A FEDERAL DE ENGENHARIA DE ITAJUBÁ

Determinação do Número de Agrupamentos em Conjuntos de Dados Multidimensionais Utilizando Algoritmos Genéticos

SANDRO CARVALHO IZIDORO

ITAJUBA
2001

ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ
Programa de Pós-Graduação em Engenharia Elétrica

DETERMINAÇÃO DO NÚMERO DE AGRUPAMENTOS EM CONJUNTOS DE DADOS MULTIDIMENSIONAIS UTILIZANDO ALGORITMOS GENÉTICOS

Dissertação apresentada à Escola Federal de Engenharia de Itajubá, para obtenção do título de Mestre em Engenharia Elétrica.

Sandro Carvalho Izidoro

Itajubá

2001

Sandro Carvalho Izidoro

DETERMINAÇÃO DO NÚMERO DE AGRUPAMENTOS EM CONJUNTOS DE DADOS MULTIDIMENSIONAIS UTILIZANDO ALGORITMOS GENÉTICOS

Dissertação apresentada à Escola Federal de Engenharia de Itajubá, para obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração:
Automação e Sistemas Elétricos Industriais

Orientador:
Germano Lambert Torres

Co-orientador:
Luiz Eduardo Borges da Silva

Itajubá

Dezembro/2001

À minha família pelo incentivo e carinho.

Agradecimentos

A Deus.

Ao professor Germano Lambert Torres pela orientação e amizade.

Ao professor doutor João Onofre Pereira Pinto pela ajuda, orientação e amizade.

Ao professor Daniel Furtado Ferreira pela colaboração e amizade.

A Márcia Cristina Fráguas pelo amor, carinho e paciência.

A todos, que direta ou indiretamente, contribuíram para a realização deste trabalho.

Sumário

Resumo

A análise de agrupamentos tem sido utilizada com sucesso nas mais diversas áreas de pesquisa com o objetivo de agrupar dados semelhantes segundo suas características. Vários métodos existentes efetuam o agrupamento dos dados baseados em um número aproximado. Uma técnica eficiente na análise de agrupamentos é a utilização da função de densidade de probabilidade que apresenta o número de agrupamentos graficamente.

Os algoritmos genéticos são algoritmos de busca e têm se mostrado muito eficientes para a busca de soluções ótimas ou aproximadamente ótimas. Os algoritmos genéticos foram utilizados com sucesso para informar o número de agrupamentos em um conjunto de dados unidimensional.

Os métodos existentes para a análise de agrupamentos em dados multidimensionais necessitam de um número aproximado de agrupamentos para localizá-los. O desempenho destes métodos dependem diretamente deste número de agrupamentos. O propósito deste trabalho é utilizar os algoritmos genéticos para predizer o número de agrupamentos em dados multidimensionais.

Para tal, foi implementado um algoritmo genético utilizando o estimador kernel multivariado como função de aptidão. Utilizando um número pequeno de gerações o algoritmo genético não encontra apenas um máximo global, mas sim vários máximos locais e a quantidade destes máximos indica o número de agrupamentos existentes no conjunto de dados analisado.

Abstract

The cluster analysis has been used with success in the most several research areas with the objective to group similar data according to its characteristics. Several existent methods make the grouping of the data based on an approximate number. An efficient technique in the cluster analysis is the use of the function of density of probability that presents the clusters number graphically.

The genetic algorithms are search algorithms and have been showing very efficient for the search of solutions excellent or approximately excellent. The genetic algorithms were used with success to inform the number of clusters in a group of data unidimensional.

The existent methods for the cluster analysis in data multidimensional need an approximate number of groupings to locate them. The acting of these methods depends directly on this number of groupings. The purpose of this work is to use the genetic algorithms to predict the number of groupings in data multidimensional.

For such, a genetic algorithm was implemented using the kernel estimator multivariate as fitness function. Using a small number of generations the genetic algorithm doesn't just find a global maximum but several local maximum and the amount of these maximum indicates the number of existent clusters in the group of data analyzed.

Lista de Figuras

Lista de Gráficos

Lista de Tabelas

CAPÍTULO 1

INTRODUÇÃO

A análise de agrupamentos (*"clusters"*) vem sendo utilizada com sucesso em várias áreas de pesquisa tais como na arqueologia e na biologia, e seu objetivo está em juntar dados semelhantes segundo suas características, gerando classes. Este tipo de análise tem se demonstrado muito útil no reconhecimento de caracteres, símbolos, figuras, imagens biomédicas, eletrocardiogramas, ondas sísmicas e ondas sonoras.

Existem vários métodos (*hierárquicos* e *não hierárquicos*) para efetuar o agrupamento dos dados, mas para todos os casos existe a necessidade de saber aproximadamente o número de agrupamentos existentes no conjunto de dados analisado. De posse deste número aproximado é que os métodos vão começar a agrupar os dados semelhantes (RUSSEL, 1995).

Uma técnica muito eficiente na análise de agrupamentos é a utilização da função de densidade de probabilidade. Após a estimação da densidade dos dados é possível apresentá-los graficamente. Os picos (*peaks*) desta função representam o número de agrupamentos (PINTO, 1998).

Os algoritmos genéticos são algoritmos de busca baseados em mecanismos da seleção natural e genética. Sem as limitações encontradas nos métodos tradicionais, os algoritmos genéticos se mostram muito eficientes para busca de soluções ótimas, ou aproximadamente ótimas, em uma grande variedade de problemas (MENDES FILHO, 1998).

Os algoritmos genéticos foram utilizados com sucesso para predizer o número de agrupamentos em um conjunto de dados unidimensional (PINTO, 1998). Nenhuma informação foi encontrada na literatura para uma extrapolação para casos bidimensionais e multidimensionais.

O objetivo deste trabalho é o de avaliar a eficiência dos algoritmos genéticos na busca de soluções ótimas para determinar o número de agrupamentos em dados bidimensionais e multidimensionais.

Este trabalho está organizado da seguinte maneira:

- Capítulo 2 – São apresentados os fundamentos teóricos de Análise de Agrupamentos e Estimação de Densidade;
- Capítulo 3 – São apresentados os fundamentos teóricos de Algoritmos Genéticos;
- Capítulo 4 – Descreve a técnica de Análise de Agrupamentos utilizando Algoritmos Genéticos para dados unidimensionais;
- Capítulo 5 – Apresenta a técnica, a implementação e vários testes da utilização dos Algoritmos Genéticos na Análise de Agrupamentos para dados bidimensionais e multidimensionais;
- Capítulo 6 – Apresenta as conclusões e desenvolvimentos futuros que poderão orientar novas pesquisas e dar continuidade aos resultados obtidos neste trabalho.

CAPÍTULO 2

ANÁLISE DE AGRUPAMENTOS E ESTIMAÇÃO DE DENSIDADE

2.1 Introdução

O objetivo principal da análise de agrupamentos está em dividir um determinado conjunto de dados em um número de agrupamentos (*"clusters"*) ou classes. No entanto, não existe nenhuma informação prévia sobre o conjunto de dados. Os dados, sem ajuda, definirão quantos agrupamentos existem e a que regras estarão submetidos nestes agrupamentos.

Várias técnicas de análise de agrupamentos são baseadas em achar semelhanças entre padrões dentro dos dados. Uma técnica muito eficiente é a utilização da função de densidade de probabilidade, onde é possível estimar a densidade dos dados e apresentá-los graficamente. Os picos (*peaks*) desta função representam os agrupamentos (PINTO, 1998).

2.2 Estimação de Densidade

Considerando um conjunto aleatório X em uma função de densidade de probabilidade *f*, a função irá fornecer uma descrição da distribuição desse conjunto permitindo encontrar probabilidades associadas com X a partir da equação:

$$\mathrm{P}(a < \mathrm{X} < b) = \int_a^b f(x)dx \quad \text{para todo } a < b$$

O objetivo da estimação de densidade é construir uma estimativa da função de densidade dos dados em questão uma vez que, freqüentemente, essa função é desconhecida.

Existem dois tipos de estimação de densidade: o paramétrico e o não paramétrico. O primeiro tipo considera que os dados são retirados de um conjunto conhecido, por exemplo: uma distribuição normal com média μ e variância σ^2. Portanto, a estimação de densidade pode ser feita encontrando-se a estimativa de μ e σ^2. O segundo tipo, que será abordado nesse trabalho, considera que os dados são obtidos de um conjunto que não se conhece (SILVERMAN, 1990).

2.3 Histograma

O histograma é o estimador de densidade mais simples e mais usado. A distribuição de densidade de probabilidade é construída através de barras com largura h distribuídas ao longo do intervalo onde os dados estão. O histograma é definido pela seguinte equação:

$$f = \frac{1}{nh}(\text{número de } X_i \text{ na mesma barra que } x)$$

em que,

n – número total do experimento;

h – largura do intervalo.

Por exemplo, seja imaginar um experimento cujos dados foram obtidos conforme a Tabela 2.1. Pode-se observar que no experimento o valor $x = 1$ foi encontrado 2 vezes, o valor $x = 2$ foi encontrado 5 vezes e assim sucessivamente. Agora estima-se uma função de probabilidade para esse experimento utilizando uma função constante, isto é, construir um histograma.

TABELA 2.1

Conjunto de pontos com suas respectivas freqüências

X	*Freqüência*
1	2
2	5
3	3
4	1
5	2
6	4

Para o experimento em questão tem-se $n = 17$, ou seja, 17 pontos. O parâmetro h irá definir quantas barras tem-se no histograma (*suavidade do histograma*) e pode ser definido arbitrariamente. Para simplificar o exemplo define-se o parâmetro $h = 1$. De acordo com os valores de n, h e da tabela de dados pode-se calcular a altura da barra para cada X. O resultado consta na Tabela 2.2.

TABELA 2.2

Conjunto de pontos com a altura de cada barra

X	*f(X)*
1	2/17
2	5/17
3	3/17
4	1/17
5	2/17
6	4/17

Através dos resultados da Tabela 2.2 pode-se construir o histograma (Gráfico 2.1) e observar que ele, basicamente, representa a quantidade de vezes que cada x ocorreu.

A divisão por n é necessária porque o histograma é uma função de densidade de probabilidade e, portanto sua área tem que ser igual a 1. Percebe-se também que o desenho do histograma pode ser alterado dependendo da escolha da origem e largura das barras.

O histograma fornece uma boa apresentação dos dados e é uma excelente ferramenta. No entanto, sua descontinuidade (ausência de suavidade) pode gerar dificuldades no seu entendimento, o que leva a ser necessário uma melhoria.

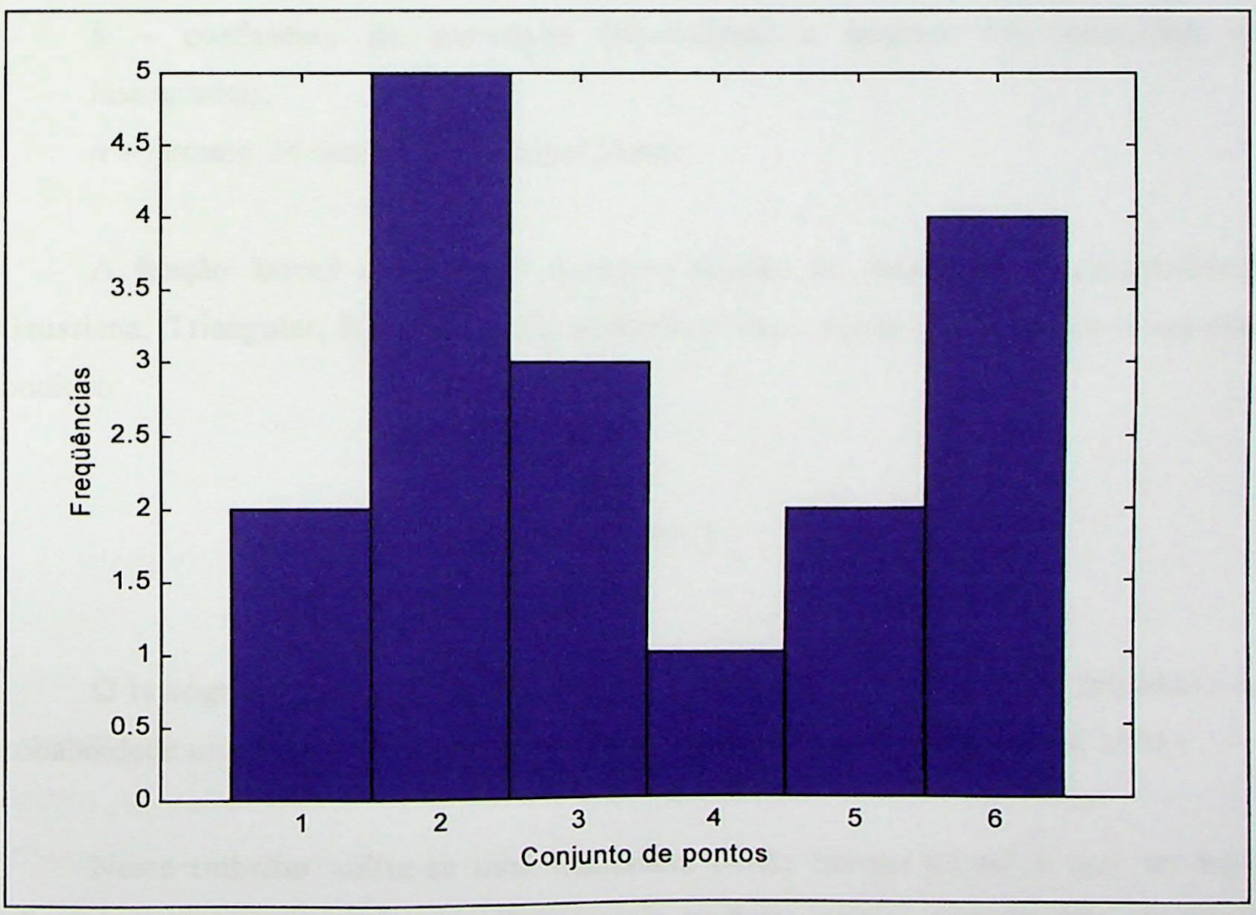

GRÁFICO 2.1 – Histograma gerado com dados da Tabela 2.2

2.4 Utilizando um Estimador Kernel

Um estimador kernel pode ser visto como uma melhoria do histograma e em vez de usar uma função constante, uma função kernel K é usada para gerar um *novo histograma*. O estimador kernel é definido por:

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - Xi}{h}\right)$$

em que,

X_i – i-ésimo ponto do conjunto de dados do experimento;

x – ponto onde será calculada a função de densidade de probabilidade;

K – uma função escolhida arbitrariamente;

h – coeficiente de suavidade (equivalente a largura dos retângulos no histograma);

n – número de resultados do experimento.

A função kernel K pode ser qualquer função de densidade de probabilidade (Gaussiana, Triangular, Retangular, Epanechnikov etc.) desde que satisfaça a seguinte condição:

$$\int_{-\infty}^{\infty} K(x)dx = 1$$

O histograma nada mais é do que a construção de uma função de densidade de probabilidade utilizando uma função retangular como função kernel (PINTO, 1998).

Neste trabalho utiliza-se uma Gaussiana como função kernel K por ser mais suave e apresentar os dados de forma mais realista, pois a maioria dos processos analisados apresenta este tipo de distribuição. A Gaussiana como função kernel é definida pela seguinte equação:

$$K(x) = \frac{1}{\sqrt{2 * \pi}} e^{\frac{x^2}{2}}$$

Quando se utiliza uma função Gaussiana como função kernel está se colocando uma pequena Gaussiana centrada em cada um dos pontos do conjunto de dados analisado. Posteriormente, soma-se todas as Gaussianas a fim de chegar na função de densidade de probabilidade de todos os pontos (Gráfico 2.2).

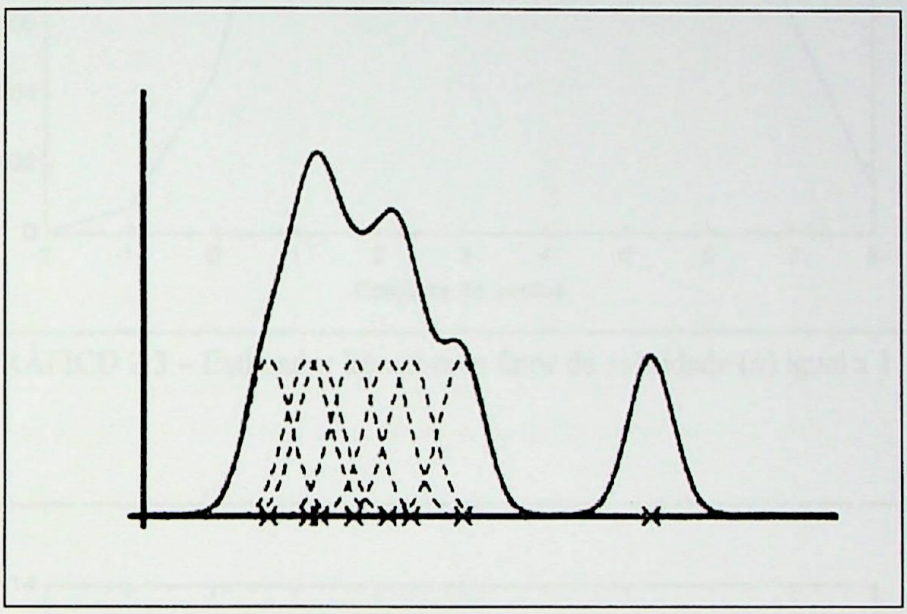

GRÁFICO 2.2 – Função de densidade de probabilidade utilizando estimador kernel
FONTE - PINTO, 1998. p.5.

Pode-se perceber que todos os dados são utilizados para calcular a função de densidade de probabilidade para um único ponto. Como exemplo, se utiliza os dados da Tabela 2.1. Para calcular o valor da função de densidade de probabilidade no ponto x = 1, necessita-se calcular a função K para todos os 17 pontos, somá-los e dividi-los pelo produto *nh*. Através dos Gráficos 2.3, 2.4 e 2.5 é possível observar o resultado do estimador utilizando uma Gaussiana como função kernel e o parâmetro h com os valores 1, 0,5 e 0,1, respectivamente.

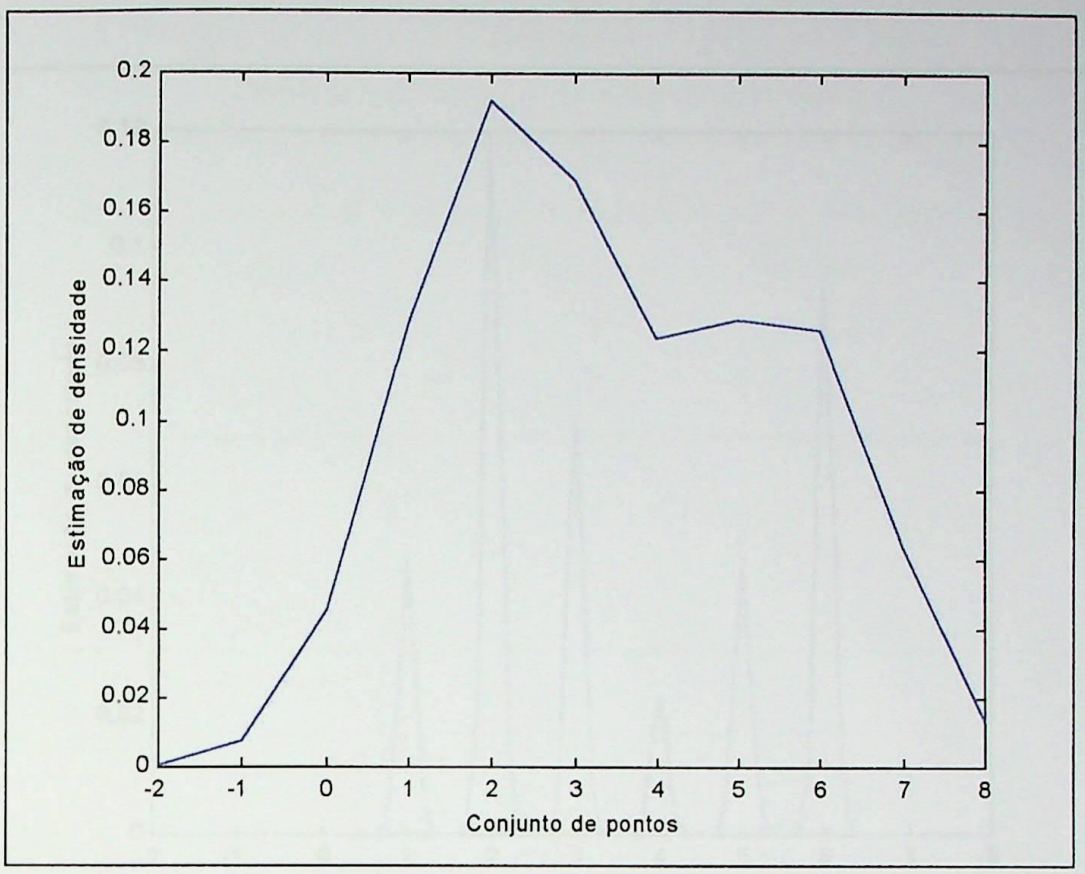

GRÁFICO 2.3 – Estimador kernel com fator de suavidade (*h*) igual a 1

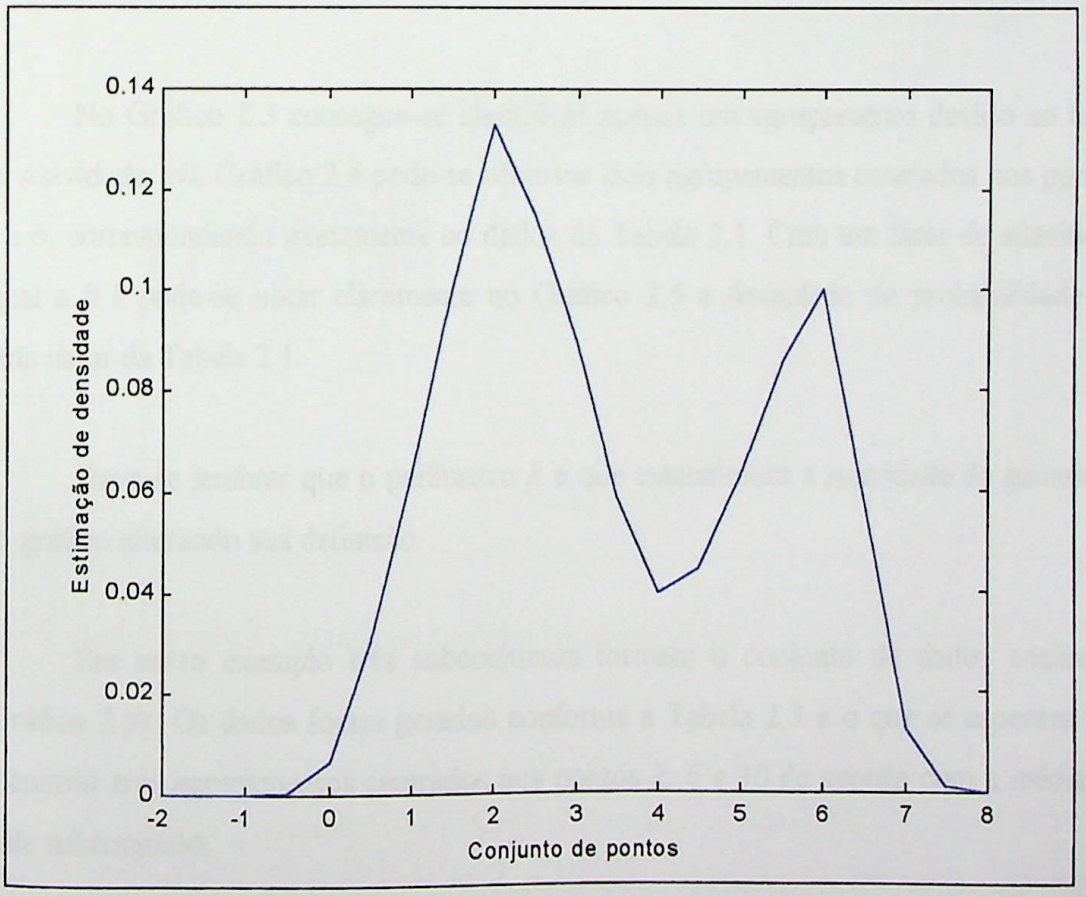

GRÁFICO 2.4 – Estimador kernel com fator de suavidade (*h*) igual a 0.5

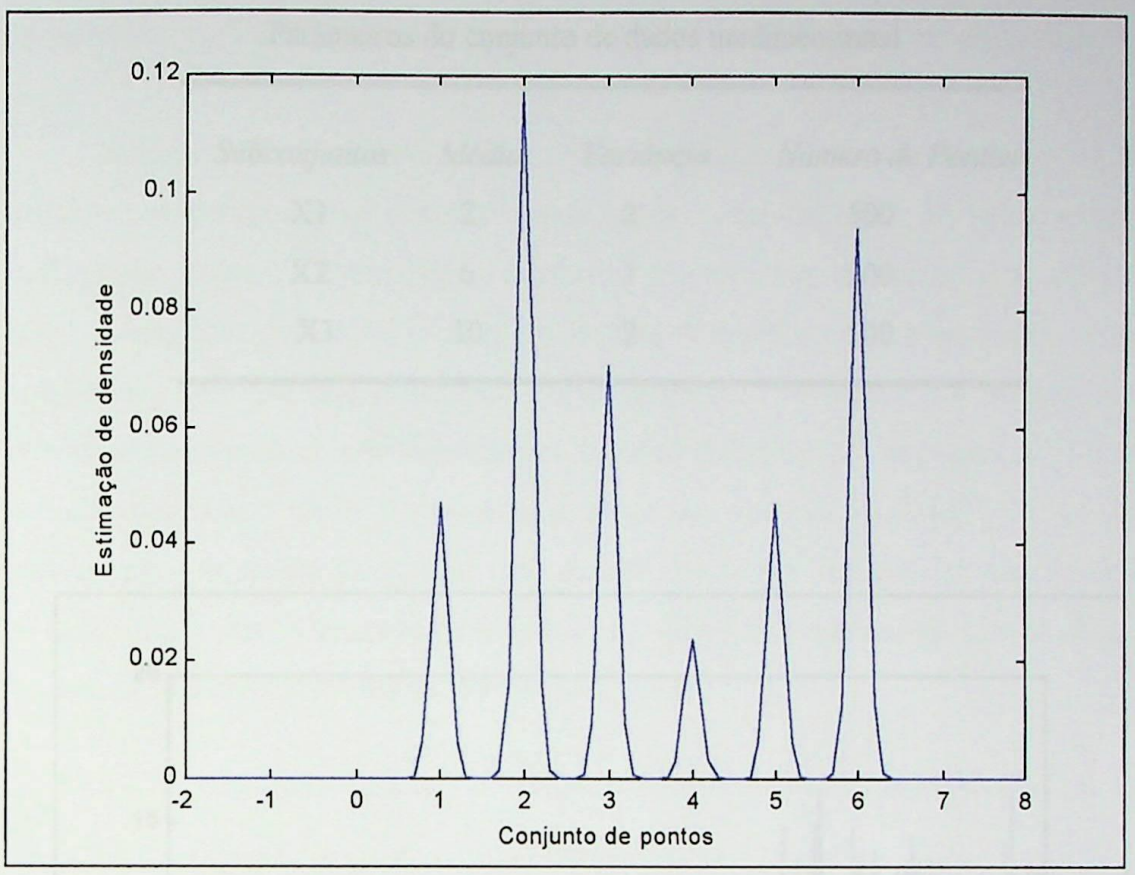

GRÁFICO 2.5 – Estimador kernel com fator de suavidade (h) igual a 0.1

No Gráfico 2.3 consegue-se identificar apenas um agrupamento devido ao fator de suavidade. No Gráfico 2.4 pode-se observar dois agrupamentos centrados nos pontos 2 e 6, correspondendo exatamente ao dados da Tabela 2.1. Com um fator de suavidade igual a 0.1 pode-se notar claramente no Gráfico 2.5 a densidade de probabilidade de cada valor da Tabela 2.1.

Deve-se lembrar que o parâmetro h é que estabelecerá a suavidade da gaussiana no gráfico alterando sua definição.

Em outro exemplo três subconjuntos formam o conjunto de dados analisado (Gráfico 2.6). Os dados foram gerados conforme a Tabela 2.3 e o que se esperava era encontrar três agrupamentos centrados nos pontos 2, 6 e 10 de acordo com a média de cada subconjunto.

TABELA 2.3

Parâmetros do conjunto de dados unidimensional

Subconjuntos	*Média*	*Variância*	*Número de Pontos*
X1	2	2	500
X2	6	1	500
X3	10	2	500

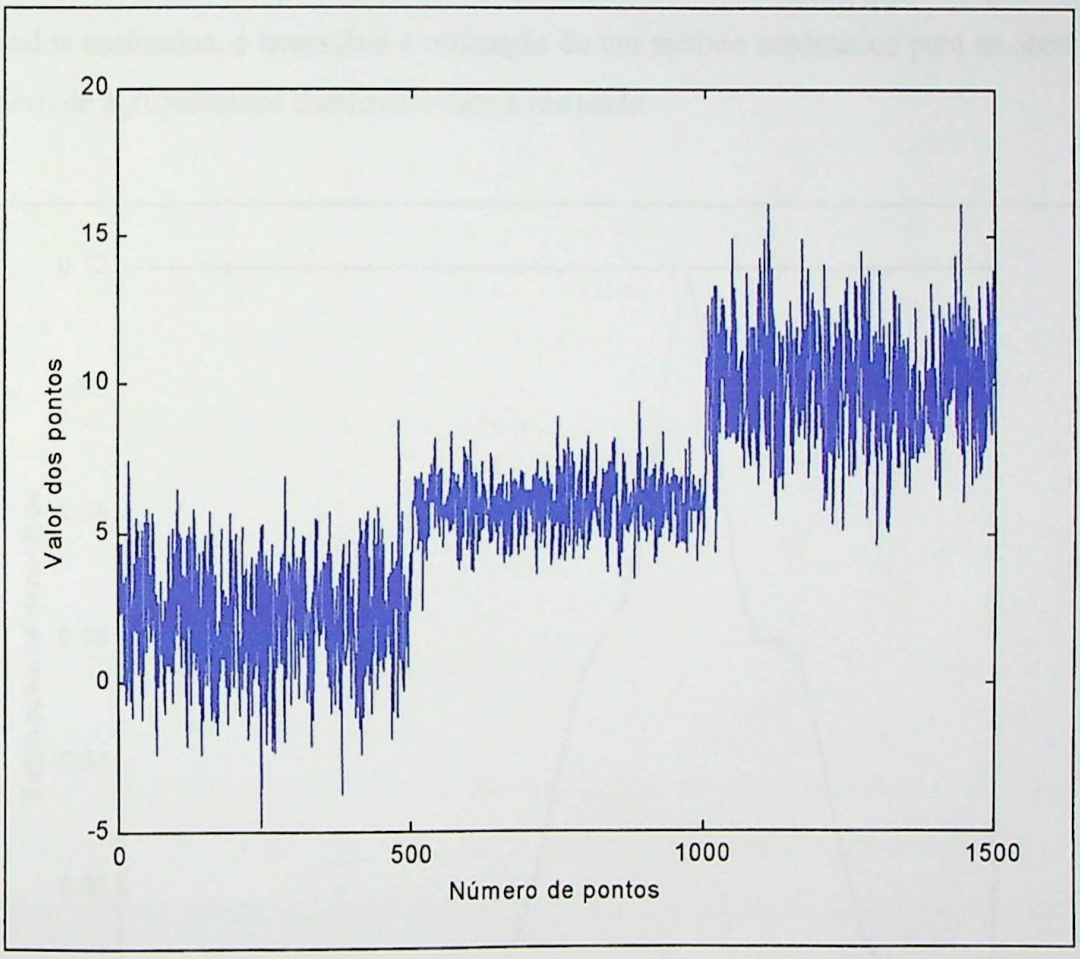

GRÁFICO 2.6 – Conjunto de dados unidimensional conforme a Tabela 2.3

A função de densidade dos dados foi realizada utilizando o estimador kernel com o parâmetro de suavidade (h) igual a 1 e 0.5. Os resultados são apresentados nos Gráfico 2.7 e 2.8 onde pode-se notar mais uma vez a importância da escolha do

parâmetro de suavidade. No Gráfico 2.7 consegue-se identificar apenas um agrupamento e no Gráfico 2.8 os três agrupamentos esperados aparecem claramente.

Existem vários métodos para escolher o parâmetro de suavidade e nenhuma unanimidade quanto a esses métodos. Nesse trabalho adota-se o *método visual* onde os gráficos são gerados e é escolhida a estimativa que está mais de acordo com as idéias sobre a densidade dos dados. Apesar de simples, esse método pode ser perfeitamente satisfatório, uma vez que, analisando vários gráficos com parâmetros de suavidade diferentes pode-se obter mais informações sobre os dados do que se consegue com um *método automático* como nos exemplos propostos em (SILVERMAN, 1990). No entanto, em uma análise de um caso real, quando não se tem nenhuma informação sobre os dados analisados, é necessária a utilização de um método automático para se obter o número de agrupamentos condizente com a realidade.

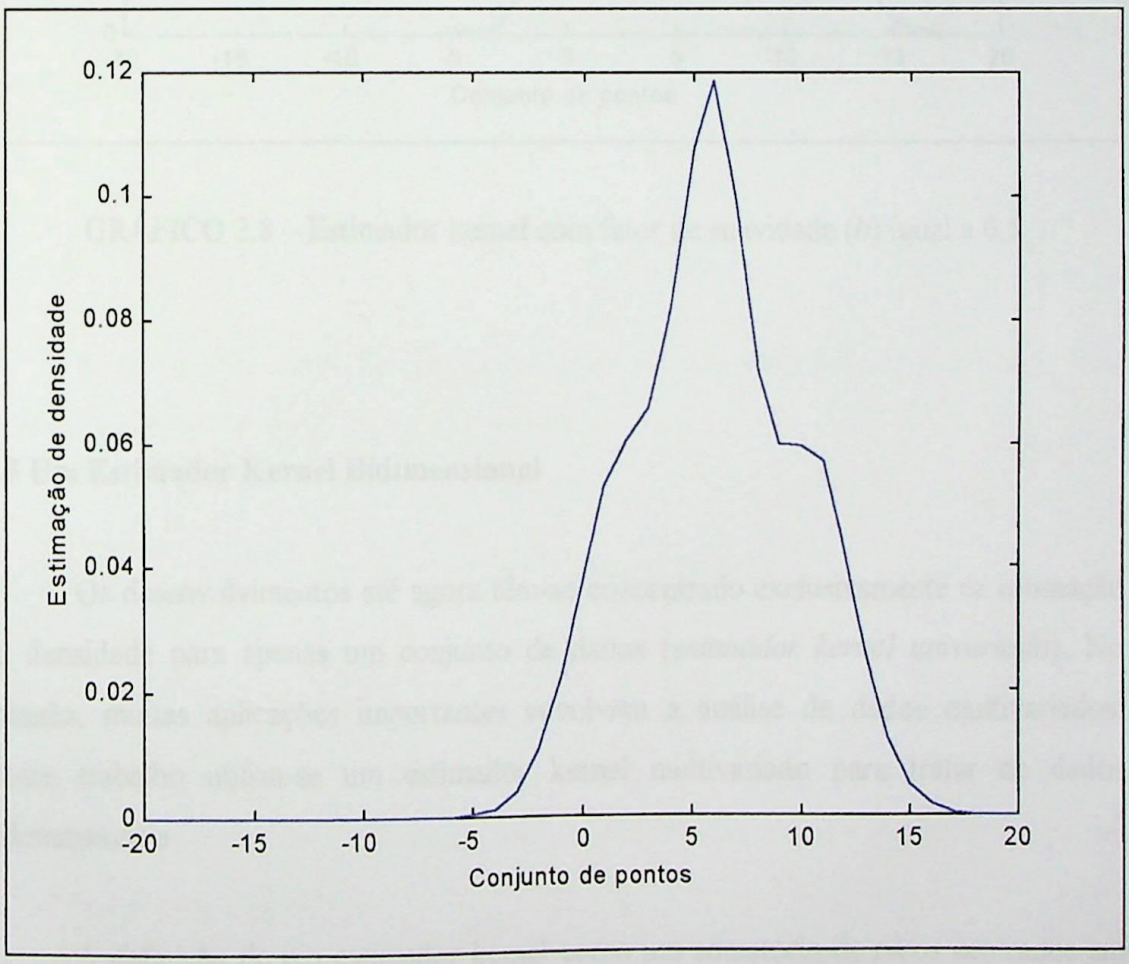

GRÁFICO 2.7 – Estimador kernel com fator de suavidade (h) igual a 1

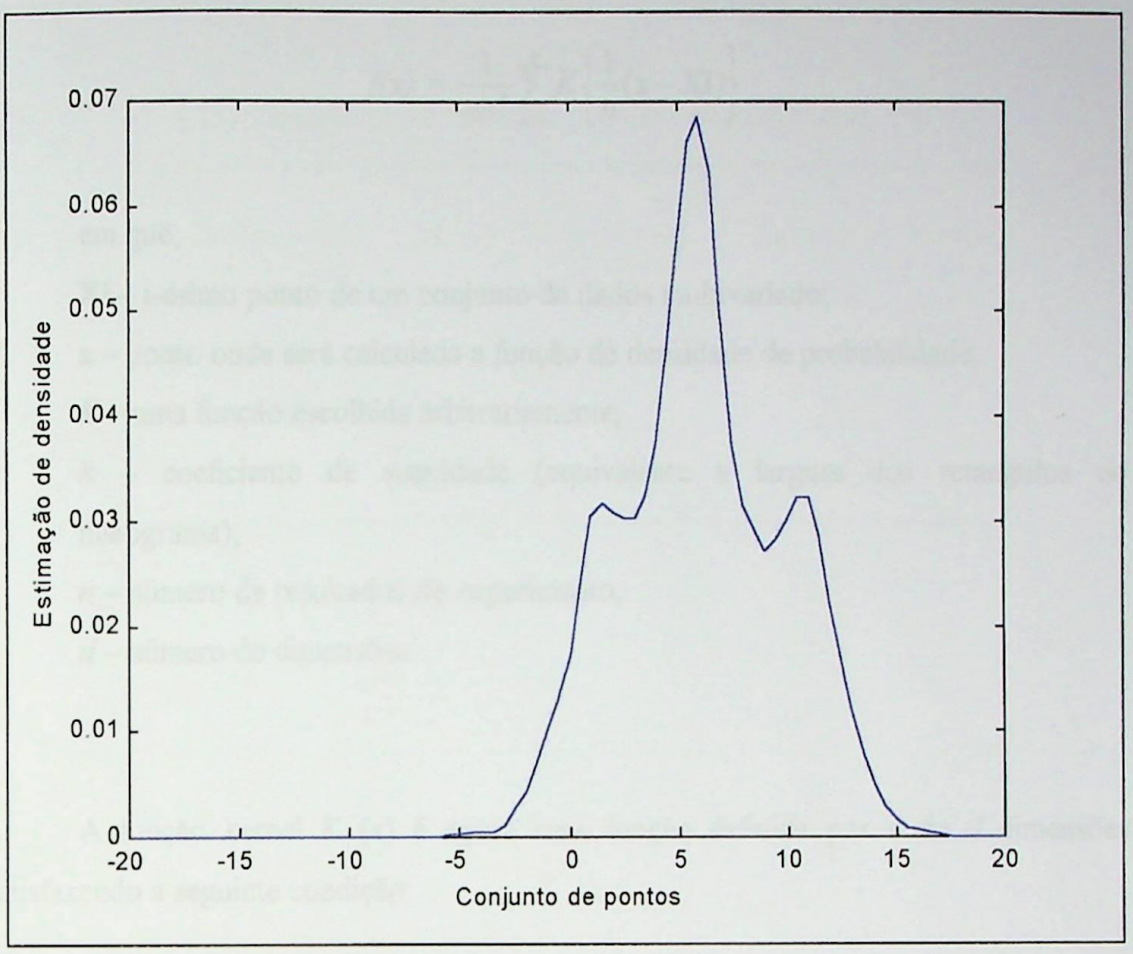

GRÁFICO 2.8 – Estimador kernel com fator de suavidade (h) igual a 0.5

2.5 Um Estimador Kernel Bidimensional

Os desenvolvimentos até agora têm-se concentrado exclusivamente na estimação de densidade para apenas um conjunto de dados (*estimador kernel univariado*). No entanto, muitas aplicações importantes envolvem a análise de dados multivariados. Nesse trabalho utiliza-se um estimador kernel multivariado para tratar de dados bidimensionais.

A definição de um estimador kernel como um somatório de *picos* centrados em cada um dos pontos do conjunto de dados analisado é facilmente generalizado para o caso multivariado. Para tal, é adotada a notação **x** (*negrito*) para um conjunto de dados multivariados de d dimensões. O estimador kernel multivariado é definido por:

$$f(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left\{\frac{1}{h}(\mathbf{x} - \mathbf{Xi})\right\}$$

em que,

Xi – i-ésimo ponto de um conjunto de dados multivariado;

x – ponto onde será calculada a função de densidade de probabilidade;

K – uma função escolhida arbitrariamente;

h – coeficiente de suavidade (equivalente a largura dos retangulos no histograma);

n – número de resultados do experimento;

d – número de dimensões.

A função kernel K (**x**) é agora uma função definida por **x** de d dimensões satisfazendo a seguinte condição:

$$\int_{R^d} K(\mathbf{x})d\mathbf{x} = 1$$

A Gaussiana como função kernel para o caso multivariado é definida pela seguinte equação:

$$K(\mathbf{x}) = (2\pi)^{-d/2}\exp(-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{x})$$

Para exemplificar o caso bivariado foram utilizados dois conjuntos de dados (Gráfico 2.9) conforme a Tabela 2.4. Para o conjunto A as médias são 3 e 9 e para o conjunto B as médias são 2 e 8. Assim, espera-se encontrar um agrupamento na posição 3 e 2 e outro na posição 9 e 8.

TABELA 2.4

Parâmetros do conjunto de dados bidimensional

Subconjuntos	*Média*	*Variância*	*Número de Pontos*
A1	3	2	300
A2	9	2	300
B1	2	2	300
B2	8	2	300

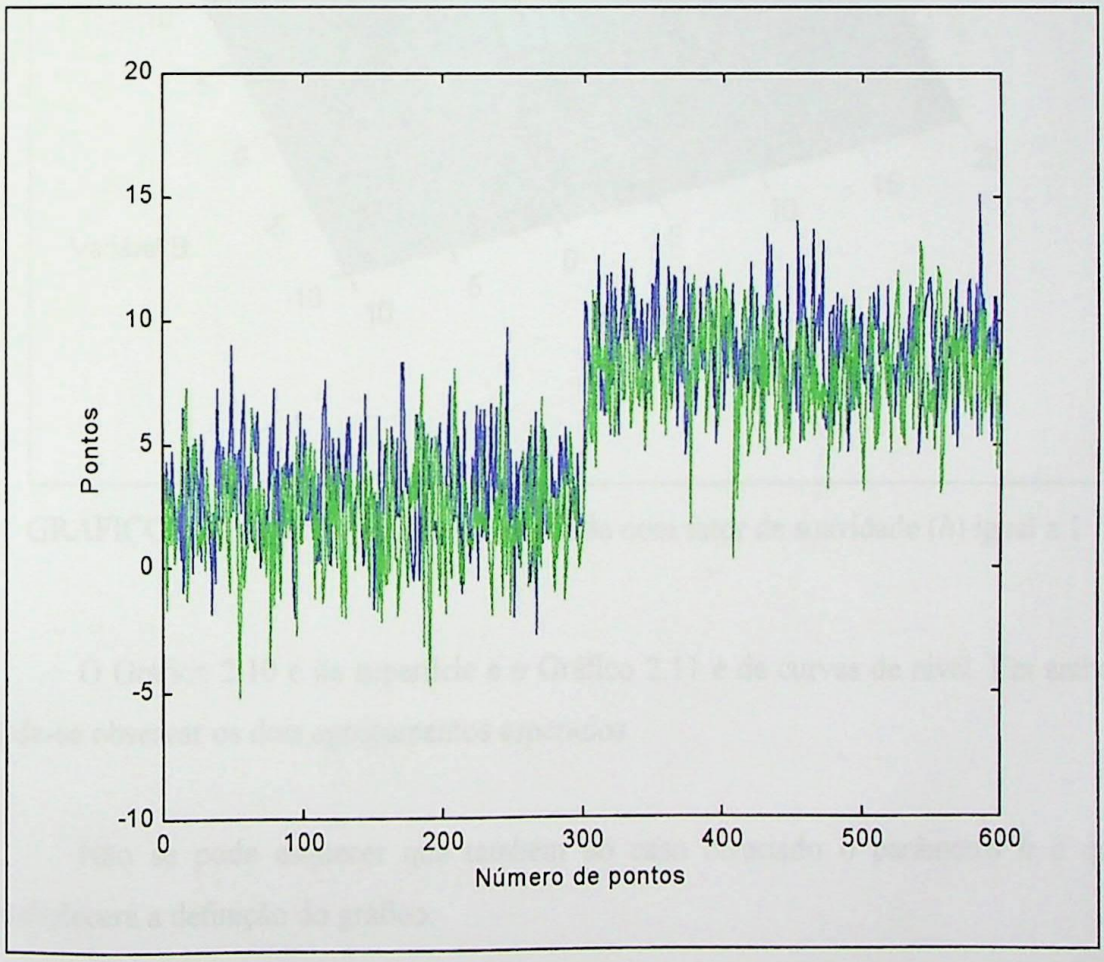

GRÁFICO 2.9 – Conjunto de dados bidimensional conforme a Tabela 2.4

A função de densidade dos dados foi realizada utilizando o estimador kernel bivariado com o parâmetro de suavidade (h) igual a 1. O resultado é apresentado nos Gráficos 2.10 e 2.11.

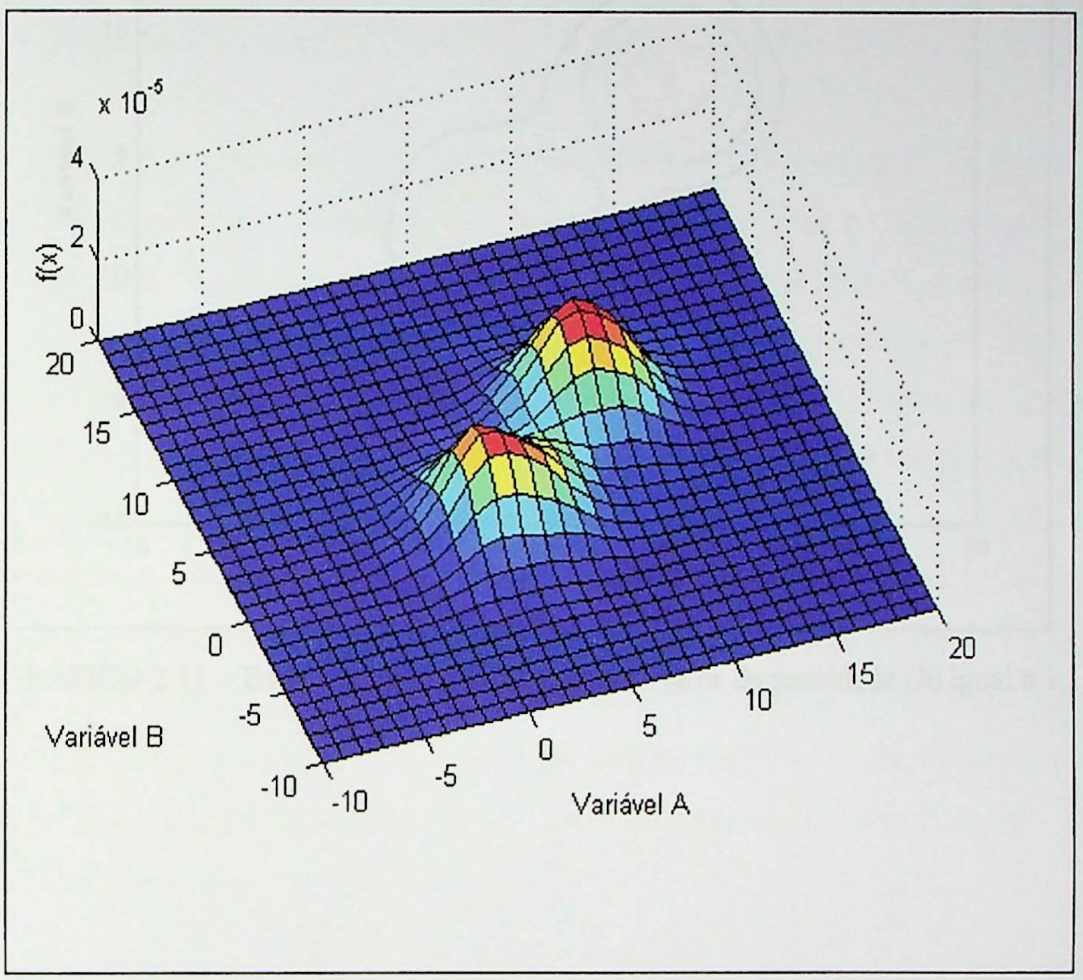

GRÁFICO 2.10 – Estimador kernel bivariado com fator de suavidade (h) igual a 1

O Gráfico 2.10 é de superfície e o Gráfico 2.11 é de curvas de nível. Em ambos pode-se observar os dois agrupamentos esperados.

Não se pode esquecer que também no caso bivariado o parâmetro h é que estabelecerá a definição do gráfico.

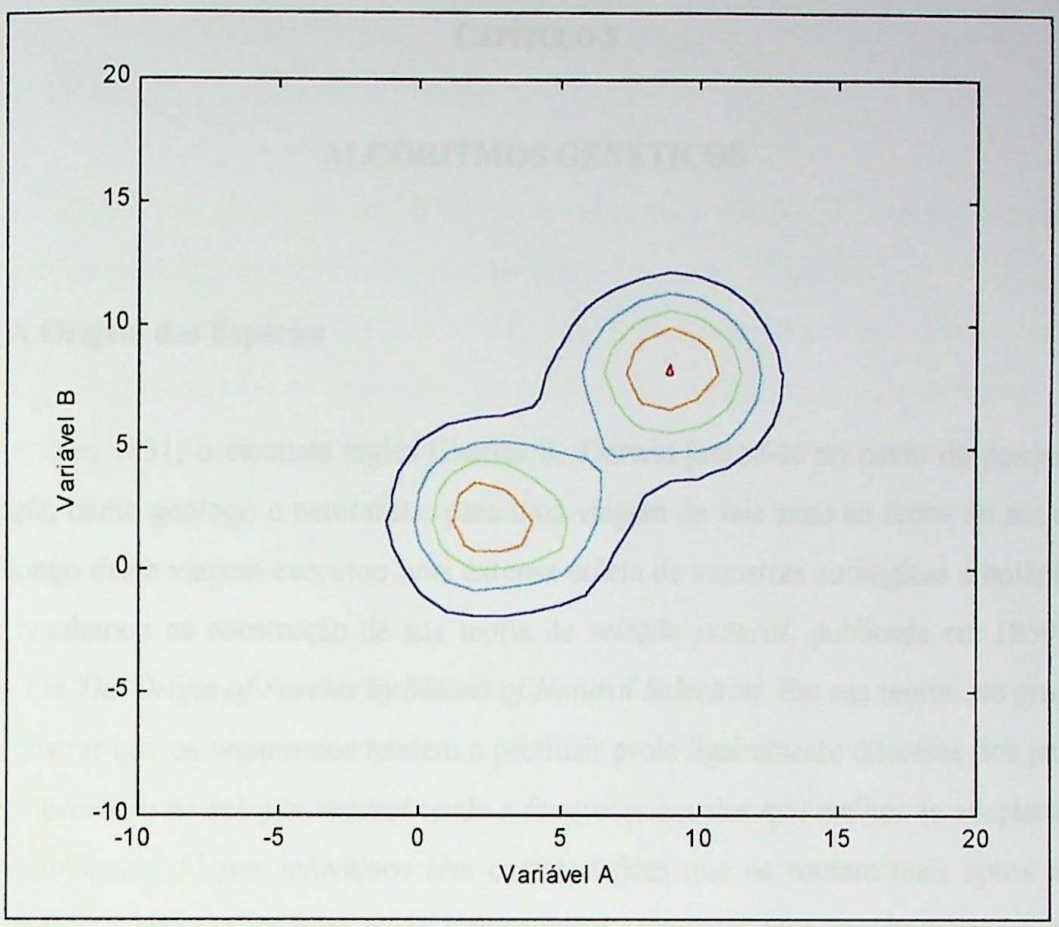

GRÁFICO 2.11 – Estimador kernel bivariado com fator de suavidade (*h*) igual a 1

CAPÍTULO 3

ALGORITMOS GENÉTICOS

3.1 A Origem das Espécies

Em 1831, o cientista inglês Charles R. Darwin juntou-se ao navio de pesquisas *Beagle*, como geólogo e naturalista, para uma viagem de seis anos ao redor do mundo. Ao longo dessa viagem executou uma extensa coleta de amostras zoológicas e botânicas que resultaram na construção de sua teoria de *seleção natural*, publicada em 1859 no livro *On The Origin of Species by Means of Natural Selection.* Em sua teoria, ele propôs demonstrar que os organismos tendem a produzir prole ligeiramente diferente dos pais e que o processo de *seleção natural* tende a favorecer aqueles que melhor se adaptam ao meio ambiente. Alguns indivíduos têm características que os tornam mais aptos para sobreviver e maiores chances de se reproduzir e transmitir suas características a seus descendentes e, com o tempo, espécies distintas vêm a se desenvolver. Essas idéias foram muito criticadas por muitos naturalistas daquele século que acreditavam que as espécies eram criadas separadamente por meio de geração espontânea ou por um princípio divino.

No início do século XX, começaram a surgir muitos trabalhos sobre evolução baseados nos princípios de *herança genética.* Estes trabalhos uniam a genética e a *seleção natural* criando o princípio básico de Genética Populacional: uma população de organismos que tem sua reprodução realizada sexualmente produzirá indivíduos diferentes através do cruzamento genético e de mutações (MENDES FILHO, 1998).

A partir da década de 50, muitos biólogos começaram a estudar e a desenvolver simulações de sistemas genéticos utilizando computadores. Em 1975, após muita pesquisa, John Holland publicou o livro *Adaptation in Natural and Artificial Systems* que hoje é usado como bibliografia básica no estudo de algoritmos genéticos. A partir deste trabalho, vários outros surgiram com sucesso utilizando os algoritmos genéticos em problemas de busca e otimização.

3.2 Definição de Algoritmos Genéticos

Algoritmos genéticos (AGs) são algoritmos de busca baseados em mecanismos da seleção natural e genética e foram desenvolvidos por John Holland e sua equipe na Universidade de Michigan (GOLDBERG, 1989). Sua pesquisa tinha como objetivo explicar rigorosamente os processos adaptativos de sistemas naturais e montar um software de um sistema artificial que implementasse os mecanismos importantes destes sistemas naturais. Esta abordagem conduziu importantes descobertas para a ciência de sistemas naturais e artificiais.

Uma tarefa de busca e otimização abrange, entre vários componentes, o espaço de busca e a função de avaliação. Técnicas tradicionais têm início com *um único candidato* que, iterativamente, é manipulado utilizando algumas heurísticas diretamente associadas ao problema a ser solucionado. Utilizadas com sucesso em várias aplicações, estas técnicas não são robustas o bastante e sua simulação em computador pode se tornar muito complexa. Os AGs são muito simples do ponto de vista computacional entretanto são métodos de busca extremamente eficientes. Partindo de uma *população de candidatos*, os AGs realizam uma busca paralela em diferentes áreas do espaço de soluções.

Pode-se identificar também, em relação às técnicas tradicionais, que os AGs trabalham com uma codificação do conjunto de parâmetros e não com eles próprios. Outra comparação seria que eles utilizam informações de recompensa ou custo e não derivadas ou outro conhecimento auxiliar. Eles são muito eficientes para busca de soluções ótimas, ou aproximadamente ótimas, em uma grande variedade de problemas, pois não impõem muitas das limitações encontradas nos métodos de busca tradicionais (MENDES FILHO, 1998).

Os AGs são métodos de busca cega por não terem conhecimento específico do problema a ser resolvido, tendo como guia apenas a função de avaliação. Aleatórios, não executam buscas sem rumo, pois através de processos iterativos (*gerações*) eles

exploram informações históricas de cada *geração* para encontrar novos pontos de busca onde são esperados melhores desempenhos (YEPES, 2000).

3.3 Conceitos Básicos

Os AGs formam uma classe de procedimentos com várias etapas distintas, e cada uma destas etapas possui muitas variações, por isso sempre são utilizados os termos "algoritmos genéticos" ou "um algoritmo genético" e não "ao algoritmo genético".

Pode-se explicar o funcionamento de um algoritmo genético clássico expondo naturalmente alguns conceitos básicos. O primeiro passo é gerar uma população inicial onde seus indivíduos representam possíveis soluções para um determinado problema. Esta população inicial pode ser gerada a partir de valores aleatórios ou a partir de valores predefinidos (*sementes*). Cada indivíduo é avaliado de acordo com o problema em questão onde os mais aptos são mantidos e os demais são eliminados. Por meio de operadores genéticos (*cruzamento* e *mutação*) os indivíduos restantes geram descendentes (*reprodução*) os quais tem uma grande possibilidade de serem mais aptos do que seus genitores. A *reprodução* é repetida até que uma condição de parada seja satisfeita. Esta condição pode estar relacionada com uma solução satisfatória, o número de gerações ou até mesmo o tempo de processamento.

Em um algoritmo genético clássico um indivíduo é representado por uma *string* binária (***0,1***) onde cada elemento é chamado de *gene* (Figura 3.1). Cada elemento da *string* pode indicar a presença (***1***) ou ausência (***0***) de uma determinada característica que na genética é referenciada como genótipo. Os elementos combinados formam as características reais do indivíduo ou o seu fenótipo.

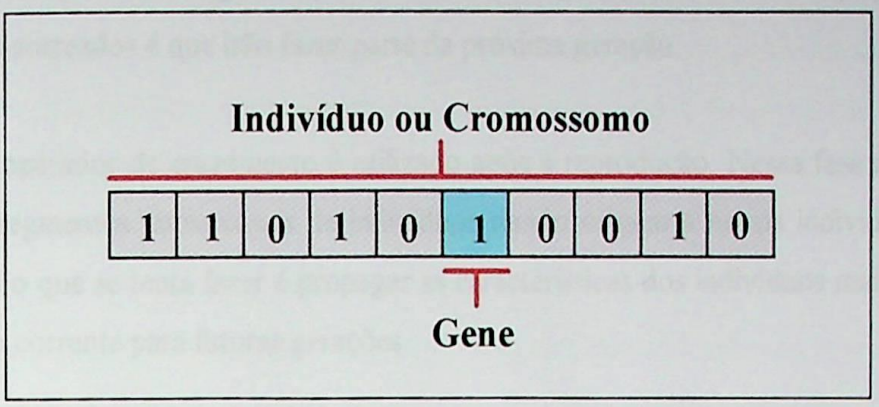

FIGURA 3.1 – Representação de um cromossomo de genes binários

3.4 Operadores e Parâmetros Genéticos

A função dos operadores genéticos é, por meio de um processo recursivo, transformar a população inicial em uma população que represente um resultado satisfatório. Um algoritmo genético clássico é composto de três operações (GOLDBERG, 1989):

1 . Reprodução ou Seleção;
2 . Cruzamento;
3 . Mutação.

A idéia básica da reprodução é selecionar os melhores indivíduos da população corrente através de uma função de aptidão. Os indivíduos com um alto valor de aptidão terão uma alta probabilidade de contribuir com um ou mais descendentes na próxima geração.

A operação de reprodução pode ser implementada de várias formas, porém, o método mais utilizado é o método da roleta. Neste método cada indivíduo da população corrente tem sua representação na roleta de acordo com o seu valor de aptidão. Indivíduos com valores de aptidão altos terão um segmento maior dentro da roleta e os indivíduos com valores menores terão segmentos menores (Tabela 3.1 e Gráfico 3.1).

Posteriormente a roleta é girada *n* vezes e de acordo com o tamanho da população os indivíduos sorteados é que irão fazer parte da próxima geração.

O operador de cruzamento é utilizado após a reprodução. Nessa fase acontece a troca de segmentos entre casais de indivíduos dando origem a novos indivíduos. Com essa troca o que se tenta fazer é propagar as características dos indivíduos mais aptos da população corrente para futuras gerações.

TABELA 3.1

Exemplo de uma população com respectivos valores de aptidão

Nº	*Indivíduos*	*Aptidão*	*% do Total*
1	10011	361	21
2	10101	441	26
3	11110	900	52
4	00011	9	1
Total	**-**	**1711**	**100**

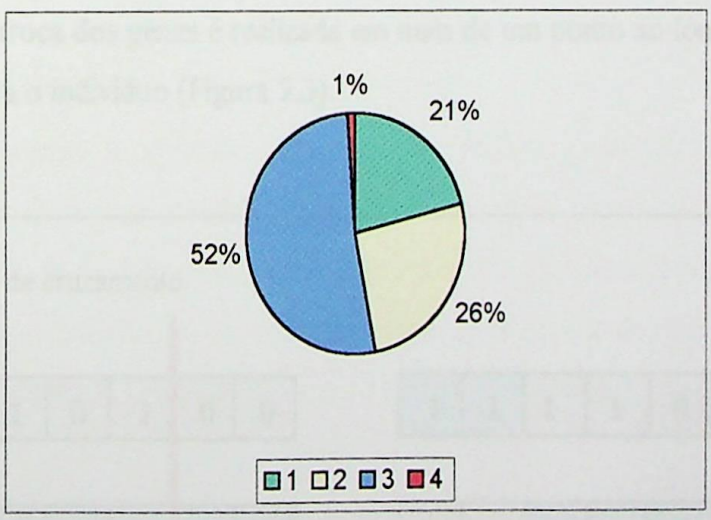

GRÁFICO 3.1 – Roleta de seleção de acordo com os valores de aptidão da Tabela 3.1

Os indivíduos selecionados pela roleta serão transferidos para uma *piscina de acasalamento* (*"mating pool"*) onde o cruzamento é realizado em dois passos. O

primeiro passo consiste em definir os casais de indivíduos de forma aleatória. No segundo um ponto de quebra do indivíduo é escolhido de forma aleatória ao longo da *string* que o representa. A partir deste ponto é realizada a troca de genes entre o par de indivíduos. O operador de cruzamento pode ser implementado de várias formas, entre as mais usadas estão:

- Um ponto de cruzamento: o ponto de quebra do cromossomo é escolhido de forma aleatória e a partir deste ponto as informações genéticas do par de indivíduos serão trocadas (Figura 3.2).

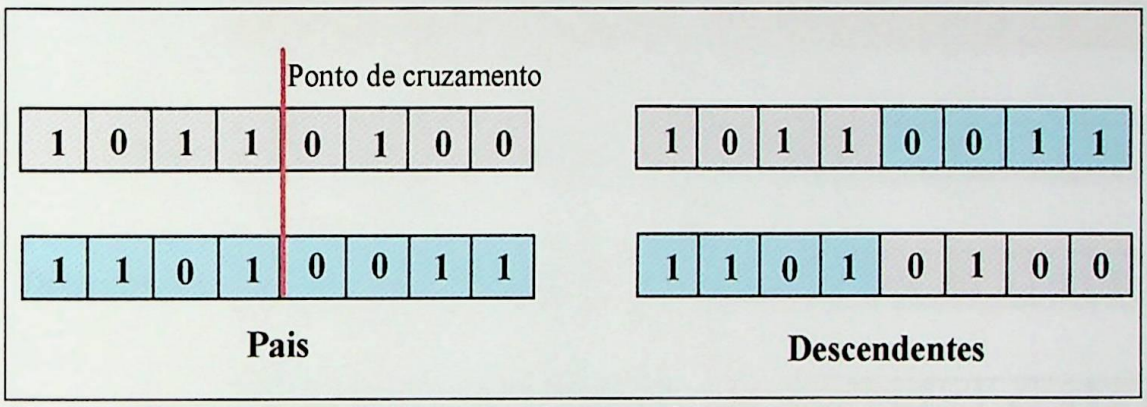

FIGURA 3.2 – Um ponto de cruzamento

- Múltiplos pontos: realizada de maneira similar ao cruzamento de um ponto, porém a troca dos genes é realizada em mais de um ponto ao longo da *string* que representa o indivíduo (Figura 3.3).

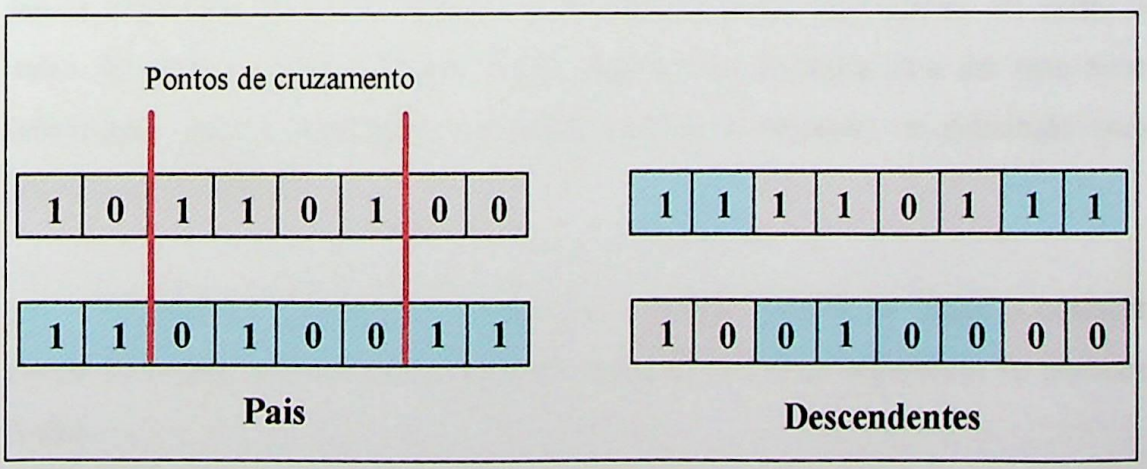

FIGURA 3.3 – Múltiplos pontos de cruzamento

- Cruzamento Uniforme: o cruzamento é feito baseado em uma máscara gerada de forma aleatória com o mesmo número de genes dos indivíduos que serão cruzados. Se houver 1 na máscara de cruzamento o gene correspondente será copiado do primeiro pai e se houver 0 será copiado do segundo pai. Uma vez formado o primeiro descendente o processo será repetido com os pais trocados para se formar o segundo descendente (Figura 3.4).

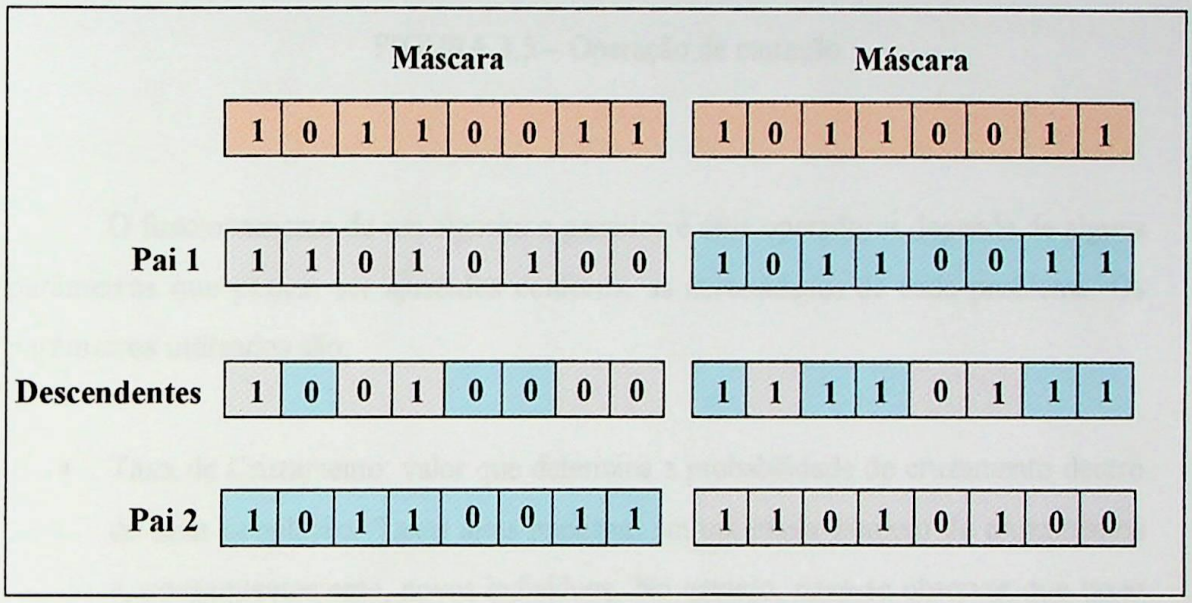

FIGURA 3.4 – Cruzamento uniforme

Após o cruzamento a operação de mutação é aplicada para cada gene de todos os novos indivíduos de forma aleatória. A operação consiste simplesmente em alterar o valor do gene (1 para 0 e vice versa) (Figura 3.5). Utilizada para dar uma nova informação para a população, a mutação previne a saturação da população com indivíduos semelhantes.

O operador de mutação garante que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca será zero, além de contornar o problema de mínimos locais.

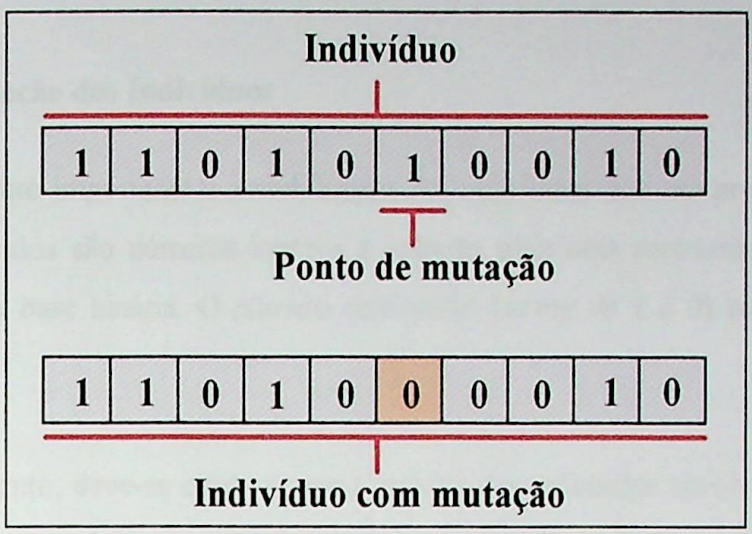

FIGURA 3.5 – Operação de mutação

O funcionamento de um algoritmo genético e seus operadores depende de alguns parâmetros que podem ser ajustados conforme as necessidades de cada problema. Os parâmetros utilizados são:

- Taxa de Cruzamento: valor que determina a probabilidade de cruzamento dentro de uma população. Taxas altas implicam em um maior número de cruzamentos e, conseqüentemente, novos indivíduos. No entanto, deve-se observar que taxas muito altas podem resultar em perda das características dos indivíduos nas gerações futuras. Indivíduos com boa aptidão podem desaparecer em uma próxima geração. Valores baixos podem tornar seu funcionamento extremamente lento.

- Taxa de Mutação: determina a probabilidade que uma mutação ocorrerá. Valores baixos são utilizados para deslocar o foco de busca de um algoritmo genético. Taxas altas tornam a busca essencialmente aleatória podendo implicar na possibilidade de que uma boa solução seja destruída.

- Tamanho da População: populações pequenas apresentam um baixo desempenho, pois atuam em um pequeno espaço de busca do problema. Populações grandes efetuam uma cobertura significativa do espaço de busca do problema. Desta forma evita-se também o problema de mínimos locais.

3.5 Representação dos Indivíduos

Um ponto importante é a codificação dos indivíduos. Em um problema onde os valores envolvidos são números inteiros a solução seria uma conversão direta de um número para a base binária. O número convertido (*string de 1 e 0*) representaria um indivíduo.

No entanto, deve-se observar que a maioria das aplicações envolve valores reais. Para trabalhar com números reais não se pode executar uma simples conversão da base decimal para a base binária. A técnica mais utilizada é efetuada através de uma representação discreta dos dados dentro de um intervalo $[x_{min}, x_{max}]$ em uma quantidade de pontos 2^t, tal que à distância entre pontos consecutivos seja menor que um valor de tolerância especificado, ou seja:

$$\frac{x_{\max} - x_{\min}}{2^t - 1} < TOL$$

Portanto, cada ponto do espaço de busca será representado por um número binário de tamanho t, começando por 0...0 que representa x_{min} e terminando por 1...1 que representa x_{max}.

O ponto principal da representação está em calcular o tamanho (t) dos indivíduos. Com base no valor de tolerância este tamanho pode ser calculado a partir da seguinte equação:

$$t = \log_2\left(1 + \frac{x_{\max} - x_{\min}}{TOL}\right)$$

Por exemplo, em um intervalo $x \in [0,1]$ e uma precisão de duas casas decimais ($TOL = 5 \times 10^{-3}$), então o tamanho do indivíduo seria:

$$t = \log_2\left(1 + \frac{1-0}{0.005}\right) = 8$$

Desta forma, pode-se utilizar indivíduos com o tamanho de 8 bits para representar o intervalo $x \in [0,1]$ com precisão menor igual a 0.005.

Com a definição do tamanho dos indivíduos e do intervalo $[x_{min}, x_{max}]$ pode-se realizar as codificações necessárias. Um valor real deverá ser convertido em um valor inteiro e este por sua vez deverá ser codificado em binário para sofrer as operações de cruzamento e mutação. Contudo, para o processo inverso deve-se converter o valor binário para um valor inteiro e, finalmente, efetuar a conversão de inteiro para real. A codificação para real seria:

$$x_{real} = x_{bin} \cdot \left(\frac{x_{\max} - x_{\min}}{2^t - 1}\right) + x_{\max}$$

CAPÍTULO 4

ANÁLISE DE AGRUPAMENTOS UTILIZANDO ALGORITMOS GENÉTICOS

4.1 Introdução

A técnica de análise de agrupamentos utilizando algoritmos genéticos (AGs) é muito simples e eficiente. A idéia principal consiste em utilizar os AGs para encontrar os máximos da função de densidade de um conjunto de dados (PINTO, 1998).

O algoritmo genético para esta implementação utiliza o estimador kernel como função de aptidão. O objetivo é achar todos os máximos locais obtidos pela função de aptidão uma vez que o agrupamento dos dados não está apenas no máximo global da função de densidade. Uma das características dos AGs é que eles podem achar os máximos locais de um conjunto de dados a partir de uma população pequena com um número pequeno de gerações (SERRADA, 1996).

A execução do algoritmo genético em apenas uma vez não garante que todos os máximos locais serão encontrados. Executa-se o algoritmo genético N vezes, armazenando a população final após M gerações. Em seguida, calcula-se a função de densidade da população de soluções e os picos apresentados representarão os agrupamentos. Os passos desta técnica são:

- Definir o estimador kernel como função de aptidão;
- Definir uma pequena população inicial;
- Definir um valor pequeno para o número máximo de gerações;
- Executar o algoritmo genético N vezes e salvar a população final a cada vez;
- Utilizar o estimador kernel para estimar a função de densidade da população final obtida depois da execução do algoritmo genético N vezes;

- O número de picos será o número de agrupamentos e as variáveis de cada pico serão o centro dos agrupamentos.

4.2 Um Algoritmo Genético Passo a Passo

Um algoritmo genético clássico foi implementado com o propósito de ser utilizado como base para outros programas pertinentes a este trabalho. O programa foi escrito em linguagem C devido a sua popularidade, portabilidade e fácil comunicação com outros softwares (como o Matlab, por exemplo).

O programa foi implementado para resolver um problema de otimização. O problema exemplo abordado foi maximizar a função $f(x) = x^2$, onde x pode variar entre 0 e 31 (GOLDBERG, 1989). Inicialmente as variáveis do problema foram definidas como *strings* de tamanho finito. No problema abordado a codificação dos indivíduos foi feita convertendo um número decimal em binário. Assim, o valor 0 passa a representar o indivíduo 00000, e o valor 31 o indivíduo 11111. O próximo passo está em selecionar uma população inicial aleatoriamente ou pré-definida (*sementes*). O tamanho da população foi definido com 4 indivíduos com 5 genes cada um.

Ainda no processo de configuração deve-se definir as taxa de mutação e cruzamento. Os valores para a taxa de cruzamento são tipicamente definidos entre 0.6 e 1.0 e, para a taxa de mutação os valores ficam em torno de 0.001 (BEASLEY, 1993). No programa foram configurados valores padrões (*default*) que podem sem alterados. Através das Figuras 4.1 e 4.2 observa-se a parte de configuração do programa.

Uma vez configurado, o programa exibe sua tela principal (Figura 4.3). Os valores iniciais e as *strings* binárias que representam cada indivíduo são apresentados respectivamente nas colunas x e *População*. Posteriormente, o programa utiliza a *função custo* para verificar a aptidão de cada indivíduo, exibida na coluna $f(x) = x^2$. Na coluna $fi(x) / S\, fi(x)$ é apresentado à porcentagem de cada indivíduo em relação ao somatório dos valores de aptidão da população.

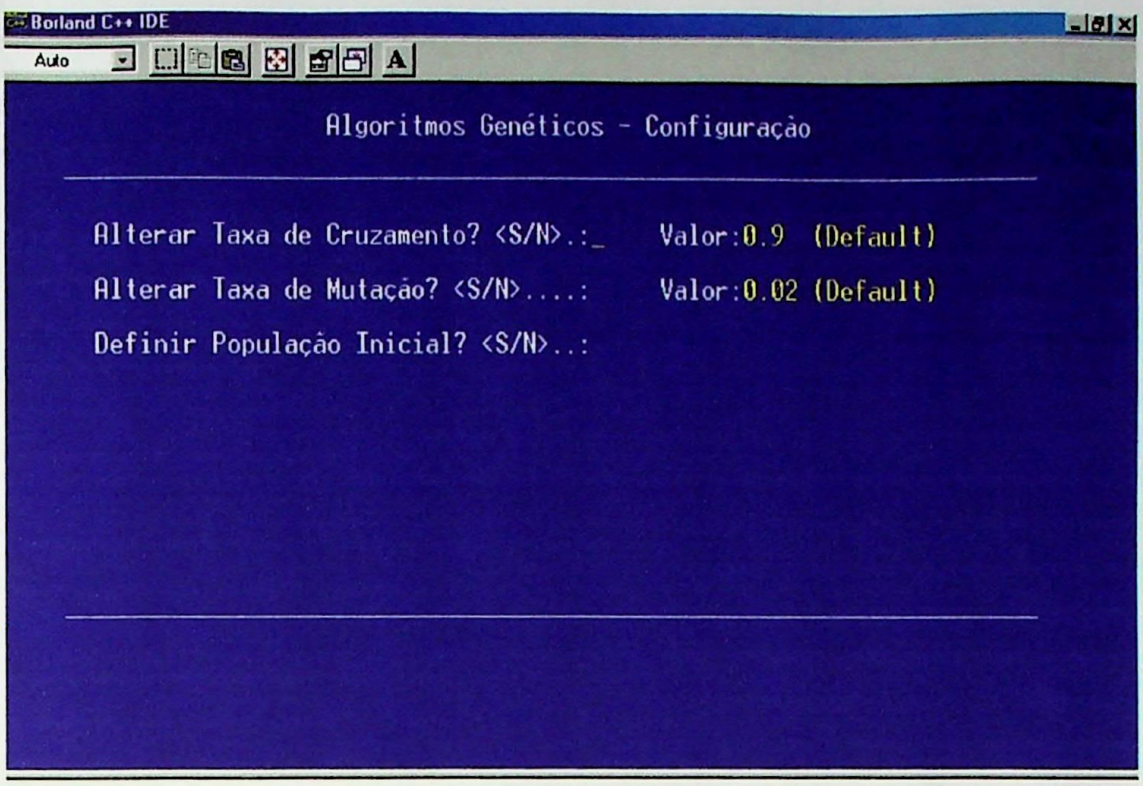

FIGURA 4.1 – Configuração do programa AG

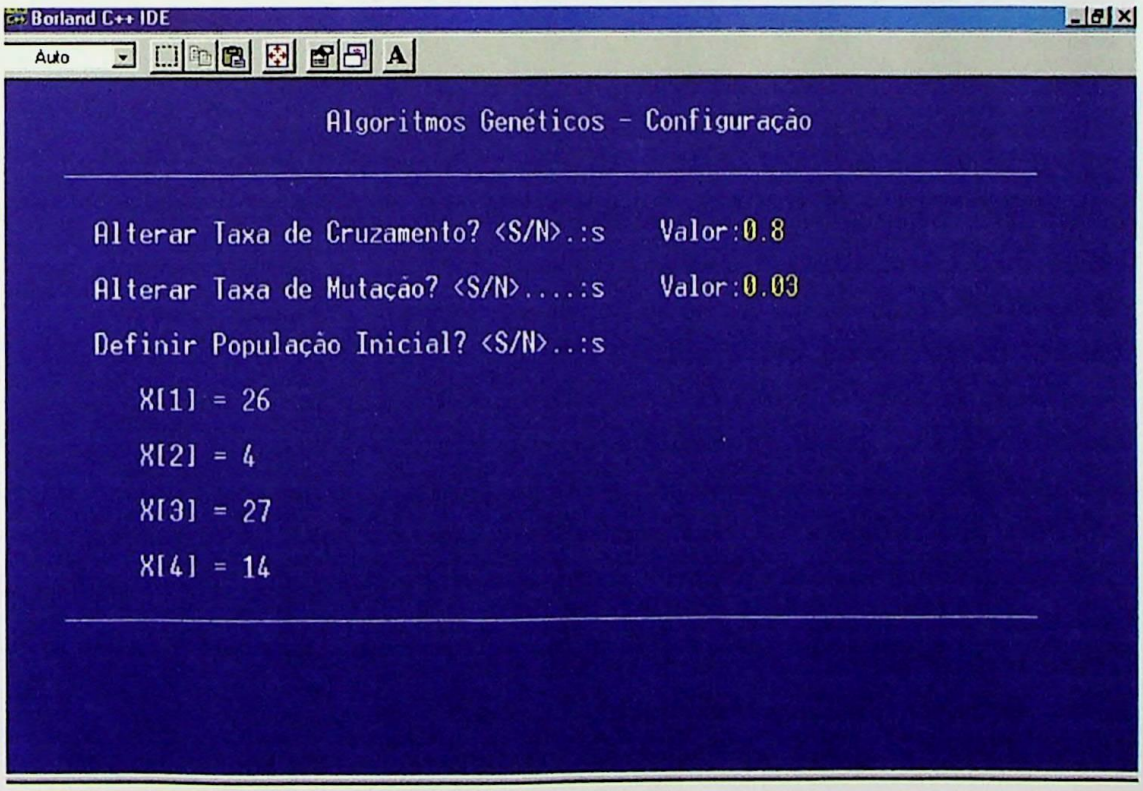

FIGURA 4.2 – Exemplo de configuração de taxas e da população inicial

A seleção dos indivíduos que irão para a *piscina de acasalamento* é feita utilizando-se o método da roleta. Os indivíduos terão seu segmento na roleta conforme os valores da coluna *fi(x) / S fi(x)*. Computacionalmente, esses valores são dispostos em uma reta e é sorteado um valor dentro deste intervalo. O número de sorteios é feito de acordo com o número de indivíduos da população. Na coluna *Roleta* o programa apresenta quais os indivíduos que irão para a *piscina de acasalamento* e em que quantidade.

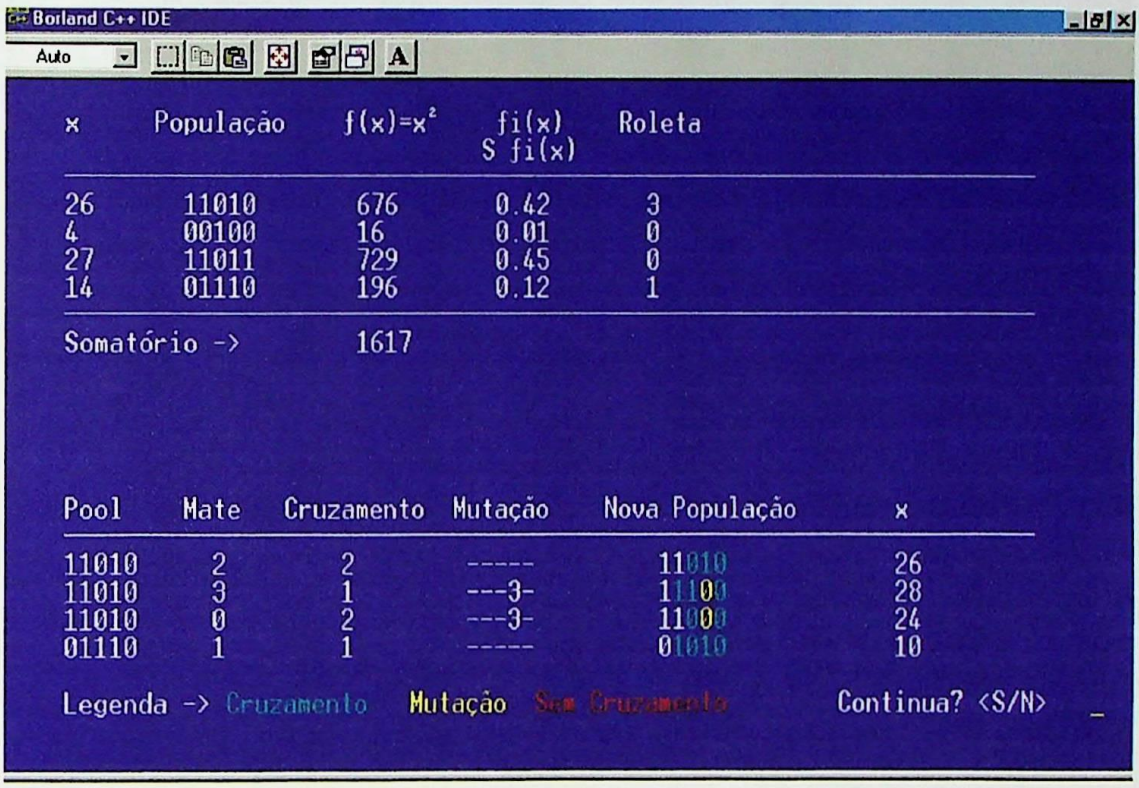

FIGURA 4.3 – Tela principal do programa AG

Na *piscina de acasalamento* são definidos, de forma aleatória, os casais, o ponto de cruzamento e os genes que sofrerão mutação (colunas *Mate*, *Cruzamento* e *Mutação*). Após aplicar os operadores genéticos (*cruzamento* e *mutação*), o programa exibe a nova população utilizando cores para indicar onde os operadores agiram. Ao final, a nova população é decodificada em decimal e o processo é reiniciado (Figura 4.4).

Nesta implementação a condição de parada fica a cargo do usuário que irá definir se continua o processo ou não.

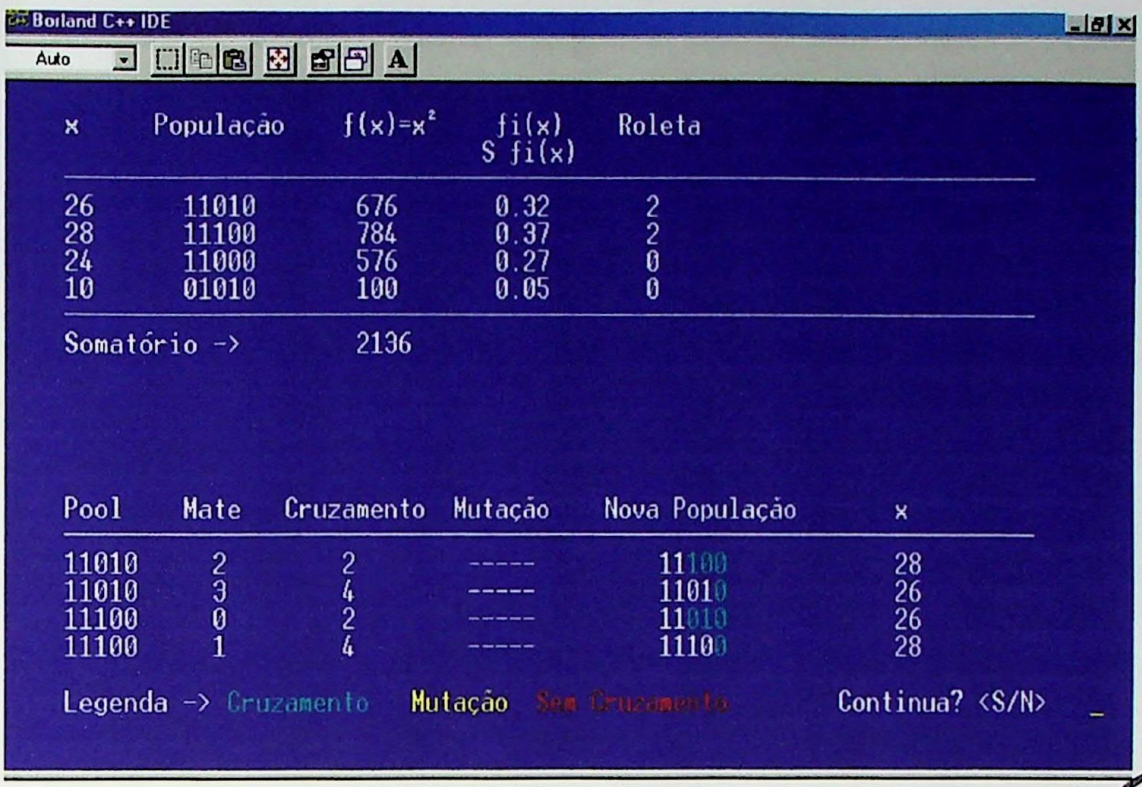

FIGURA 4.4 – O programa reinicia o processo com a nova geração

4.3 Análise de Agrupamento Utilizando AGs para Dados Unidimensionais

Para avaliar e exemplificar a utilização dos algoritmos genéticos na análise de agrupamentos foi primeiramente desenvolvida uma aplicação para dados unidimensionais. Foram gerados três subconjuntos de dados com uma distribuição normal (Gaussiana) baseados nos parâmetros apresentados na Tabela 4.1. Os subconjuntos foram gerados utilizando o software MATLAB e são apresentados no Gráfico 4.1. Os três subconjuntos formam o conjunto de dados que será analisado. O que se espera é encontrar três agrupamentos centrados nos pontos 2, 7 e 11 conforme a média de cada subconjunto apresentado na Tabela 4.1.

Inicialmente, a função de densidade dos dados foi estimada utilizando apenas o estimador kernel apresentado no Capítulo 2. Os resultados apresentados nos Gráficos 4.2 e 4.3 foram obtidos com o parâmetro de suavidade (h) igual a 1 e 0.5

respectivamente. Analisando o Gráfico 4.2 pode-se observar a dificuldade em localizar os agrupamentos com o parâmetro de suavidade igual a 1. O que ocorre é uma sobreposição dos pontos devido aos valores de média e variância. No Gráfico 4.3, com o parâmetro de suavidade igual a 0.5, pode-se observar mais claramente os três agrupamentos (picos) próximos dos valores esperados.

TABELA 4.1

Parâmetros do conjunto de dados unidimensional

Subconjuntos	*Média*	*Variância*	*Número de Pontos*
1	2	2	600
2	7	2	600
3	11	2	600
Total	**-**	**-**	**1800**

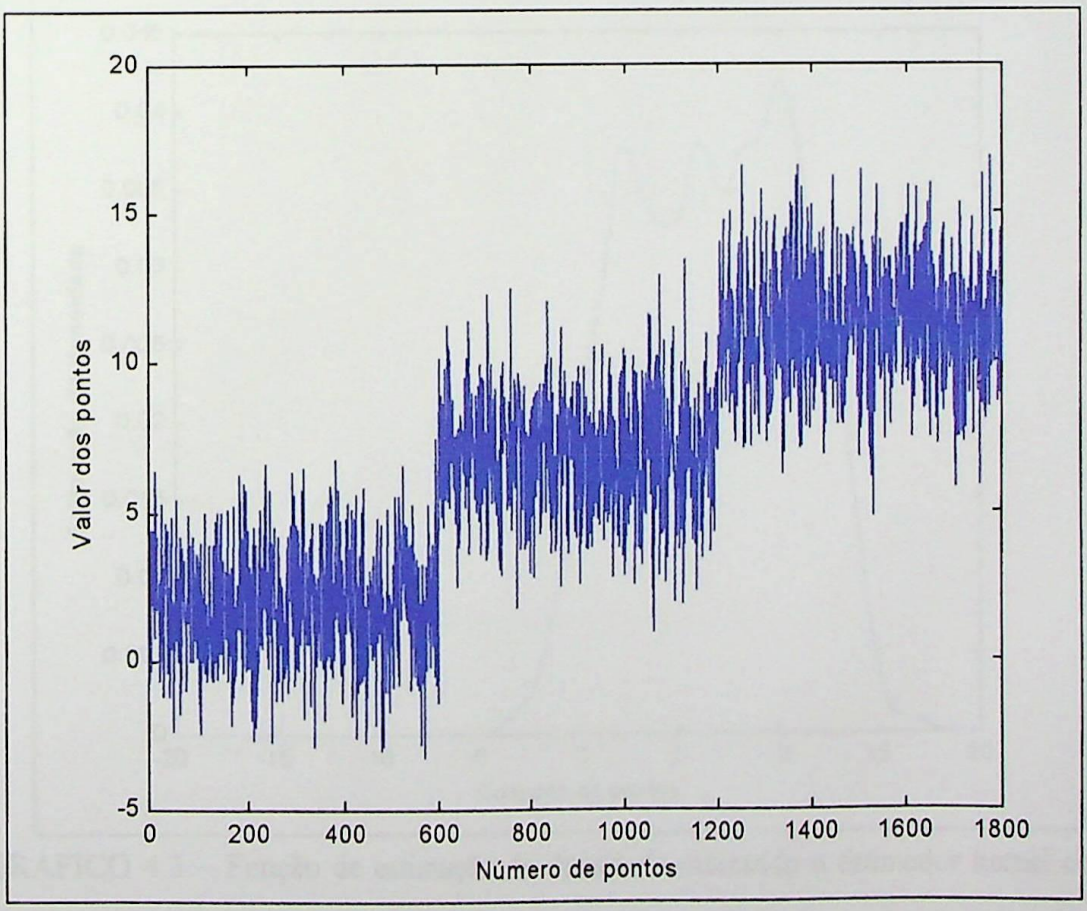

GRÁFICO 4.1 – Conjunto de dados unidimensional conforme a Tabela 4.1

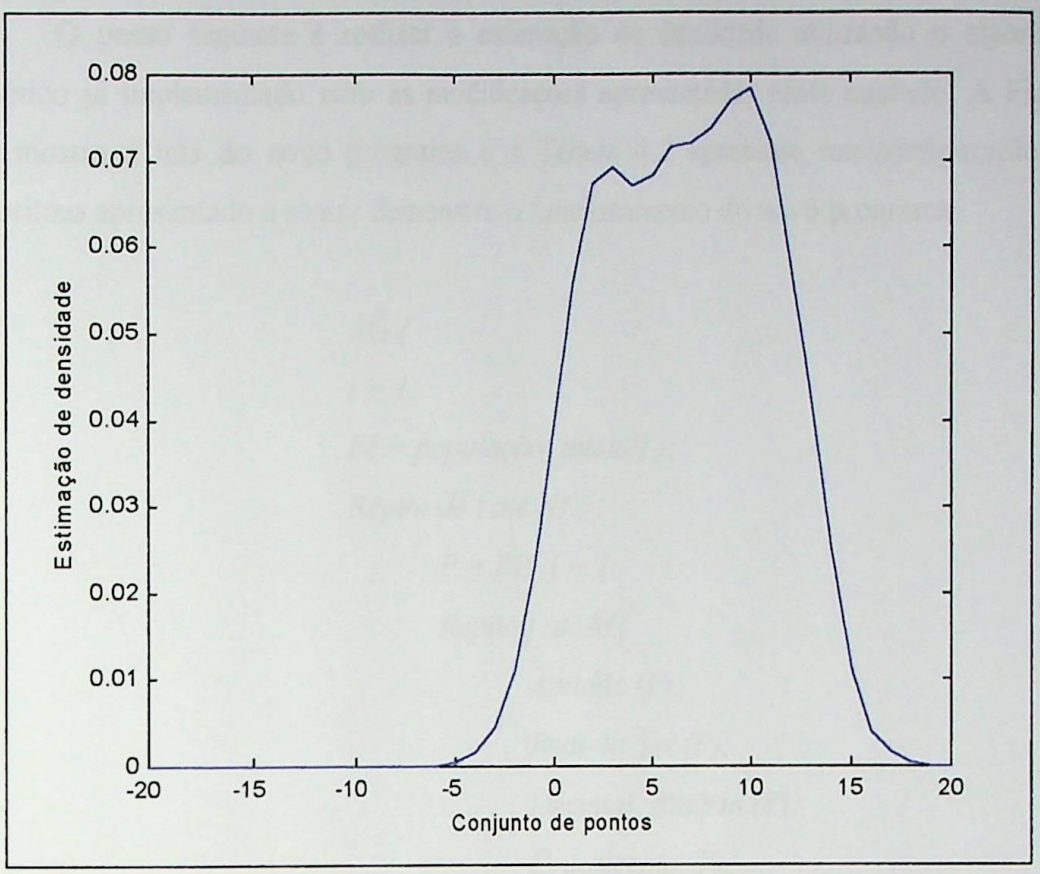

GRÁFICO 4.2 – Função de estimação de densidade utilizando o estimador kernel com parâmetro de suavidade (h) igual a 1

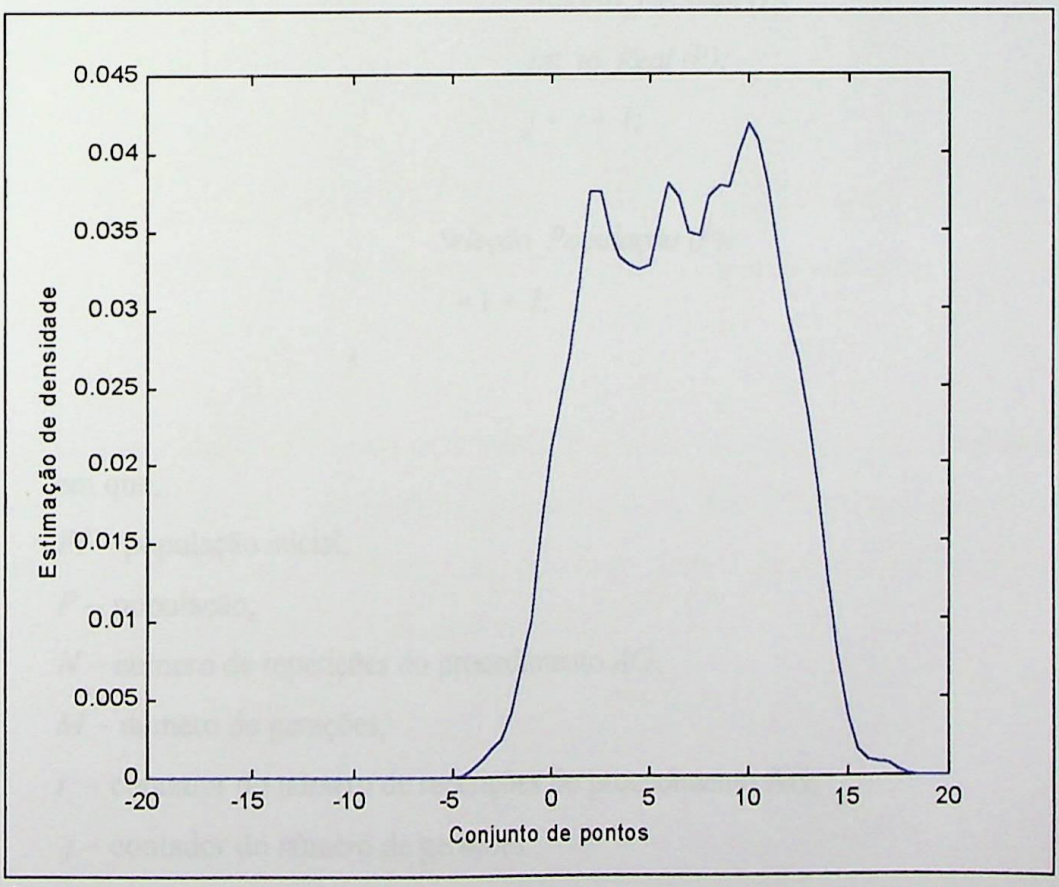

GRÁFICO 4.3 – Função de estimação de densidade utilizando o estimador kernel com parâmetro de suavidade (h) igual a 0.5

O passo seguinte é realizar a estimação de densidade utilizando o algoritmo genético já implementado com as modificações apresentadas neste capítulo. A Figura 4.5 mostra a tela do novo programa e a Tabela 4.2 apresenta sua configuração. O algoritmo apresentado a seguir demonstra o funcionamento do novo programa:

```
AG {
i = 1;
PI = população_inicial( );
Repita de i até N{
        P = PI;  j = 1;
        Repita j até M{
                Aptidão (P);
                Real_to_Int (P);
                Decimal_Binário (P);
                Cruzamento (P);
                Mutação (P);
                Binário_Decimal (P);
                Int_to_Real (P);
                j = j + 1;
        }
        Seleção_População (P);
        i = i + 1;
}
```

em que,

PI – população inicial;

P – população;

N – número de repetições do procedimento *AG*;

M – número de gerações;

i – contador do número de repetições do procedimento AG;

j – contador do número de gerações.

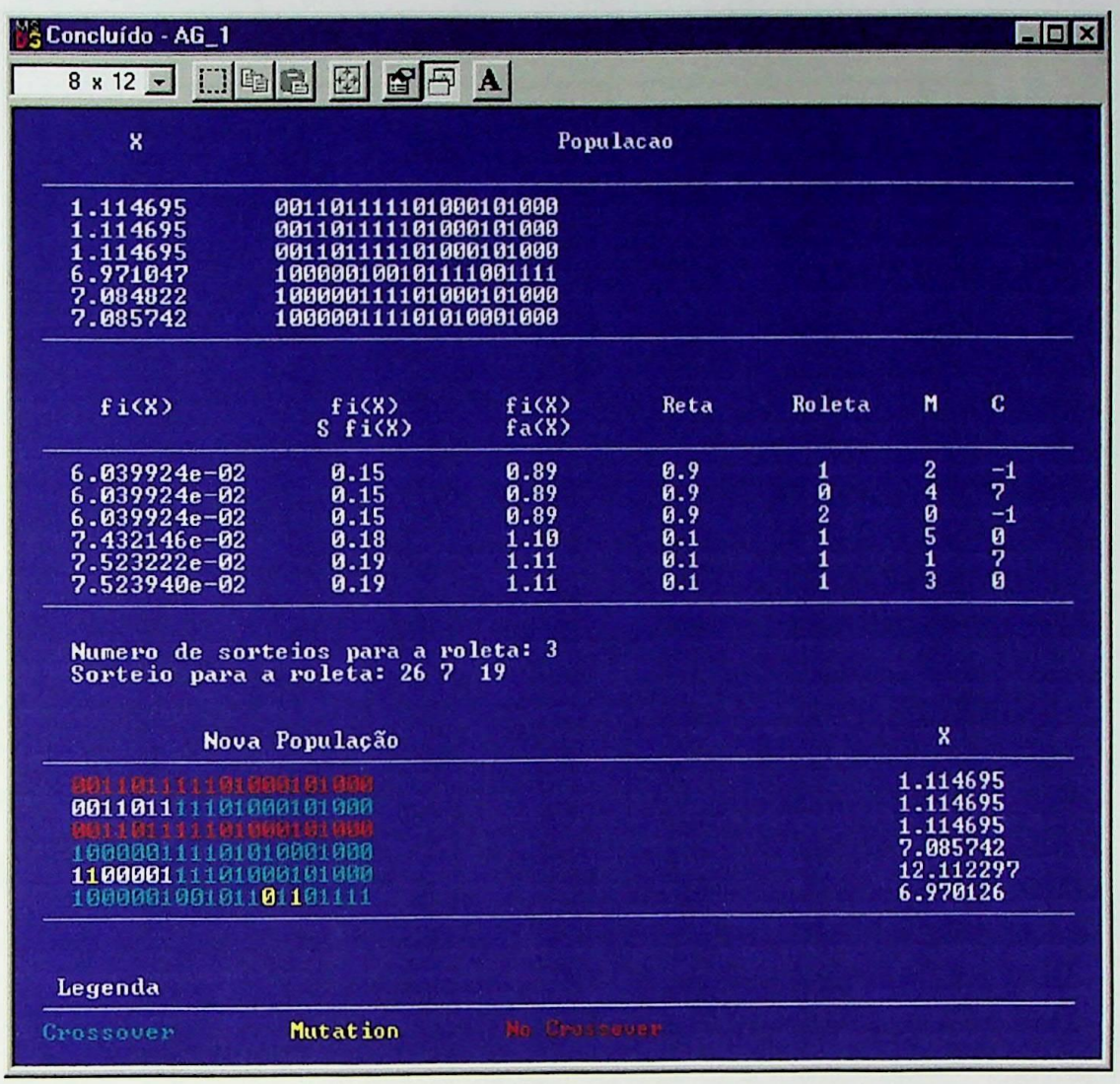

FIGURA 4.5 – Tela do programa AG para o caso unidimensional

O programa é executado cem vezes e a cada dez gerações ele armazena a população final. Com a configuração apresentada, ao final do processo tem-se um conjunto de 600 pontos. A função de densidade deste conjunto é estimada e os resultados são apresentados nos Gráficos 4.4 e 4.5.

TABELA 4.2

Configuração do programa AG para o caso unidimensional

População Inicial	*Aleatória*
Tamanho da População	*6*
Número de Gerações	*10*
Número de Execuções do AG	*100*
Número de Pontos	*1800*
Taxa de Cruzamento	*0.6*
Taxa de Mutação	*0.02*
Função de Aptidão	*Estimador Kernel (Gaussiana)*
Precisão	*0.00001*
Tamanho do Indivíduo	*21*

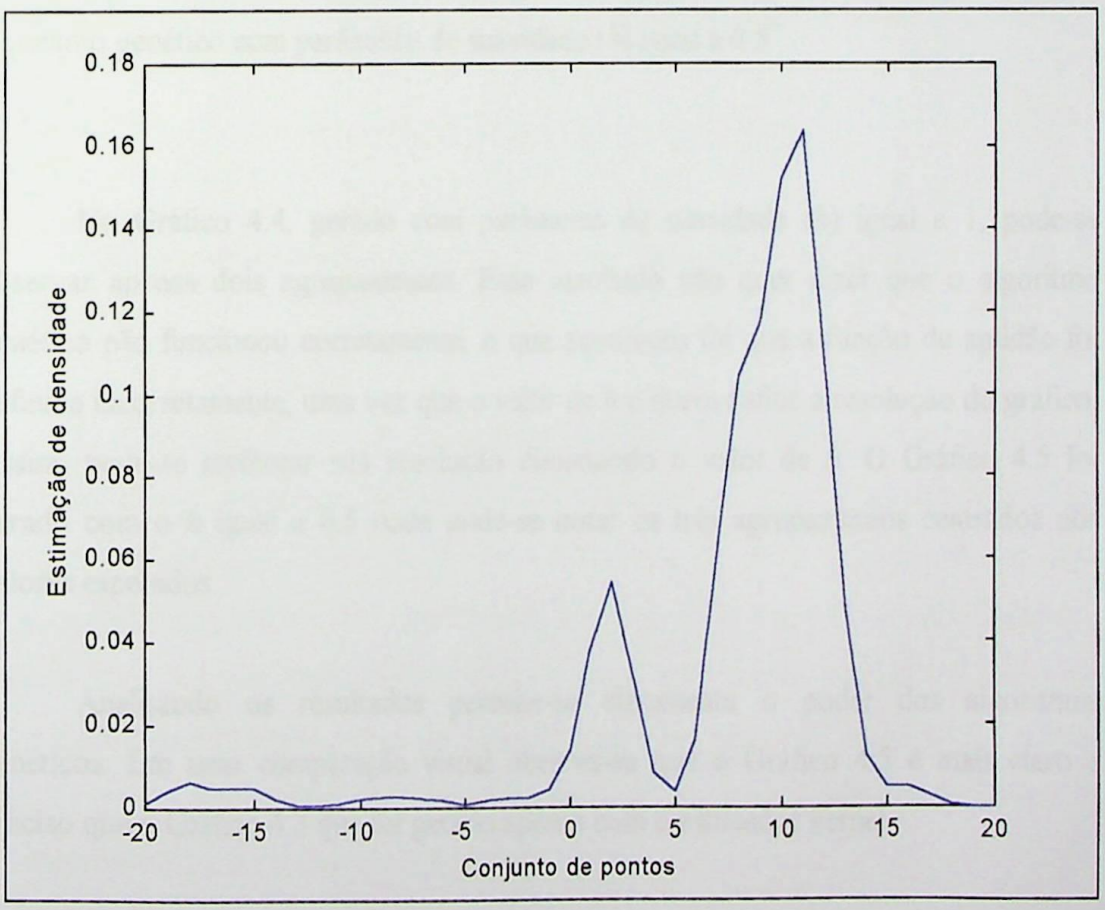

GRÁFICO 4.4 – Função de estimação de densidade utilizando a população final do algoritmo genético com parâmetro de suavidade (h) igual a 1

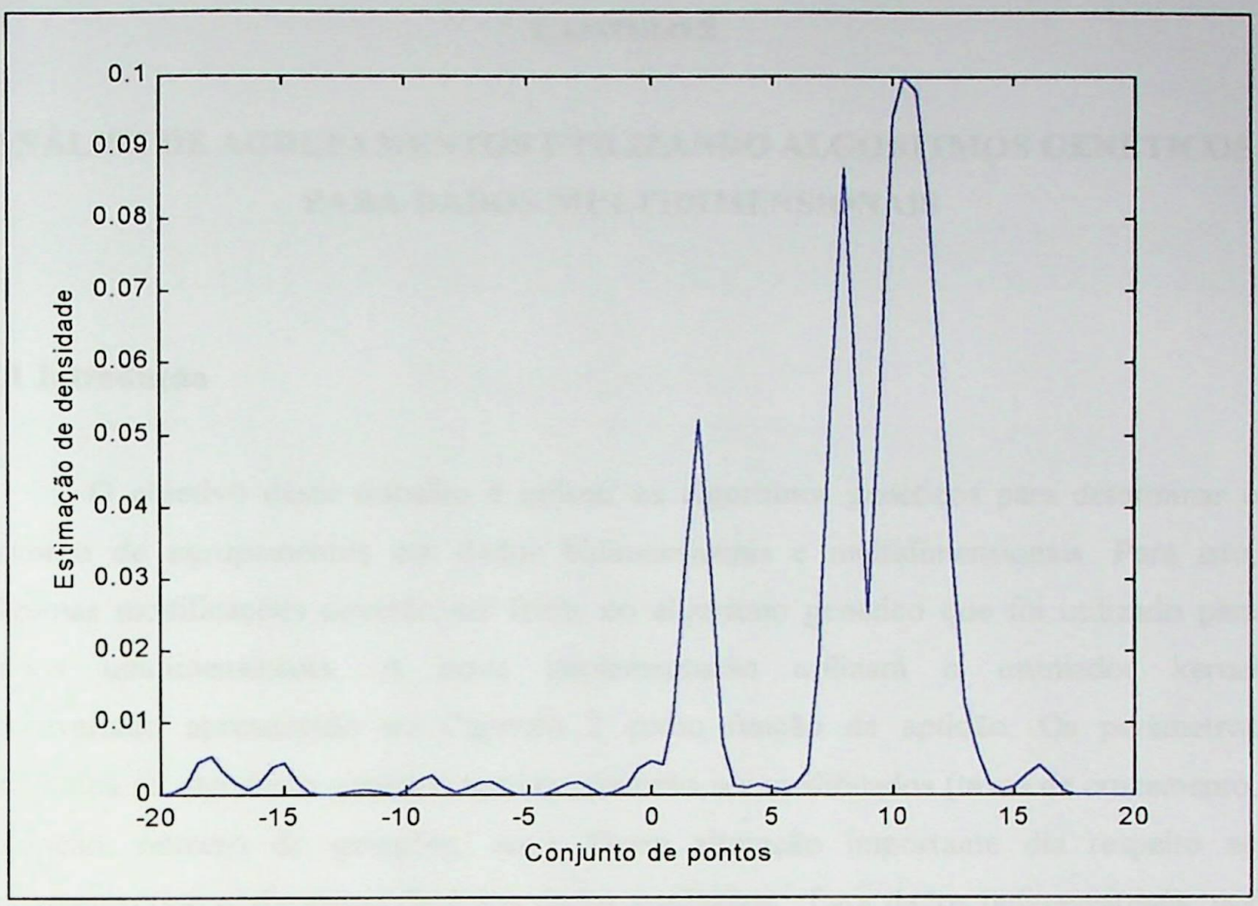

GRÁFICO 4.5 – Função de estimação de densidade utilizando a população final do algoritmo genético com parâmetro de suavidade (h) igual a 0.5

No Gráfico 4.4, gerado com parâmetro de suavidade (h) igual a 1, pode-se observar apenas dois agrupamentos. Esse resultado não quer dizer que o algoritmo genético não funcionou corretamente, o que aconteceu foi que a função de aptidão foi definida incorretamente, uma vez que o valor de h é quem define a resolução do gráfico. Assim, pode-se melhorar sua resolução diminuindo o valor de h. O Gráfico 4.5 foi gerado com o h igual a 0.5 onde pode-se notar os três agrupamentos centrados nos valores esperados.

Analisando os resultados percebe-se claramente o poder dos algoritmos genéticos. Em uma comparação visual observa-se que o Gráfico 4.5 é mais claro e preciso que o Gráfico 4.3 que foi gerado apenas com o estimador kernel.

CAPÍTULO 5

ANÁLISE DE AGRUPAMENTOS UTILIZANDO ALGORITMOS GENÉTICOS PARA DADOS MULTIDIMENSIONAIS

5.1 Introdução

O objetivo deste trabalho é utilizar os algoritmos genéticos para determinar o número de agrupamentos em dados bidimensionais e multidimensionais. Para isto, algumas modificações deverão ser feitas no algoritmo genético que foi utilizado para dados unidimensionais. A nova implementação utilizará o estimador kernel multivariado apresentado no Capítulo 2 como função de aptidão. Os parâmetros utilizados no algoritmo genético também deverão ser modificados (taxas de cruzamento, mutação, número de gerações, etc.). Outra alteração importante diz respeito ao tratamento que o algoritmo dará aos dados analisados. Para dados bidimensionais, por exemplo, a nova aplicação terá um conjunto de N dados da seguinte forma:

$$(x1, y1), (x2, y2), (x3, y3), \ldots (xN, yN)$$

Cada par ordenado (xi, yi) deverá ter um valor para a função de densidade de probabilidade.

O algoritmo genético não pode tratar os dados separadamente. Após a aplicação dos operadores genéticos (*cruzamento* e *mutação*) em um conjunto de dados bidimensional um valor de x pode *evoluir* e o mesmo pode não acontecer com um valor de y. A idéia é concatenar os indivíduos de x e y, gerando um novo e único indivíduo (Figura 5.1). Assim os operadores genéticos serão aplicados no par (xi, yi) como um todo e a evolução será do par e não apenas de uma das variáveis. A técnica é a mesma para os casos multidimensionais, ou seja, todas as variáveis (*indivíduos*) devem ser concatenadas para gerar um único indivíduo.

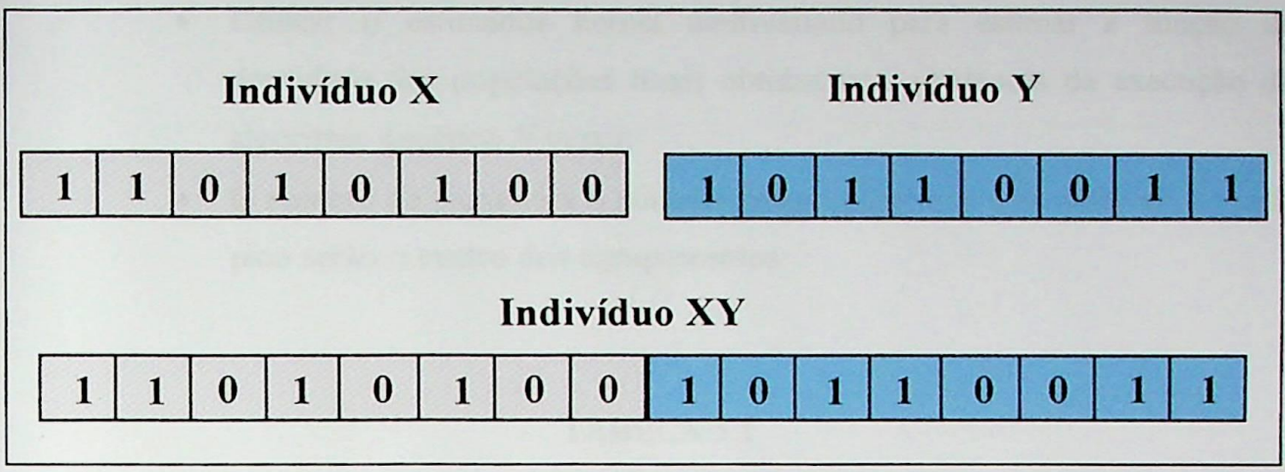

FIGURA 5.1 – Concatenação dos indivíduos x e y

5.2 Caso Bivariado

O algoritmo genético para esta implementação utiliza o estimador kernel multivariado para duas variáveis como função de aptidão.

No caso univariado o algoritmo genético foi configurado com um pequeno número de gerações. Para o caso bivariado este número deve ser aumentado. Para trabalhar com duas variáveis o algoritmo genético precisa de mais gerações para conseguir evoluir sua população. As taxas de cruzamento e mutação também foram alteradas para permitir que o algoritmo genético possa atingir todo o espaço de soluções. A configuração do algoritmo genético para o caso bivariado está definida na Tabela 5.1. Os passos desta técnica foram definidos da seguinte maneira:

- Definir o estimador kernel multivariado como função de aptidão;
- Definir duas populações inicias de tamanho pequeno (x e y);
- Definir um valor pequeno para o número máximo de gerações;
- Concatenar os indivíduos de x e y gerando um único indivíduo;
- Executar o algoritmo genético N vezes;
- Separar o indivíduo da população final obtendo novamente os indivíduos de x e y;

- Salvar a populações finais (x e y) a cada vez;
- Utilizar o estimador kernel multivariado para estimar a função de densidade das populações finais obtidas (x e y) depois da execução do algoritmo genético N vezes;
- O número de picos será o número de agrupamentos e as variáveis de cada pico serão o centro dos agrupamentos.

TABELA 5.1

Configuração do programa AG para o caso bidimensional

População Inicial	*Aleatória*
Tamanho da População	*16*
Número de Gerações	*100*
Número de Execuções do AG	*100*
Taxa de Cruzamento	*0.4*
Taxa de Mutação	*0.08*
Função de Aptidão	*Estimador Kernel Multivariado*
Precisão	*0.00001*
Tamanho do Indivíduo	*21*
Tamanho do Indivíduo xy	*42*

Para testar o algoritmo genético para o caso bivariado foram utilizados dois conjuntos de dados (Gráfico 5.1 e 5.2) gerados conforme as Tabela 5.2 e 5.3.

No primeiro conjunto de dados tem-se para as variáveis A e B médias 1, 5 e 10. Assim, espera-se encontrar três agrupamentos nas posições 1 e 1, 5 e 5, e 10 e 10.

TABELA 5.2

Parâmetros do conjunto de dados bidimensional – Primeiro caso

Variáveis	Média	Variância	Número de Pontos
A	1	1	300
	5	1	300
	10	1	300
B	1	1	300
	5	1	300
	10	1	300

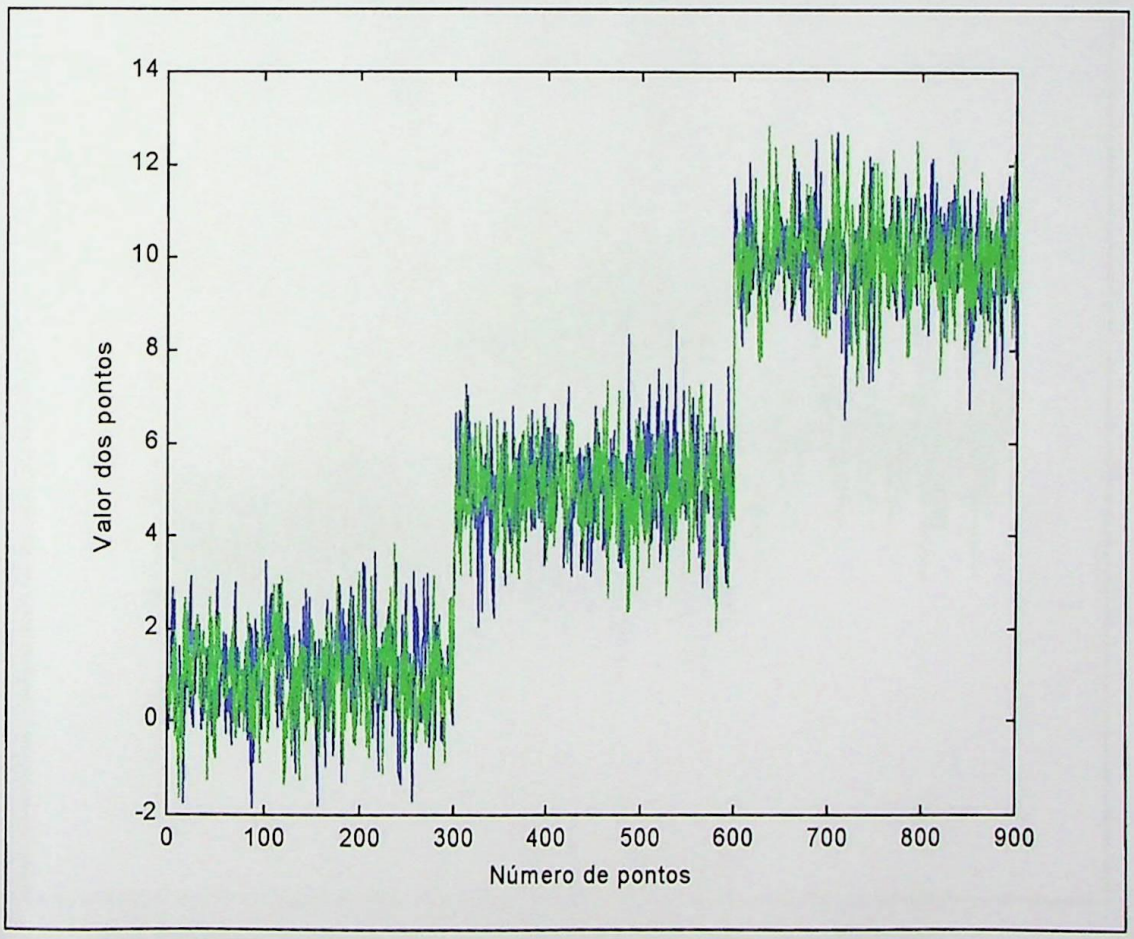

GRÁFICO 5.1 – Conjunto de dados bidimensionais – Primeiro caso

Para o segundo conjunto de dados tem-se para a variável A médias 5, 5 e 10, e para a variável B médias 5, 9 e 7. Assim, espera-se encontrar três agrupamentos nas posições 5 e 5, 5 e 9, e 10 e 7.

TABELA 5.3

Parâmetros do conjunto de dados bidimensional – Segundo caso

Variáveis	*Média*	*Variância*	*Número de Pontos*
A	5	1	300
	5	1	300
	10	1	300
B	5	1	300
	9	1	300
	7	1	300

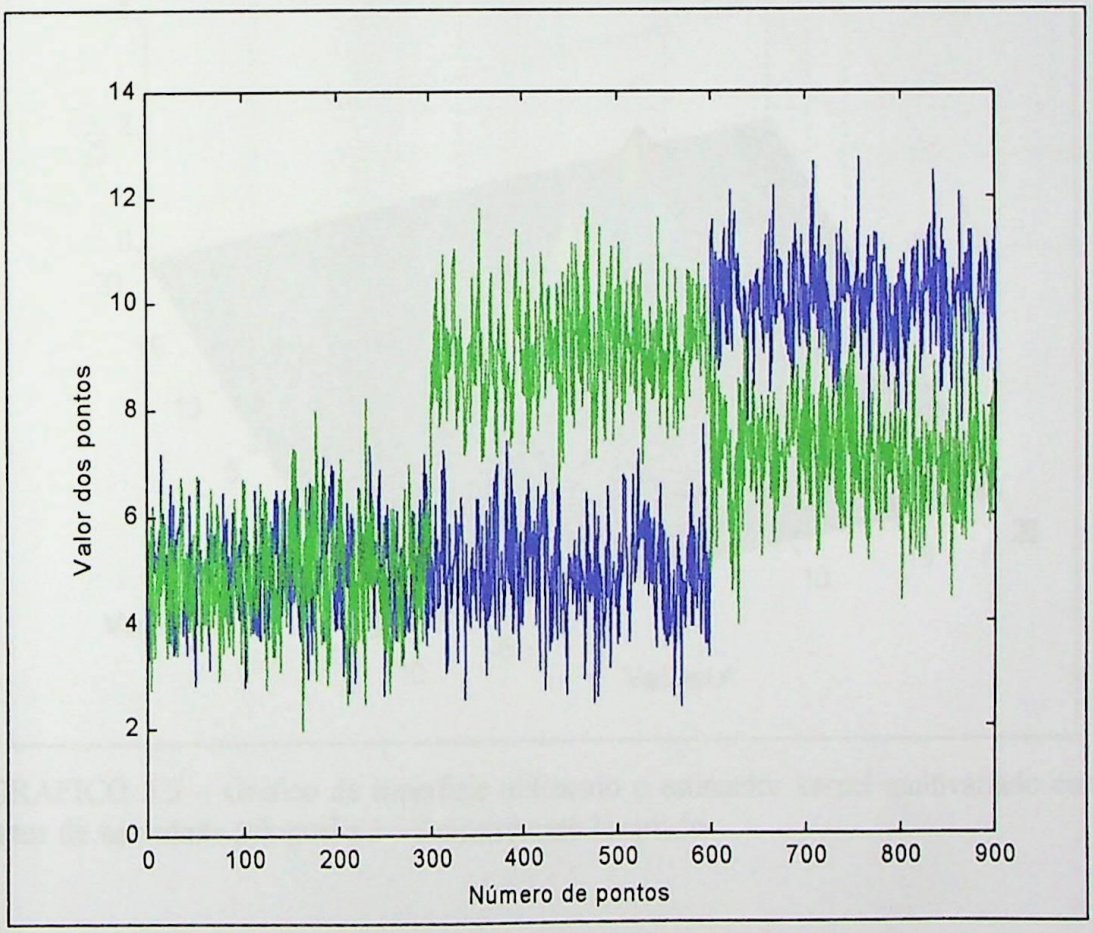

GRÁFICO 5.2 – Conjunto de dados bidimensionais – Segundo caso

Os resultados apresentados nos Gráficos 5.3 e 5.4 foram obtidos utilizando os dados do primeiro caso e apenas o estimador kernel multivariado.

Os resultados apresentados nos Gráficos 5.5 e 5.6 foram obtidos utilizando o estimador kernel multivariado e o algoritmo genético com a configuração que consta na Tabela 5.1. Pode-se observar que os agrupamentos apresentados nos Gráficos 5.3 e 5.4 também foram encontrados nos Gráficos 5.5 e 5.6.

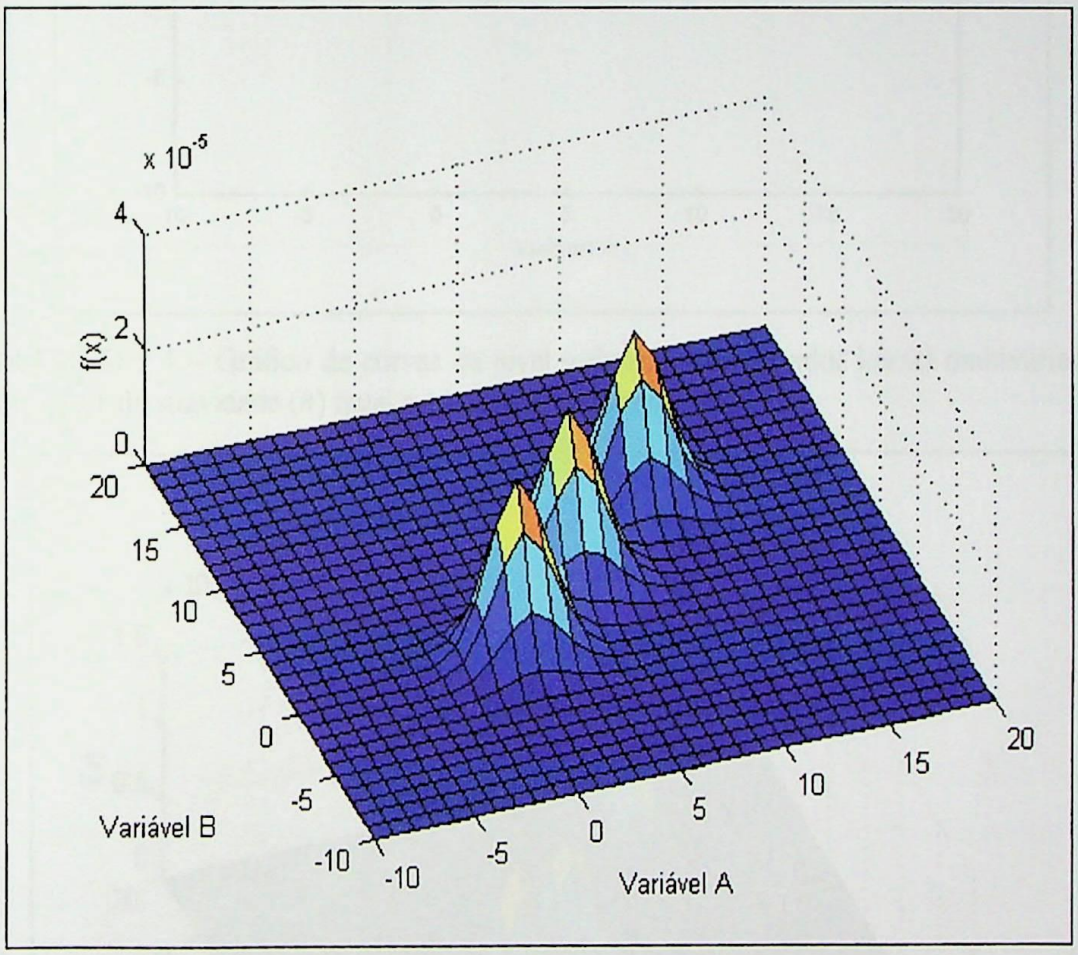

GRÁFICO 5.3 – Gráfico de superfície utilizando o estimador kernel multivariado com fator de suavidade (*h*) igual a 1 – Primeiro caso bivariado

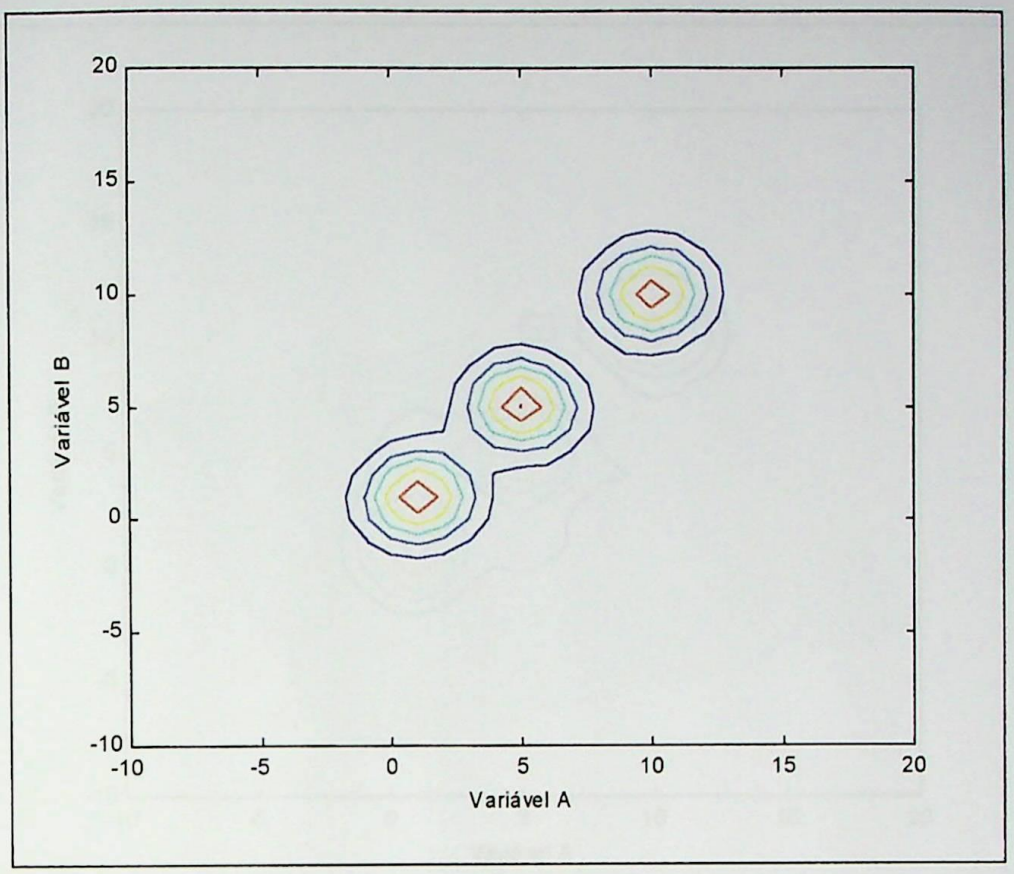

GRÁFICO 5.4 – Gráfico de curvas de nível utilizando o estimador kernel multivariado com fator de suavidade (*h*) igual a 1 – Primeiro caso bivariado

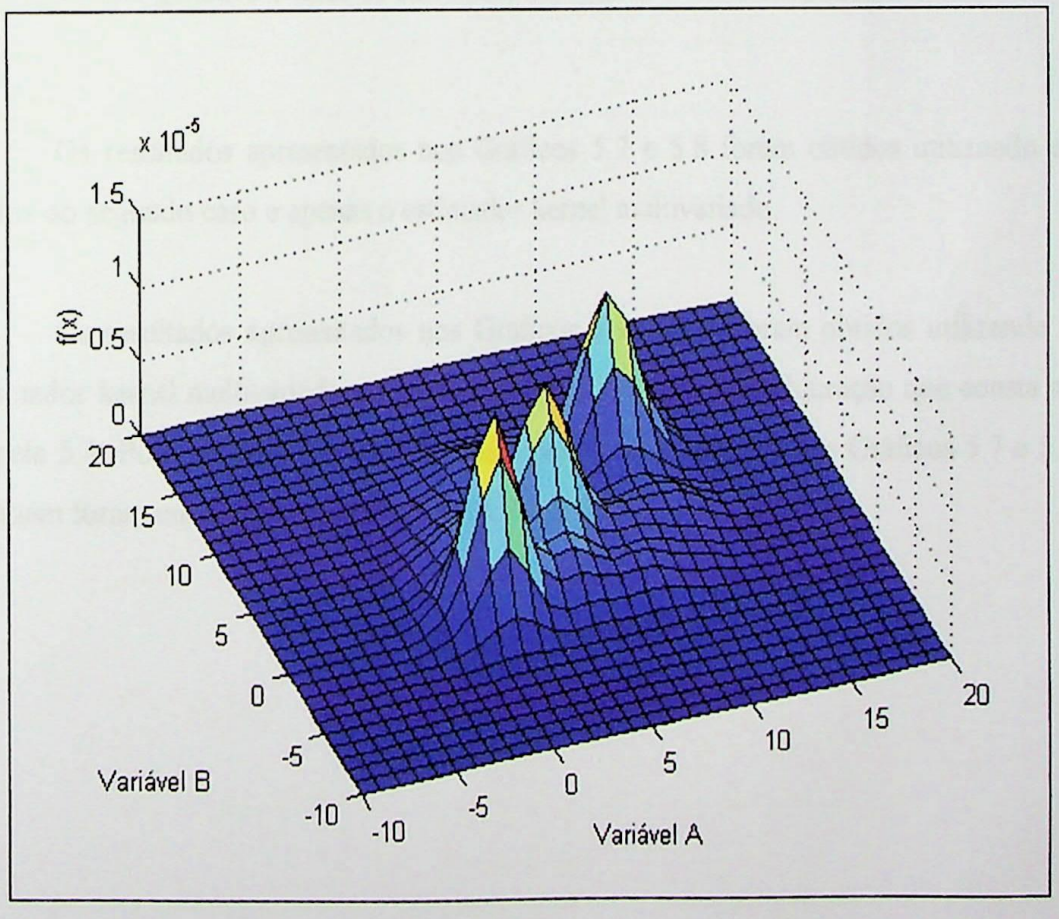

GRÁFICO 5.5 – Gráfico de superfície utilizando o estimador kernel multivariado com algoritmo genético e fator de suavidade (*h*) igual a 1 – Primeiro caso bivariado

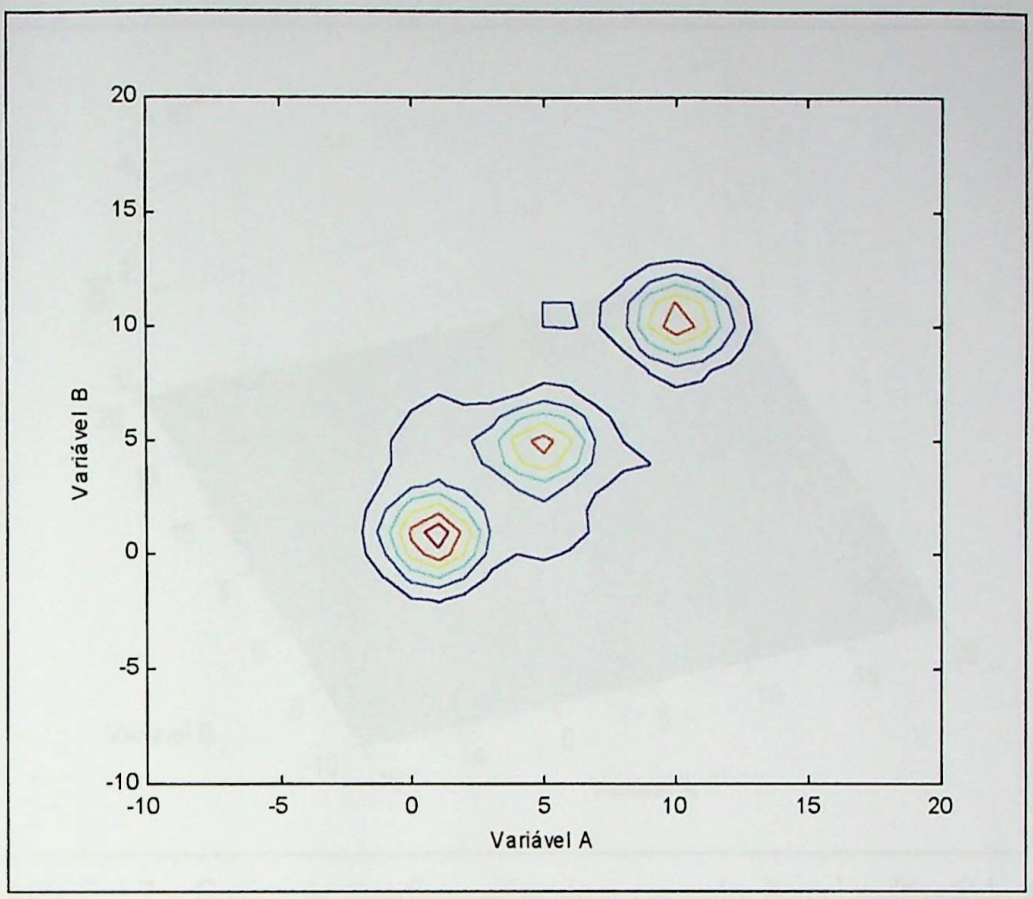

GRÁFICO 5.6 – Gráfico de curvas de nível utilizando o estimador kernel multivariado com algoritmo genético e fator de suavidade (*h*) igual a 1 – Primeiro caso bivariado

Os resultados apresentados nos Gráficos 5.7 e 5.8 foram obtidos utilizando os dados do segundo caso e apenas o estimador kernel multivariado.

Os resultados apresentados nos Gráficos 5.9 e 5.10 foram obtidos utilizando o estimador kernel multivariado e o algoritmo genético com a configuração que consta na Tabela 5.1. Pode-se observar que os agrupamentos apresentados nos Gráficos 5.7 e 5.8 também foram encontrados nos Gráficos 5.9 e 5.10.

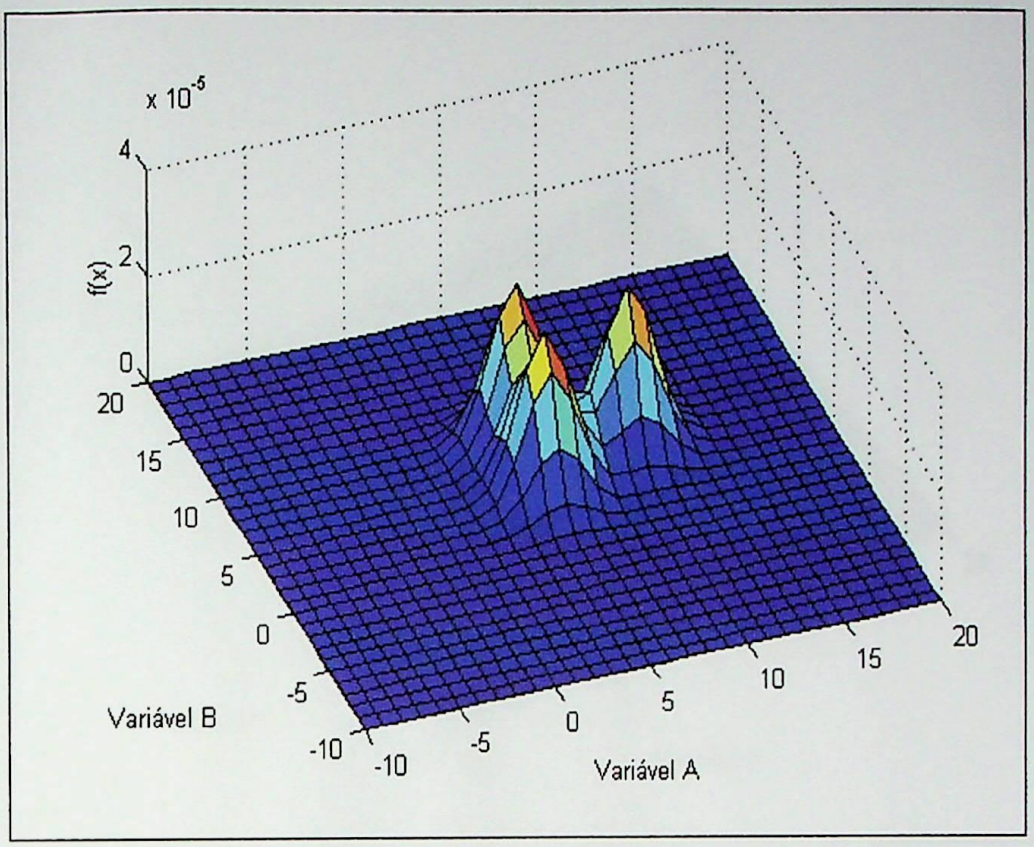

GRÁFICO 5.7 – Gráfico de superfície utilizando o estimador kernel multivariado com fator de suavidade (*h*) igual a 1 – Segundo caso bivariado

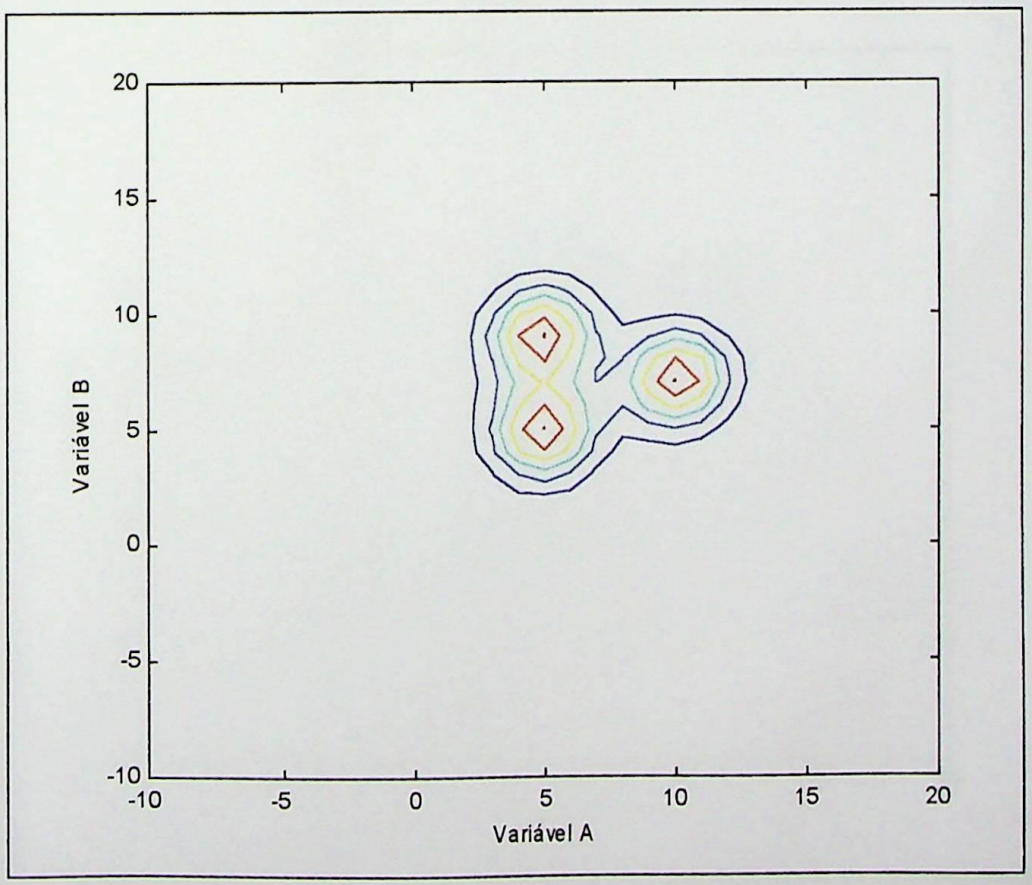

GRÁFICO 5.8 – Gráfico de curvas de nível utilizando o estimador kernel multivariado com fator de suavidade (*h*) igual a 1 – Segundo caso bivariado

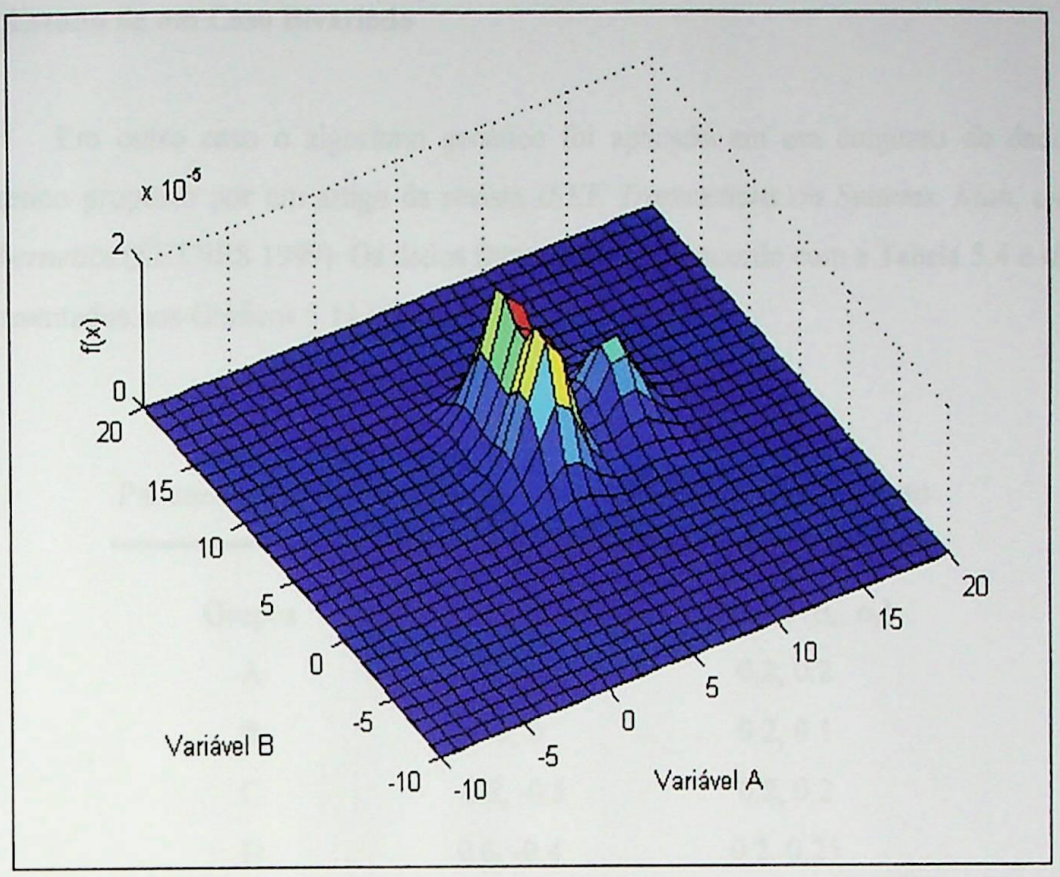

GRÁFICO 5.9 – Gráfico de superfície utilizando o estimador kernel multivariado com algoritmo genético e fator de suavidade (*h*) igual a 1 – Segundo caso bivariado

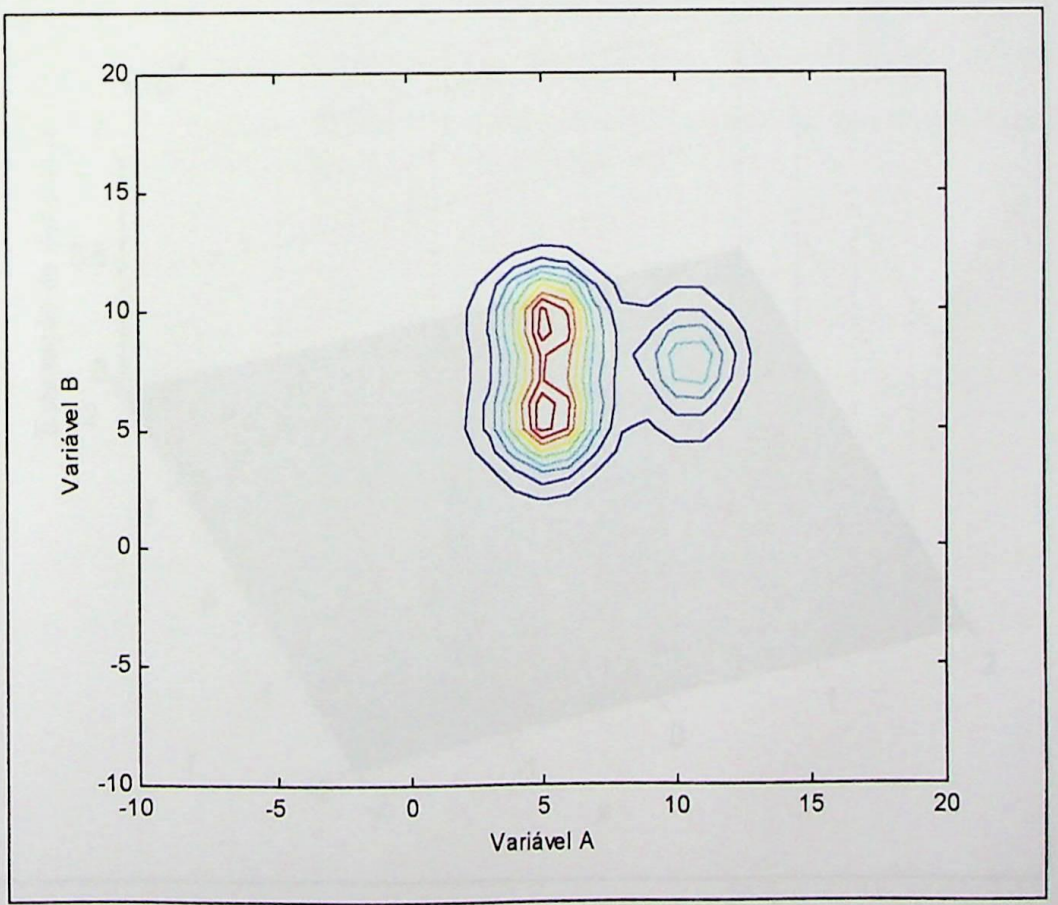

GRÁFICO 5.10 – Gráfico de curvas de nível utilizando o estimador kernel multivariado com algoritmo genético e fator de suavidade (*h*) igual a 1 – Segundo caso bivariado

5.3 Estudo de um Caso Bivariado

Em outro caso o algoritmo genético foi aplicado em um conjunto de dados sintético proposto por um artigo da revista *IEEE Transactions on Systems, Man, and Cybernetics* (SETNES 1999). Os dados foram gerados de acordo com a Tabela 5.4 e são apresentados nos Gráficos 5.11 e 5.12.

TABELA 5.4

Parâmetros do conjunto de dados bidimensional – Caso proposto

Grupos	Centro do grupo (x, y)	Variância (σ_x, σ_y)
A	0.5, 0.5	0.2, 0.2
B	0.1, 0	0.2, 0.1
C	-0.5, -0.5	0.2, 0.2
D	0.6, -0.4	0.2, 0.25

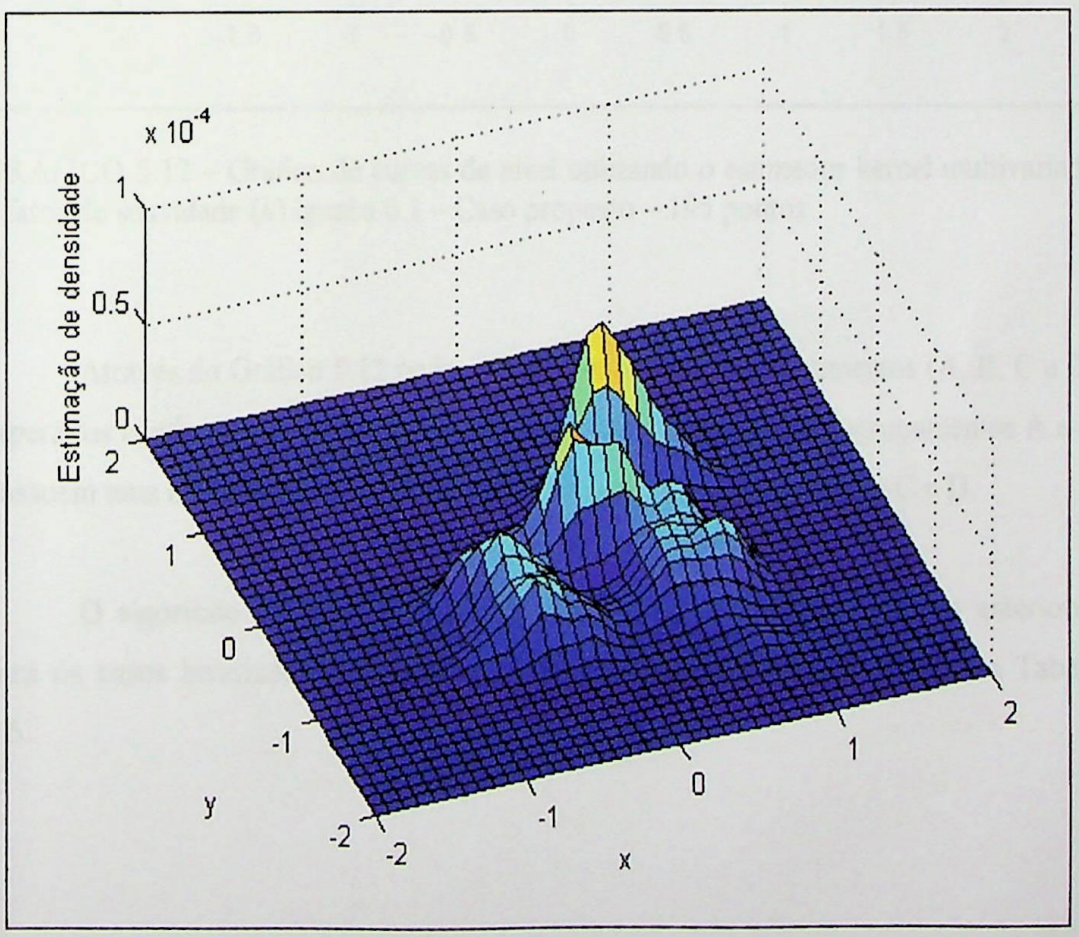

GRÁFICO 5.11 – Gráfico de superfície utilizando o estimador kernel multivariado com fator de suavidade (h) igual a 0.1 – Caso proposto – 195 pontos

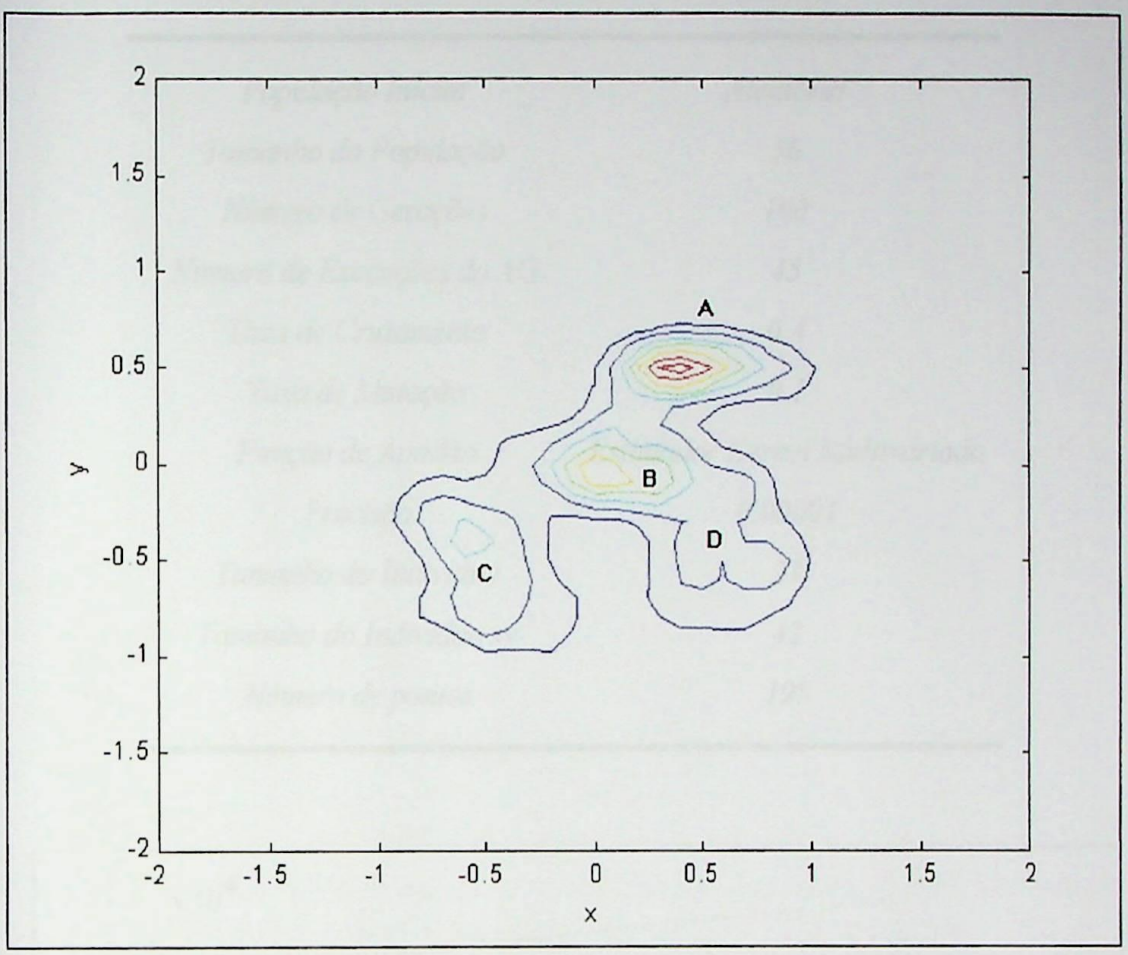

GRÁFICO 5.12 – Gráfico de curvas de nível utilizando o estimador kernel multivariado e fator de suavidade (h) igual a 0.1 – Caso proposto – 195 pontos

Através do Gráfico 5.12 pode-se perceber os quatro agrupamentos (A, B, C e D) esperados conforme a Tabela 5.4. Pode-se observar também que os agrupamentos A e B possuem uma maior densidade e definição em relação aos agrupamentos C e D.

O algoritmo genético aplicado foi o mesmo utilizado nos exemplos anteriores para os casos bivariados com algumas alterações na configuração conforme a Tabela 5.5.

TABELA 5.5

Configuração do programa AG para o caso bidimensional proposto

População Inicial	*Aleatória*
Tamanho da População	*36*
Número de Gerações	*100*
Número de Execuções do AG	*45*
Taxa de Cruzamento	*0.4*
Taxa de Mutação	*0.1*
Função de Aptidão	*Estimador Kernel Multivariado*
Precisão	*0.00001*
Tamanho do Indivíduo	*21*
Tamanho do Indivíduo xy	*42*
Número de pontos	*195*

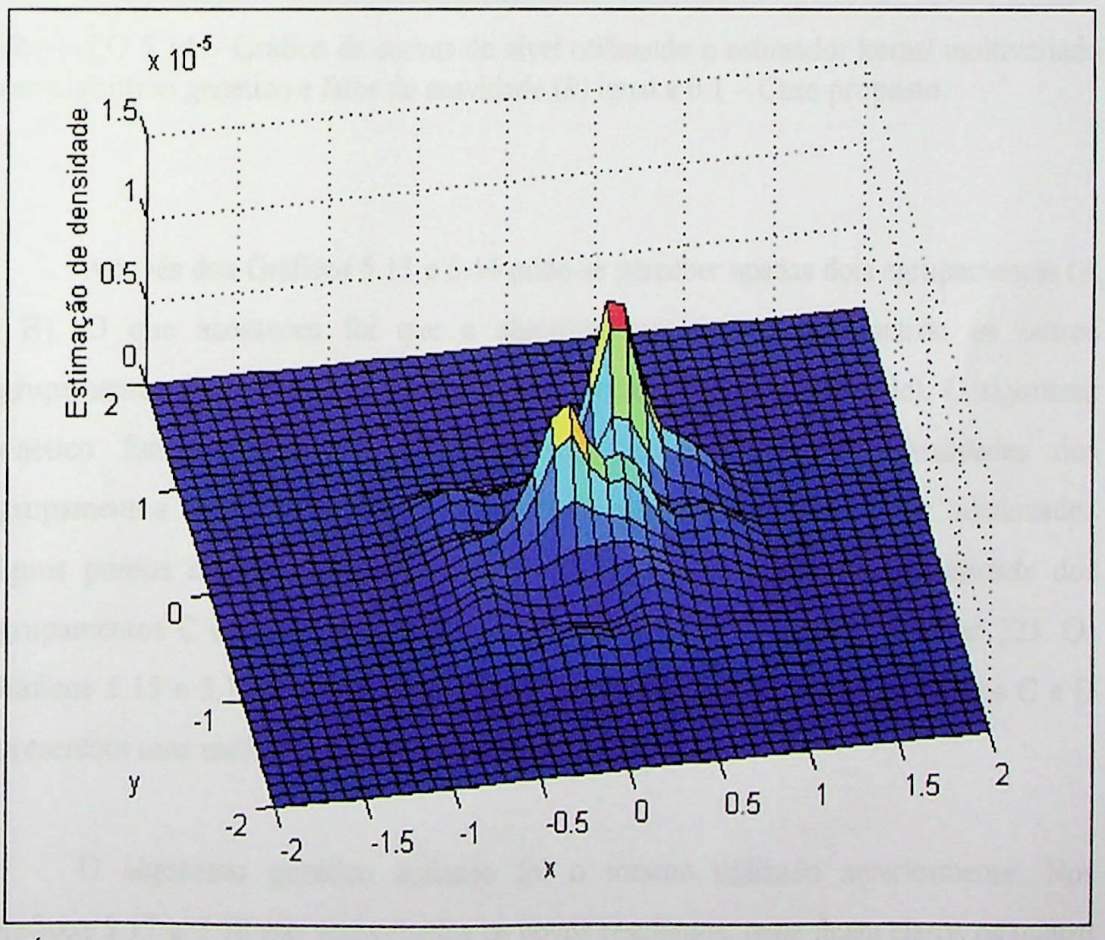

GRÁFICO 5.13 – Gráfico de superfície utilizando o estimador kernel multivariado com algoritmo genético e fator de suavidade (*h*) igual a 0.1 – Caso proposto

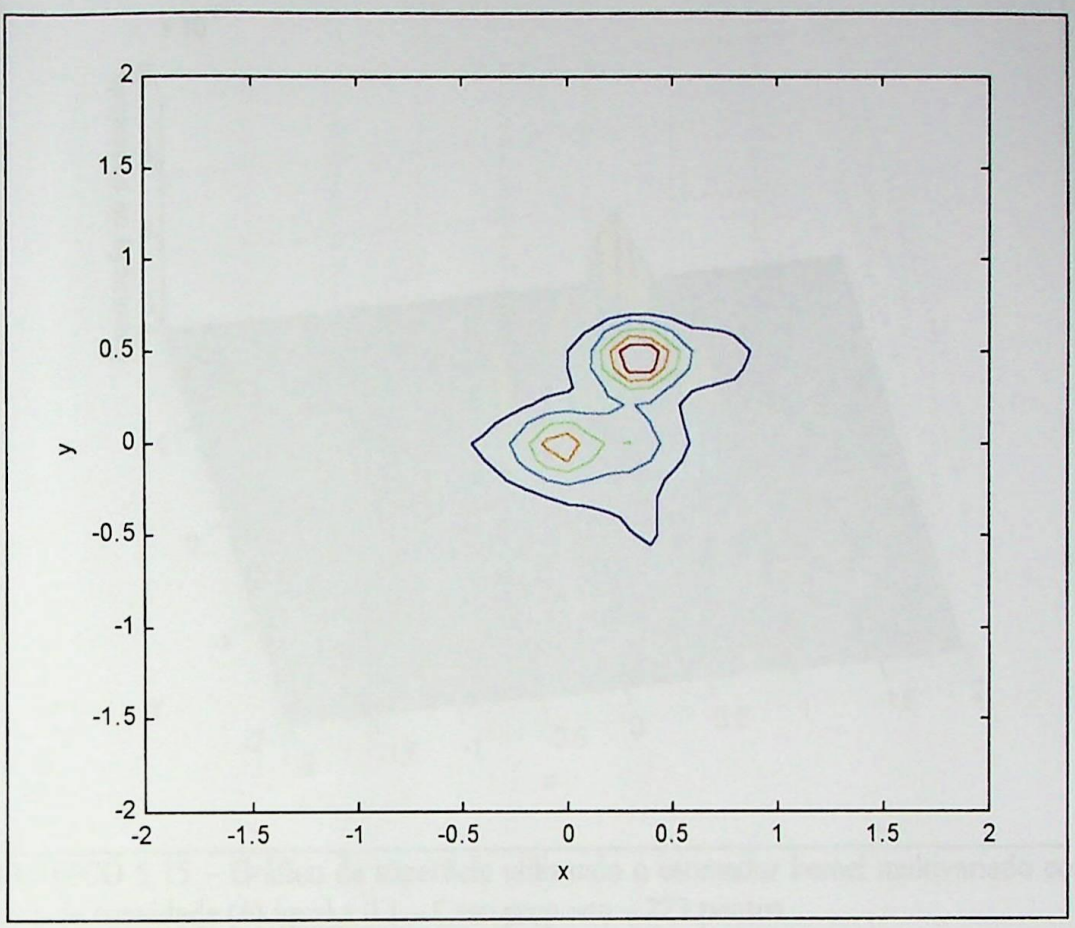

GRÁFICO 5.14 – Gráfico de curvas de nível utilizando o estimador kernel multivariado com algoritmo genético e fator de suavidade (h) igual a 0.1 – Caso proposto

Através dos Gráficos 5.13 e 5.14 pode-se perceber apenas dois agrupamentos (A e B). O que aconteceu foi que o algoritmo genético não encontrou os outros agrupamentos (C e D) por não estarem bem definidos (baixa densidade). O algoritmo genético funciona bem quando existe um equilíbrio entre as densidades dos agrupamentos analisados. Para se comprovar esta característica foram adicionados alguns pontos no conjunto de dados com o objetivo de aumentar a densidade dos agrupamentos C e D. O conjunto de dados que tinha 195 pontos passou a ter 223. Os Gráficos 5.15 e 5.16 mostram o novo conjunto de dados onde os agrupamentos C e D apresentam uma melhor definição (maior densidade).

O algoritmo genético aplicado foi o mesmo utilizado anteriormente. Nos Gráficos 5.17 e 5.18 são apresentados os novos resultados, onde ficam claros os quatro agrupamentos esperados.

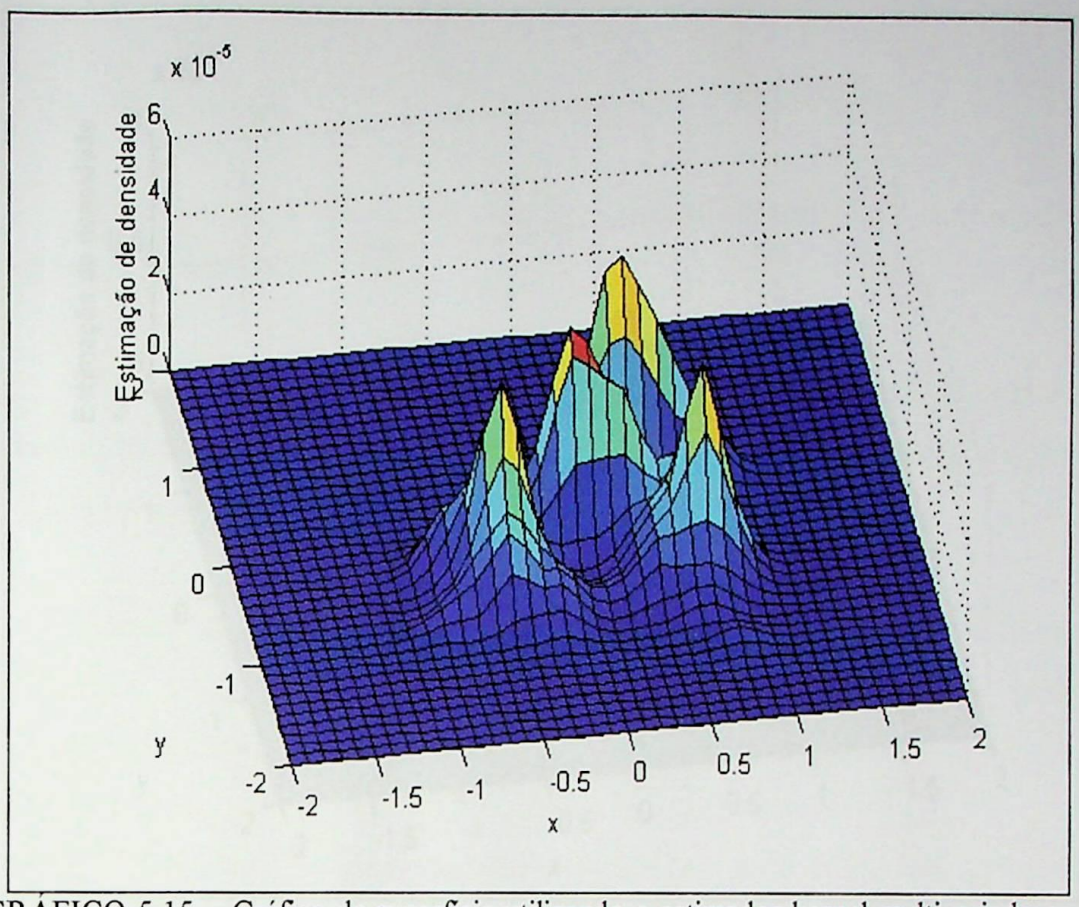

GRÁFICO 5.15 – Gráfico de superfície utilizando o estimador kernel multivariado com fator de suavidade (*h*) igual a 0.1 – Caso proposto – 223 pontos

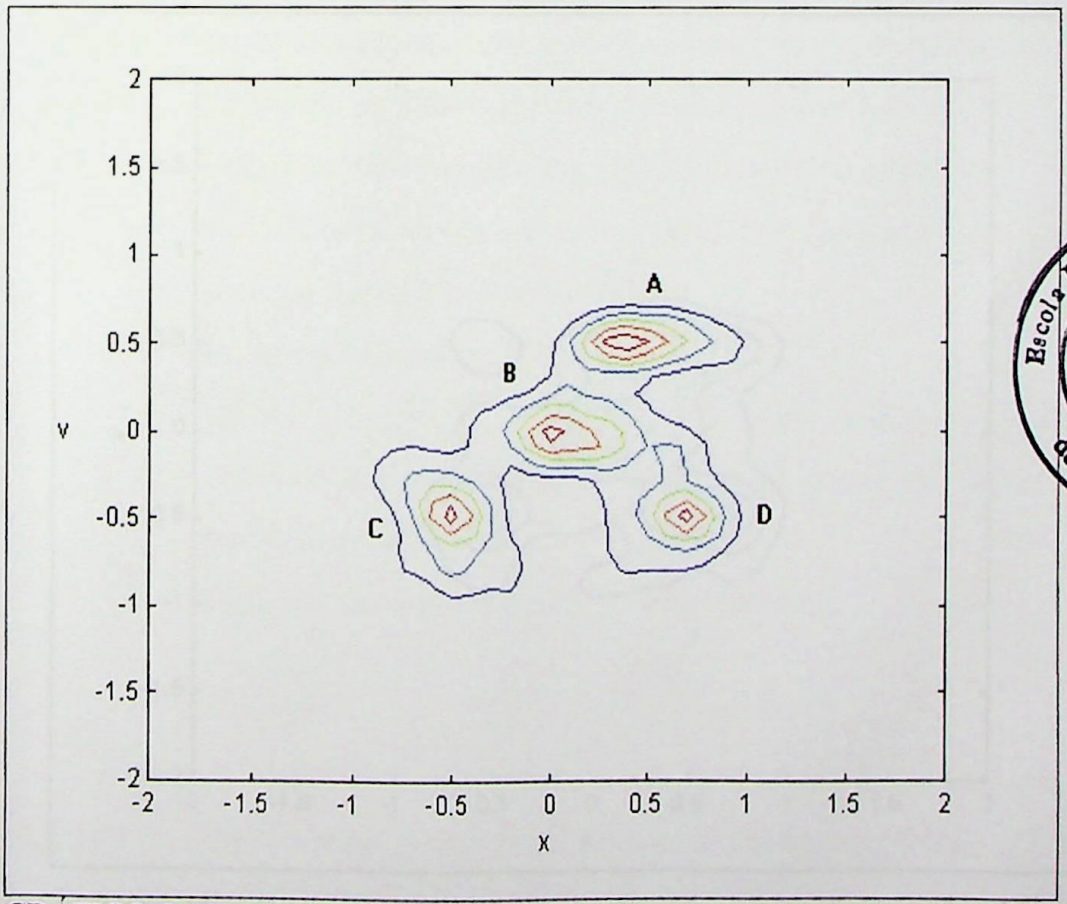

GRÁFICO 5.16 – Gráfico de curvas de nível utilizando o estimador kernel multivariado e fator de suavidade (*h*) igual a 0.1 – Caso proposto – 223 pontos

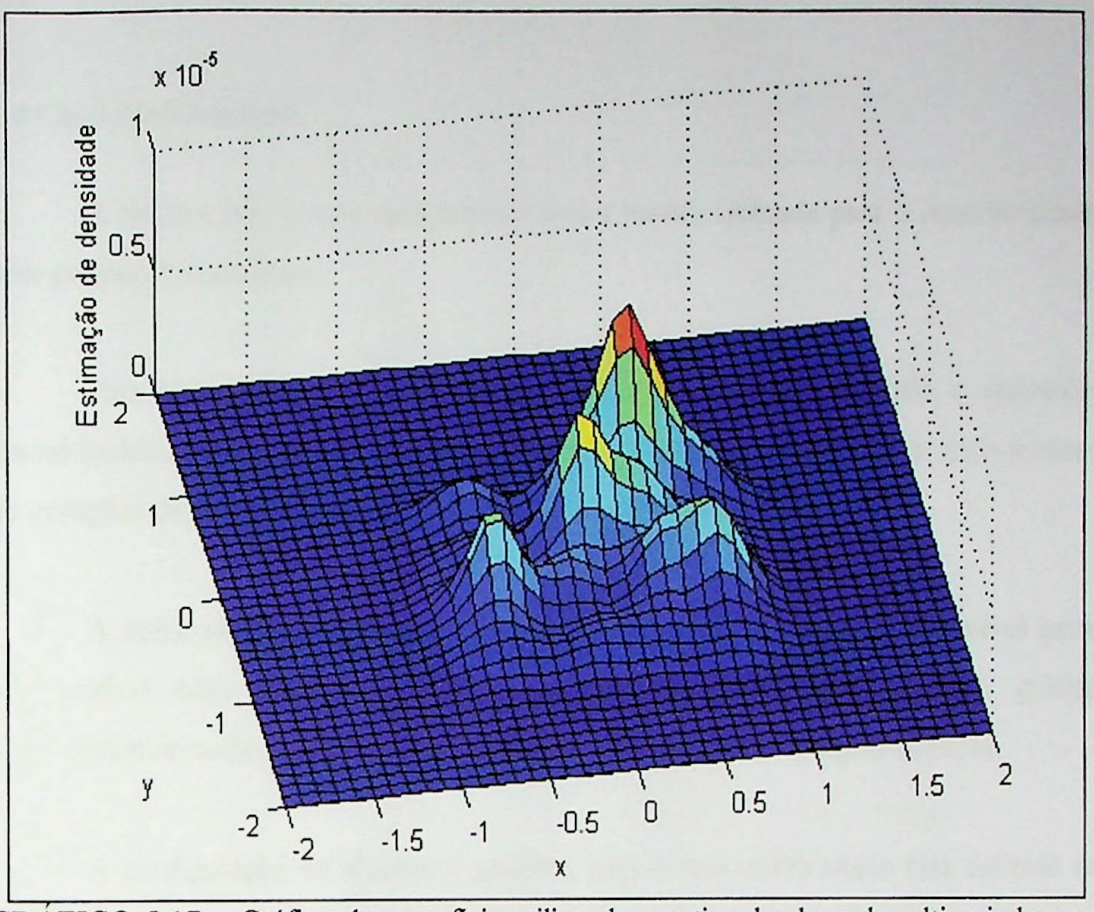

GRÁFICO 5.17 – Gráfico de superfície utilizando o estimador kernel multivariado com algoritmo genético e fator de suavidade (*h*) igual a 0.1 – Caso proposto

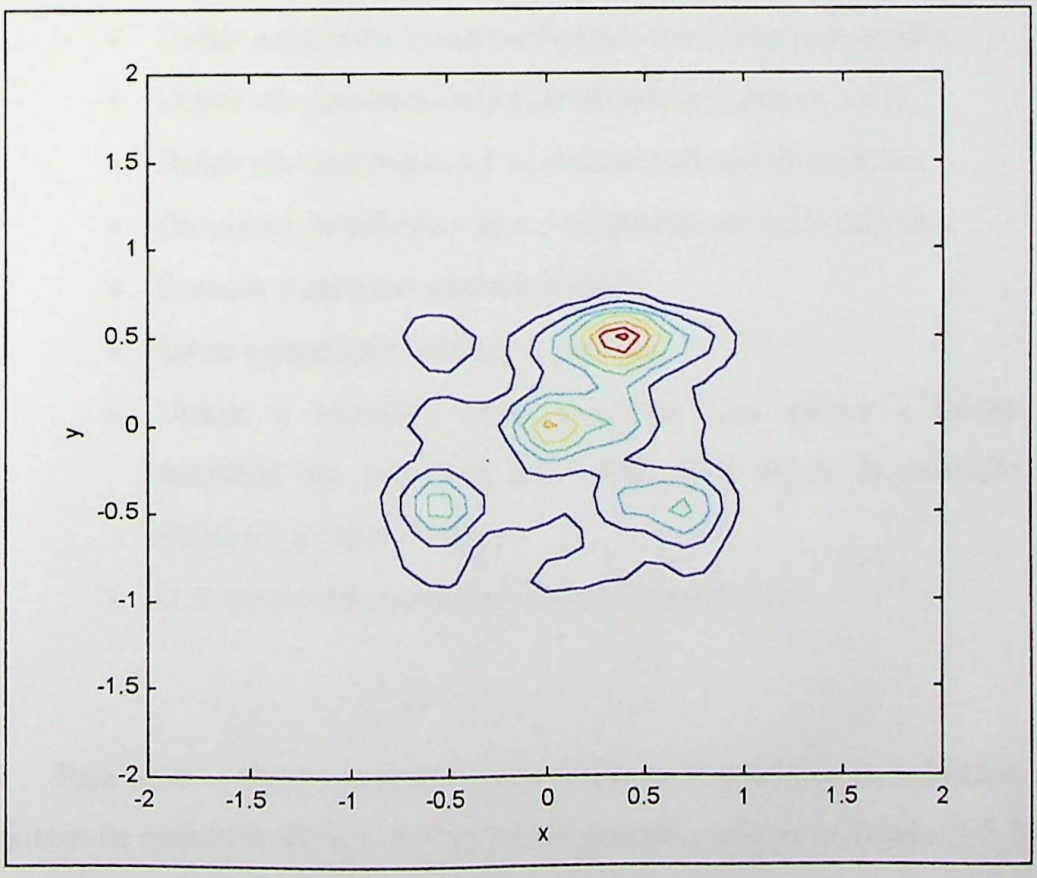

GRÁFICO 5.18 – Gráfico de curvas de nível utilizando o estimador kernel multivariado com algoritmo genético e fator de suavidade (*h*) igual a 0.1 – Caso proposto

5.4 Caso Multivariado

A técnica para o caso multivariado será a mesma utilizada para o caso bivariado com pequenas alterações.

Para os testes o algoritmo genético foi implementado utilizando o estimador kernel multivariado para três variáveis. Por causa deste aumento de variáveis o número de gerações deverá ser aumentado em relação ao caso bivariado.

A outra alteração está relacionada à geração dos gráficos. Não é possível gerar um gráfico com mais de 2 variáveis. Neste caso será utilizado um gráfico unidimensional onde os picos apresentados indicarão o número de agrupamentos.

A configuração do algoritmo genético para o caso multivariado está definida na Tabela 5.6. Os passos desta técnica foram definidos da seguinte maneira:

- Definir o estimador kernel multivariado como função de aptidão;
- Definir três populações inicias de tamanho pequeno (x, y e z);
- Definir um valor pequeno para o número máximo de gerações;
- Concatenar os indivíduos de x, y e z gerando um único indivíduo;
- Executar o algoritmo genético N vezes;
- Salvar a população final (xyz) a cada vez;
- Utilizar o estimador kernel univariado para estimar a função de densidade das população final obtida (xyz) depois da execução do algoritmo genético N vezes;
- O número de picos será o número de agrupamentos.

Para testar o algoritmo genético para o caso multivariado foram utilizados três conjuntos de dados (Gráficos 5.19, 5.21 e 5.23) gerados conforme as Tabelas 5.7, 5.8 e 5.9.

TABELA 5.6

Configuração do programa AG para o caso multidimensional

População Inicial	*Aleatória*
Tamanho da População	*12*
Número de Gerações	*1000*
Número de Execuções do AG	*100*
Taxa de Cruzamento	*0.4*
Taxa de Mutação	*0.08*
Função de Aptidão	*Estimador Kernel Multivariado*
Precisão	*0.00001*
Tamanho do Indivíduo	*21*
Tamanho do Indivíduo xyz	*63*

No primeiro conjunto de dados (Gráfico 5.19) foram geradas duas médias para cada variável. Assim, espera-se encontrar dois agrupamentos. No Gráfico 5.20 pode-se perceber claramente dois picos correspondentes aos agrupamentos esperados.

TABELA 5.7

Parâmetros do conjunto de dados multidimensional – Primeiro caso

Variáveis	Média	Variância	Número de Pontos
A	1	1	50
	9	1	50
B	5	1	50
	5	1	50
C	3	1	50
	7	1	50

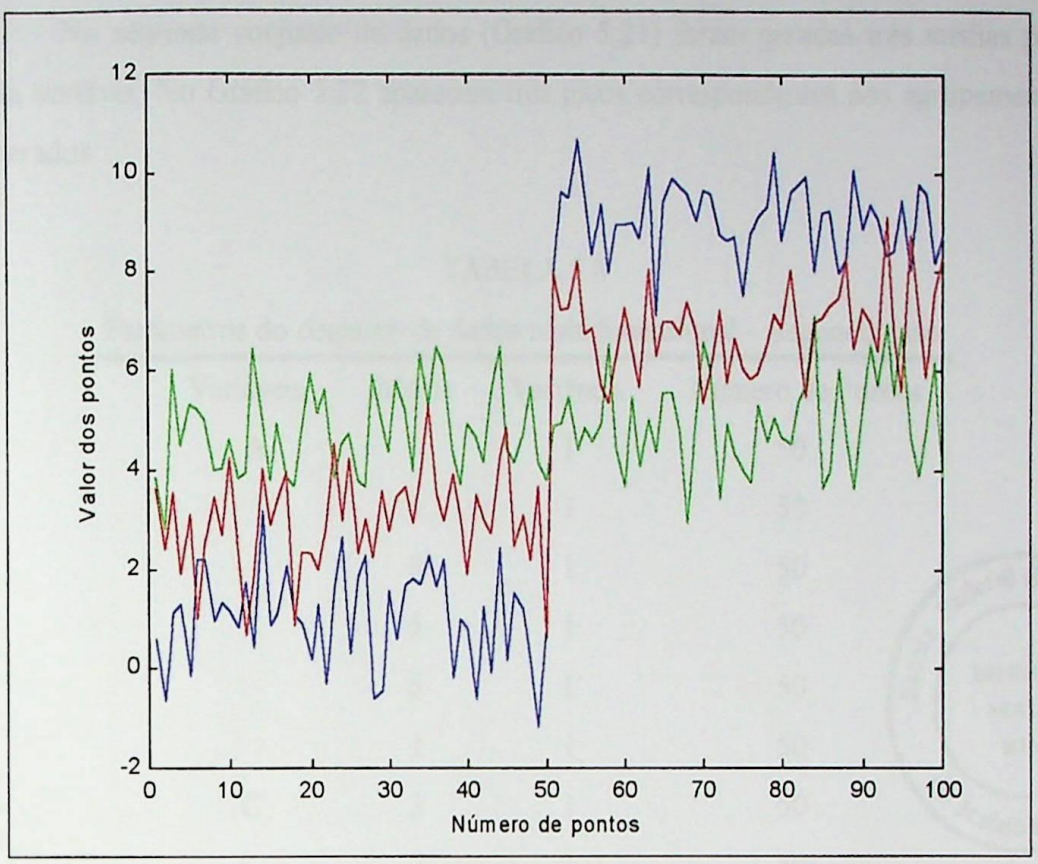

GRÁFICO 5.19 – Conjunto de dados multidimensional – Primeiro caso

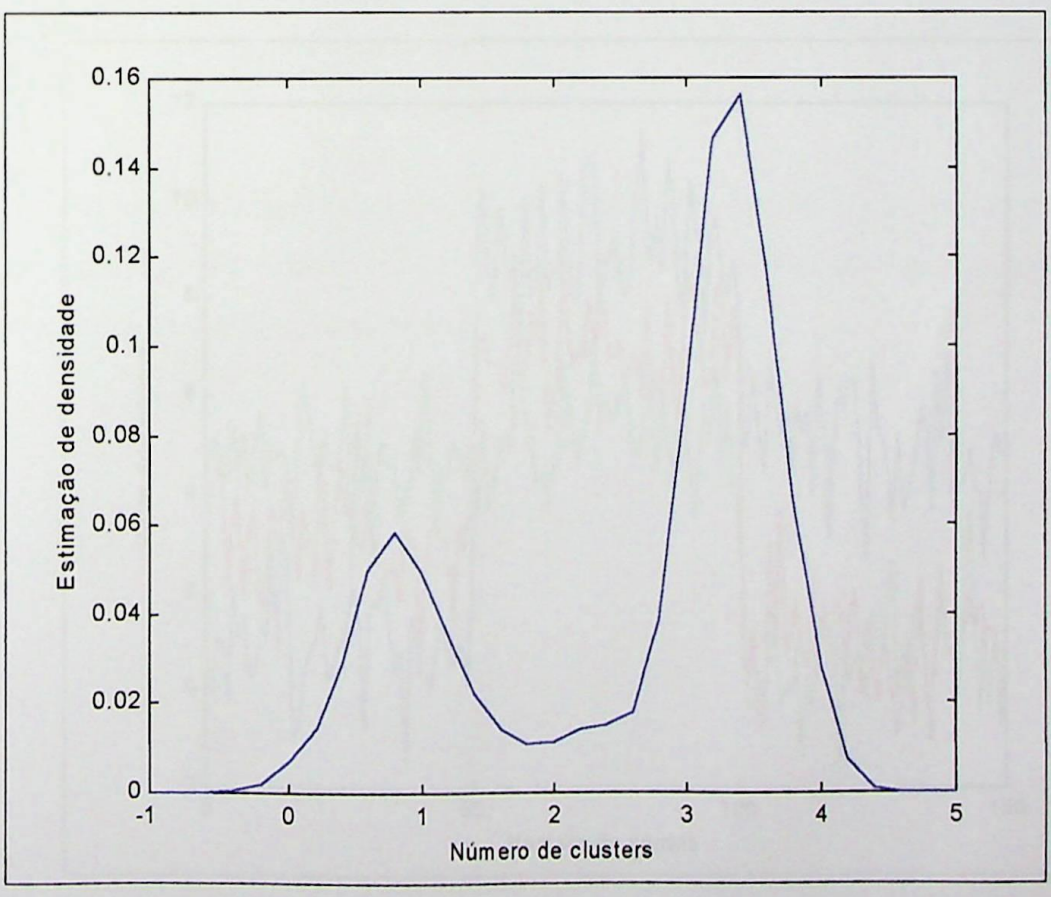

GRÁFICO 5.20 – Número de agrupamentos do primeiro caso – Parâmetro de suavidade (*h*) igual a 0.2

No segundo conjunto de dados (Gráfico 5.21) foram geradas três médias para cada variável. No Gráfico 5.22 aparecem três picos correspondentes aos agrupamentos esperados.

TABELA 5.8

Parâmetros do conjunto de dados multidimensional – Segundo caso

Variáveis	Média	Variância	Número de Pontos
A	1	1	50
	9	1	50
	5	1	50
B	5	1	50
	5	1	50
	1	1	50
C	3	1	50
	7	1	50
	1	1	50

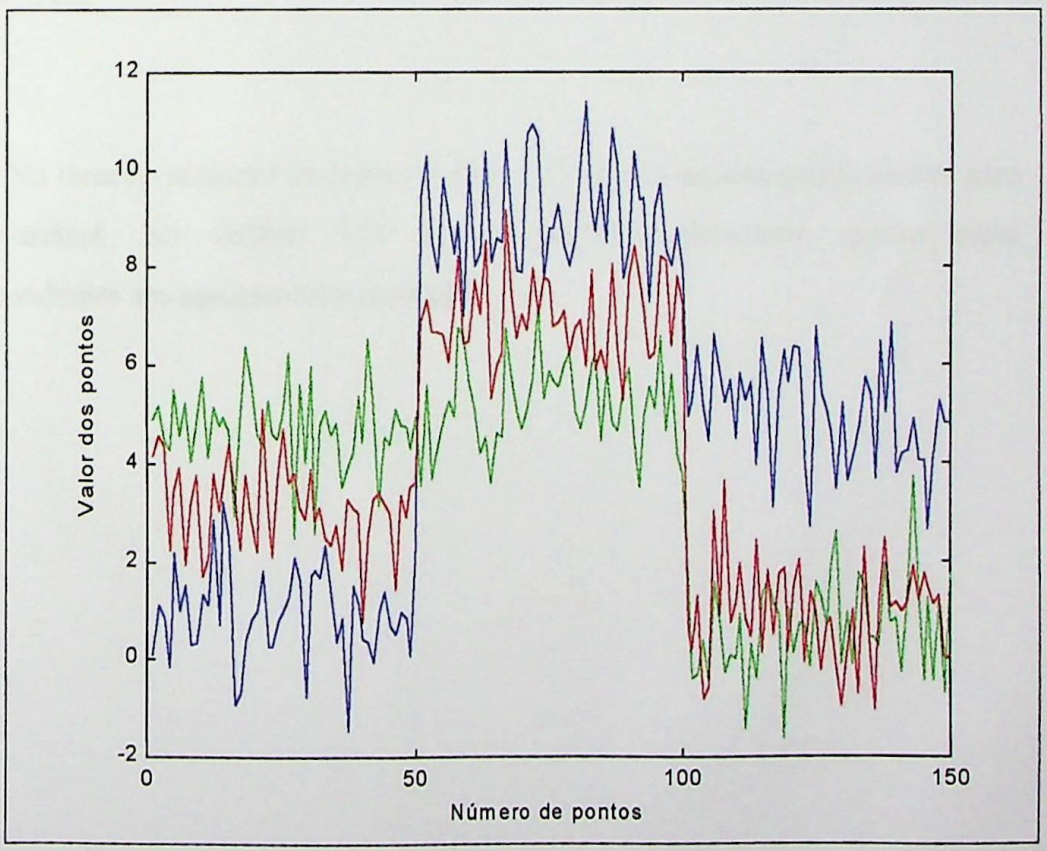

GRÁFICO 5.21 – Conjunto de dados multidimensional – Segundo caso

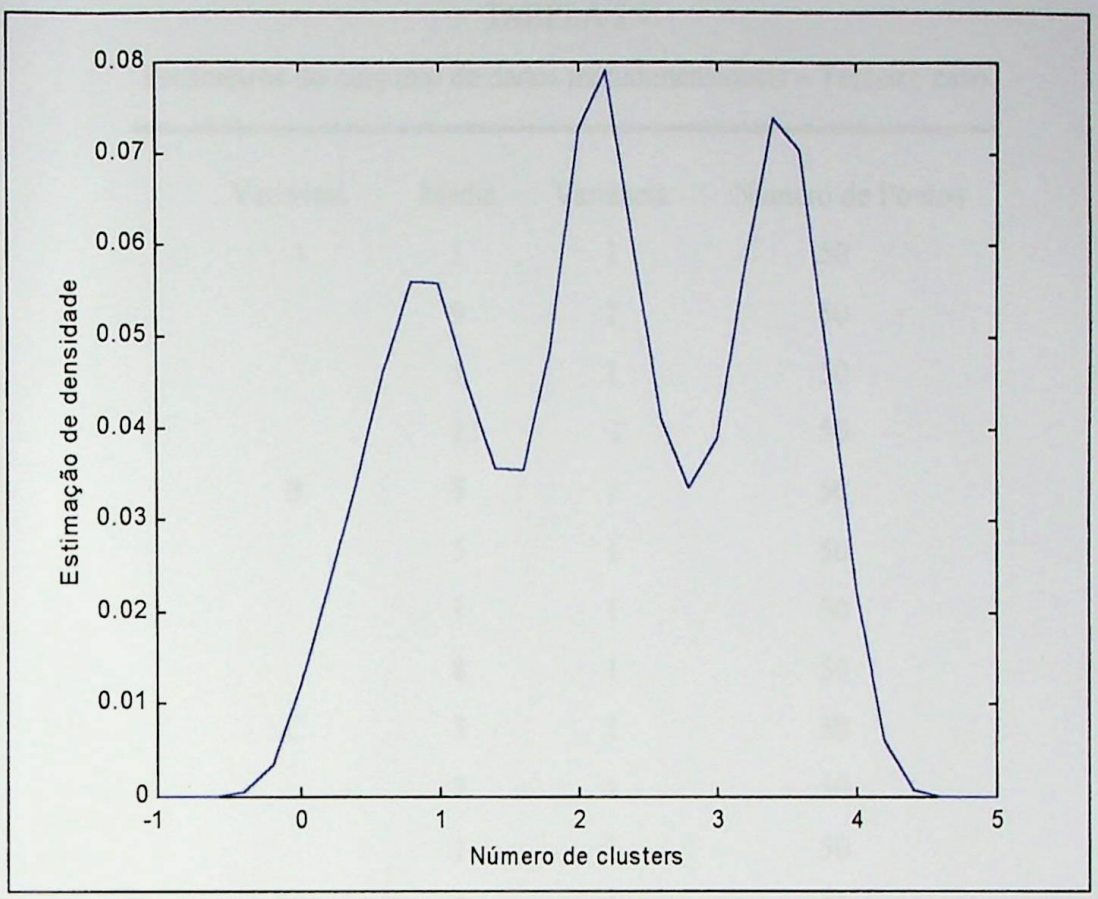

GRÁFICO 5.22 – Número de agrupamentos do segundo caso – Parâmetro de suavidade (*h*) igual a 0.2

No terceiro conjunto de dados (Gráfico 5.23) foram geradas quatro médias para cada variável. No Gráfico 5.24 pode-se perceber claramente quatro picos correspondentes aos agrupamentos esperados.

TABELA 5.9

Parâmetros do conjunto de dados multidimensionais – Terceiro caso

Variáveis	Média	Variância	Número de Pontos
A	1	1	50
	9	1	50
	5	1	50
	13	1	50
B	5	1	50
	5	1	50
	1	1	50
	8	1	50
C	3	1	50
	7	1	50
	1	1	50
	1	1	50

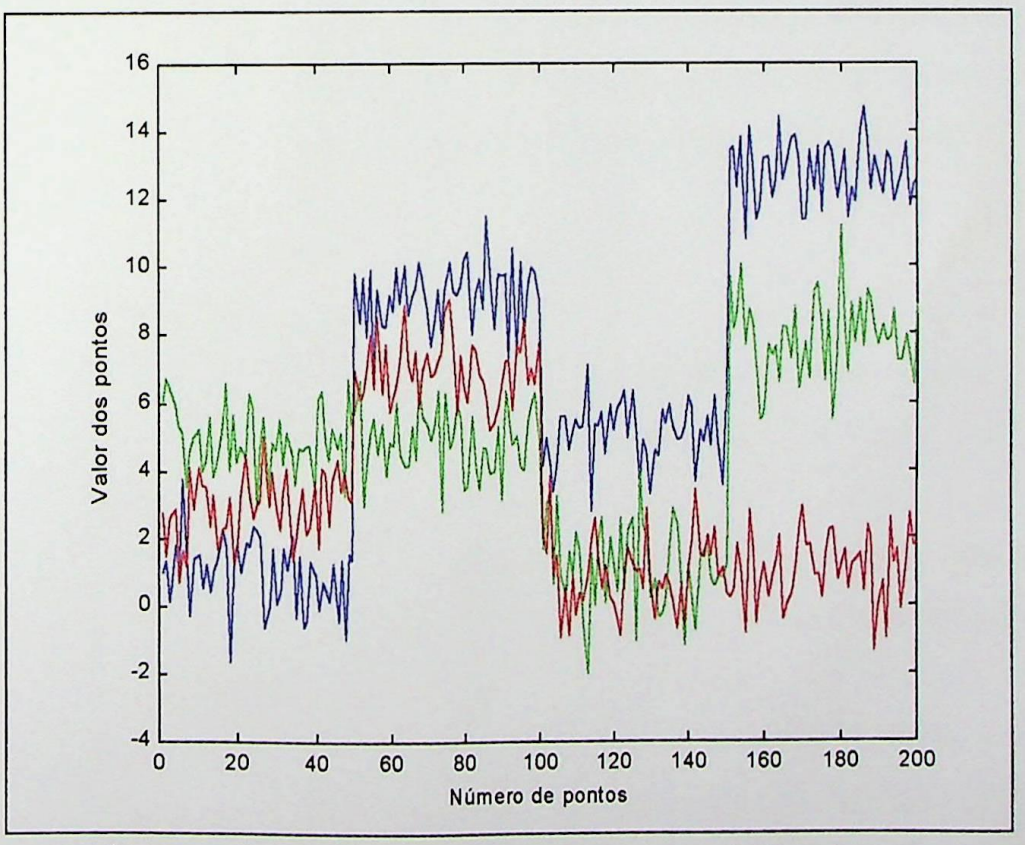

GRÁFICO 5.23 – Conjunto de dados multidimensional – Terceiro caso

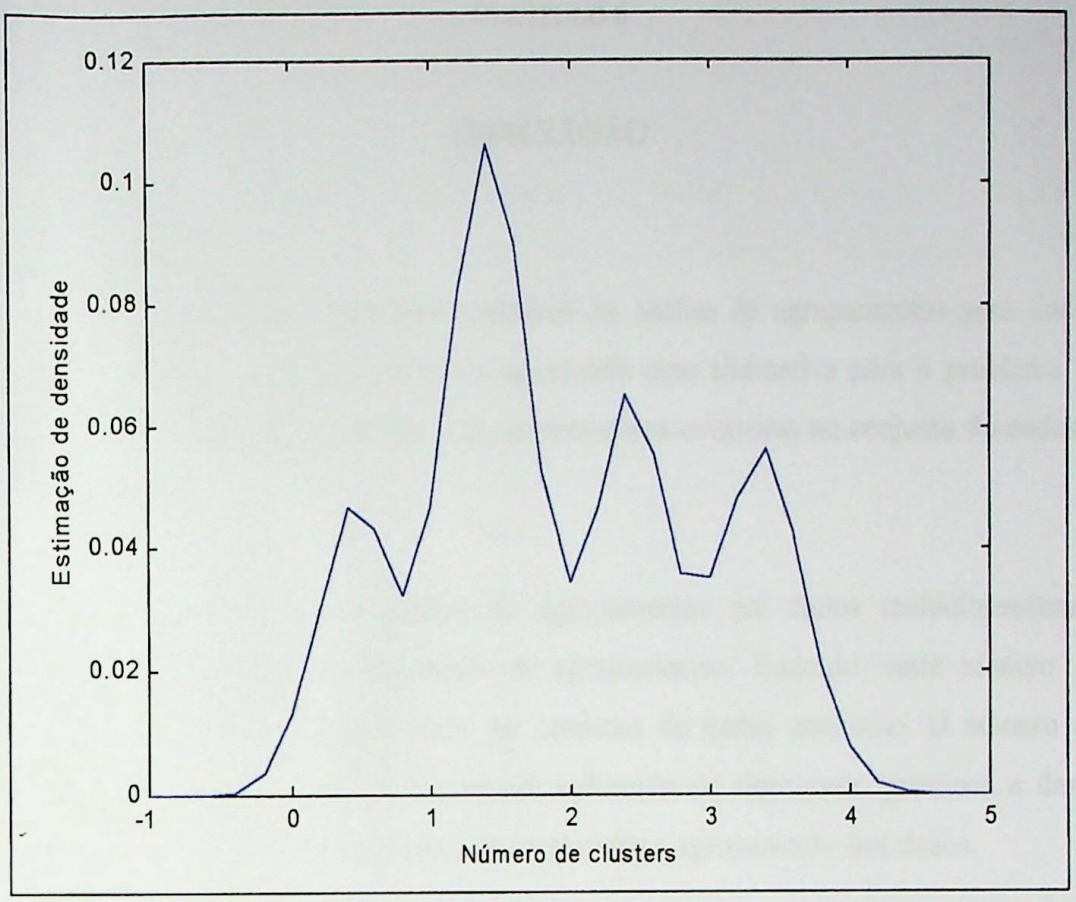

GRÁFICO 5.24 – Número de agrupamentos do terceiro caso – Parâmetro de suavidade (*h*) igual a 0.2

CAPÍTULO 6

CONCLUSÃO

A aplicação dos algoritmos genéticos na análise de agrupamentos para dados multidimensionais se mostrou eficiente apontando uma alternativa para o problema da falta de informação sobre o número de agrupamentos existentes no conjunto de dados a ser analisado.

Os métodos para a análise de agrupamentos em dados multidimensionais utilizam de um número aproximado de agrupamentos. Baseado neste número os métodos localizam os agrupamentos no conjunto de dados analisado. O número de agrupamentos agora pode ser conseguido utilizando os algoritmos genéticos e dessa forma conseguir um melhor resultado na localização e agrupamento dos dados.

No caso bidimensional os algoritmos genéticos encontraram uma pequena dificuldade com relação à definição dos agrupamentos. A desigualdade da densidade entre os agrupamentos mostrou que em certos casos a função kernel apresenta uma melhor realidade dos dados estudados que com o uso dos algoritmos genéticos.

Como desenvolvimentos futuros pode-se sugerir:

a) Utilizar o algoritmo genético apresentado neste trabalho com outras funções kernel que não a função Gaussiana (Triangular, Epanechnikov, etc.) para verificar e comparar a precisão e eficiência do método;
b) Efetuar alterações no algoritmo genético proposto como, por exemplo, estabelecer múltiplos pontos de cruzamento. O objetivo seria de melhorar o desempenho do algoritmo genético;
c) Tentar uma fusão do algoritmo genético proposto neste trabalho com a lógica difusa (sistema híbrido) com o objetivo de sanar a dificuldade dos algoritmos genéticos com relação à desigualdade de densidade entre os agrupamentos.

REFERÊNCIAS BIBLIOGRÁFICAS

BEASLEY, David, BULL, David R., MARTIN, Ralph R. *An overview of genetic algorithms*: Part 1, Fundamentals [online] and Part 2, Research Topics [online]. 1993. Disponível na Internet via World Wide Web: <http://www.geocities.com/igoryepes/>.

GOLDBERG, David E. *Genetic algorithms in search, optimization, and machine learning*. Reading: Addison Wesley, 1989. 412p.

MENDES FILHO, Elson Felix. *Algoritmos genéticos* [online]. 1998. Disponível na Internet via World Wide Web: <http://www.icms.sc.usp.br/~prico/index.html>.

PINTO, João Onofre Pereira. *Cluster Analysis using GA*, University of Tenessee, 1998.

RUSSEL, Stuart, NORVIG, Peter. *Artificial intelligence*: a modern approach. Upper Saddle River: Prentice Hall, 1995. 932p.

SERRADA, Anselmo Pérez. *Una introducción a la computación evolutiva* [online]. 1996. Disponível na Internet via World Wide Web: <http://www.geocities.com/igoryepes/>.

SETNES, Magne, BABUŠKA, Robert. *Fuzzy relational classifier trained by fuzzy clustering*. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 29, no. 5, pp. 619-625, October 1999.

SILVERMAN, B. W. *Density estimation for statistics and data analysis* (Monographs on statistics and applied probability). London: Chapman and Hall, 1990. 175p.

YEPES, Igor. *Uma incursão aos algoritmos genéticos* [online]. Disponível na Internet via World Wide Web: <http://www.geocities.com/igoryepes/ >.