograma apresentado na Figura 5.1.

Após o pré-processamento de dados, a base de dados deve ser separada de acordo com as diferentes classes que possui. Essa etapa é importante, pois é construída uma CP para cada uma das classes que compõem o conjunto de dados. A construção dessas curvas é feita de maneira paralela, com a utilização da meta-heurística utilizada, no caso aqui apresentado, TLBO. Para isso a população é modelada com os parâmetros $(k_1, f_1, k_2, f_2, \dots, k_n, f_n)$, em que cada par (k_i, f_i) constitui os parâmetros de uma das n CP que por sua vez representam uma das n classes do conjunto de dados.

Uma vez criado o modelo do classificador, como mostrado na Figura 5.1, a classificação dos eventos é feita por meio do cálculo do quadrado da distância euclidiana entre as n curvas e o evento a ser classificado. Desse modo o evento é classificado como pertencente à classe representada pela curva mais próxima, como apresentado pelo fluxograma da Figura 5.2.

Figura 5.1 – Fluxograma da criação do Modelo de Classificação Supervisionada por meio da minimização do comprimento da curva e distância aos eventos como função-custo



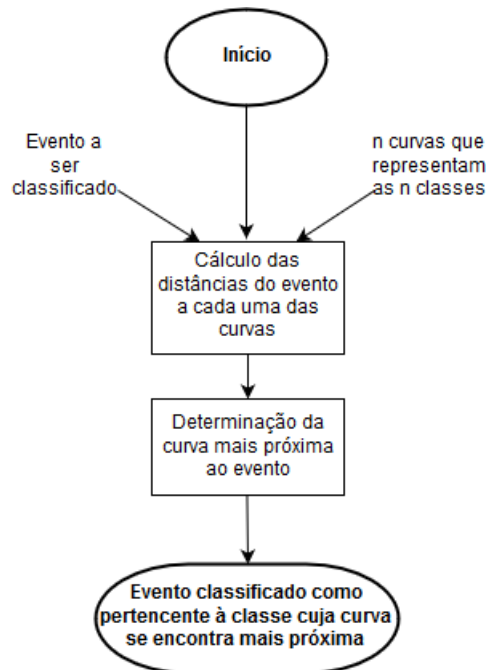
Fonte: Acervo do Autor

Esse processo de classificação demonstra a necessidade do correto dimensionamento das CP que representam cada uma das classes, uma vez que uma curva bem dimensionada é capaz de representar de maneira significativa a distribuição e contorno formado pelos dados. Curvas subdimensionadas ou superdimensionadas podem gerar erros no processo de classificação, pois não são capazes de representar corretamente as classes, fazendo com que eventos pertencentes a determinada classe possam estar mais próximos das curvas de outras, sendo assim erroneamente classificados.

De modo a verificar a viabilidade de aplicação do método de seleção automática de parâmetros do algoritmo k-segmentos no modelo de classificador com curvas foram realizados testes com as bases de dados apresentadas na Seção 5.1.

Os testes foram realizados com a utilização da validação cruzada *k-fold*, com $k = 10$. Desse modo a cada iteração do algoritmo 90% dos dados foram utilizados para construir as

Figura 5.2 – Processo de classificação supervisionada de dados com utilização de CP



Fonte: Acervo do Autor

CP, enquanto os 10% restantes foram utilizados para verificar a capacidade de classificação do modelo gerado.

Na Tabela 5.2 se encontram os resultados dos testes, sendo apresentadas as médias e desvio padrão da acurácia de classificação do método proposto e do classificador *k-nearest neighbors* (k-NN), com valores de *k* evidenciados na tabela.

Tabela 5.2 – Desempenho da classificação supervisionada, em %, para as bases de teste

Base de Dados	Método Proposto	k-NN ($k = 3$)	k-NN ($k = 5$)
<i>Diabetes</i>	63,41 ± 3,51	72,65 ± 3,87	73,56 ± 4,16
<i>Half Rings</i>	100,00 ± 0,00	100,00 ± 0,00	100,00 ± 0,00
<i>Iris</i>	96,67 ± 3,51	94,67 ± 4,22	95,33 ± 4,50
<i>Libras Movement</i>	83,06 ± 5,15	78,61 ± 8,89	75,56 ± 9,69
<i>Seeds</i>	91,90 ± 3,92	92,86 ± 2,51	92,86 ± 4,05
<i>Spiral</i>	100,00 ± 0,00	100,00 ± 0,00	100,00 ± 0,00
<i>Wine</i>	95,49 ± 5,20	94,93 ± 6,19	97,22 ± 5,40
<i>Compact</i>	99,10 ± 0,74	99,50 ± 0,71	99,30 ± 0,82
<i>Elongated</i>	99,70 ± 0,67	99,60 ± 0,52	99,70 ± 0,48
<i>Spherical</i>	99,90 ± 0,31	99,90 ± 0,32	99,70 ± 0,48

Assim como as CP, o k-NN é um modelo não-paramétrico que apresenta diversas aplicações em reconhecimento de padrões. A classificação por esse modelo é feita de modo que ao evento que está sendo classificado é atribuída a classe mais comum entre seus *k* vizinhos mais próximos. Nos testes, a métrica de distância utilizada pelo k-NN é a distância euclidiana.

Como pode ser observado na Tabela 5.2, os resultados obtidos pelo método proposto são, para a maioria das bases de dados testadas, similares aos obtidos pelos classificadores k-NN, com $k = 3$ e $k = 5$. Observa-se que para a base de dados *Diabetes* a acurácia obtida pelo classificador em curvas é menor do que os classificadores k-NN. Por outro lado, para a base *Libras Movement* tem-se uma média de acurácia maior e menor desvio padrão dos resultados, o que indica menor dispersão das acurácias obtidas durante os testes de validação cruzada realizados. Os bons resultados obtidos nessa base demonstram a capacidade do método proposto de lidar com dados de altas dimensões e múltiplas classes, uma vez que a base *Libras Movement* apresenta 90 dimensões e 15 classes.

Ao se observar os valores dos parâmetros encontrados pelo algoritmo durante os testes, é possível perceber a importância da definição automática dos mesmos. De modo a exemplificar tal afirmação, toma-se os resultados do teste de uma das partições da base *Iris*, composta por 3 classes. Neste teste, a acurácia de classificação foi de 100%. Esses resultados foram obtidos com a definição de valores de k iguais a 9, 17 e 8 e f iguais a 1,37, 1,10 e 0,78 para as classes que compõem essa base de dados.

A definição de parâmetros diferentes para cada uma das classes, como observado nos testes, é inviável de ser realizada manualmente, principalmente em bases de dados em que o número de classes é maior, tal como a base *Libras Movement*, que possui 15 classes. Uma vez que a definição de tais parâmetros mostra-se fundamental para a garantia de resultados satisfatórios na classificação supervisionada de dados, a otimização realizada pelo método proposto é de grande importância e, pelos resultados obtidos, é possível concluir que esse método de classificação apresenta, de maneira geral, boa capacidade de generalização, mostrando ser capaz de construir CP que representam as diferentes classes das bases de dados de maneira correta, de modo que mesmo os eventos não utilizadas na construção das curvas se projetam de maneira satisfatória nas mesmas, o que por sua vez garante boa acurácia de classificação.

Na Seção 5.3 é apresentado um modelo de classificação supervisionada de dados com o algoritmo k-segmentos com parâmetros determinados pela minimização do erro de classificação da base de validação e realizadas comparações com o modelo apresentado nesta seção.

5.3 Minimização do erro de classificação como função-custo

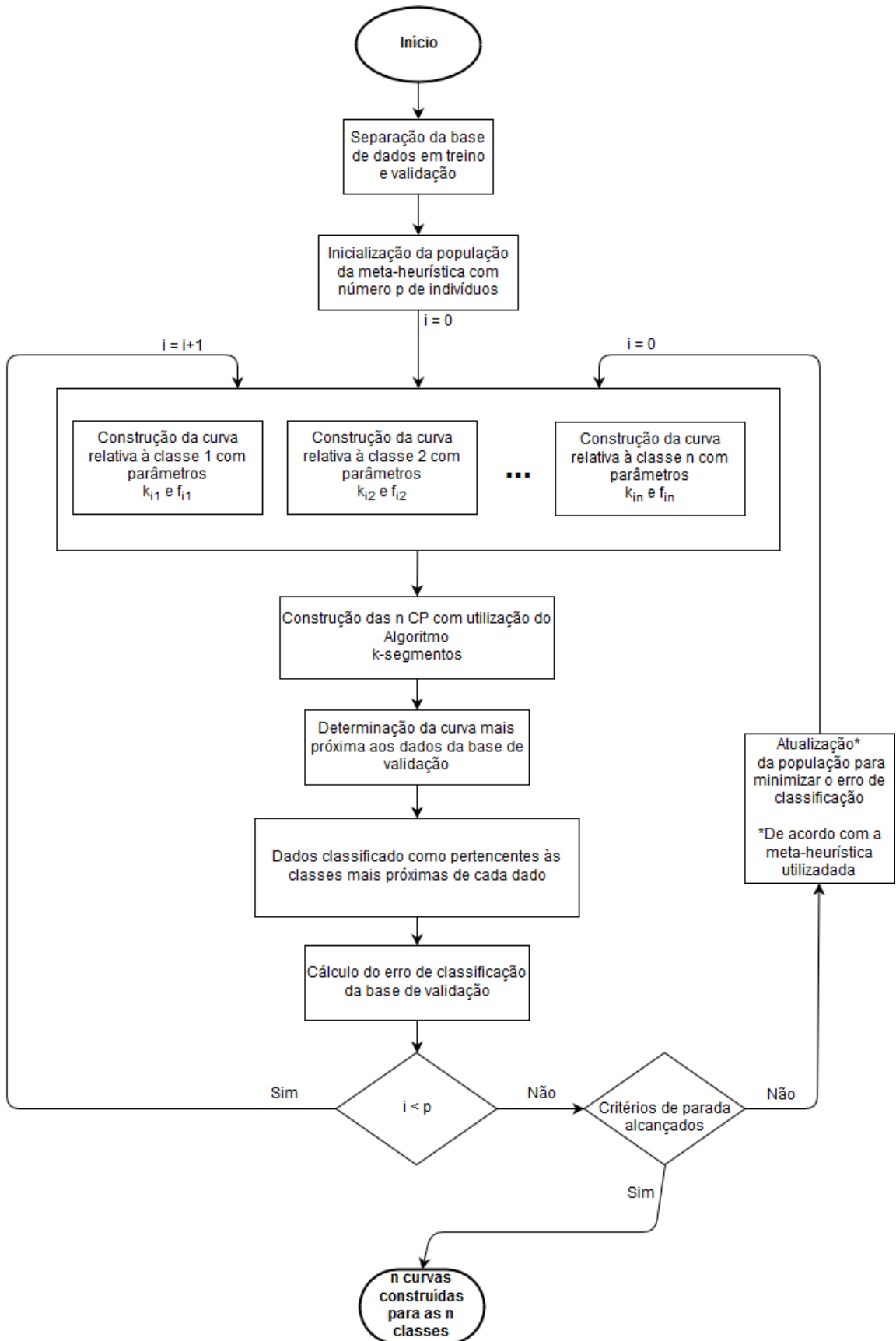
A criação do modelo utilizado para classificação supervisionada de dados com o algoritmo k-segmentos com parâmetros determinados pela minimização do erro de classificação da base de validação é feita como apresentado na Figura 5.3.

Para que o processo de criação do modelo seja possível é necessário dividir a base de dados, para a qual se deseja construir as curvas, em base de treino e base de validação. Uma vez realizada essa divisão, otimiza-se a quantidade e comprimento dos segmentos das curvas a partir da minimização do erro de classificação da base de validação. Para isso, deve-se construir paralelamente as curvas que representam as classes da base e verificar se são capazes de classificar corretamente os eventos utilizados para validação. A partir da acurácia de classificação dessa base, atualiza-se a população de acordo com os critérios da meta-heurística utilizada para que as curvas sejam corretamente dimensionadas e capazes de representar de maneira satisfatória as classes do banco de dados.

Uma vez modelado o classificador, o processo de classificação de dados é feita de maneira similar ao classificador apresentado na Seção 5.2, ou seja, como apresentado no fluxograma da Figura 5.2.

Os testes do classificador, modelado da maneira apresentada nesta seção, foram realizados com a utilização da validação cruzada *k-fold*, com $k = 10$. Para atender os critérios de construção do modelo, a base de treino foi dividida em treino e validação de modo que a cada iteração 90% dos dados foram utilizados para gerar as CP, sendo 80% na construção dos contornos da curva e 10% para testar e otimizar os parâmetros da mesma, ambos conjuntos escolhidos aleatoriamente dentre os dados de treino. Os 10% restantes foram utilizados para verificar a capacidade de classificação do modelo gerado. Foram utilizadas 5 bases de dados reais para verificar a viabilidade do método proposto nesta seção. Essas bases foram escolhidas por serem as bases de dados reais em que a acurácia foi menor que 100% nos testes realizados na Seção 5.2. Uma vez que deseja-se verificar a potencialidade de se utilizar a minimização do erro como função-custo da otimização dos parâmetros do algoritmo k-segmentos, a utilização das bases escolhidas mostra-se suficiente para tal análise, como pode ser observado nas discussões que seguem.

Figura 5.3 – Fluxograma da criação do Modelo de Classificação Supervisionada por meio da minimização do erro de classificação como função custo



Fonte: Acervo do Autor

Os resultados obtidos são apresentados na Tabela 5.3, onde se encontram as médias e desvio padrão da acurácia de classificação do método proposto nesta seção e do classificador com curvas apresentados na Seção 5.2.

Tabela 5.3 – Comparação de desempenho da classificação supervisionada, em %, para diferentes funções-custo

Base de Dados	Comprimento da curva e distância aos eventos como função-custo	Erro de classificação como função custo
<i>Diabetes</i>	63,41 ± 3,51	68,63 ± 6,40
<i>Iris</i>	96,67 ± 3,51	92,00 ± 6,13
<i>Libras Movement</i>	83,06 ± 5,15	74,17 ± 5,72
<i>Seeds</i>	91,90 ± 3,92	91,90 ± 7,46
<i>Wine</i>	95,49 ± 5,20	95,49 ± 4,50

Pelos resultados apresentados na Tabela 5.3, observa-se que para as bases *Seeds* e *Wine* a média das acurácias dos testes são semelhantes para ambos os modelos analisados. Por outro lado, o modelo determinado pela minimização de erro de classificação apresenta maior desvio padrão, ou seja, maior dispersão dos resultados.

Para as bases *Iris* e *Libras Movement* os resultados do modelo proposto nesta seção são piores que o modelo proposto na Seção 5.2. Além disso observa-se aumento do desvio padrão da acurácia dos testes, com exceção da base *Wine*. A grande dispersão dos resultados pode ser explicada pela diminuição da quantidade de dados utilizado para a construção da curva, uma vez que parte dos dados de treinamento são utilizados na validação das curvas construídas, o que prejudica a determinação dos vértices dos segmentos que compõem a curva. Isso faz com que a CP não seja capaz de realizar o contorno dos dados de maneira ideal, o que leva à diminuição da capacidade de generalização do modelo proposto nesta seção.

A baixa capacidade de generalização pode fazer com que as acurácias de classificação sejam menores, como observado com as bases *Iris* e *Libras Movement*. Além disso, observa-se que houve aumento do desvio padrão dos resultados para as 5 bases de dados analisadas. Por outro lado, para base de dados *Diabetes* observa-se aumento significativo da acurácia média de classificação, apesar dos resultados ainda apresentarem-se inferiores aos obtidos pelo classificador k-NN, exibidos na Tabela 5.2.

Em relação aos parâmetros sugeridos pelo algoritmo, observou-se que, assim como na Seção 5.2, a quantidade e comprimento dos segmentos foram, na maioria dos casos, diferentes entre as classes de cada base de dados. Ao se comparar os resultados dos dois métodos de otimização testados, percebe-se que, na maioria dos casos os parâmetros das mesmas classes são

diferentes. Isso ocorre uma vez que o método apresentado nesta seção utiliza uma quantidade menor de dados para construção da curva. Esse fato faz com que, mesmo nos casos em que os parâmetros entre ambos os métodos são iguais, a curva construída é diferente devido à diferença dos dados utilizados em sua construção, ou seja, a diminuição da quantidade de dados utilizados faz com que a posição dos vértices dos segmentos seja diferente.

Por esses resultados, verifica-se que o método de classificação supervisionada de dados com a utilização do algoritmo k-segmentos com parâmetros determinados pela minimização do comprimento geral da CP e da distância dos eventos à curva é, de modo geral, melhor que o modelo proposto nesta seção, uma vez que apresenta maior capacidade de construir curvas que generalizam o conjunto de dados. Por outro lado, as potencialidades de ambos os modelos não podem ser descartadas, sendo necessário, principalmente para o modelo que tem como função-custo o erro de classificação da base de validação, estudos mais aprofundados e realização de possíveis adaptações de modo a garantir-lhe maior robustez durante a classificação de dados.

A potencialidade da utilização de CP pode ser comprovada pelas diversas aplicações encontradas na literatura, desde a primeira utilização de curvas para tal propósito, por Chang e Ghosh (1998), à classificação de navios por imagens de sonar, por Oliveira e Filho (2017). Desse modo a otimização dos parâmetros apresentada neste documento servem como uma maneira de se buscar o aumento as potencialidades das CP para tais aplicações, tal como a possibilidade de otimizar o algoritmo de classificação não supervisionada de dados, proposto por Moraes e Ferreira (2016), apresentado no Capítulo 6.

6 CLUSTERIZAÇÃO COM A UTILIZAÇÃO DO ALGORITMO K-SEGMENTOS COM PARÂMETROS INICIAIS OTIMIZADOS

A utilização de Curvas Principais na classificação não supervisionada de dados utilizando o algoritmo k-segmentos foi proposta por Moraes (2015), Moraes e Ferreira (2016) e Moraes et al. (2019). A classificação não supervisionada de dados ou clusterização consiste no agrupamento automático de dados segundo seu grau de semelhança. Aos agrupamentos dá-se o nome de *clusters*.

O método de clusterização de dados baseado em CP que utiliza o algoritmo k-segmentos constrói a curva que representa a totalidade dos dados. É importante destacar que, diferentemente da classificação supervisionada de dados, em que para cada classe é construída uma CP, o primeiro passo da classificação não supervisionada é a construção de uma única CP para todo o conjunto de dados.

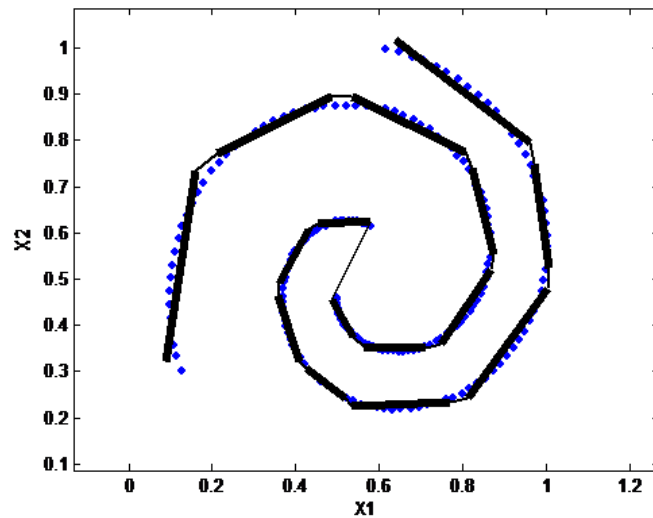
Após a construção dessa CP, calculam-se os comprimentos das linhas que ligam os segmentos que compõem a curva. Para isso utiliza-se o quadrado da distância euclidiana entre os vértices de interconexão dos segmentos. Uma vez determinado pelo usuário o número n de agrupamentos, são excluídas as $n - 1$ linhas de maior comprimento de modo a separar a CP original em n curvas diferentes, de modo que cada curva resultante representa uma das classes. Esse processo baseia-se no fato de que a construção da CP pelo algoritmo k-segmentos garante que os segmentos sejam alocados nas regiões de Voronoi, enquanto as linhas de interconexão entre os segmentos são alocados em regiões com menor densidade de dados, o que ocorre nos espaços entre os diferentes agrupamentos. Na Figura 6.1 é possível observar o funcionamento desse algoritmo de clusterização.

O contorno de dados da figura representa dois diferentes agrupamentos, a espiral interna e a espiral externa. Na Figura 6.1(a) observa-se uma única CP construída sobre o contorno de dados. É possível perceber de maneira clara que a maior linha de interconexão (linha mais fina que os segmentos) se encontra entre os segmentos das diferentes espirais, ou seja, entre os diferentes agrupamentos.

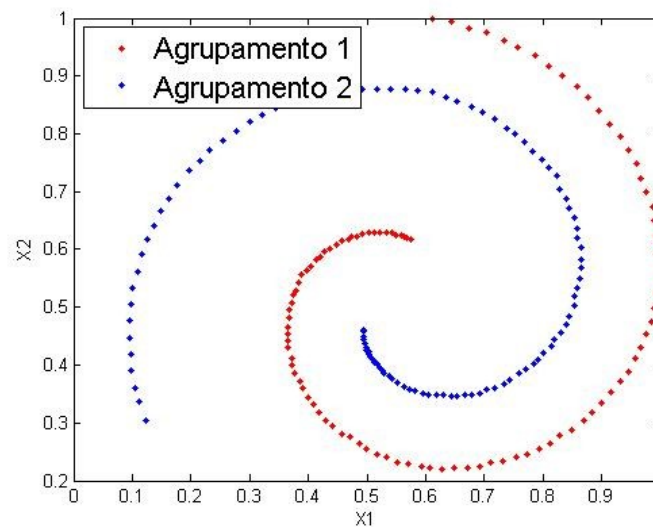
De modo a separar os agrupamentos existentes, a maior linha é excluída, gerando duas curvas distintas, cada uma representando um dos *clusters*, como visto na Figura 6.1(b).

Segundo Moraes et al. (2019), é de grande relevância para esse algoritmo de clusterização determinar a quantidade e comprimento ideais de segmentos que compõem a curva, uma vez que a acurácia do método está diretamente ligada ao dimensionamento da CP, de modo

Figura 6.1 – Funcionamento do algoritmo de clusterização de dados com a utilização do algoritmo k-segmentos.



(a) CP construída de modo a representar a base de dados



(b) Separação da base de dados em dois agrupamentos, após a exclusão da maior linha de ligação entre segmentos

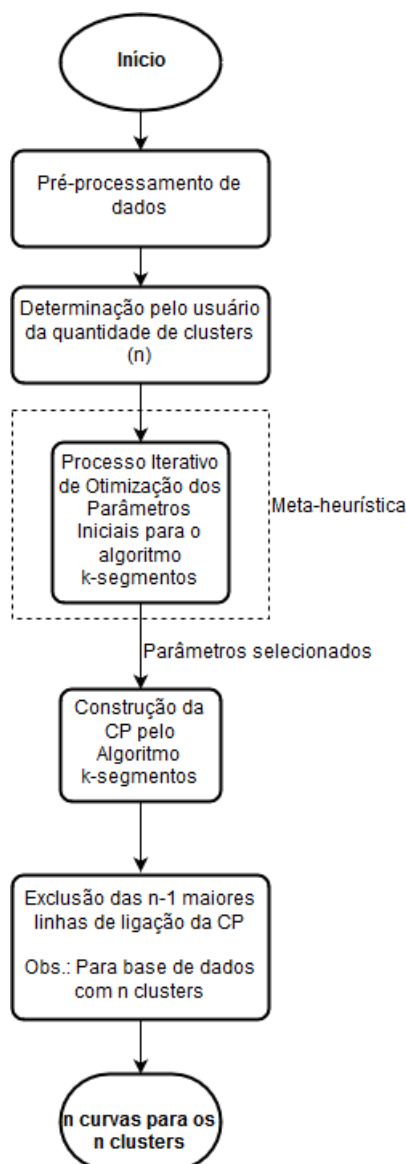
Fonte: Adaptado de Moraes (2015)

a, posteriormente, garantir sua separação em curvas menores, cada qual representando um dos diferentes agrupamentos do banco de dados.

Com o objetivo de se dimensionar corretamente a CP para a classificação não supervisionada de dados, propõem-se a determinação automática dos parâmetros do algoritmo k-segmentos com a utilização de meta-heurísticas, como apresentada no fluxograma da Figura 6.2. O primeiro passo desse método é o pré-processamento dos dados a serem classificados. Essa parte do processo é realizada de maneira semelhante à detalhada na Seção 5.1.1, uma vez

que assim são atendidas as necessidades do algoritmo k-segmentos para a correta construção da curva.

Figura 6.2 – Fluxograma da criação do Modelo de Clusterização com seleção automática dos parâmetros do algoritmo k-segmentos



Fonte: Acervo do Autor

Após a determinação do número de agrupamentos pelo usuário, inicia-se o processo iterativo da determinação dos parâmetros do algoritmo k-segmentos. Esse processo é igual ao descrito no Capítulo 3, uma vez que nesta etapa apenas uma curva deve ser construída, diferente da aplicação apresentada no Capítulo 5, em que o número de curvas construídas é igual ao número de classes da base de dados. A definição da população da meta-heurística é realizada de modo que os parâmetros se encontrem dentro de faixas de valores específicas. Assim, a quantidade de segmentos se encontra no intervalo $(n; 30 \times n)$, onde n é o número de

clusters da base de dados. Por outro lado o comprimento dos segmentos se encontra no intervalo $(0,0001;2)$, destacando que quando o comprimento do segmento é pequeno o suficiente, ele pode ser considerado como um ponto, fazendo com que o método de clusterização baseado no algoritmo *k*-segmentos seja semelhante ao algoritmo *k-means*.

Como última etapa, são excluídas as $n - 1$ maiores linhas de interconexão entre os segmentos de modo a serem geradas n curvas, cada qual representando um dos n *clusters*, finalizando, assim, o processo.

Os testes do método de clusterização proposto foram realizados com a utilização das mesmas bases de dados reais e simuladas descritas na Seção 5.1, com exceção da base *Libras Movement*, uma vez que seu elevado número de classes faz com que a verificação da acurácia dos resultados, que depende das permutações dos rótulos de cada agrupamento, se torne computacionalmente inviável.

Os resultados obtidos pelo método proposto são comparados com o algoritmo de clusterização *k-means*, tendo como métrica a distância euclidiana. Esse algoritmo divide os dados em diferentes agrupamentos a partir da minimização da distância total entre os dados de um grupo e o seu respectivo centro.

De modo a verificar a viabilidade do método apresentado na Figura 6.2, foram realizadas duas fases de testes. Na primeira fase, apresentada na Seção 6.1, utilizou-se técnicas de validação cruzada de modo a analisar os resultados, enquanto na segunda etapa, apresentada na Seção 6.2, foram utilizadas as bases de dados completas para a realização das análises.

6.1 Resultados da classificação não supervisionada nos testes com validação cruzada

Nesta seção são apresentados os resultados dos testes que foram realizados com a utilização de técnicas de validação cruzada. Da mesma maneira que na Seção 5.2, as bases de dados foram divididas em 10 partições, de modo que a cada iteração do algoritmo 90% dos dados foram utilizados para construir as CP, enquanto os 10% restantes foram utilizados para verificar se os agrupamentos construídos pelo algoritmo são, de fato, capazes de representar as diferentes classes de cada base de dados.

Durante a realização dos testes iniciais, observou-se que para o método de clusterização apresentado aqui, a função-custo utilizada nos Capítulos 3, 4 e 5 não retorna resultados satisfatórios. Por esse motivo, decidiu-se modificar as ponderações relativas aos termos da função-custo, aumentando a importância do termo relativo ao comprimento total da curva. Desse modo,

utilizou-se como função-custo a equação:

$$0,1 n \log l + 0,9 \sum_{i=1}^k \sum_{x \in V_i} d(s_i, \mathbf{x})^2 / (2\sigma^2). \quad (6.1)$$

As novas ponderações tem como objetivo a geração de curvas mais compactas, ou seja, curvas formadas por uma quantidade menor de segmentos, de modo que ao serem retiradas as maiores linhas de interconexão, os dados possam ser divididos corretamente em *clusters*.

Os valores de média e desvio padrão dos testes são apresentados na Tabela 6.1, juntamente com os resultados obtidos pelo algoritmo de clusterização *k-means*.

Tabela 6.1 – Desempenho da classificação não supervisionada, em %, para as bases de teste com validação cruzada

Base de Dados	Método Proposto	<i>k-means</i>
<i>Diabetes</i>	60,67 ± 5,99	66,80 ± 3,50
<i>Half Rings</i>	89,00 ± 8,99	87,13 ± 4,08
<i>Iris</i>	73,33 ± 12,96	75,33 ± 12,20
<i>Seeds</i>	60,47 ± 10,54	89,05 ± 5,04
<i>Spiral</i>	99,00 ± 3,16	55,79 ± 3,90
<i>Wine</i>	47,75 ± 7,84	92,75 ± 11,72
<i>Compact</i>	75,60 ± 16,45	99,60 ± 0,70
<i>Elongated</i>	82,00 ± 12,87	62,30 ± 2,75
<i>Spherical</i>	72,90 ± 16,37	85,00 ± 1,94

Como pode ser observado na Tabela 6.1, os resultados obtidos pelo método proposto são inferiores aos encontrados pelo método de clusterização *k-means* para as bases de dados *Seeds*, *Wine*, *Spherical* e *Compact*. Esses resultados mostram que as CP geradas na primeira fase do algoritmo não foram divididas em curvas menores de modo adequado. Com isso, os agrupamentos gerados mostraram-se incapazes de representar os *clusters* reais dessas bases de dados.

Por outro lado, para as bases *Spiral* e *Elongated*, a acurácia média de clusterização dos dados mostra-se significativamente maior do que a clusterização utilizando *k-means*. Os resultados melhores para essas duas últimas bases citadas pode ser explicada pela distribuição dos dados no espaço multidimensional, formado pelos seus parâmetros, de modo que para essas distribuições, os agrupamentos gerados pelo algoritmo *k-means* não se aproximam dos agrupamentos reais de dados, enquanto o algoritmo baseado em curvas é capaz de fazer isso de maneira satisfatória.

Para as demais bases de dados testadas, os resultados obtidos por ambos os métodos de clusterização são similares. De modo geral, é possível observar grande dispersão dos resultados obtidos durante os testes com o método proposto. Isso ocorre devido ao fato de que a curva construída com a utilização da base de treinos, em alguns casos, não representa de maneira adequada os eventos da base de testes.

Por outro lado, observa-se que as CP geradas por meio da minimização da função-custo, Equação 6.1, podem não apresentar de maneira adequada as dimensões necessárias para sua posterior divisão em curvas menores, sendo necessário verificar a utilização de outras funções-custo. Na Seção 6.2, são apresentados os resultados dos testes com as bases de dados completas de modo a verificar a potencialidade do método proposto com a utilização de duas diferentes funções-custo.

6.2 Resultados da classificação não supervisionada nos testes com bases de dados completas

Nesta seção são apresentados os resultados dos testes realizados com as bases de dados completas, de modo a verificar a acurácia do método proposto na separação das bases de dados em agrupamentos. Para isso, todos os dados de cada base são utilizados na construção das CP que representam os agrupamentos, que são, posteriormente, comparados com os *clusters* reais de cada base, de modo a verificar a capacidade do algoritmo em questão na clusterização de dados. Assim como na Seção 6.1, os resultados obtidos pelo método proposto são comparados com os resultados do método *k-means*.

Uma vez que os resultados obtidos durante os testes de validação cruzada não foram satisfatórios para todas as bases de dados, decidiu-se utilizar outra função-custo nos testes. Desse modo, além da Equação 6.1, é também utilizada na otimização, como função objetivo, a equação:

$$0,98n \log l + 0,02 \sum_{i=1}^k \sum_{x \in V_i} d(s_i, \mathbf{x})^2 / (2\sigma^2). \quad (6.2)$$

As ponderações dos termos da Equação (6.2) foram tomadas de modo a dar maior importância à minimização do tamanho da curva gerada, com o intuito de verificar os resultados obtidos pelo método proposto neste capítulo, quando quantidade e comprimento dos segmentos são menores.

A acurácia na classificação não supervisionada dos dados do método proposto, com a utilização das duas funções-custo, juntamente com os resultados obtidos pelo algoritmo de clusterização *k-means*, são apresentados na Tabela 6.2.

Tabela 6.2 – Desempenho da classificação não supervisionada, em %, para as bases de dados completas

Base de Dados	Método Proposto tendo Equação (6.1) como função-custo	Método Proposto tendo Equação (6.2) como função-custo	<i>k-means</i>
<i>Diabetes</i>	57,16	64,84	63,54
<i>Half Rings</i>	80,70	90,08	86,60
<i>Iris</i>	73,47	85,03	81,63
<i>Seeds</i>	56,19	89,52	88,57
<i>Spiral</i>	100,00	50,97	54,37
<i>Wine</i>	53,37	62,92	94,94
<i>Compact</i>	84,50	98,90	99,60
<i>Elongated</i>	97,40	73,30	62,20
<i>Spherical</i>	54,40	87,00	85,00

Como pode ser visto na Tabela 6.2, ao se utilizar como função-custo a Equação (6.1), a acurácia de clusterização das bases *Spiral* e *Elongated* pelo método proposto é alta, sendo expressivamente superior aos resultados obtidos pelo *k-means*, resultados similares ao obtidos por essas mesmas bases nos testes da Seção 6.1. Quando utilizada a Equação (6.2) como função objetivo, a acurácia da classificação supervisionada dessas bases de dados diminui de maneira significativa, apesar dos resultados da base *Elongated* manterem-se melhores que os obtidos pelo *k-means*.

Por outro lado, para todas as demais bases, a utilização da Equação (6.2) como função-custo traz melhorias de acurácia, de modo que com a exceção das bases *Compact* e *Wine*, as demais apresentam resultados superiores àqueles apresentados pelo algoritmo *k-means*.

Para a base *Compact* os resultados do método proposto e do algoritmo *k-means* são próximos, tendo ambos acurácia superior a 98,5%. Por outro lado, a acurácia para *Wine* obtida pelo método de clusterização baseado em curvas, é inferior à obtida pelo *k-means*.

De modo a avaliar melhor os resultados obtidos pelo método, e avaliar o impacto da utilização das duas diferentes funções-custo, é necessário verificar os parâmetros definidos pelo algoritmo. Na tabela 6.3 são apresentados os valores de k e f , sendo $f\sigma$ o comprimento dos segmentos, definidos para todas as bases de dados pelo método proposto, para as duas funções objetivo utilizadas na otimização.

Tabela 6.3 – Quantidade e comprimento dos segmentos obtidos durante os testes de classificação não supervisionada

Base de Dados UCI	Método Proposto tendo Equação (6.1) como função-custo		Método Proposto tendo Equação (6.2) como função-custo	
	k	f	k	f
<i>Diabetes</i>	60	1,878	20	1,933
<i>Half Rings</i>	28	0,509	3	0,506
<i>Iris</i>	147	1,327	3	0,327
<i>Seeds</i>	80	1,004	3	0,532
<i>Spiral</i>	24	0,508	4	0,509
<i>Wine</i>	54	1,951	28	1,001
<i>Compact</i>	56	0,652	3	0,774
<i>Elongated</i>	11	0,746	3	0,581
<i>Spherical</i>	24	0,521	4	0,687

Como pode ser observado na Tabela 6.3, os valores de k e f tem variações significativas quando se varia a função-custo utilizada na otimização. A diminuição da quantidade média de segmentos com a utilização da Equação (6.2) deve-se ao fato de se dar maior importância ao tamanho total da curva. Para as bases *Half Rings*, *Iris* e *Seeds* observa-se que a quantidade de segmentos definida pelo algoritmo é 3, valor mínimo para as bases de 3 agrupamentos, de modo que cada agrupamento é representado por um único segmento.

Essa baixa quantidade de segmentos gera representações simples das diferentes classes das bases de dados. Apesar da simplicidade, essas curvas são capazes de separar o conjunto de dados em agrupamentos com melhor capacidade de representar os *clusters* reais do que aqueles formados pelo algoritmo *k-means*. O método proposto com utilização da Equação (6.2) gera classificadores mais simples computacionalmente, tanto na fase de projeto (treino) quanto na fase operacional, uma vez que gera curvas mais simples, o que demanda menos iterações de inserção de segmentos pelo algoritmo *k-segmentos*. Durante os testes observou-se que, para as bases onde houve melhoria dos resultados com a utilização desta equação, também houve melhora na acurácia média nos testes de validação cruzada. Se por um lado a simplicidade das curvas geradas melhorou a acurácia do método para essas bases de dados, para as bases *Compact* e *Spiral*, as curvas mais complexas, ou seja, aquelas que apresentam maior número de segmentos, mostram-se significativamente melhores para o processo de clusterização.

Esses fatos demonstram que para o processo de classificação não supervisionada de dados, a possibilidade da CP ser dividida em curvas menores de maneira ideal é mais importante que sua capacidade de representar o contorno da distribuição dos dados. Isso ocorre porque em

alguns casos, a interseção entre diferentes agrupamentos contém densidade de dados suficiente para que o algoritmo k-segmentos construa linhas de interconexão menores ou mesmo segmentos onde se projetam esses dados. Isso faz com que a divisão do espaço em agrupamentos seja feita de maneira equivocada. Por outro lado, para esse mesmo tipo de distribuição de dados, quando a curva é construída com baixa quantidade de segmentos, esses se concentram mais próximos aos aglomerados de dados, fazendo que as linhas de interconexão estejam, de fato, no espaço entre os agrupamentos.

Tal fato se explica, ainda, pelo motivo do primeiro passo do algoritmo ser a construção da curva para todos os dados, fazendo com que a representação do contorno dos dados como um todo não seja dividida de maneira ideal em curvas que representam os agrupamentos de maneira correta. Dessa maneira, a distribuição dos eventos de cada base de dados no espaço multidimensional tem influência direta na ponderação dos termos da função-custo a ser utilizada na otimização dos parâmetros do algoritmo k-segmentos.

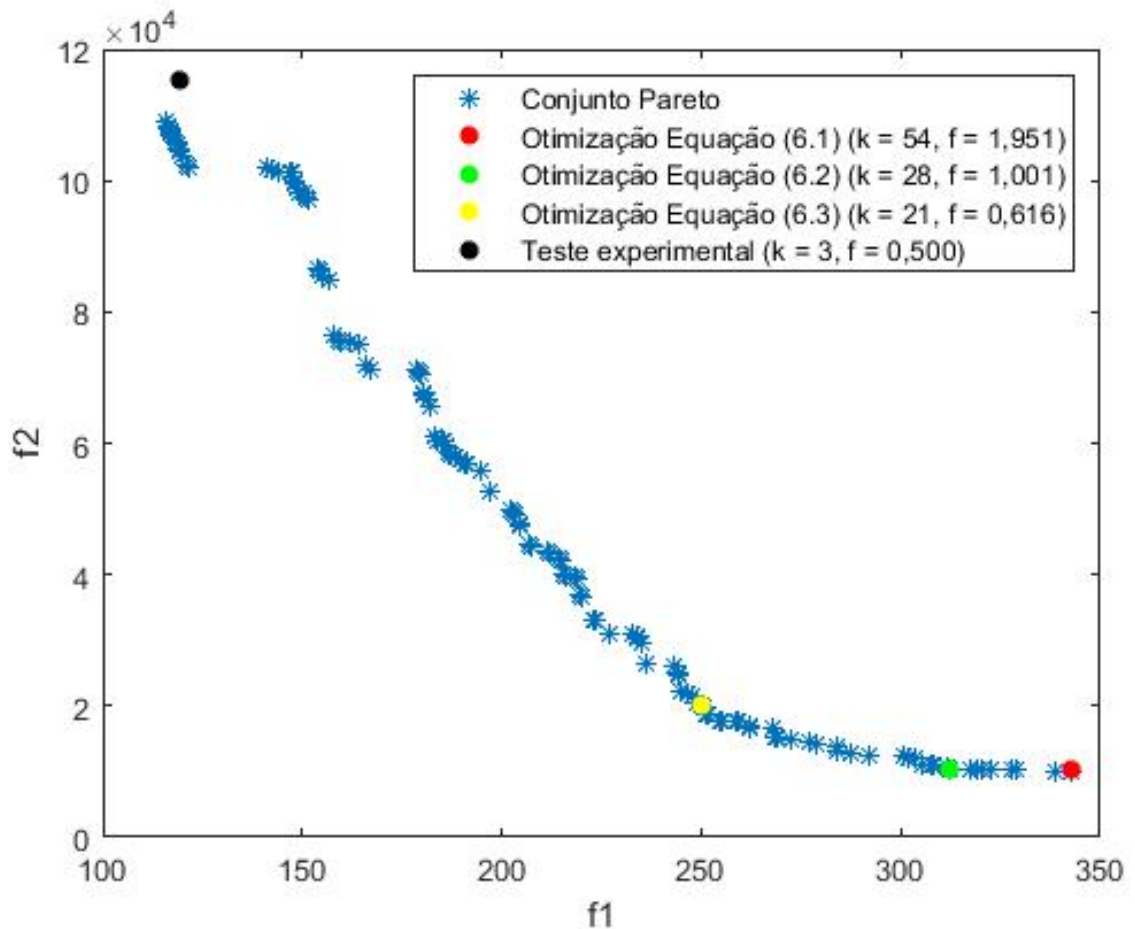
Ao se observar a Tabela 6.2, percebe-se que para a base *Wine* não se tem boa acurácia de clusterização para nenhuma das duas funções-custo utilizadas. Com o objetivo de observar o comportamento do método de clusterização proposto, para a base *Wine*, com a utilização de diferentes ponderações, fez-se uso da otimização multiobjetivo, tal como a realizada na Seção 4.2. O conjunto Pareto obtido nesse teste pode ser visto na Figura 6.3.

Por meio de testes experimentais, observou-se que para essa base de dados, quando se tomam os parâmetros $k = 3$ e $f = 0,5$, fora da fronteira de Pareto, é possível se obter uma acurácia de 93,82%, valor expressivamente maior do que os resultados obtidos com a utilização das outras funções-custo, apresentadas na Tabela 6.2. Por outro lado, quando se tomam os pares Pareto da otimização multiobjetivo, observa-se que um deles, destacado na Figura 6.3, cujos parâmetros são dados por $k = 21$ e $f = 0,616$, apresentam acurácia de 95,51%, superando os valores obtidos pelo *k-means*. Esse ponto do Conjunto Pareto é relativo às ponderações de 85% e 15% dadas, respectivamente, aos termos relativos ao comprimento total da CP e distância dos eventos à curva, ou seja, pela minimização da equação:

$$0,85n \log l + 0,15 \sum_{i=1}^k \sum_{x \in V_i} d(s_i, \mathbf{x})^2 / (2\sigma^2). \quad (6.3)$$

Se por um lado, a maior importância dada à minimização do termo relativo ao comprimento total da curva traz bons resultados para a base de dados *Wine*, por outro, para a base *Spiral*, os melhores resultados são alcançados quando se tem como ponderações da função-custo

Figura 6.3 – Fronteira de Pareto para base de dados Wine.



Fonte: Acervo do Autor

10% e 90% dadas, respectivamente, aos termos relativos ao comprimento total da CP e distância dos eventos à curva, com a qual se obtém 100% de acerto, com valores de $k = 24$ e $f = 0,508$, como observado na Tabela 6.2. Percebe-se, então, que para essas bases de dados os melhores resultados são obtidos com a utilização de funções-custo com ponderações contrárias, o que demonstra que a função-custo a ser utilizada depende da distribuição espacial dos dados em seu espaço multidimensional, destacando que esse método de clusterização, ainda sem parâmetros otimizados, apresenta, segundo Moraes (2015) e Moraes e Ferreira (2016), melhores resultados para agrupamentos com distribuições alongadas e circulares no espaço de parâmetros.

O fato da ponderação depender do problema tratado demonstra a importância da otimização multiobjetivo, pois como ocorre com a base de dados *Wine*, provavelmente com a utilização de um dos membros do conjunto Pareto é possível modelar um classificador não supervisionado que seja capaz de obter resultados melhores que aqueles apresentados pelo *k-means*.

A utilização da otimização multiobjetivo traz como desafio a escolha de qual elemento do conjunto Pareto deve ser utilizado para a obtenção do melhor classificador. Desse modo, deve-se definir o decisor de acordo com o problema tratado e suas particularidades. Para as bases de dados utilizadas neste capítulo, observa-se que, de maneira específica, uma das 3 funções-custo apresentadas é capaz de definir os parâmetros do algoritmo k-segmentos que determina um classificador não supervisionado, para o qual os resultados obtidos são satisfatórios.

Desse modo, não se pode descartar a importância de se otimizar os parâmetros do algoritmo k-segmentos para clusterização, seja essa otimização feita por uma meta-heurística mono-objetivo ou multiobjetivo. Destaca-se, ainda, que, como pode ser observado na Tabela 6.3, para nenhuma das bases de dados utilizadas, o valor definido para o comprimento dos segmentos é igual a $3\sigma/2$, como o valor fixado no algoritmo k-segmentos original, proposto por Verbeek, Vlassis e Kröse (2002). Ainda, de modo geral, o método aqui utilizado mostra-se uma boa alternativa à definição manual dos parâmetros do algoritmo k-segmentos para fins de clusterização.

7 CONSIDERAÇÕES FINAIS E CONCLUSÃO

Como pôde ser observado neste trabalho, os parâmetros do algoritmo k-segmentos impactam de maneira significativa as Curvas Principais por ele construídas, de modo a afetar os resultados de suas aplicações, seja para representação, classificação supervisionada ou não supervisionada. O método de seleção automática de parâmetros iniciais do algoritmo k-segmentos com a utilização de meta-heurísticas proposto neste trabalho apresenta resultados satisfatórios nas diversas aplicações em que as CP podem ser utilizadas. Observou-se que, diferentemente do que afirmam Verbeek, Vlassis e Kröse (2002), o comprimento ideal dos segmentos não é igual a $3\sigma/2$, de modo que esse parâmetro varia de acordo com a aplicação e base de dados em que se utiliza o algoritmo k-segmentos.

Dentre as meta-heurísticas mono-objetivo utilizadas, a definição pela utilização da TLBO ocorreu devido à simplicidade de sua utilização, uma vez que seu uso não depende da definição de parâmetros pelo usuário, além de sua rápida convergência. A escolha de outra meta-heurística, por outro lado, não causaria prejuízos ao método proposto, como pôde ser visto no Capítulo 4. É importante destacar que a eficiência da otimização dos parâmetros do algoritmo k-segmentos deve-se diretamente à função-custo utilizada nesse processo. Por esse motivo, a definição de tal função foi apresentada de maneira detalhada no Capítulo 3.

Uma vez comprovada a capacidade do método mono-objetivo proposto em construir curvas com boa capacidade de representação, mostrou-se que sua potencialidade na aplicação como classificador supervisionado é alta. Não deve ser descartada, porém, a possibilidade de melhoria de tal método com a utilização de outras funções objetivo, tais como a variação das ponderações da equação cujos termos definem o comprimento total da curva e a distância dos eventos à ela, ou a modificação da função-custo que tem como objetivo a diminuição dos erros de classificação nas bases de validação, ou mesmo a inserção de dados artificiais gerados com o acréscimo de ruído aos eventos reais, de modo a aumentar a quantidade de dados, tornando possível a construção de curvas mais robustas. Quanto à primeira função, a utilização de meta-heurísticas multiobjetivo mostra-se uma alternativa robusta da definição de melhores ponderações, uma vez que a fronteira de Pareto apresenta grande quantidade de parâmetros ótimos para diferentes ponderações, como pode ser visto na Seção 4.2, quando tal algoritmo é utilizado na representação, e na Seção 6.2, quando sua potencialidade é observada na classificação não supervisionada de dados.

Se por um lado, a utilização da mesma função-custo para representação e classificação supervisionada retorna resultados satisfatórios, para clusterização de dados não se tem os mesmos resultados, uma vez que para tal finalidade não basta que a curva represente o contorno dos dados como um todo de maneira correta. É necessário, ainda, que a CP construída possa ser dividida em curvas menores, cada uma representando um dos agrupamentos. Isso só é feito de maneira ideal quando as maiores linhas de interconexão da curva estão localizadas entre os agrupamentos de dados. A definição de uma função-custo, que ao ser minimizada retorna parâmetros capazes de determinar curvas que cumpram tais requisitos, depende diretamente da distribuição dos dados. Por esse motivo, a utilização de meta-heurísticas multi-objetivos se apresenta como uma alternativa robusta na definição de tais parâmetros.

Nesse caso, passa a ser necessário definir um decisor, capaz de determinar qual dos elementos do conjunto Pareto deve ser utilizado na definição dos parâmetros de construção da CP pelo algoritmo k-segmentos para clusterização de dados. Para isso faz-se necessário utilizar alguma medida de desempenho dos *cluster* gerados, tal como *silhouettes*, método de interpretação e validação de consistência dentro de aglomerados de dados. Essa definição apresenta-se como próxima etapa da pesquisa, assim como a modelagem de um método automático da definição da quantidade de *clusters* de cada base de dados e a atualização dos agrupamentos em tempo real, à medida que novos dados são apresentados ao algoritmo.

A grande aplicabilidade do algoritmo k-segmentos pode ser observada nos diversos exemplos apresentados neste trabalho, não restringindo-se, porém, somente a esses. Por essa razão, os resultados obtidos durante a pesquisa são considerados satisfatórios por apresentarem robustez na definição dos parâmetros desse algoritmo, para as diferentes aplicações em que é utilizado.

REFERÊNCIAS

- BAI, Q.; ZHU, X. Features of off-line handwritten digit recognition based on principal curves. In: **Symposium on ICT and Energy Efficiency and Workshop on Information Theory and Security (CIICT 2012)**. [S.l.: s.n.], 2012. p. 209–214.
- BANFIELD, J. D.; RAFTERY, A. E. Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. **Journal of the American Statistical Association**, [American Statistical Association, Taylor Francis, Ltd.], v. 87, n. 417, p. 7–16, 1992.
- BIAU, G.; FISCHER, A. Parameter selection for principal curves. **IEEE Transactions on Information Theory**, v. 58, n. 3, p. 1924–1939, March 2012.
- CHANG, K.; GHOSH, J. Principal curve classifier-a nonlinear approach to pattern classification. In: **1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)**. [S.l.: s.n.], 1998. v. 1, p. 695–700 vol.1.
- CLEJU, I.; FRANTI, P.; WU, X. Clustering by principal curve with tree structure. In: **International Symposium on Signals, Circuits and Systems, 2005. ISSCS 2005**. [S.l.: s.n.], 2005. v. 2, p. 617–620 Vol. 2.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 2, p. 182–197, April 2002.
- DELICADO, P.; SMREKAR, M. Mixture of nonlinear models: a bayesian fit for principal curves. In: **2007 International Joint Conference on Neural Networks**. [S.l.: s.n.], 2007. p. 195–200.
- DUA, D.; GRAFF, C. **UCI Machine Learning Repository**. 2017. Disponível em: <<http://archive.ics.uci.edu/ml>>.
- EIBEN, A.; SMITH, J. **Introduction To Evolutionary Computing**. [S.l.: s.n.], 2003. v. 45.
- FAIER, J. M. **Curvas principais aplicadas na identificação de descargas parciais em equipamentos de potência**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, 2006.
- FERNANDES, T. S. P.; Souza Filho, J. B. de Oliveira e; ALVES, E. dos S. Identificação de navios em sinais de sonar passivo por curvas principais. In: Bastos Filho, C. J. A.; POZO, A. R.; LOPES, H. S. (Ed.). **Anais do 12 Congresso Brasileiro de Inteligência Computacional**. Curitiba, PR: ABRICOM, 2015. p. 1–6.
- FERNANDEZ, H. L. **Classificação de Navios Baseada em Curvas Principais**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, 2005.
- FERREIRA, D. D. et al. Exploiting principal curves for power quality monitoring. **Electric Power Systems Research**, v. 100, p. 1 – 6, 2013.
- FERREIRA, D. D. et al. A new power quality deviation index based on principal curves. **Electric Power Systems Research**, v. 125, p. 8 – 14, 2015.

FERREIRA, D. D. et al. A direct approach for disturbance detection based on principal curves. In: **2014 16th International Conference on Harmonics and Quality of Power (ICHQP)**. [S.l.: s.n.], 2014. p. 747–751.

GARCIA, C. M. et al. Teaching-learning-based optimization no treinamento de redes neurais artificiais para problemas de classificação. 2017.

GHASSABEH, Y. A.; RUDZICZ, F. Incremental algorithm for finding principal curves. **IET Signal Processing**, v. 9, n. 7, p. 521–528, 2015.

HASTIE, T.; STUETZLE, W. Principal curves. **Journal of the American Statistical Association**, [American Statistical Association, Taylor Francis, Ltd.], v. 84, n. 406, p. 502–516, 1989.

HAUBERG, S. Principal curves on riemannian manifolds. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 38, n. 9, p. 1915–1921, Sept 2016.

KEGL, B. et al. Learning and design of principal curves. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 22, n. 3, p. 281–297, Mar 2000.

KOHLI, M.; ARORA, S. Chaotic grey wolf optimization algorithm for constrained optimization problems. **Journal of Computational Design and Engineering**, v. 5, n. 4, p. 458 – 472, 2018.

LEE, T. C. M.; WANG, H. On a k-segments algorithm for computing principal curves. In: **2006 IEEE Southwest Symposium on Image Analysis and Interpretation**. [S.l.: s.n.], 2006. p. 183–187.

MA, C. et al. Fingerprint skeleton extraction based on improved principal curve. In: **2009 Eighth IEEE/ACIS International Conference on Computer and Information Science**. [S.l.: s.n.], 2009. p. 605–610.

MIRJALILI, S.; MIRJALILI, S. M.; LEWIS, A. Grey wolf optimizer. **Advances in Engineering Software**, v. 69, p. 46 – 61, 2014.

MORAES, E. C. C. **Método não supervisionado baseado em curvas principais para reconhecimento de padrões**. Dissertação (Mestrado) — Universidade Federal de Lavras, 2015.

MORAES, E. C. C.; FERREIRA, D. D. A principal curve-based method for data clustering. In: **2016 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2016. p. 3966–3971.

MORAES, E. C. C. et al. Data clustering based on principal curves. **Advances in Data Analysis and Classification**, Jun 2019. ISSN 1862-5355. Disponível em: <<https://doi.org/10.1007/s11634-019-00363-w>>.

OLIVEIRA, T. C. C.; FILHO, J. B. d. O. S. Otimização de classificadores de sinais de sonar passivo baseado em curvas principais hierárquicas. In: **Anais do XIII Simpósio Brasileiro de Automação Inteligente**. Porto Alegre, RS: [s.n.], 2017. p. 569–574.

OZERTEM, U.; ERDOGMUS, D. Locally defined principal curves and surfaces. **J. Mach. Learn. Res.**, JMLR.org, v. 12, p. 1249–1286, jul. 2011. ISSN 1532-4435.

- RAO, R.; SAVSANI, V.; VAKHARIA, D. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. **Computer-Aided Design**, v. 43, n. 3, p. 303 – 315, 2011.
- RAO, R. V.; PATEL, V. An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. **Scientia Iranica**, v. 20, n. 3, p. 710 – 720, 2013.
- SHI, J.; LUO, Z. Missing value estimation for dna microarray gene expression data with principal curves. In: **2010 International Conference on Bioinformatics and Biomedical Technology**. [S.l.: s.n.], 2010. p. 262–265.
- STORN, R.; PRICE, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, Dec 1997.
- THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition**. Fourth edition. [S.l.]: Elsevier, 2009.
- VERBEEK, J.; VLASSIS, N.; KRÖSE, B. A k-segments algorithm for finding principal curves. **Pattern Recognition Letters**, v. 23, n. 8, p. 1009 – 1017, 2002.
- VERBEEK, J. J.; VLASSIS, N.; KRÖSE, B. A soft k-segments algorithm for principal curves. In: **Artificial Neural Networks — ICANN 2001**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. p. 450–456.
- WANG, H.; LEE, T. C. Automatic parameter selection for a k-segments algorithm for computing principal curves. **Pattern Recognition Letters**, v. 27, n. 10, p. 1142 – 1150, 2006.
- WANG, Y. et al. An algorithm for learning principal curves with explicit formulation based on neural network. In: **2008 Congress on Image and Signal Processing**. [S.l.: s.n.], 2008. v. 1, p. 399–404.
- WANG, Y. H. et al. An algorithm for learning principal curves with principal component analysis and back-propagation network. In: **Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)**. [S.l.: s.n.], 2007. p. 447–453.
- WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. **IEEE Transactions on Evolutionary Computation**, v. 1, n. 1, p. 67–82, April 1997.
- YOU, S. et al. Principal curved based retinal vessel segmentation towards diagnosis of retinal diseases. In: **2011 IEEE First International Conference on Healthcare Informatics, Imaging and Systems Biology**. [S.l.: s.n.], 2011. p. 331–337.
- YU, K.; WANG, X.; WANG, Z. An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems. **Journal of Intelligent Manufacturing**, v. 27, n. 4, p. 831–843, Aug 2016.
- YUSOFF, Y.; NGADIMAN, M. S.; ZAIN, A. M. Overview of nsga-ii for optimizing machining process parameters. **Procedia Engineering**, v. 15, p. 3978 – 3983, 2011. CEIS 2011.
- ZHANG, H.; MIAO, D.; PEDRYCZ, W. Fuzzy granular principal curves algorithm for large data sets. In: **2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)**. [S.l.: s.n.], 2013. p. 956–961.

ZHANG, H. et al. From principal curves to granular principal curves. **IEEE Transactions on Cybernetics**, v. 44, n. 6, p. 748–760, June 2014.

ZHANG, J.; CHEN, D.; KRUGER, U. Adaptive constraint k-segment principal curves for intelligent transportation systems. **IEEE Transactions on Intelligent Transportation Systems**, v. 9, n. 4, p. 666–677, Dec 2008.

ZHANG, J. et al. Principal curve algorithms for partitioning high-dimensional data spaces. **IEEE Transactions on Neural Networks**, v. 22, n. 3, p. 367–380, March 2011.

ZHANG, L.; LUO, Z. Adaptive smooth principal curves design. In: **International Conference on Computer Graphics, Imaging and Visualization (CGIV'05)**. [S.l.: s.n.], 2005. p. 241–244.

APÊNDICE A – OTIMIZAÇÃO DOS PARÂMETROS DO ALGORITMO K-SEGMENTOS SUAVE

Algoritmo k-segmentos suave

Proposto por Verbeek, Vlassis e Kröse (2001), o algoritmo k-segmentos suave apresenta diferenças em relação ao algoritmo k-segmentos não suave. Diferentemente deste último algoritmo, no qual se utiliza particionamento rígido, o k-segmentos suave se utiliza de um esquema de ponderação suave para determinar quanto um ponto contribui para a configuração de um segmento de linha.

Dessa maneira, a construção dos segmentos é feita como um problema de estimativa de densidade de mistura. As atualizações do comprimento dos segmentos é feita heurísticamente, de modo que o centro dos segmentos seja igual ao centro da Gaussiana. A direção é tomada ao longo do autovetor com maior autovalor, λ_{max} , da matriz de covariância da Gaussiana. Experimentalmente definiu-se o comprimento do segmento como $3\sqrt{\lambda_{max}}$. É importante destacar que neste trabalho o comprimento dos segmentos é tomado como um parâmetro inicial a ser otimizado, não sendo mantido fixo como no algoritmo original.

Assim como no algoritmo k-segmentos não suave o parâmetro λ determina a suavidade de ligação entre os segmentos, e, segundo Verbeek, Vlassis e Kröse (2001), bons resultados são obtidos quando tal parâmetro é mantido fixado em 1, como é feito nesse trabalho.

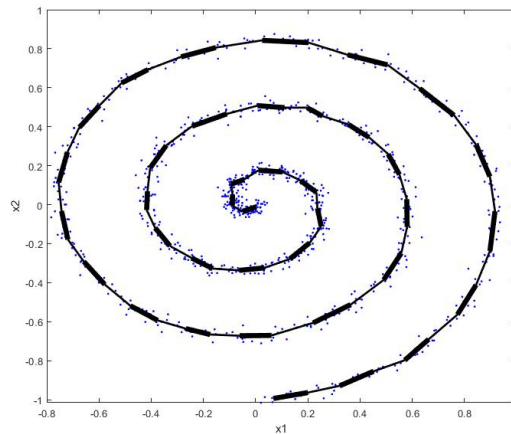
Assim como no algoritmo k-segmentos não suave, a inserção de segmentos é feita de maneira gradual até o número máximo k de segmentos definidos pelo usuário. Segundo Verbeek, Vlassis e Kröse (2001) o esquema de particionamento utilizado neste algoritmo garante maior robustez, fazendo-o menos sensível a detalhes de configuração da base de dados, quando comparado ao particionamento em regiões de Voronoi do algoritmo k-segmentos não suave.

Apesar das vantagens do algoritmo k-segmentos suave, o método proposto por este trabalho é analisado e aplicado com a utilização do algoritmo k-segmentos não suave devido à sua ampla utilização nas bibliografias do meio. Por outro lado, a robustez do método proposto garante a capacidade de otimização dos parâmetros iniciais do algoritmo k-segmentos suave, sem a necessidade de adaptações à meta-heurística utilizada e tampouco à função-custo do mesmo. Os resultados dessa otimização são apresentados a seguir.

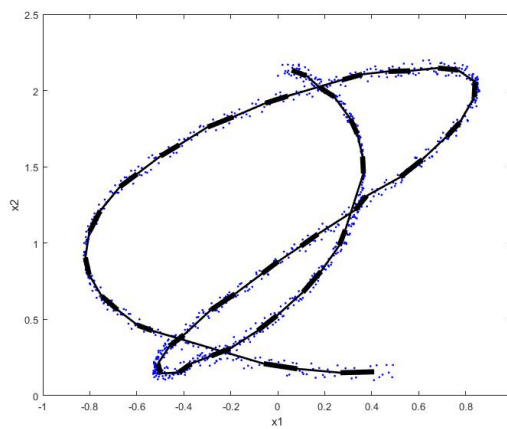
Definição dos Parâmetros Iniciais do algoritmo k-segmentos suave

Os resultados da otimização dos parâmetros iniciais do algoritmo k-segmentos suave podem ser observados na Figura 1.

Figura 1 – Curvas geradas a partir da minimização da Equação (3.4) com a utilização do algoritmo k-segmentos suave.



(a) $k = 38$; $f = 0,914363$



(b) $k = 31$; $f = 0.770581$

Fonte: Acervo do Autor

A curva construída com os parâmetros otimizados pelo método proposto são capazes de representar adequadamente o contorno dos dados. Para os dados em espiral, apresentados na Figura 1(a) é possível observar suavidade nos contornos, com resultados similares aos obtidos pelo algoritmo k-segmentos não suave.

Na Figura 1(b) é possível observar que as ligações entre os segmentos que compõem a curva são feitos sem equívocos. Desse modo, a representação do contorno de dados é feita de maneira satisfatória.

Apesar do método proposto por esse trabalho ser aplicado e testado para o algoritmo k-segmentos não suave, é importante observar a efetividade do mesmo para o algoritmo k-segmentos suave, sem que haja a necessidade de mudanças no modelo proposto. Isso garante a possibilidade de sua utilização, como feita por Bai e Zhu (2012) ou em eventuais casos para os quais esse último algoritmo de construção de CP se mostre mais adequado.

APÊNDICE B – ARTIGO PUBLICADO EM ANAIS DE CONGRESSO

Braga, A.A., Ferreira, D.D., Barbosa, B.H.G. Seleção automática de parâmetros iniciais do algoritmo k-segmentos com teaching-learning-based optimization. **XXII Congresso Brasileiro de Automática**, João Pessoa-PB, Setembro 2018.

Sendo uma generalização não linear de análise de componentes principais, a técnica de curvas principais é uma ferramenta robusta para análise e classificação de dados. Em reconhecimento de padrões, um dos algoritmos mais populares para construção de Curvas Principais é o algoritmo k-segmentos. Esse algoritmo apresenta bons resultados e ótima aplicabilidade por sua convergência garantida e robustez. Contudo, sua utilização depende de parâmetros definidos pelo usuário. Este trabalho apresenta uma técnica de seleção automática da quantidade e comprimento dos segmentos do algoritmo k-segmentos com a utilização da meta-heurística TLBO (Teaching-Learning-Based Optimization). É utilizada uma função-custo que leva em consideração o comprimento da curva e a distância dos eventos aos segmentos onde se projetam. Testes experimentais feitos com bases de dados sintéticos são apresentados para demonstrar a eficiência do método proposto em problemas de representação.