

**Casamento de Padrões de Pontos com  
Perturbação**

*Eduardo A. Patrocínio*

**Dissertação de Mestrado**

# Casamento de Padrões de Pontos com Perturbação

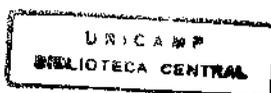
Este exemplar corresponde à redação final da  
Dissertação devidamente corrigida e defendida  
por Eduardo A. Patrocínio e aprovada pela  
Banca Examinadora.

Campinas, 18 de junho de 1997.



Prof. Dr. Pedro Jussieu de Rezende  
(Orientador)

Dissertação apresentada ao Instituto de Com-  
putação, UNICAMP, como requisito parcial para  
a obtenção do título de Mestre em Ciência da  
Computação.



# Casamento de Padrões de Pontos com Perturbação

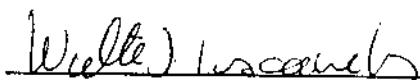
**Eduardo A. Patrocínio**

Outubro de 1996

**Banca Examinadora:**

- Prof. Dr. Pedro Jussieu de Rezende (Orientador)
- Profa. Dra. Célia Picinin de Mello
- Prof. Dr. Walter Mascarenhas  
IMECC — UNICAMP
- Prof. Dr. Ricardo Dahab

Tese de Mestrado defendida e aprovada em 27 de novembro de 1996 pela Banca Examinadora composta pelos Professores Doutores



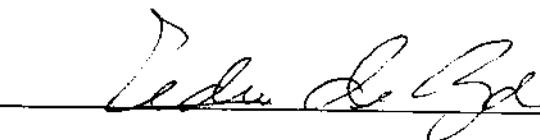
---

Prof. Dr. Walter Mascarenhas



---

Profa. Dra. Célia Picinin de Mello



---

Prof. Dr. Pedro Jussieu de Rezende

© Eduardo A. Patrocínio, 1996.  
Todos os direitos reservados.

# Prefácio

O problema de casamento de pontos já foi discutido em diversos trabalhos anteriores da literatura: há estudos de casamento aproximado entre conjuntos de mesma cardinalidade assim como estudos de casamento exato entre conjuntos de cardinalidades diferentes.

O propósito deste trabalho é o de estender um algoritmo polinomial proposto em [3] para casamento exato a um algoritmo também eficiente para casamento de conjuntos de pontos com perturbação.

A abordagem utilizada aqui consiste de, através de demonstrações construtivas por indução da existência de solução para os problemas estudados, obter algoritmos eficientes. Tais demonstrações estabelecem a corretude destes algoritmos ao mesmo tempo em que sugerem uma versão (inicialmente) recursiva deles.

Inicialmente, damos ênfase ao estudo do caso de uma dimensão, por ser este a base dos demais casos estudados. O problema para duas dimensões é resolvido em detalhe, apresentando-se um algoritmo recursivo inspirado no algoritmo iterativo de [3] pois nos parece assim mais claro e porque grandemente auxilia a subsequente exposição do caso bi-dimensional com perturbação. Já o algoritmo para dimensões maiores, que é uma adaptação relativamente simples e natural dos procedimentos descritos para duas dimensões, não é explicitamente enunciado.

Durante o desenvolvimento deste trabalho, nos deparamos com diferentes e interessantes situações: um problema cuja solução pareceu inicialmente requerer solução através de programação não linear foi resolvido através da demonstração de um teorema que apresenta a solução ótima (Teorema 9) calculável de maneira exata; para se determinar o valor máximo da perturbação permitida em duas dimensões mediante certas restrições, encontramos um algoritmo de complexidade  $O(n^2)$ , vencendo uma aparente necessidade de consideração de  $O(n^3)$  triplas de pontos. Ademais, utilizando um resultado anterior, demonstramos que este algoritmo é ótimo.

# Agradecimentos

A Deus Supremo.

Ao Prof. Pedro, por sua inestimável contribuição à minha formação acadêmica.

A Miriam, por me apoiar e me acompanhar em todos os instantes e situações.

A todos os docentes do Instituto de Computação que me ajudaram a concretizar este trabalho.

Aos meus pais, a quem dedico este trabalho, por terem me ensinado tudo o que sei, se alguma coisa sei.

# Conteúdo

<b>Prefácio</b>	<b>v</b>
<b>Agradecimentos</b>	<b>vi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Trabalhos Anteriores . . . . .	5
<b>2 Casamento Exato em uma Dimensão</b>	<b>7</b>
2.1 Definição do Problema . . . . .	7
2.2 Construção de Solução por Indução . . . . .	7
2.3 Algoritmo Recursivo para Casamento Exato em uma Dimensão . . . . .	9
2.4 Análise de Complexidade do Algoritmo . . . . .	10
2.5 O Problema de Casamento Exato em Dimensões Maiores . . . . .	11
<b>3 Casamento com Perturbação em uma Dimensão</b>	<b>12</b>
3.1 Definição do Problema . . . . .	12
3.2 Determinação de um Casamento com Perturbação em uma Dimensão . . . . .	14
3.3 Caracterização de Casamento com Perturbação em uma Dimensão . . . . .	16
<b>4 Algoritmos para Casamento com Perturbação em uma Dimensão</b>	<b>19</b>
4.1 Construção de Solução por Indução . . . . .	19
4.2 Algoritmo Recursivo para Casamento com Perturbação em uma Dimensão . . . . .	21
4.3 Análise de Complexidade do Algoritmo . . . . .	23
4.4 Generalizando para Dimensões Maiores . . . . .	24
4.5 Outro Algoritmo Recursivo para Casamento com Perturbação em uma Dimensão . . . . .	25
4.6 Análise de Complexidade do Algoritmo . . . . .	27
<b>5 Casamento Exato em Duas Dimensões</b>	<b>28</b>
5.1 Algumas Definições . . . . .	28

5.2	Algoritmo para Casamento Exato de Pontos em Duas Dimensões . . . . .	29
5.3	Análise de Complexidade do Algoritmo . . . . .	31
<b>6</b>	<b>Casamento com Perturbação em Duas Dimensões</b>	<b>33</b>
6.1	Definição de Casamento com Perturbação em Duas Dimensões . . . . .	33
6.2	Análise do Problema . . . . .	34
6.3	Região de Perturbação e de Casamento . . . . .	34
6.4	Determinação de um Raio $\rho$ Conveniente . . . . .	35
6.5	Análise do Problema de Casamento com Perturbação em Duas Dimensões .	39
6.6	Representação da Região de Perturbação . . . . .	40
6.7	Construção de Solução por Indução . . . . .	40
6.8	Algoritmo para Casamento com Perturbação em Duas Dimensões . . . . .	42
6.9	Análise de Complexidade do Algoritmo . . . . .	47
<b>7</b>	<b>Casamento de Pontos em Dimensões Maiores</b>	<b>48</b>
7.1	Definição do Casamento com Perturbação em Dimensões Maiores . . . . .	48
7.2	Determinação de um Raio $\rho$ Conveniente . . . . .	49
7.3	Análise do Problema de $d$ Dimensões . . . . .	50
7.4	Construção de Solução por Indução . . . . .	51
7.5	Algoritmo para Casamento com Perturbação em Dimensão $d$ . . . . .	52
<b>8</b>	<b>Conclusão</b>	<b>53</b>
8.1	Contribuições deste Trabalho . . . . .	53
8.2	Tratamento de Escala . . . . .	54
8.3	Determinação de Melhor Casamento . . . . .	54
	<b>Bibliografia</b>	<b>56</b>

# Lista de Figuras

1.1	Representação de um conjunto de estrelas como um conjunto de pontos . . .	2
1.2	Um conjunto de pontos padrão . . . . .	3
1.3	Uma composição de rotação e translação do conjunto $P$ que leva a um casamento em $S$ . . . . .	4
2.1	Ilustração da execução do algoritmo <i>Casamento Exato</i> <sub>1</sub> . . . . .	9
3.1	Um conjunto de pontos na reta e os respectivos intervalos . . . . .	13
3.2	Uma situação em que não existe ponto no intervalo $[p_i + \tau + \varepsilon_e, p_i + \tau]$ . . .	15
3.3	Uma situação em que um ponto de $S$ é a média aritmética de dois pontos adjacentes de $P$ . . . . .	17
3.4	Ilustração do lema 3 . . . . .	17
3.5	Ilustração do lema 3 . . . . .	18
4.1	Situação em que o processo apresentado não encontra um casamento . . . .	20
4.2	Uma translação do conjunto $P$ que possibilitou encontrar um casamento . .	20
4.3	Ilustração de uma configuração de pontos de $P$ e $S$ que produziriam um casamento se tentássemos generalizar o algoritmo acima . . . . .	24
6.1	Valor máximo permitido para $\rho$ . . . . .	35
6.2	Ilustração do Teorema 9 . . . . .	36
6.3	Situação em que a ordem circular não é mantida . . . . .	37
6.4	Representação da região $R$ como uma lista de pontos . . . . .	41
6.5	Uma situação em que $(p_1, p_2, p_i) \approx_\rho (s_{m_1}, s_{m_1, m_2}, s_{m_1, m_i})$ . . . . .	43

# Capítulo 1

## Introdução

Dados uma amostra e um padrão, desejamos saber se o padrão está presente na amostra. Frequentemente estes objetos, padrão e amostra, são representados por conjuntos de pontos, ou podem ser mapeados para tais conjuntos.

Considere, por exemplo, que temos a imagem de um conjunto de estrelas. Podemos representar esta imagem como uma matriz bi-dimensional de valores inteiros que representam diversos níveis de cinza, ou ainda, utilizando técnicas de processamento de imagem, podemos transformá-la em um conjunto de pontos no plano, onde cada ponto representa uma estrela (figura 1.1). Pretendemos saber se um conjunto (padrão)  $P$  apresenta um *casamento* em um conjunto (amostra)  $S$ . No exemplo, após mapear a constelação e a imagem para conjuntos de pontos, pretendemos verificar se o primeiro (padrão) está presente no segundo (amostra) (figura 1.2).

Um *casamento exato* de  $P$  em  $S$  consiste de um par de funções  $f : P \rightarrow S$  e  $\phi : P \rightarrow \mathbb{R}^d$ , onde  $f$  é injetora e  $\phi$  é uma composição de rotação, translação e reflexão tais que  $f(p) = \phi(p)$  para todo  $p \in P$  (figura 1.3).

A princípio, para verificarmos se existe um casamento exato de  $P$  em  $S$ , teríamos que tomar todos os subconjuntos de  $k = |P|$  elementos de  $S$  e verificar se existe alguma função  $f$  bijetora de  $P$  neste subconjunto e uma combinação de rotação, translação e reflexão de  $P$  em  $\mathbb{R}^2$  que satisfaçam à condição acima. O número de subconjuntos de  $S$  com  $k$  elementos é exponencial em  $k$ . Então, se fôssemos analisar todos os subconjuntos, gastaríamos tempo exponencial. Apresentamos um algoritmo que encontra, se existir, um casamento exato de  $P$  em  $S$  em tempo polinomial em  $k$  e em  $n$  estando estes conjuntos em uma ou duas dimensões, adaptado do algoritmo apresentado em [3]. A alteração em relação a este é que o algoritmo aqui apresentado advém de uma prova por indução e por ser recursivo, nos parece mais claro.

Por outro lado, existem muitos casos em que há uma perturbação local, isto é, existe uma pequena diferença entre a posição real de um ponto e a posição obtida. Por menor

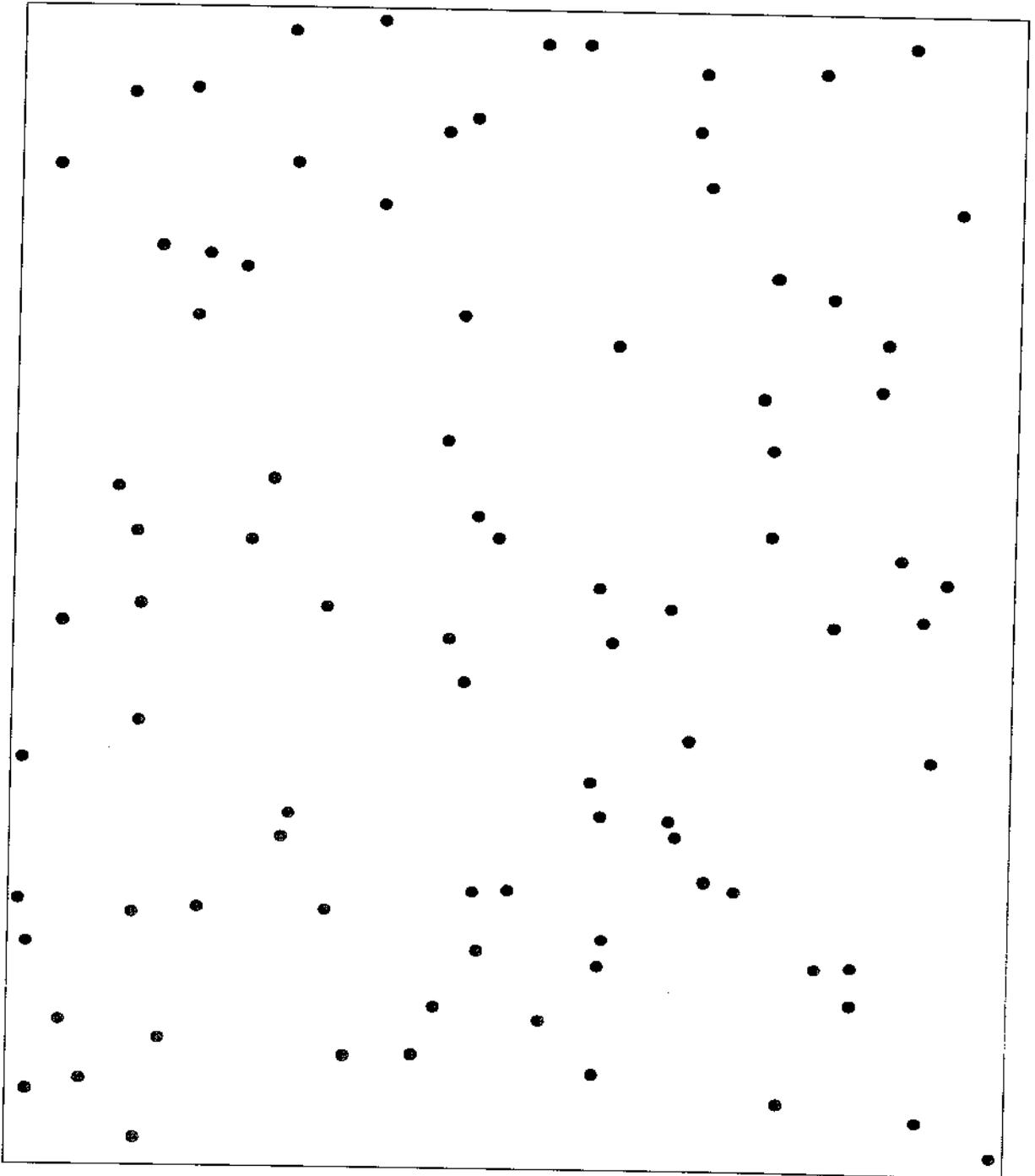


Figura 1.1: Representação de um conjunto de estrelas como um conjunto de pontos



Figura 1.2: Um conjunto de pontos padrão

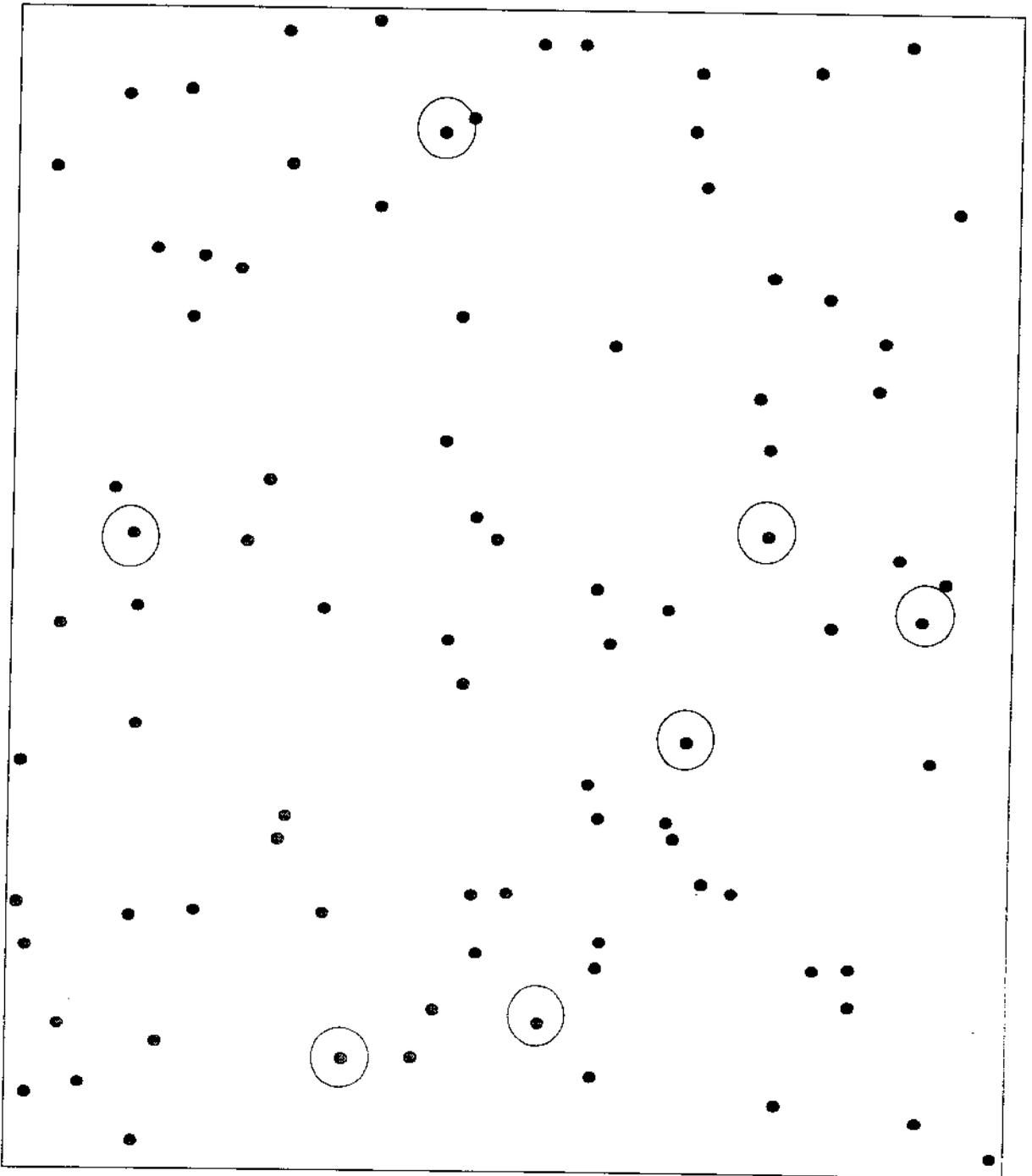


Figura 1.3: Uma composição de rotação e translação do conjunto  $P$  que leva a um casamento em  $S$

que seja esta perturbação, um casamento exato não será encontrado. Porém, gostaríamos de encontrar um casamento, mesmo tendo esta perturbação.

O nosso objetivo é encontrar pares de pontos dados pelas funções  $f(p)$  e  $\phi(p)$  que estejam relativamente próximos, mas não necessariamente coincidentes.

O propósito deste trabalho é o de estender um algoritmo polinomial para casamento exato a um algoritmo também eficiente para casos com perturbação.

Uma característica imprescindível para que possamos utilizar o algoritmo de casamento exato é que os conjuntos de pontos a serem casados devem ter a mesma ordem circular, isto é, a ordem em que um dado ponto “enxerga os demais” pontos do conjunto deve ser mantida. Ou seja, os conjuntos  $f(P)$  e  $\phi(P)$  devem ter a mesma ordem circular.

O algoritmo do caso exato para uma dimensão é apresentado no segundo capítulo. No terceiro capítulo, discutimos o problema com perturbação em uma dimensão. No quarto capítulo, descrevemos dois algoritmos para casamento com perturbação em uma dimensão. A seguir, apresentamos os algoritmos para duas dimensões: o caso exato, no quinto capítulo, e o caso com perturbação, no sexto capítulo. Depois disto, descrevemos como obter um algoritmo para dimensões arbitrárias no sétimo capítulo. Apresentamos, então, uma conclusão dos resultados obtidos.

## 1.1 Trabalhos Anteriores

Para o leitor interessado, citamos a seguir alguns dos principais trabalhos anteriores presentes na literatura que lidam com problemas de natureza semelhante aos problemas tratados aqui. Não é nosso objetivo resumi-los, mas apenas mencionar alguma característica de cada trabalho.

- [3] apresenta um algoritmo polinomial de casamento de pontos em  $d$ -dimensões, para conjuntos de cardinalidades diferentes, porém sem considerar a existência de perturbação;
- [2] apresenta alguns algoritmos e cotas inferiores de complexidade para o problema de casamento de pontos em duas dimensões, para conjuntos de mesma cardinalidade;
- em [8], considera-se a existência de um pequeno fator de tolerância (pré-estabelecido) para encontrar conjuntos *aproximadamente* congruentes;
- [1] apresenta diversos algoritmos para o cálculo de transformações geométricas que mapeiam exata ou aproximadamente dois conjuntos de pontos;

- o artigo [10] discorre sobre uma série de estruturas geométricas que podem ser utilizadas para reconhecimento de padrões;
- o conceito de matriz- $\lambda$  é apresentado no artigo [6]. Este conceito é importante para o cálculo de arranjos de forma eficiente;
- uma nova estrutura é apresentada em [7] para resolver o problema da congruência entre dois conjuntos de pontos;
- uma aplicação da utilização de algoritmos heurísticos para o reconhecimento de caracteres numéricos é apresentado em [9].

# Capítulo 2

## Casamento Exato em uma Dimensão

### 2.1 Definição do Problema

Neste capítulo, apresentamos um algoritmo para o problema de casamento exato em uma dimensão. Este algoritmo será produzido a partir da prova indutiva de que é possível resolver o seguinte problema, estudado em [3]:

#### Casamento Exato em uma Dimensão

**Instância:** Um conjunto  $P$  de  $k$  pontos em  $\mathbb{R}$  (padrão) e um conjunto  $S$  de  $n$  pontos em  $\mathbb{R}$  (amostra), com  $k \leq n$ .

**Questão:** Encontrar um casamento exato de  $P$  em  $S$ , ou mostrar que não existe um.

### 2.2 Construção de Solução por Indução

Suponhamos que os conjuntos  $P$  e  $S$  estejam ordenados. A técnica que utilizaremos é a prova por indução em  $k$ : dados os conjuntos  $P$  e  $S$  com  $k$  e  $n$  pontos respectivamente, tomamos, por hipótese de indução, que é possível resolver o problema para sub-conjuntos  $P'$  e  $S'$  de tamanho  $k - 1$  e  $n - 1$  respectivamente e, então, procuramos uma maneira de casar o último ponto de  $P$ . A hipótese de indução que temos é a seguinte:

**Hipótese:** *É possível resolver o problema de Casamento Exato de Pontos para um conjunto  $P$  de  $k - 1$  pontos e um conjunto  $S$  de  $n - 1$  pontos.*

Considere um conjunto  $P$  com  $k > 1$  pontos. Por hipótese de indução, é possível resolver o problema para os conjuntos  $P'$  e  $S'$ , formados pelos  $k - 1$  e  $n - 1$  primeiros pontos, respectivamente.

Se não existe casamento exato de  $P'$  em  $S'$ , podemos afirmar que não existe casamento exato de  $P$  em  $S$ .

Existindo casamento exato de  $P'$  em  $S'$ , seja  $\{m_1, \dots, m_{k-1}\}$  o conjunto de índices que definem este casamento exato. Temos duas possibilidades: se existe um ponto de  $S$  na posição  $p_k + s_{m_1} - p_1$ , podemos definir como  $m_k$  o índice deste ponto e afirmar que o conjunto  $\{m_1, \dots, m_k\}$  representa um conjunto de índices que definem um casamento exato de  $P$  em  $S$ . Se tal ponto não existe, *não* podemos tirar nenhuma conclusão sobre o casamento exato, pois poderíamos ter um outro casamento exato de  $P'$  em  $S'$  que permitisse associar  $p_k$  a um ponto de  $S$ .

Neste caso, a aplicação de indução simples requer que verifiquemos que nenhum dos casamentos podem ser estendidos para a totalidade dos pontos do padrão. Portanto, nesse caso, faz-se necessário computar todos os casamentos parciais. Assim, é natural explicitar na hipótese de indução que sejam encontrados *todos* os casamentos. Logo, é não mais difícil mostrar o seguinte resultado mais geral:

**Proposição 1** *Dados um conjunto  $P$  de  $k$  pontos em  $\mathbf{R}$  (padrão) e um conjunto  $S$  de  $n$  pontos em  $\mathbf{R}$  (amostra), com  $k \leq n$ , podemos encontrar todos os casamentos exatos de  $P$  em  $S$ , ou mostrar que não existe um.*

### Demonstração:

Por indução em  $k$ :

**Base:** Se  $k = 1$ , basta tomar os  $n$  conjuntos dados por  $M_i = \{i\}$ .

**Hipótese:** Podemos determinar *todos* os conjuntos  $M$  que definem os casamentos exatos de  $P'$  em  $S'$ , onde  $P'$  tem  $k - 1$  pontos, e  $S'$ ,  $n - 1$  pontos.

**Prova:** Se  $k > 1$ , por indução, é possível resolver o problema para um conjunto  $P'$  de  $k - 1$  pontos e um conjunto  $S'$  de  $n - 1$  pontos. Se não existe casamento exato de  $P'$  em  $S'$ , podemos afirmar que não existe um casamento exato de  $P$  em  $S$ . Se existem casamentos exatos de  $P'$  em  $S'$ , para cada um destes, verificamos se é possível estendê-lo para um casamento exato de  $P$  em  $S$ . Para isto, devemos verificar se existe um ponto de  $S$  na posição  $p_k + s_{m_1} - p_1$ , onde  $s_{m_1}$  é o ponto que está sendo associado a  $p_1$  por este casamento exato. Todos aqueles que forem extensíveis desta forma determinam casamentos exatos de  $P$  em  $S$ .

Com isto, com uma aplicação da hipótese de indução e uma varredura em  $S$ , conseguimos determinar todos os casamentos exatos possíveis para  $P$  em  $S$ .  $\diamond$

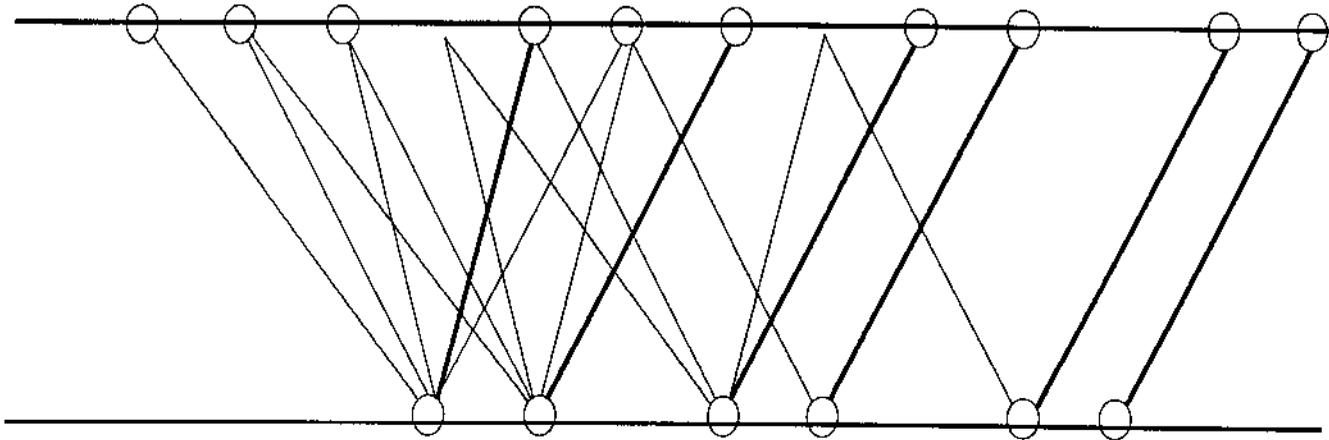


Figura 2.1: Ilustração da execução do algoritmo *Casamento Exato<sub>1</sub>*

## 2.3 Algoritmo Recursivo para Casamento Exato em uma Dimensão

Podemos, agora, escrever o algoritmo que verifica se existe um casamento exato de  $P$  em  $S$ . Este algoritmo está baseado na prova por indução apresentada na seção anterior e constitui-se basicamente da chamada de um procedimento recursivo *EncontraÍndice* que determina os conjuntos de índices  $M_k$ . A figura 2.1 apresenta uma ilustração da execução deste algoritmo.

**Algoritmo** *CasamentoExato<sub>1</sub>* ( $P, S$ )

1. Ordene os conjuntos  $P$  e  $S$ , tal que  $p_i \leq p_{i+1}$  para  $1 \leq i \leq k - 1$  e  $s_i \leq s_{i+1}$  para  $1 \leq i \leq n - 1$ ;
2. [Chama o procedimento recursivo que devolve o conjunto de índices ]  
Seja  $\Gamma = \text{EncontraÍndice}(P, S)$ ;
3. [Se o conjunto for vazio, retorna falha ]  
se  $\Gamma = \{\}$ , então retorne falha;
4. senão
  - 4.1 [Podemos tomar qualquer conjunto retornado ]  
Seja  $M_k \in \Gamma$ ;
  - 4.2 [Retorna o conjunto  $M_k$  ]  
retorne  $M_k$ .

**procedimento** *EncontraÍndice* ( $P = \{p_1, \dots, p_k\}, S = \{s_1, \dots, s_n\}$ )

1. [Caso base: retorna todos os índices de  $S$  como um casamento exato possível ]  
se  $k = 1$ , então retorne  $\{\{1\}, \{2\}, \dots, \{n\}\}$ ;
2. senão
  - 2.1 [Obtém todos os casamentos exatos para  $k - 1$  pontos ]  
Seja  $\Gamma = \text{EncontraÍndice}(\{p_1, \dots, p_{k-1}\}, \{s_1, \dots, s_{n-1}\})$ ;
  - 2.2 [Inicializa o valor de  $m_k$  ]  
Seja  $m_k = k$ ;
  - 2.3 [Inicializa o valor do conjunto que será retornado ]  
Seja  $\Gamma' = \{\}$ ;
  - 2.4 [Itera em cima de todos os valores retornados pela recursão ]  
para cada  $\{m_1, \dots, m_{k-1}\} \in \Gamma$ , faça
    - 2.4.1 [Define o valor inicial de  $m_k$  ]  
Seja  $m_k = \max\{m_k, m_{k-1} + 1\}$ ;
    - 2.4.2 [Atualiza convenientemente  $m_k$  ]  
enquanto  $s_{m_k} < p_k + s_1 - p_1$  e  $m_k \leq n$  faça  $m_k = m_k + 1$ ;
    - 2.4.3 [O procedimento *Casa* verifica se os pontos  $p_k$  e  $s_{m_k}$  casam ]  
se  $m_k \leq n$  e *Casa* ( $p_k, s_{m_k}, p_1, s_{m_1}$ ) então  
faça  $\Gamma' = \Gamma' \cup \{\{m_1, \dots, m_k\}\}$ ;
  - 2.5 retorne  $\Gamma'$ .

Temos que apresentar também o procedimento *Casa*:

**procedimento *Casa*** ( $p_i, s_j, p_1, s_{m_1}$ )

retorne  $p_i + s_{m_1} - p_1 = s_j$ .

A corretude do algoritmo *CasamentoExato*<sub>1</sub> segue imediatamente da seção anterior.

Devemos observar que, para a determinação de todos os casamentos, ao invés de apenas um deles, bastaria retornarmos todo o conjunto  $\Gamma$  e não apenas um de seus elementos.

## 2.4 Análise de Complexidade do Algoritmo

A complexidade do algoritmo se resume à complexidade do procedimento *EncontraÍndice*. Este procedimento é recursivo e apresenta em cada passo um laço que é executado tantas vezes quantos forem os elementos do conjunto  $\Gamma$ . Este conjunto tem, no máximo, cardinalidade  $n - k + 1$ . Dentro deste laço, temos ainda um outro laço para determinar o valor de  $m_k$ . Podemos analisar a complexidade deste laço interno da seguinte forma: o valor de  $m_k$  pode ser avançado para cada  $s_l$  com  $l = k, \dots, n$  no máximo uma vez e nunca

recua. Com isto, cada passo consome tempo proporcional a  $n - k + 1$ . Chegamos, então à seguinte relação de recorrência:

$$T(n, k) = T(n - 1, k - 1) + (n - k + 1)$$

com a qual concluímos que o procedimento *EncontraÍndice* possui complexidade  $O((n - k + 1)k)$ . Assim, temos:

**Teorema 2** *O problema de casamento exato de pontos em uma dimensão de um conjunto  $P$  de  $k$  elementos em um conjunto  $S$  de  $n$  elementos pode ser resolvido em tempo  $O(n \log n + (n - k + 1)k)$ .*

## 2.5 O Problema de Casamento Exato em Dimensões Maiores

O algoritmo para casamento exato em uma dimensão visto na seção 2.3 pode ser adaptado para dimensões maiores que um. Em uma dimensão, temos o conceito de ordenação total dos pontos. Em dimensões maiores, precisamos estender este conceito. Para isto, usaremos o conceito de ordenação circular dos pontos, que permite utilizar o algoritmo acima como base para um algoritmo para dimensões maiores.

# Capítulo 3

## Casamento com Perturbação em uma Dimensão

### 3.1 Definição do Problema

Neste capítulo, consideramos a existência de perturbação local nos conjuntos de pontos que nos leve a um casamento com perturbação de  $P$  em  $S$ .

Sejam  $P$  um conjunto de  $k$  pontos em uma reta e  $S$  um conjunto de  $n$  pontos nesta mesma reta com  $k \leq n$ . O nosso objetivo é encontrar um subconjunto de  $k$  pontos de  $S$ , tal que, para cada ponto  $p$  de  $P$ , exista um ponto deste subconjunto *relativamente* próximo de uma translação de  $P$ .

Temos que formalizar o conceito de pontos *relativamente* próximos, que depende do vínculo que queremos manter entre estes dois conjuntos. Sem este conceito, não existe nenhum processo de busca sistemática de casamento com perturbação. Nesse caso, não vemos como evitar de ter que considerar  $C_k^n$  subconjuntos de  $S$ . Se mantivermos a mesma ordem, isto é, se  $s_{m_i} \leq s_{m_j}$  quando estamos casando  $p_i$  e  $p_j$  com  $s_{m_i}$  e  $s_{m_j}$ , respectivamente e  $p_i < p_j$ , podemos adaptar o algoritmo de casamento exato para este caso.

Porém se requerermos apenas a manutenção da ordem, podemos ter situação de casamento onde os pontos casados estão arbitrariamente distantes entre si. Por isso, outra característica importante é a proximidade entre os pontos casados. Podemos alcançar isto, impondo uma restrição sobre a distância entre eles. Ou seja, imponhamos que o ponto de  $S$  que está sendo casado com um ponto  $p_i$  de  $P$  esteja dentro de um intervalo que inclua  $\phi(p_i)$  e que estes intervalos sejam disjuntos.

Como estamos tratando de perturbações locais, devemos permitir a liberdade de movimento em todas as direções, sem favorecer nenhuma delas. Desta forma, é natural que escolhamos, para limitar a distância entre os pontos casados, intervalos de mesma amplitude e centrados em cada ponto de  $P$ .

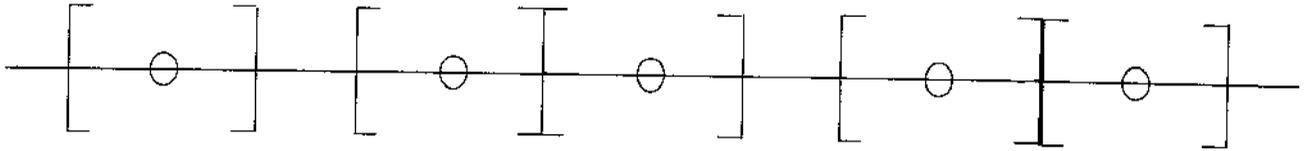


Figura 3.1: Um conjunto de pontos na reta e os respectivos intervalos

Sejam  $p_i$  e  $p_{i+1}$  pontos de  $P$  que formam um par mais próximo. Como queremos intervalos de mesma amplitude e que preservem a ordem entre os pontos casados, podemos tomar os intervalos como tendo raio no máximo  $d(p_i, p_{i+1})/2$ . Chamemos de  $\rho$  esta distância:

$$\rho = \min_{1 \leq i \leq k-1} \frac{d(p_i, p_{i+1})}{2}$$

Este valor é o máximo possível de forma que dois pontos de  $S$  associados a dois pontos de  $P$  não estejam em ordem inversa.

A figura 3.1 apresenta um conjunto de pontos e os intervalos definidos pela regra acima.

**Definição 1** *Sejam  $P$  um conjunto com  $k$  pontos e  $S$  um conjunto com  $n \geq k$  pontos na reta real. Dizemos que existe um casamento com perturbação de  $P$  em  $S$ , se existem funções  $f$  e  $\phi$ , tais que:*

1.  $f$  seja injetora de  $P$  em  $S$ ;
2.  $\phi$  seja uma translação de  $P$  em  $\mathbf{R}$ ; e
3.  $f$  e  $\phi$  satisfaçam à seguinte relação:

$$d(f(p), \phi(p)) \leq \rho, \forall p \in P,$$

onde  $\rho$  é definido pela relação acima.

O valor de  $\rho$  pode ser calculado, fazendo-se uma ordenação do conjunto  $P$ , seguida de uma varredura. Isto pode ser feito em tempo  $O(n \log n)$ , que é ótimo.

Podemos agora enunciar o seguinte problema:

### Casamento com Perturbação em Uma Dimensão

**Instância:** Um conjunto  $P$  de  $k$  pontos e um conjunto  $S$  de  $n$  pontos, com  $k \leq n$ .

**Questão:** Encontrar um casamento com perturbação dos pontos de  $P$  em  $S$ , ou mostrar que não existe um.

### 3.2 Determinação de um Casamento com Perturbação em uma Dimensão

Estamos procurando uma translação  $\phi$  para o conjunto  $P$ , de tal forma que, para todo ponto  $p \in P$ , o intervalo centrado em  $\phi(p)$  de comprimento  $\rho$  contenha (pelo menos) um ponto de  $S$  e estamos procurando  $f$  tal que  $d(f(p), \phi(p)) \leq \rho$ , para todo  $p \in P$ .

Suponha que tenhamos os conjuntos  $P$  e  $S$  ordenados. Procuremos esta translação casando o primeiro ponto  $p_1 \in P$  com um ponto  $s_{m_1} \in S$ . Para que  $f(p_1)$  seja  $s_{m_1}$ , podemos tomar  $\phi(p_1) = s_{m_1}$ . Seja  $\tau = s_{m_1} - p_1$  o deslocamento inicial.

Apesar de estarmos colocando  $p_1$  exatamente sobre  $s_{m_1}$ , temos a liberdade de realizar a translação  $\phi(p_1)$  dentro do intervalo:  $[s_{m_1} - \rho, s_{m_1} + \rho]$ , se necessário, para encontrar casamento para pontos subseqüentes a  $p_1$ .

Inicialmente, os candidatos a se associarem a  $p_i$  são os pontos de  $S$  que distam no máximo  $2\rho$  de  $p_i + \tau$ . Sejam, então:  $\varepsilon_e = -2\rho$ ,  $\varepsilon_d = 2\rho$ , as amplitudes esquerda e direita do intervalo de pontos que temos que analisar. Como veremos mais à frente, estes valores poderão se tornar diferentes um do outro em módulo. Estamos tomando estes valores iniciais para  $\varepsilon_e$  e  $\varepsilon_d$  a fim de incorporar a liberdade que temos de associar o ponto  $p_i$  ao ponto  $s_{m_i}$ .

O cerne da busca por um casamento com perturbação consiste em verificar se é possível encontrar *compativelmente* pontos de  $S$  nos intervalos  $[p_i + \tau + \varepsilon_e, p_i + \tau + \varepsilon_d]$ , para cada  $i$ . Se não encontrarmos ponto algum de  $S$  em um destes intervalos, temos a garantia de que não temos um casamento, tal que o ponto  $p_1$  seja associado a  $s_{m_1}$ . Deveremos, então, tentar encontrar um casamento, associando  $p_1$  ao próximo ponto de  $S$ .

A compatibilidade a que nos referimos está relacionada ao fato de que começamos com um intervalo de amplitude  $4\rho$ , porém estamos procurando um casamento onde a distância entre os pontos casados seja no máximo  $\rho$ .

Se encontramos pelo menos um ponto em cada um destes intervalos, podemos prosseguir, tentando encontrar par para o próximo ponto de  $P$ . Só que, para isto, possivelmente teremos que atualizar os valores de  $\varepsilon_e$  e  $\varepsilon_d$ .

Esta atualização é necessária porque, se o ponto  $p_i$  e seu correspondente em  $S$  não forem coincidentes, após fazer a translação de  $\tau$  no conjunto  $P$ , teremos, possivelmente, que limitar a região onde estamos procurando pontos a serem associados. Para isto, temos, a princípio que considerar todos os pontos de  $S$  que estão neste intervalo. Se existe algum ponto de  $S$  em  $[p_i + \tau + \varepsilon_e, p_i + \tau + \varepsilon_d]$ , temos as seguintes situações:

1. Se houver pelo menos um ponto de  $S$  no intervalo  $[p_i + \tau + \varepsilon_e, p_i + \tau]$  e pelo menos um ponto de  $S$  no intervalo  $[p_i + \tau, p_i + \tau + \varepsilon_d]$ , mantemos os mesmos valores para  $\varepsilon_e$  e  $\varepsilon_d$ .

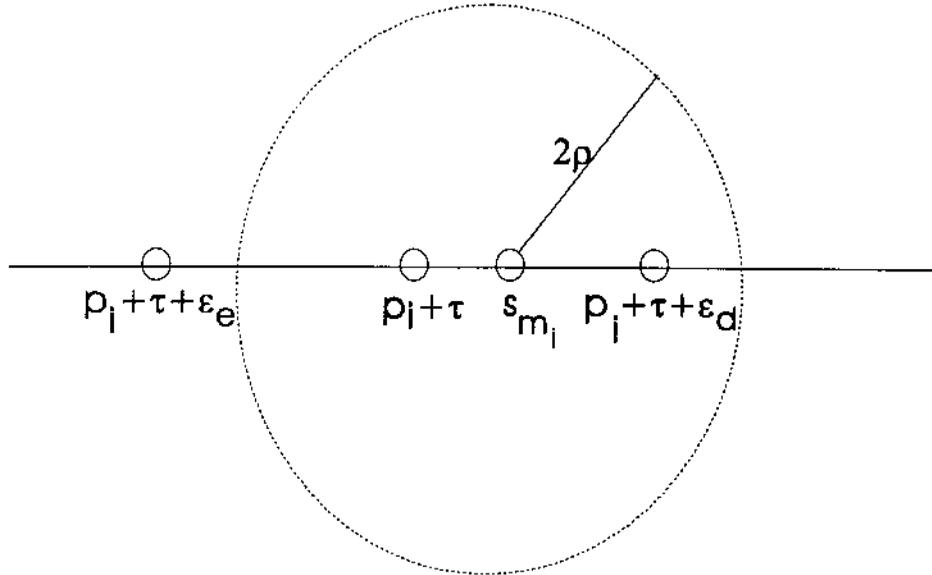


Figura 3.2: Uma situação em que não existe ponto no intervalo  $[p_i + \tau + \varepsilon_e, p_i + \tau]$

2. Se não houver nenhum ponto de  $S$  no intervalo  $[p_i + \tau + \varepsilon_e, p_i + \tau]$ , temos que atualizar o valor de  $\varepsilon_e$  da seguinte forma: seja  $s_{m_i}^d$  o ponto de  $S$  no intervalo  $[p_i + \tau, p_i + \tau + \varepsilon_d]$  mais próximo de  $p_i + \tau$ ; atualize (ver figura 3.2):

$$\varepsilon_e = \max\{\varepsilon_e, s_{m_i}^d - (p_i + \tau) - 2\rho\}$$

Esta atualização é necessária por que não é possível encontrarmos um casamento que associe  $p_i$  a  $s_{m_i}^d$  e  $p_j$  a um ponto  $s_{m_j}$ ,  $j > i$ , tal que  $s_{m_j} < p_j - (s_{m_i}^d - 2\rho - p_i)$ . Portanto, devemos limitar o intervalo de busca à esquerda.

3. A atualização de  $\varepsilon_d$ , no caso de haver pontos somente no intervalo  $[p_i + \tau + \varepsilon_e, p_i + \tau]$ , é análoga a 2.

Se conseguirmos manter os valores de  $\varepsilon_e$  e  $\varepsilon_d$  de forma que o intervalo  $[\varepsilon_e, \varepsilon_d]$  permaneça não-vazio até o último ponto de  $P$ , podemos afirmar que temos um casamento entre os dois conjuntos.

Não sabemos, porém, precisar que subconjunto de pontos de  $S$  será casado com  $P$ ; para isto, temos que fazer uma segunda varredura verificando quais pontos de  $S$  permanecem no intervalo  $[p_i + \tau + \varepsilon_e, p_i + \tau + \varepsilon_d]$  para cada  $i$ . Temos a garantia de que, em cada um destes intervalos, haverá pelo menos um ponto de  $S$ . Porém este intervalo pode ter amplitude maior que  $2\rho$ . Mas estamos procurando um casamento que associe pontos de  $P$  e  $S$  cuja distância máxima dos pontos associados seja  $\rho$ .

Sabemos que, em cada um dos intervalos  $[p_i + \tau + \varepsilon_e, p_i + \tau + \varepsilon_d], \forall p_i \in P$ , existe um ponto de  $S$  e que os valores de  $\varepsilon_e$  e  $\varepsilon_d$  foram ajustados de forma tal que exista pelo menos um ponto de  $S$  em qualquer intervalo de tamanho  $2\rho$  contido inteiramente no intervalo  $[p_i + \tau + \varepsilon_e - \rho, p_i + \tau + \varepsilon_e + \rho]$ , para cada  $i$ . Portanto, podemos tomar a translação como sendo:

$$\phi(p_i) = p_i + \tau + \frac{\varepsilon_d + \varepsilon_e}{2}$$

Com esta varredura, podemos encontrar um ponto  $f(p_i)$  que está no intervalo  $[\phi(p_i) - \rho, \phi(p_i) + \rho]$ . Com isto, encontramos um casamento com perturbação para  $P$  em  $S$ , pois  $d(f(p_i), \phi(p_i)) \leq \rho, \forall i$ .

### 3.3 Caracterização de Casamento com Perturbação em uma Dimensão

A seguir, apresentamos alguns lemas que nos auxiliarão a mostrar a corretude do algoritmo para casamento com perturbação em uma dimensão baseado nos conceitos apresentados acima, a ser apresentado no próximo capítulo.

**Lema 3** *Uma vez que fixamos um ponto de  $S$  a ser casado com  $p_1 \in P$ , então um ponto  $s \in S$  pode ser casado com no máximo dois pontos de  $P$ .*

**Demonstração:**

Note, em primeiro lugar, que  $s$  pode se casar com dois pontos  $p_i$  e  $p_{i+1}$  se  $s = (p_i + p_{i+1})/2 + \tau$  e  $d(p_i, p_{i+1}) = 2\rho$ . Reciprocamente, provemos que  $s \in S$  só pode se casar com dois pontos de  $P$  se ele for a média aritmética destes pontos. Se  $s$  pode se casar com os pontos  $p_i$  e  $p_{i+1}$  de  $P$ , temos que:

$$d(s, p_i + \tau) \leq \rho$$

e

$$d(s, p_{i+1} + \tau) \leq \rho.$$

Logo,  $s = (p_{i+1} + p_i)/2 + \tau$ , pois sabemos que  $d(p_{j+1}, p_j) \geq 2\rho, \forall j = 1, \dots, k - 1$  (ver figura 3.3).  $\diamond$

**Lema 4** *Se fixamos a associação do ponto  $p_1 \in P$  com um ponto  $s_{m_1} \in S$ , então, para cada  $p_i \in P \setminus \{p_1\}$ , é suficiente analisar dois pontos de  $S$  para determinar com que ponto de  $S$   $p_i$  será associado. Estes pontos são o ponto  $s_{m_i}^e$  de  $S$  que está mais próximo e à esquerda de  $p_i + \tau$ , e o ponto  $s_{m_i}^d$  de  $S$  que está mais próximo e à direita de  $p_i + \tau$ . Reciprocamente, tanto  $s_{m_i}^e$  quanto  $s_{m_i}^d$  podem fazer parte de um casamento com perturbação de  $P$  em  $S$ .*

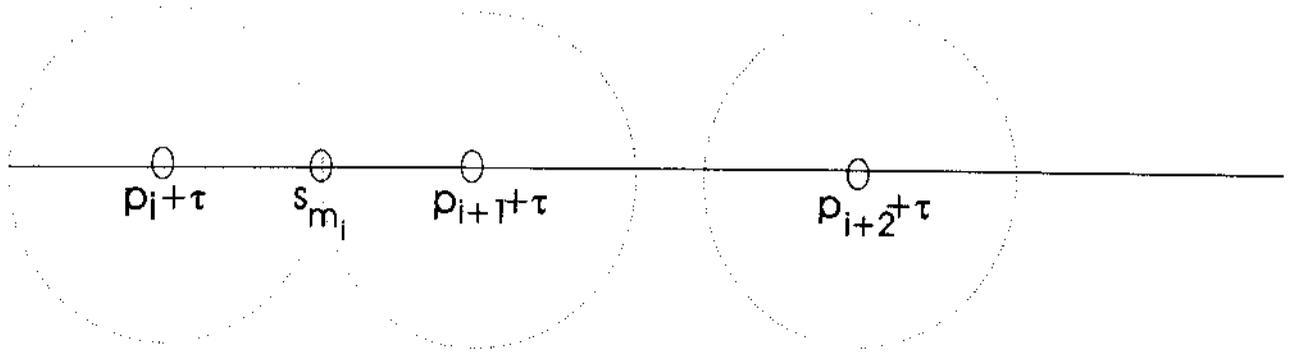


Figura 3.3: Uma situação em que um ponto de  $S$  é a média aritmética de dois pontos adjacentes de  $P$

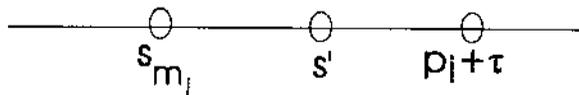


Figura 3.4: Ilustração do lema 3

**Demonstração:**

Provaremos por construção. Seja  $s_{m_i}$  o ponto associado ao ponto  $p_i$  por este casamento e suponha que exista um outro ponto de  $S$ ,  $s'$  entre  $p_i + \tau$  e  $s_{m_i}$  (ver figura 3.4). Suponha, sem perda de generalidade, que  $s_{m_i} < p_i + \tau$ . Sendo um casamento, temos que:  $d(p_i + \tau, s_{m_i}) \leq \rho$  e  $d(p_i + \tau, s_{m_i}) \leq \rho$ . Como  $s'$  está entre  $p_i + \tau$  e  $s_{m_i}$ , temos que  $d(p_i + \tau, s') \leq \rho$ . Portanto, podemos substituir  $s_{m_i}$  pelo ponto  $s'$  na associação com  $p_i$ . Podemos, dessa forma, tomar  $s' = s_{m_i}^e$ , onde  $s_{m_i}^e$  é o ponto entre  $p_i + \tau$  e  $s_{m_i}$ , mais próximo de  $p_i + \tau$ .

Sempre que temos um casamento com perturbação de  $P$  em  $S$ , tal que o ponto  $s$  de  $S$  sendo associado a um ponto  $p_i \in P$  não seja o mais próximo e à esquerda de  $p_i + \tau$  ou o mais próximo e à direita de  $p_i + \tau$ , podemos encontrar outro casamento com perturbação que substitui a associação entre  $p_i$  e  $s$  pela associação entre  $p_i$  e um dos dois pontos citados acima.

Logo, basta analisar o ponto de  $S$  mais próximo de  $p_i + \tau$  em cada uma das direções:  $s_{m_i}^e$  e  $s_{m_i}^d$ .

Reciprocamente, suponha que tenhamos o ponto  $s_{m_i}^e$  e  $s_{m_i}^d$  a uma distância menor de  $2\rho$  de  $p_i + \tau$ . Suponha que tenhamos, também, um ponto  $s_{m_{i+1}}$  de  $S$  na posição  $p_{i+1} + \tau - 2\rho$  e que os demais pontos de  $S$  estejam a uma distância de  $p_{i+1} + \tau$  maior que  $2\rho$  (ver figura 3.5). Desta forma, o único ponto que poderia ser associado a  $p_{i+1}$ , uma vez que associamos  $p_i$  a  $s_{m_i}$ , é o ponto  $s_{m_{i+1}}$ . Porém, para isto, temos que associar ao ponto  $p_i$  o ponto  $s_{m_i}^e$  e tomar para  $\phi(p_i) = p_i + \tau - \rho$ .

Portanto, ele é o único que permite encontrar uma associação para  $p_{i+1}$  independente de  $p_i + \tau$  estar ou não mais próximo do ponto  $s_{m_i}^d$ , do que do ponto  $s_{m_i}^e$ .  $\diamond$

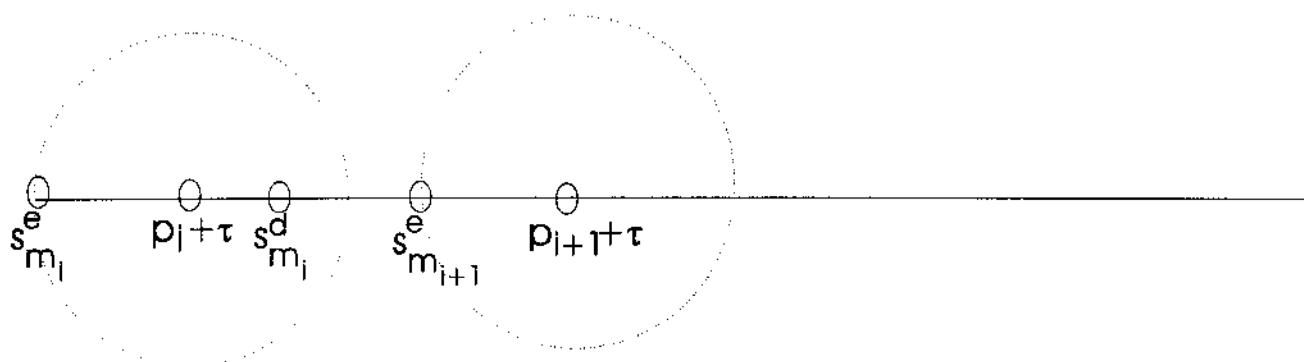


Figura 3.5: Ilustração do lema 3

# Capítulo 4

## Algoritmos para Casamento com Perturbação em uma Dimensão

Queremos apresentar um algoritmo que resolva o problema de casamento com perturbação em uma dimensão, produzindo um conjunto  $M_k = \{m_1, \dots, m_k\}$  de índices tal que os pares  $(p_i, s_{m_i}), \forall 1 \leq i \leq k$  formem um casamento com perturbação.

### 4.1 Construção de Solução por Indução

Se  $k = 1$ , devemos novamente produzir todos os conjuntos de índices  $\{\{1\}, \{2\}, \dots, \{n\}\}$ .

Utilizemos novamente a técnica de indução para resolver este problema. Dados os conjuntos  $S$  e  $P$  com  $n$  e  $k$  pontos respectivamente, assumamos que seja possível resolver o problema para conjuntos de tamanho  $n$  e  $k - 1$  respectivamente e então procuremos uma maneira de casar o último ponto de  $P$ . Da mesma forma que no caso exato, será necessário reforçar a hipótese de indução. Baseando-se na hipótese de indução que temos para o caso exato, podemos escrever a seguinte hipótese:

**Hipótese:** *É possível determinar todos os conjuntos  $M_{k-1}$  que definem os casamentos com perturbação possíveis de  $P'$  em  $S'$ , onde  $P'$  tem  $k - 1$  pontos e  $S'$  tem  $n - 1$  pontos.*

Tivemos que colocar na hipótese de indução muito mais do que a determinação de um conjunto de índices. Por hipótese de indução, sabemos resolver o problema para um conjunto  $P'$  com  $k - 1$  pontos. Para cada conjunto de índices  $\{m_1, \dots, m_{k-1}\}$  que temos, devemos verificar se existe um ponto de  $S$  no intervalo  $[p_k + \tau - \rho, p_k + \tau + \rho]$ , onde  $\tau = s_{m_1} - p_1$  e  $\rho = \min\{p_{i+1} - p_i\}/2, \forall 1 \leq i \leq k - 1$ . Se houver, podemos definir como  $m_k$  o índice deste ponto e afirmar que  $\{m_1, \dots, m_k\}$  define um casamento com perturbação válido de  $P$  em  $S$ , e o conjunto de todos os índices será obtido fazendo-se esta verificação em cada conjunto de índices que define um casamento com perturbação de  $P'$  em  $S'$ .

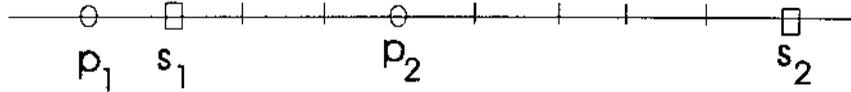


Figura 4.1: Situação em que o processo apresentado não encontra um casamento

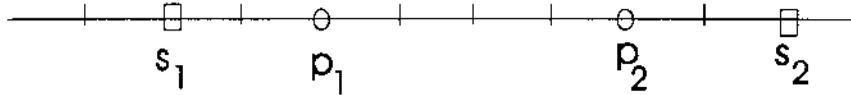


Figura 4.2: Uma translação do conjunto  $P$  que possibilitou encontrar um casamento

Se procedermos desta forma até o final e encontrarmos pelo menos um mapeamento, podemos dizer que temos um casamento com perturbação, onde a translação que devemos tomar é  $\tau$ . Caso contrário, *não* podemos afirmar que não existe casamento com perturbação de  $P$  em  $S$ . Isto por que quando fixamos o casamento de  $p_1$  com  $s_{m_1}$ , estamos desperdiçando perturbação, pois poderíamos tomar uma translação qualquer no intervalo  $[\tau - \rho, \tau + \rho]$  e teríamos ainda um casamento que associe  $p_1$  a  $s_{m_1}$ . A figura 4.1 mostra uma situação em que não encontramos um casamento pelo processo acima.

Porém, se considerássemos a liberdade que temos quando associamos  $p_1$  a  $s_{m_1}$ , podemos encontrar um casamento na figura acima, como mostra a figura 4.2, na qual fez-se uma translação do conjunto  $P$  que possibilitou o casamento de  $P$  em  $S$ .

A fim de incorporar esta perturbação, dividamos o problema em duas partes: a primeira parte consiste em determinar conjuntos de índices  $\{m_1, \dots, m_k\}$  tais que seja possível definir um casamento que associe  $p_i$  ao ponto  $s_{m_i}$  ou ao ponto  $s_{m_i-1}$ . Na segunda parte, iremos determinar com qual destes dois pontos é possível realizar o casamento. Como a segunda parte pode ser feita fazendo-se uma simples varredura no conjunto de índices, concentremo-nos na seguinte questão:

### Determinação de um Conjunto de Índices que Defina um Casamento

**Instância:** Um conjunto  $P = \{p_1, \dots, p_k\}$  de  $k$  pontos e um conjunto  $S = \{s_1, \dots, s_n\}$  de  $n$  pontos, com  $k \leq n$ .

**Questão:** Existe um conjunto de índices  $M$  que determina um casamento que associe a  $p_i$  o ponto  $s_{m_i}$  ou  $s_{m_i-1}$ . (Note que  $s_{m_i-1}$  é o ponto de  $S$  anterior a  $s_{m_i}$  e que  $m_i - 1$  pode não pertencer a  $M$ .)

Sejam  $M_k = \{m_1, \dots, m_k\}$  o conjunto de índices que define um casamento com perturbação de  $P$  em  $S$ , e  $\varepsilon_e$  e  $\varepsilon_d$ , que me garantem que no intervalo  $[p_i + \tau + \varepsilon_e, p_i + \tau + \varepsilon_d]$  existe pelo menos um ponto de  $S$ . Chamemos de  $W_k$  esta tripla de valores  $(M_k, \varepsilon_e, \varepsilon_d)$ . Podemos enunciar a seguinte proposição:

**Proposição 5** *Dados os conjuntos  $P$  com  $k$  elementos e  $S$  com  $n$  elementos,  $k \leq n$ , é possível determinar todos os conjuntos  $W = \{M, \varepsilon_e, \varepsilon_d\}$  que definem os casamentos possíveis de  $P$  em  $S$ .*

**Demonstração:**

**Base:** Se  $k = 1$ , devemos retornar os  $n$  conjuntos dados por  $(\{i\}, -2\rho, 2\rho), 1 \leq i \leq n$ .

Reforcemos, mais uma vez, a hipótese de indução:

**Hipótese:** É possível determinar *todos* os conjuntos  $W = \{M, \varepsilon_e, \varepsilon_d\}$  que definem os casamentos possíveis de  $P'$  em  $S'$ , onde  $P'$  tem  $k - 1$  pontos e  $S'$  tem  $n - 1$  pontos.

**Passo:** Se  $k > 1$ , por indução, é possível resolver o problema para  $k - 1$  pontos no conjunto  $P'$  e  $n - 1$  pontos no conjunto  $S'$ . Se não existir casamento com perturbação de  $P'$  em  $S'$ , podemos afirmar que não existe casamento com perturbação de  $P$  e  $S$ . Se existirem casamentos, obtemos uma coleção de triplas  $W_{k-1}$ . Para cada conjunto  $W_{k-1}$ , devemos verificar se é adequado para estender o casamento a  $p_k$  ou não. Os que forem adequados, depois de fazer o ajuste de  $\varepsilon_e$  e  $\varepsilon_d$  convenientemente, formarão os conjuntos de valores que definem os casamentos com perturbação de  $P$  em  $S$ .

Com isto, com uma aplicação da hipótese de indução e uma varredura em  $S$ , conseguimos determinar todos os casamentos com perturbação possíveis para  $P$  com  $k$  pontos.

## 4.2 Algoritmo Recursivo para Casamento com Perturbação em uma Dimensão

Podemos, agora, escrever o algoritmo que verifica se existe um casamento com perturbação de  $P$  em  $S$ . Este algoritmo, conforme foi detalhado na seção anterior, possui duas partes: uma parte recursiva, para determinação dos índices  $M_k$  e outra que utiliza este índices para produzir um casamento com perturbação de  $P$  em  $S$ .

**Algoritmo CasamentoComPerturbação<sub>1</sub> ( $P, S$ )**

1. [Determina o tamanho do intervalo onde procuramos os pontos a serem casados ]  
Determine o valor de  $\rho = \min_{1 \leq i \leq k-1} (p_{i+1} - p_i)/2$ ;
2. [Chama o procedimento recursivo que devolve uma coleção de conjuntos de índices que definem um casamento com perturbação e os valores de  $\varepsilon_e$  e  $\varepsilon_d$  para cada um destes casamentos ]

Seja  $\Gamma = \text{EncontraÍndice}(P, S, \rho)$

3. [Se o conjunto for vazio, retorna falha ]  
se  $\Gamma = \{\}$ , então retorne falha;
4. **senão**
  - 4.1 [Podemos tomar qualquer conjunto retornado ]  
Seja  $(M, \varepsilon_e, \varepsilon_d) \in \Gamma$ ;
  - 4.2 [Chamemos o procedimento *Busca* para obter o casamento com perturbação ]  
retorne *Busca*  $(M, \frac{\varepsilon_e + \varepsilon_d}{2})$ .

**procedimento** *EncontraÍndice* ( $P = \{p_1, \dots, p_k\}, S = \{s_1, \dots, s_n\}, \rho$ )

1. [Caso base: retorne todos os índices de  $S$  como um casamento com perturbação possível ]  
se  $k = 1$ , então retorne  $\{(\{1\}, -2\rho, 2\rho), (\{2\}, -2\rho, 2\rho), \dots, (\{n\}, -2\rho, 2\rho)\}$ ;
2. **senão**
  - 2.1 [Obtém todos os casamentos com perturbação para  $k - 1$  pontos ]  
Seja  $\Gamma = \text{EncontraÍndice}(\{p_1, \dots, p_{k-1}\}, \{s_1, \dots, s_{n-1}\}, \rho)$ ;
  - 2.2 [Inicializa o conjunto a ser retornado ]  
Seja  $\Gamma' = \{\}$ ;
  - 2.3 [Inicializa o valor de  $m_k$  ]  
Seja  $m_k = k$ ;
  - 2.4 [Itera em cima de todos os valores retornados pela recursão ]  
**para** cada  $(\{m_1, \dots, m_{k-1}\}, \varepsilon_e, \varepsilon_d) \in \Gamma$ , **faça**
    - 2.4.1 [Define o valor inicial de  $m_k$  ]  
Seja  $m_k = \max\{m_k, m_{k-1} + 1\}$ ;
    - 2.4.2 [Atualiza convenientemente  $m_k$  ]  
**enquanto**  $s_{m_k} < p_k + s_1 - p_1$  e  $m_k \leq n$  **faça**  $m_k = m_k + 1$ ;
    - 2.4.3 [O procedimento *Casa* atualiza os valores de  $\varepsilon_e$  e  $\varepsilon_d$  ]  
se  $m_k \leq n$  ou não *Casa*  $(p_k, m_k, \tau, \varepsilon_e, \varepsilon_d)$  então  
**faça**  $\Gamma' = \Gamma' \cup \{(\{m_1, \dots, m_k\}, \varepsilon_e, \varepsilon_d)\}$ ;
  - 2.5 retorne  $\Gamma'$ .

O procedimento *Casa* verifica se o ponto  $s_j$  ou  $s_{j-1}$  está dentro do intervalo  $[p_i + \tau + \varepsilon_e, p_i + \tau + \varepsilon_d]$  e atualiza os valores de  $\varepsilon_e$  e  $\varepsilon_d$  convenientemente:

**procedimento** *Casa* ( $p_i, j, \tau, \varepsilon_e, \varepsilon_d$ )

1. [Verifica se existe um ponto à esquerda ]  
Defina  $\text{ExisteEsquerda} = s_{j-1} \in [p_i + \tau + \varepsilon_e, p_i + \tau]$ ;

2. [Verifica se existe um ponto à direita ]  
 Defina  $\text{ExisteDireita} = s_j \in [p_i + \tau, p_i + \tau + \varepsilon_d]$  ;
3. [Se não existir nenhum ponto no intervalo, então retorne falha; ]  
**se não ExisteEsquerda e não ExisteDireita, então retorne falha;**
4. [Atualiza os limites do intervalo de busca ]  
**se não ExisteEsquerda, então atualize  $\varepsilon_e = \max\{\varepsilon_e, -2\rho + d(s_j, p_i + \tau)\}$ ;**
5. **se não ExisteDireita, então atualize  $\varepsilon_d = \min\{\varepsilon_d, 2\rho - d(p_i + \tau, s_{j-1})\}$ ;**
6. **retorne sucesso.**

O procedimento *Busca* determina o casamento com perturbação de  $P$  em  $S$  depois que temos o conjunto de índices  $\{m_1, m_2, \dots, m_k\}$  que indica que  $s_{m_i}$  ou  $s_{m_{i-1}}$  pode ser casado com  $p_i$ :

**procedimento** *Busca* ( $M, \varepsilon$ )

1. Faça  $i = 1$ ;
2. **enquanto**  $i \leq k$  **faça**
  - 2.1 [Se o ponto de  $S$  imediatamente à direita de  $p_i + \varepsilon$  não serve ]  
**se  $d(s_{m_i}, p_i + \varepsilon) > \rho$  então**  
 [Considere o ponto de  $S$  à esquerda de  $p_i + \varepsilon$  ]  
 Faça  $m_i := m_i - 1$ ;
  - 2.2 Faça  $i := i + 1$ ;
3. **retorne**  $\{m_i | 1 \leq i \leq k\}$ .

### 4.3 Análise de Complexidade do Algoritmo

O algoritmo é uma adaptação do algoritmo do caso exato. As únicas diferenças são que estamos retornando os valores de  $\varepsilon_e$  e  $\varepsilon_d$  no procedimento *EncontraÍndice*, o procedimento *Casa* possui uma lógica mais complexa, porém continua sendo executado em tempo constante e, depois de chamar o procedimento *EncontraÍndice*, estamos chamando o procedimento *Busca*, que, por sua vez, tem complexidade  $O(k)$ . Podemos então concluir que o procedimento *EncontraÍndice* possui complexidade  $O((n - k + 1)k)$

**Teorema 6** *O problema de casamento com perturbação de um conjunto  $P$  de  $k$  elementos em um conjunto  $S$  de  $n$  elementos pode ser resolvido em tempo  $O(n \log n + (n - k + 1)k)$ .*

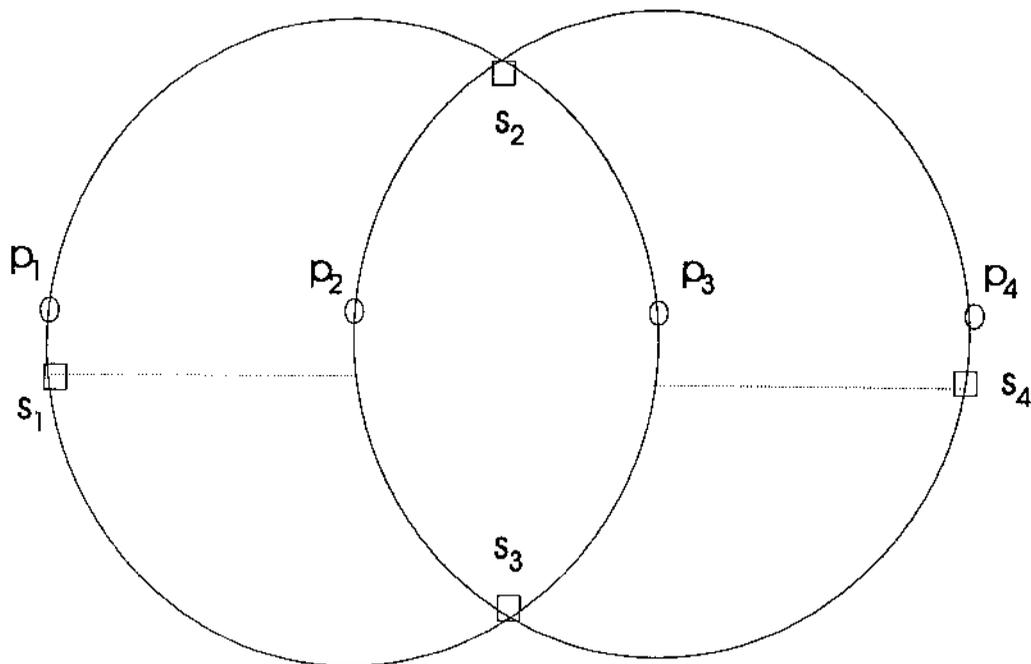


Figura 4.3: Ilustração de uma configuração de pontos de  $P$  e  $S$  que produziriam um casamento se tentássemos generalizar o algoritmo acima

## 4.4 Generalizando para Dimensões Maiores

Apesar de o algoritmo descrito acima encontrar um casamento com perturbação para qualquer instância de uma dimensão, ele não pode ser adaptado para dimensões maiores pois envolveria o cálculo da intersecção de círculos de raio  $2\rho$  que resulta numa *lente* que poderia reportar casamento quando não existe um.

Um exemplo é apresentado na figura 4.3. Nele, temos quatro pontos em cada um dos conjuntos, e o algoritmo acima acusaria a presença de um casamento, quando ele, na verdade, não existe. Pelo algoritmo acima, após associar  $p_2$  a  $s_2$ , temos que limitar a região de perturbação, que passa a ser uma lente formada por dois arcos de circunferência. Ao transladar esta região para o ponto  $p_3$ , verificamos que o ponto  $s_3$  pode se casar com  $p_3$ . Ao fazer a segunda varredura para determinar o casamento, vemos que a única transformação isomórfica que casa os pontos  $p_1$  e  $p_2$  com  $s_1$  e  $s_2$  é uma transladando-se o conjunto  $P$  de  $\rho$ , de forma que os pontos  $p_1$  e  $p_2$  fiquem a uma distância  $\rho$  de  $s_1$  e  $s_2$ , respectivamente. Porém esta transformação faz com que  $p_3$  fique a uma distância maior que  $\rho$  de  $s_3$ .

Por isso, construamos um algoritmo semelhante e de mesma complexidade, que pode ser adaptado para dimensões maiores.

No algoritmo acima, estamos trabalhando com um intervalo, onde procuramos pontos

a serem casados com um dado ponto  $p_i$ . Esta região tem inicialmente tamanho  $4\rho$  e, à medida que varremos os pontos de  $P$ , ela tende a diminuir, podendo chegar até ao tamanho  $2\rho$ . Isto por que a atualização do intervalo de busca está sendo feita mediante a intersecção do intervalo original  $I$  com um intervalo de tamanho  $2\rho$ , centrado em um ponto de  $I$ . Logo, a atualização de  $I$  não pode produzir um intervalo de tamanho menor que  $2\rho$ .

Quando esta região chega ao tamanho  $2\rho$ , não temos mais liberdade para a escolha da translação do conjunto  $P$ . Prosseguimos com a varredura dos pontos de  $P$ , verificando se no intervalo de tamanho  $2\rho$  em torno de cada ponto de  $P$  temos pelo menos um ponto de  $S$ .

O próximo algoritmo que propomos não se baseia em um intervalo que se translada pelos pontos de  $P$ , procurando pontos de  $S$  dentro deste intervalo. Ele se baseia no intervalo de liberdade que temos para transladar o conjunto  $P$ . Este intervalo inicia-se com tamanho  $2\rho$  e depois de transladá-lo para um ponto  $p_i$ , procuramos pontos de  $S$  que estejam dentro deste intervalo ou a uma distância máxima  $\rho$  do mesmo, evitando assim o problema de encontrarmos um casamento quando ele não existe, levantado acima. Baseado no conjunto de pontos de  $S$  que estejam dentro deste intervalo, podemos atualizá-lo, possivelmente diminuindo seu tamanho.

Ao final teremos um intervalo de comprimento maior ou igual a zero. A translação de cada ponto de  $P$ , será dada pela soma de  $\tau = s_{m_1} - p_1$  com qualquer ponto, por exemplo o ponto médio, do intervalo  $I = [\varepsilon_e, \varepsilon_d]$ :

$$\phi(p_i) = p_i + \tau + \frac{\varepsilon_e + \varepsilon_d}{2}$$

## 4.5 Outro Algoritmo Recursivo para Casamento com Perturbação em uma Dimensão

Para este algoritmo, temos um intervalo de busca  $I$  menor que o do algoritmo anterior. Porém estamos procurando não só pontos dentro deste intervalo, mas também pontos que estejam a uma distância máxima  $\rho$  do mesmo. Dado o conjunto de pontos candidatos a se casarem com um dado ponto  $p_i$ , temos que atualizar o intervalo  $I$ . Para cada ponto  $s_{m_i}$  de  $S$  nesta situação, podemos construir um intervalo, centrado em  $s_{m_i}$ , e de tamanho  $2\rho$ . Como estamos interessados em casar o ponto  $p_i$  com um destes pontos de  $S$ , a atualização do intervalo  $I = [\varepsilon_e, \varepsilon_d]$  será dada por:

$$I = I \cap \left\{ \bigcup [s_{m_i} - \rho, s_{m_i} + \rho] \right\}$$

onde a união é tomada para todo  $s_{m_i} \in [p_i + \tau + \varepsilon_e - \rho, p_i + \tau + \varepsilon_d + \rho]$ .

Estamos, então fazendo uma interseção do intervalo  $I$ , de tamanho máximo  $2\rho$  com a união de vários intervalos de tamanho  $2\rho$ . Com isto, iremos possivelmente diminuir o tamanho do intervalo  $I$ .

A atualização do intervalo  $I$  pode ser feita de uma maneira muito mais simples do que calculando o intervalo dado pela união de vários intervalos e, finalmente, fazendo a interseção com o intervalo original. Podemos encontrar pontos em três posições válidas: pontos dentro do intervalo, pontos à esquerda do intervalo e pontos à direita do intervalo. Temos os seguintes casos:

1. Se encontramos um ponto de  $S$  dentro do intervalo  $I$ , este permanece inalterado.
2. Se não temos nenhum ponto de  $S$  em  $I$ , temos as seguintes situações:
  - (a) Se temos pelo menos um ponto à esquerda e um ponto à direita do intervalo  $I$  e a uma distância máxima de  $\rho$  dele, este também não é alterado.
  - (b) Se temos apenas ponto de  $S$  à esquerda do intervalo a uma distância máxima  $\rho$ , seja  $s_j$  o ponto mais próximo de  $p_i + \tau$ . Então  $I$  é atualizado da seguinte forma:

$$I = I \cap [s_j - \rho, s_j + \rho]$$

- (c) Se temos apenas pontos de  $S$  à direita do intervalo a uma distância máxima  $\rho$  dele, a atualização é análoga.
- (d) Se não temos nenhum ponto de  $S$  a ser casado com  $p_i$ , podemos afirmar que não existe nenhum casamento com perturbação de  $P$  em  $S$  que associe  $p_1$  a  $s_{m_1}$ .

Reescrevamos inicialmente o procedimento *Casa*:

**procedimento** *Casa* ( $p_i, j, \tau, \varepsilon_e, \varepsilon_d$ )

1. [Verifica se existe ponto dentro do intervalo ]  
 Defina  $\text{ExistePontoInterior} = s_{j-1} \in [p_i + \tau + \varepsilon_e, p_i + \tau]$  ou  $s_j \in [p_i + \tau, p_i + \tau + \varepsilon_d]$ ;
2. se não  $\text{ExistePontoInterior}$ 
  - 2.1 [Verifica se existe ponto à esquerda do intervalo ]  
 Defina  $\text{ExisteEsquerda} = s_{j-1} \in [p_i + \tau + \varepsilon_e - \rho, p_i + \tau]$ ;
  - 2.2 [Verifica se existe ponto à direita do intervalo ]  
 Define  $\text{ExisteDireita} = s_j \in [p_i + \tau, p_i + \tau + \varepsilon_d + \rho]$ ;
  - 2.3 se  $\text{ExisteEsquerda}$  então

[Se existe apenas ponto à esquerda ]  
**se não ExisteDireita então**  
 [Atualiza o valor de  $\varepsilon_d$  ]  
 Atualize  $\varepsilon_d = \min\{\varepsilon_d, d(s_{j-1} + \rho, p_i + \tau)\};$

2.4 **senão**

[Se não existe ponto algum ]  
**se não ExisteDireita então**  
**retorne falha;**

**senão**

[Atualiza o valor de  $\varepsilon_e$  ]  
 Atualize  $\varepsilon_e = \max\{\varepsilon_e, -d(p_i + \tau, s_j - \rho)\};$

3. **retorne sucesso.**

Com relação aos demais procedimentos, eles permanecem iguais, exceto pela inicialização dos valores de  $\varepsilon_e$  e  $\varepsilon_d$  quando  $k = 1$ : ao invés de eles terem os valores  $-2\rho$  e  $2\rho$ , terão os valores  $-\rho$  e  $\rho$  respectivamente.

## 4.6 Análise de Complexidade do Algoritmo

As mudanças que foram introduzidas não alteram a complexidade do algoritmo. Portanto, o algoritmo continua tendo complexidade  $O(n \log n + (n - k + 1)k)$ .

# Capítulo 5

## Casamento Exato em Duas Dimensões

### 5.1 Algumas Definições

Podemos estender o problema enunciado no capítulo 2 para duas dimensões:

#### Casamento Exato em Duas Dimensões

**Instância:** Um conjunto  $P$  de  $k$  pontos em  $\mathbb{R}^2$  (padrão) e um conjunto  $S$  de  $n$  pontos em  $\mathbb{R}^2$  (amostra), com  $k \leq n$ .

**Questão:** Encontrar um casamento exato de  $P$  em  $S$ , ou mostrar que não existe um.

Quando estamos trabalhando em uma dimensão, temos uma ordenação total dos pontos. Trabalhando em dimensões maiores, temos que, de alguma forma, generalizar este conceito. Em [4], os autores apresentam o conceito de *ordenação circular*:

**Definição 2** *Dado um conjunto de  $n$  pontos  $S = \{s_1, s_2, \dots, s_n\}$  no plano, a ordem circular dos pontos de  $S \setminus \{s_i\}$  ao redor de  $s_i$  é definida da seguinte maneira: ordene os pontos lexicograficamente por seus ângulos polares ao redor de  $s_i$  no sentido anti-horário e por suas distâncias a  $s_i$ .*

A partir desta definição, podemos construir um algoritmo para casamento exato de padrões de pontos em duas dimensões. A grande vantagem do algoritmo de uma dimensão é que pode ser facilmente adaptado para dimensões maiores. O algoritmo para duas dimensões é fortemente baseado no algoritmo para uma dimensão, pois as listas circulares obtidas da ordenação circular fazem com que possamos aplicar o algoritmo para uma dimensão a cada uma destas listas.

$S$  é dito ordenado quando se têm as  $n$  listas circulares.

Utilizaremos as seguintes notações, definidas em [3]: a ordenação circular dos pontos  $S \setminus \{s_j\}$  ao redor de  $s_j$  será denotada por  $S_j = \{s_{j,1} = s_j, s_{j,2}, \dots, s_{j,n}\}$ .

Também iremos utilizar a seguinte relação de ordem dos pontos de  $S$ : diremos que  $s_i <_j^h s_l$  se, e somente se,  $s_i$  precede  $s_l$  na lista ordenada  $S_j$ , considerada começando com os elementos  $s_{j,1} = s_j$  e  $s_{j,2} = s_h$ .

Sejam  $s_{i_1}, s_{i_2}, s_{i_3}$  e  $p_{l_1}, p_{l_2}, p_{l_3}$  pontos. Denotamos por  $\sphericalangle(p_{l_1}, p_{l_2}, p_{l_3})$  o ângulo em  $p_{l_1}$  de  $p_{l_2}$  a  $p_{l_3}$  no sentido anti-horário. Além disso, se  $d(s_{i_1}, s_{i_2}) = d(p_{l_1}, p_{l_2})$ , podemos escrever:

1.  $(p_{l_1}, p_{l_2}, p_{l_3}) \approx (s_{i_1}, s_{i_2}, s_{i_3})$  se  $\sphericalangle(p_{l_1}, p_{l_2}, p_{l_3}) = \sphericalangle(s_{i_1}, s_{i_2}, s_{i_3})$  e  $d(p_{l_1}, p_{l_3}) = d(s_{i_1}, s_{i_3})$ ;
2.  $(p_{l_1}, p_{l_2}, p_{l_3}) \prec (s_{i_1}, s_{i_2}, s_{i_3})$  se  $\sphericalangle(p_{l_1}, p_{l_2}, p_{l_3}) < \sphericalangle(s_{i_1}, s_{i_2}, s_{i_3})$  ou se  $\sphericalangle(p_{l_1}, p_{l_2}, p_{l_3}) = \sphericalangle(s_{i_1}, s_{i_2}, s_{i_3})$  e  $d(p_{l_1}, p_{l_3}) < d(s_{i_1}, s_{i_3})$ ;
3.  $(p_{l_1}, p_{l_2}, p_{l_3}) \succ (s_{i_1}, s_{i_2}, s_{i_3})$  caso contrário.

Podemos enunciar a seguinte proposição:

**Proposição 7** *Dados um conjunto  $P$  de  $k$  pontos em  $\mathbb{R}^2$  (padrão) e um conjunto  $S$  de  $n$  pontos em  $\mathbb{R}^2$  (amostra), com  $k \leq n$ , podemos encontrar todos os casamentos exatos de  $P$  em  $S$ , ou mostrar que não existe um.*

A demonstração da proposição acima é similar à apresentada no capítulo 2, o que faz com que possamos omiti-la aqui.

## 5.2 Algoritmo para Casamento Exato de Pontos em Duas Dimensões

Utilizando as definições acima, podemos enunciar um algoritmo que funciona da seguinte maneira: ordenamos  $P$  em torno do ponto  $p_1$  e ordena-se  $S$  circularmente; ancorando o ponto  $p_1$  de  $P$  em cada um dos pontos  $s_j$  de  $S$ , percorremos a lista circular de pontos ao redor de  $p_1$  e de  $s_j$ . Como foi dito acima, o algoritmo para duas dimensões será baseado no algoritmo para uma dimensão. No caso de uma dimensão, temos uma lista (ordenada) dos pontos de  $P$  e outra dos pontos de  $S$  e fazemos uma varredura para verificar se existe um casamento exato entre estas duas listas. No caso de duas dimensões, temos  $n$  listas circulares para  $S$  e precisamos verificar se alguma delas apresenta um casamento exato com a lista circular de  $P$  em torno de  $p_1$ . Para isto, o algoritmo de duas dimensões

contém um laço para percorrer as listas circulares  $S_j$  de  $S$ . A partir disto, basta apenas verificar se uma lista  $S_j$  casa com a lista de  $P$ , de forma análoga ao que foi feito em uma dimensão, porém considerando  $S_j$  como uma lista circular.

Se o conjunto  $P$  possuir apenas um ponto, a solução do problema consiste em associar este ponto a qualquer ponto de  $S$ . Portanto, consideremos que o conjunto  $P$  possua pelo menos 2 pontos.

O algoritmo toma por base o algoritmo de casamento exato em uma dimensão e foi desenvolvido a partir do fato de que sabemos é possível resolver este problema quando  $|P| = 2$  e, por hipótese de indução, para subconjuntos de  $P$  com  $k - 1$  pontos.

### Algoritmo *Casamento Exato*<sub>2</sub>

1. Ordene circularmente o conjunto  $S$  e ordene os pontos de  $P$  lexicograficamente por coordenadas polares (ângulo, distância) ao redor do ponto  $p_1$ ;
2. [Chama o procedimento que devolve o conjunto de índices ]  
Seja  $\Gamma = \text{EncontraÍndice}(P, S)$ ;
3. [Verifica se a resposta é um conjunto vazio ]  
se  $\Gamma \neq \{\}$ , então
  - 4.1 [Podemos tomar qualquer conjunto ]  
Seja  $M_k \in \Gamma$ ;
  - 4.2 [Retorna o conjunto  $M_k$  ]  
retorne  $M_k$ ;
4. senão retorne  $\{\}$ .

Enunciemos agora o procedimento *EncontraÍndice*:

**procedimento** *EncontraÍndice* ( $P = \{p_1, \dots, p_k\}$ ,  $S = \{s_1, s_2, \dots, s_n\}$ )

1. [Caso base: retorna todos os índices de  $S$  como um casamento exato possível ]  
se  $k = 2$ , então retorne *CasoBase* ( $P, S$ );
2. senão
  - 2.1 [Obtém todos os casamentos exatos para  $k - 1$  pontos ]  
Seja  $\Gamma = \text{EncontraÍndice}(\{p_1, \dots, p_{k-1}\}, \{s_1, \dots, s_n\})$ ;
  - 2.2 [Inicializa o valor de  $m_k$ , que representa o valor do índice na lista  $S_{m_1}$  do ponto que está sendo associado a  $p_k$  ]  
Seja  $m_k = k$ ;
  - 2.3 [Inicializa o valor do conjunto que será retornado ]  
Seja  $\Gamma' = \{\}$ ;

- 2.4 [Itera em cima de todos os valores retornados pela recursão ]  
**para cada**  $\{m_1, \dots, m_{k-1}\} \in \Gamma$ , **faça**
- 2.4.1 [Define o valor inicial de  $m_k$  ]  
**Seja**  $m_k = \max\{m_k, m_{k-1} + 1\} \bmod n$ ;
- 2.4.2 [Atualiza convenientemente  $m_k$  ]  
**enquanto**  $(s_{m_1,1}, s_{m_1,m_2}, s_{m_1,m_k}) \prec (p_1, p_2, p_k)$  e  $m_k < n$  **faça**  
 $m_k = (m_k + 1) \bmod n$ ;
- 2.4.3 [Se os pontos  $p_k$  e  $s_{j,m_k}$  casam ]  
**se** *Casa*  $(p_1, p_2, p_k, s_{m_1,1}, s_{m_1,m_2}, s_{m_1,m_k})$  **então**  
 [Adiciona-os à coleção de lista de índices ]  
**faça**  $\Gamma' = \Gamma' \cup \{\{m_1, \dots, m_k\}\}$ ;
- 2.5 **retorne**  $\Gamma'$ .

Finalmente, apresentamos os procedimentos *Casa* e *CasoBase*:

**procedimento** *Casa*  $(p_1, p_2, p_i, s_1, s_2, s_j)$

**retorne**  $(p_1, p_2, p_i) \approx (s_1, s_2, s_j)$ .

**procedimento** *CasoBase*  $(P = \{p_1, \dots, p_k\}, S = \{s_1, \dots, s_k\})$

1. [Inicializa a coleção ]  
**Seja**  $\Gamma = \{\}$
2. **para**  $i = 1$  até  $n$  **faça**
  - 2.1 **para**  $j = 2$  até  $n$  **faça**

[Se os pontos  $p_1$  e  $p_2$  casam com os pontos  $s_{i,1}$  e  $s_{i,j}$  respectivamente ]  
**se** *Casa*  $(p_1, p_2, p_2, s_{i,1}, s_{i,j}, s_{i,j})$  **então**  
 [Adiciona-os à coleção ]  
**Faça**  $\Gamma = \Gamma \cup \{\{i, j\}\}$ ;

## 5.3 Análise de Complexidade do Algoritmo

Essencialmente, a única diferença que temos em termos de complexidade com relação ao caso de uma dimensão é que o caso base, que, em uma dimensão era computado em tempo linear, em duas dimensões leva tempo  $O(n^2)$ ; além disso, na chamada recursiva, ao invés de retirar o último ponto de  $S$ , passamos todo ele. No mais, os procedimentos *EncontraÍndice* e *Casa* foram adaptados para trabalharem com pontos em duas dimensões.

O primeiro passo toma tempo  $O(n^2 + k \log k)$ . Para os demais passos, no pior caso, o ponto  $p_1$  pode ser ancorado em todos os pontos de  $S$ . Em cada caso, o valor de  $m_i$  pode

avançar para todos os  $n - i$  últimos índices de pontos de  $S$  no máximo uma vez e não volta atrás. Com isto, o procedimento *EncontraÍndice* consome tempo  $O(nk)$ . Portanto, o tempo total é dado por  $O(n^2 + k \log k + nkn) = O(kn^2)$ .

**Teorema 8** *Dados um conjunto  $P$  de  $k$  pontos e em conjunto  $S$  de  $n \geq k$  pontos no plano, o problema de Casamento Exato de Pontos em Duas Dimensões pode ser resolvido em tempo  $O(kn^2)$ .*

Devemos observar que, com uma simples alteração no algoritmo, retornando  $\Gamma$  ao invés de um de seus elementos, conseguiríamos produzir todos os casamentos e não somente um deles.

# Capítulo 6

## Casamento com Perturbação em Duas Dimensões

### 6.1 Definição de Casamento com Perturbação em Duas Dimensões

A definição de casamento com perturbação em duas dimensões é similar à definição dada para uma dimensão: queremos que os pontos a serem casados preservem a ordem (circular) e também mantenham a restrição da distância entre os pontos que estão sendo casados. Porém em duas dimensões, a função  $\phi$  não deve se restringir a uma translação e reflexão: ela é definida como um mapeamento isométrico de  $P$ , isto é, uma composição de translação, rotação e reflexão.

Novamente, a reflexão pode ser considerada à parte: consideremos  $\phi$  como sendo uma combinação de translação e rotação. Se encontrarmos um casamento com perturbação, podemos parar; se não, executamos o algoritmo novamente para uma reflexão qualquer de  $P$  em  $S$ .

**Definição 3 Casamento com Perturbação em Duas Dimensões** *Sejam  $P$  um conjunto de  $k$  pontos,  $S$  um conjunto de  $n$  pontos, com  $k \leq n$  em um plano e  $\rho > 0$ . Dizemos que existe um casamento com perturbação  $\rho$  em duas dimensões de  $P$  em  $S$  se existem funções*

1.  $f$  injetora do conjunto  $P$  no conjunto  $S$ ; e
  2.  $\phi$  combinação de translação e rotação do conjunto  $P$ ,
- tais que estas duas funções satisfaçam à seguinte relação:*

$$d(f(p), \phi(p)) \leq \rho, \forall p \in P$$

Temos, então o seguinte problema:

### Casamento com Perturbação em Duas Dimensões

**Instância:** Um conjunto  $S$  de  $n$  pontos e um conjunto  $P$  de  $k$  pontos em  $\mathbb{R}^2$ , com  $k \leq n$ .

**Questão:** Encontrar um casamento com perturbação de  $P$  em  $S$  ou mostrar que não existe um.

## 6.2 Análise do Problema

Façamos uma análise do algoritmo de uma dimensão para tentar estendê-lo para duas dimensões. A adaptação do algoritmo para casamento com perturbação em duas dimensões a partir do algoritmo para casamento com perturbação em uma dimensão não é tão simples como no caso da adaptação do algoritmo para casamento exato em duas dimensões a partir do algoritmo exato em uma dimensão.

Neste caso, calculamos as ordenações circulares ao redor de cada ponto de  $S$  e ordenamos circularmente o conjunto  $P$  ao redor do ponto  $p_1$ . O algoritmo do caso exato associa o ponto  $p_1$  a um ponto  $s_j$  de  $S$  e utiliza o algoritmo do caso de uma dimensão, percorrendo duas listas: a lista de ordenação circular dos pontos de  $P$  ao redor de  $p_1$  e a lista dos pontos de  $S$  ao redor de  $s_j$ .

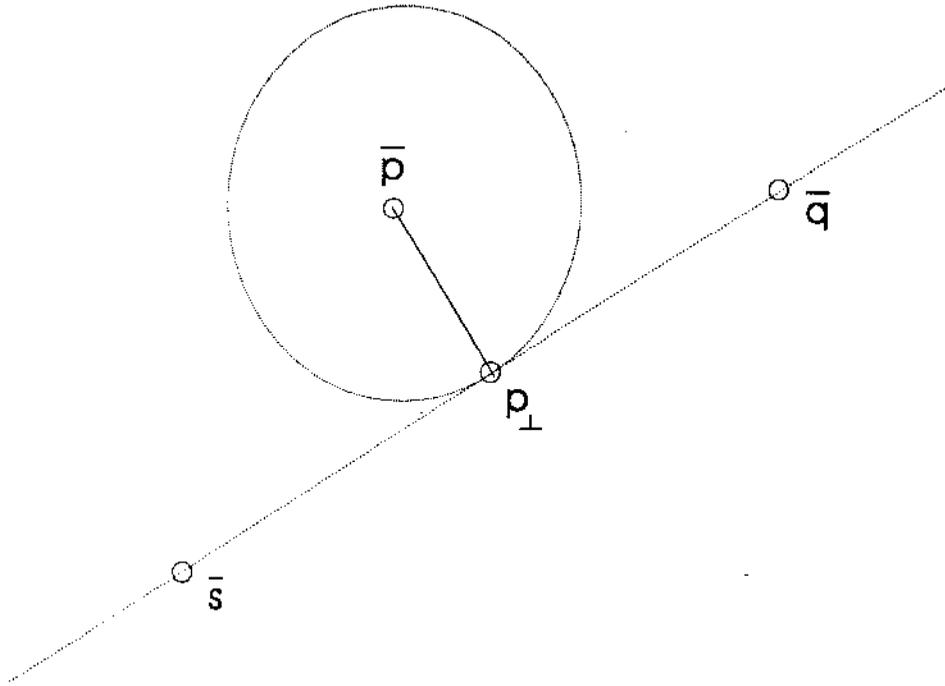
No caso de uma dimensão com perturbação, encontramos um intervalo ao redor de cada ponto de  $P$  e procuramos nele um ponto de  $S$ . Trabalhando no plano, coloquemos uma região ao redor de cada ponto de  $P$  e procuramos pontos de  $S$  nela ou que estejam a uma distância máxima  $\rho$  dela.

Fazendo uma analogia ao caso unidimensional, queremos dar igual liberdade de movimento em todas as direções. Então, tomaremos, para esta região, círculos de mesmo raio.

## 6.3 Região de Perturbação e de Casamento

Chamemos a região que será colocada ao redor de cada ponto  $p_i$  de  $P$  de *região de perturbação*  $\mathcal{R}_i$ . Esta é a região que determina a liberdade que temos para definir a função  $\phi$ .

Chamemos de *região de casamento* a região onde serão procurados os pontos a serem casados com um ponto de  $P$ . Chamemos de  $\mathcal{L}_i$  esta região. Note que  $\mathcal{R}_i \subset \mathcal{L}_i$  e mais, como queremos que a distância entre os pontos casados seja, no máximo,  $\rho$ ,  $\mathcal{L}_i$  será a região dos pontos que distam no máximo  $\rho$  de  $\mathcal{R}_i$ .

Figura 6.1: Valor máximo permitido para  $\rho$ 

## 6.4 Determinação de um Raio $\rho$ Conveniente

Estamos então procurando regiões circulares e uniformes para cada um dos pontos. O nosso objetivo é termos a maior liberdade possível para determinar o casamento; para isso, estamos procurando utilizar o maior valor de  $\rho$  possível. Por outro lado, estamos querendo adaptar o algoritmo exato para o caso com perturbação. Para isso, não podemos tomar o valor de  $\rho$  arbitrariamente grande: ele deve ser tal que a ordem circular dos pontos seja mantida. Sejam  $C(p, \rho)$  o círculo centrado em  $p$  de raio  $\rho$  e, para um par de pontos  $q$  e  $s$ , chamemos de  $r(q, s)$  a reta que passa por eles e  $C(p, \rho_1, \rho_2)$  a coroa circular centrada em  $p$  de raio menor  $\rho_1$  e raio maior  $\rho_2$ .

**Teorema 9** *O valor máximo permitido para o raio dos círculos de perturbação dos pontos de  $P$  de modo que a ordem circular destes pontos seja mantida é:*

$$\rho = \min \frac{d(p, r(q, s))}{2}, \forall p, q, s \in P$$

onde os pontos  $p$ ,  $q$  e  $s$  devem ser distintos entre si (ver figura 6.1).

### Demonstração:

Primeiramente, mostremos que este valor é válido. Sejam  $p' \in C(p, \rho)$ ,  $q' \in C(q, \rho)$  e  $s' \in C(s, \rho)$ . Para se alterar a ordem circular entre os pontos  $p$ ,  $q$  e  $s$ , é

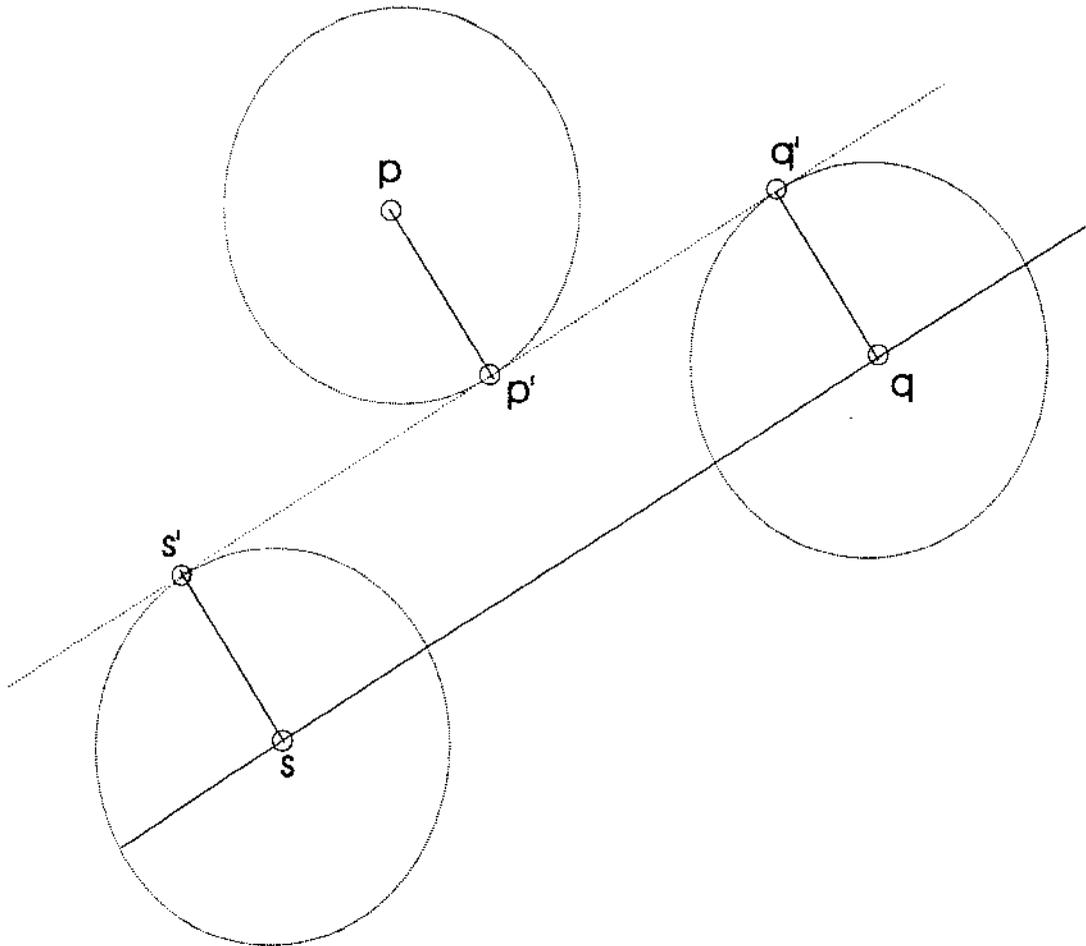


Figura 6.2: Ilustração do Teorema 9

necessário que a reta  $r(q', s')$  intersecte o interior do círculo  $C(p, \rho)$ . Mas, como estamos tomando  $\rho \leq d(p, r(q, s))/2$ , temos que qualquer ponto no interior do círculo  $C(p, \rho)$  distará de  $p$  no máximo, metade da distância  $d(p, r(q, s))$ . Isto segue do fato de que, no pior caso,  $p'$ ,  $q'$  e  $s'$  podem ser colineares, como é mostrado na figura 6.2.

Portanto, as ordens circulares não serão alteradas.

Provemos agora que este valor é máximo. Sejam  $\bar{p}$ ,  $\bar{q}$  e  $\bar{s}$  pontos de  $P$  tais que  $\rho = d(\bar{p}, r(\bar{q}, \bar{s}))/2$  e seja a projeção  $p_{\perp}$  de  $\bar{p}$  perpendicularmente sobre  $r(\bar{q}, \bar{s})$ . Assuma, sem perda de generalidade que  $\bar{p}$  está do lado esquerdo da reta  $r(\bar{q}, \bar{s})$ .

Temos que  $d(\bar{p}, p_{\perp}) = d(\bar{p}, r(\bar{q}, \bar{s})) = 2\rho$ .

Se tomarmos os raios dos círculos como sendo  $\rho' = \rho + \delta$ , com  $\delta > 0$ , teremos pontos  $q' \in C(\bar{q}, \rho')$ ,  $s' \in C(\bar{s}, \rho')$  e  $p' \in C(\bar{p}, \rho')$ , tais que  $p'$  está à direita da reta  $r(q', s')$  (ver figura 6.4), alterando a ordem original de  $\bar{p}$ ,  $\bar{q}$  e  $\bar{s}$ .  $\diamond$

Temos então uma forma simples de encontrar o raio  $\rho$  do círculo. Uma outra maneira de calcular  $\rho$  é tomar metade da menor das alturas de todos os triângulos formados por

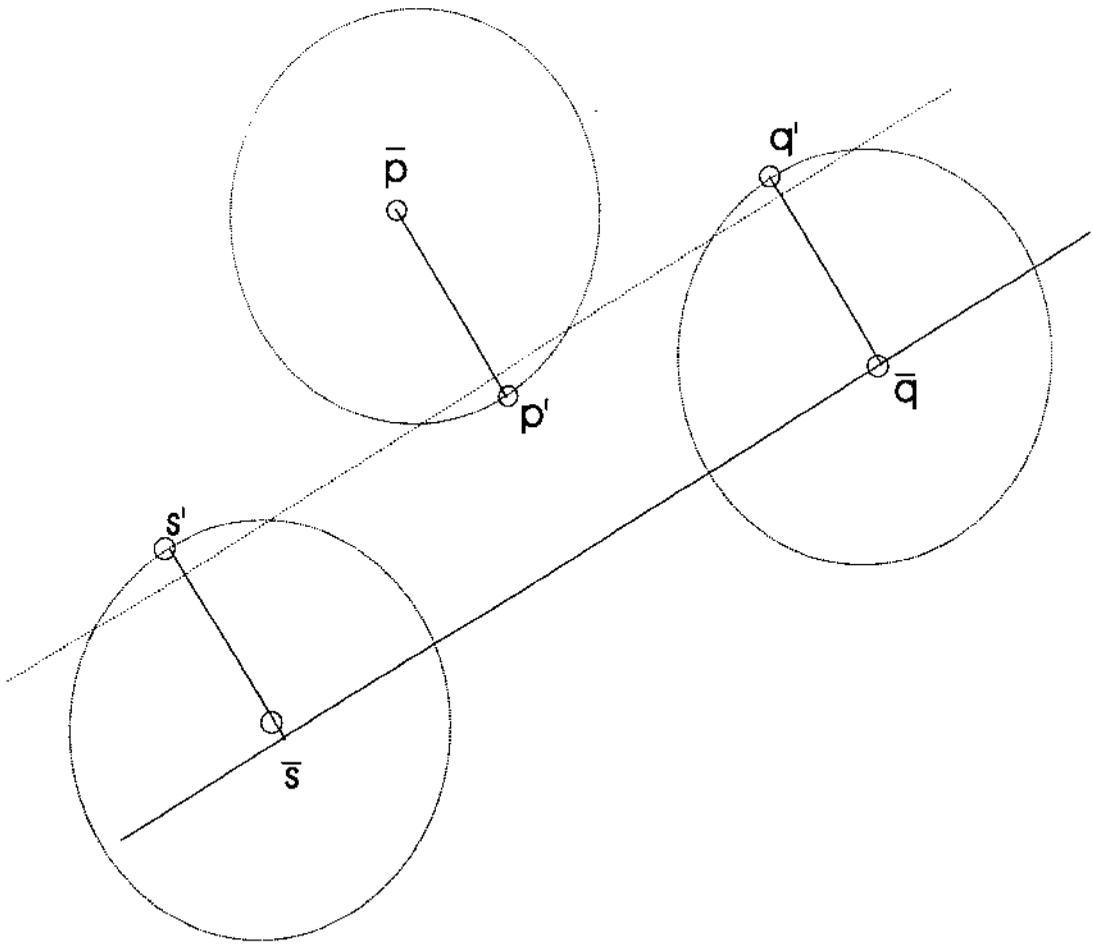


Figura 6.3: Situação em que a ordem circular não é mantida

vértices que são pontos de  $P$ . A princípio, para calcular o valor de  $\rho$ , teríamos que olhar todas as triplas de pontos de  $P$ , o que gastaria tempo  $O(k^3)$ . No entanto, podemos demonstrar o seguinte:

**Lema 10** *O valor de  $\rho$  pode ser calculado em tempo  $O(k^2)$ .*

**Demonstração:**

Seja  $P_i = \{p_i = p_{i,1}, p_{i,2}, p_{i,3}, \dots, p_{i,k}\}$  a lista circular dos pontos de  $P$  ao redor do  $p_i \in P$ .

**Lema 11** *A tripla de pontos que define  $\rho$  será dado por um ponto  $p_i$  e dois pontos consecutivos na lista circular  $P_i$ .*

**Demonstração:**

Suponha, por absurdo, que a tripla que define  $\rho$  seja dado pelos pontos  $p_i$ ,  $p_j$  e  $p_k$ , de forma que  $p_j$  e  $p_k$  não sejam consecutivos na lista  $P_i$ ,  $p_i$  e  $p_k$  não sejam consecutivos na lista  $P_j$  e  $p_i$  e  $p_j$  não consecutivos na lista  $P_k$ . Em particular, suponha que entre os pontos  $p_j$  e  $p_k$ , exista um ponto  $p_\ell$  na lista  $P_i$ . Seja  $\lambda(q_1, q_2, q_3)$  a menor das alturas do triângulo definido pelos pontos  $q_1$ ,  $q_2$  e  $q_3$ . O valor de  $\lambda(p_i, p_j, p_\ell)$  será, evidentemente, menor que  $\lambda(p_i, p_j, p_k)$ . Portanto, os pontos  $p_i$ ,  $p_j$  e  $p_k$  não podem definir a menor das alturas dos triângulos formados por vértices que são pontos de  $P$ .  $\diamond$

Usando o lema acima, para calcularmos o valor de  $\rho$ , basta percorrer todas as  $k$  listas circulares  $P_i$ , comparando as distâncias entre os  $p_{i,j}$  às retas  $r(p_i, p_{i,j+1})$  e  $r(p_i, p_{i,j-1})$ .  $\diamond$

Além disso, podemos enunciar o seguinte resultado complementar:

**Lema 12** *O problema do cálculo do valor de  $\rho$  possui cota inferior de  $\Omega(k^2)$ .*

**Demonstração:**

Em [5], é apresentado o seguinte problema:

**Instância:** Um conjunto  $P$  de  $k$  pontos.

**Questão:** Determinar se os pontos de  $P$  contêm degeneração afim, isto é,  $P$  contém 3 pontos colineares.

e apresenta-se uma demonstração de que a cota inferior para este problema é de  $\Omega(k^2)$ .

Façamos uma redução do problema acima ao problema de cálculo de  $\rho$ . A entrada dos dois problemas é a mesma. Se a saída do problema de cálculo de  $\rho$  for 0, podemos afirmar que os pontos contêm degeneração afim. Caso contrário, podemos afirmar que eles não contêm. Portanto, em tempo  $O(k)$ , podemos reduzir um problema ao outro e afirmar que a cota inferior do problema acima pode ser transferida para o problema de cálculo de  $\rho$ .  $\diamond$

## 6.5 Análise do Problema de Casamento com Perturbação em Duas Dimensões

A dificuldade da adaptação do algoritmo de casamento com perturbação do caso de uma dimensão para o caso de duas dimensões se deve à perda de uma propriedade: no caso de uma dimensão, uma vez que associamos um ponto de  $S$  ao ponto  $p_1$ , temos a garantia de que no máximo dois pontos de  $S$  estarão associados a um ponto de  $P$ : o ponto mais próximo e à esquerda de  $p_i + \tau$  e o ponto mais próximo e à direita de  $p_i + \tau$ . No caso de duas dimensões, não temos esta garantia. Temos que considerar todos os pontos que estão dentro do círculo de perturbação ao redor de  $p_i$ . Exceto por esta diferença, podemos novamente usar o algoritmo de uma dimensão para construir o algoritmo de duas dimensões.

No caso de uma dimensão, iniciávamos os valores de  $\epsilon_e$  e  $\epsilon_d$  com  $-\rho$  e  $\rho$ , respectivamente. No caso de duas dimensões, iniciaremos a região de perturbação com um círculo de raio  $\rho$ . O algoritmo de duas dimensões está baseado na colocação de uma região de perturbação  $\mathcal{R}_i$  em torno de cada ponto  $p_i$  de  $P$ . Em cada uma destas regiões estamos procurando pontos de  $S$ .

Uma outra dificuldade que temos para o casamento com perturbação em duas dimensões é que no caso exato o ponto  $s_j$  associado a  $p_1$  determina a translação, e o ponto de  $S$  associado a  $p_2$  determina a rotação. No caso com perturbação, não podemos determinar a translação, nem tampouco a rotação com os dois primeiros pontos. Podemos definir uma região de translação e um intervalo de rotação permitidos, porém não podemos determinar estes valores.

A liberdade de translação é resolvida tomando-se inicialmente a região de liberdade como sendo um círculo de raio  $2\rho$ . Quanto à liberdade de rotação que temos para realizar o casamento, podemos solucioná-la, fazendo-se a interseção do círculo como um coroa circular centrada em  $s_{m_1}$  de raio menor  $d(s_{m_2}, s_{m-1}) - \rho$  e raio maior  $d(s_{m_2}, s_{m_1}) + \rho$ .

Para todo ponto de  $S$  que se casa com  $p_2$ , devemos verificar se este casamento com perturbação pode ser estendido aos demais pontos de  $P$ . No processo de extensão destes casamentos, as regiões de perturbação devem ser atualizadas (semelhantemente à atualização de  $\epsilon_e$  e  $\epsilon_d$  no caso de uma dimensão).

A cada passo desta extensão, a região de perturbação pode sofrer encolhimento, dado pela sua interseção com a união dos círculos  $C(s_i, \rho)$ , onde os pontos  $s_i$ , são os candidatos a casamento neste passo.

Prosseguindo desta maneira, iremos procurar pontos de  $S$  nas regiões  $\mathcal{L}_i$ . Se chegarmos até  $i = k$  e existir um ponto de  $S$  dentro de  $\mathcal{L}_k$ , podemos afirmar que existe um casamento com perturbação de  $P$  em  $S$ , embora não possamos ainda determinar qual é este casamento. Para determinar o casamento, temos que fazer uma outra varredura, determinando que pontos de  $S$  estão dentro da região  $\mathcal{L}_k$ , quando transladada a cada um dos pontos de  $P$ . Podemos garantir que em cada uma destas regiões haverá um ponto de  $S$ . Portanto, determinaremos o casamento com perturbação.

## 6.6 Representação da Região de Perturbação

A região de perturbação é inicialmente um círculo e, a cada passo, estamos atualizando esta região, considerando os pontos que caem dentro de dela, ou a uma distância máxima  $\rho$  dela. Esta atualização é dada pela intersecção da região atual com a união de alguns círculos de mesmo raio. Portanto, a região  $\mathcal{R}_i$  será dada por:

$$\mathcal{R}_i = C(p_i + \tau, \rho) \cap C(s_{m_2}, d(s_{m_2}, s_{m_1}) - \rho, d(s_{m_2}, s_{m_1}) + \rho) \bigcap_{i=3}^k \left\{ \bigcup_{j=1}^{m_i} C(s_{i,j}, \rho) \right\}$$

Um aspecto que temos que ressaltar é o fato de que um ponto de  $S$  aparecerá, no máximo, uma vez nesta expressão. Isto por que como estamos pedindo a manutenção da ordem circular, cada ponto de  $S$  estará no máximo em uma região de liberdade, quando esta é colocada centrada em um ponto de  $P$ .

Se cada ponto de  $S$  estiver dentro da região de liberdade  $\mathcal{L}$  quando esta é colocada em algum ponto de  $P$ , todos eles aparecerão na expressão acima exatamente uma vez.

Já que todos os círculos que definem a região de perturbação  $\mathcal{R}_i$  têm o mesmo raio, basta-nos guardar o centro de cada círculo e o valor do raio menor da coroa circular. Representaremos a região  $\mathcal{R}_i$  através de uma coleção de listas de pontos (ver figura 6.6). A verificação de que um ponto está dentro de  $\mathcal{L}_i$  pode ser feita em tempo proporcional ao tamanho da coleção. No pior caso, gastaremos tempo proporcional a  $O(n)$ , pois a fronteira de  $\mathcal{L}_i$  é formada por  $O(n)$  arcos de círculos.

Utilizaremos indistintamente  $\mathcal{R}$  para denotar a região de perturbação ou sua representação de vetores.

## 6.7 Construção de Solução por Indução

Solucionemos, mais uma vez, o problema de casamento em duas dimensões com perturbação, utilizando a técnica de indução. Se o conjunto  $P$  possui apenas um ponto, po-

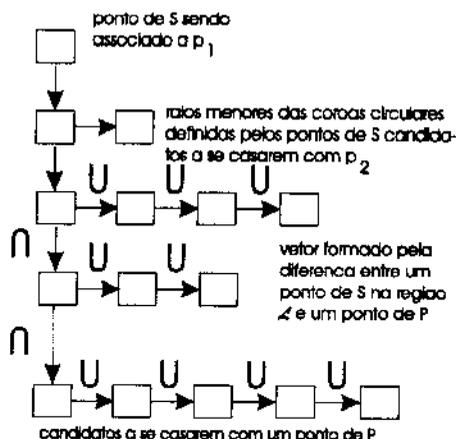


Figura 6.4: Representação da região R como uma lista de pontos

demos associá-lo a todos os pontos de  $S$ . Consideremos, portanto, que o conjunto  $P$  possui pelo menos 2 pontos.

Definamos  $\Psi$  como a composição de translação que leva  $p_1$  em  $s_{m_1}$  com a rotação que leva  $p_2$  em  $s_{m_1, m_2}$ . Dividamos o problema em duas partes. Na primeira, iremos determinar uma coleção de pares de valores  $(\{m_1, m_2, \dots, m_k\}, \mathcal{R})$ , onde os índices  $m_i, i > 2$  representam o índice do último ponto de  $S_i$  que pode ser associado a  $p_i$ . Na segunda parte, iremos fazer uma varredura para determinar que índice em cada um destes conjuntos determina um ponto que, associado a  $p_i$ , define um casamento.

Temos que dividir o problema em duas partes para considerar a liberdade que temos quando associamos um ponto de  $S$  a  $p_1$ . Diremos que uma região está ancorada em um ponto  $p_i$  quando o transladamos, de forma que sua origem coincida com  $p_1$ . Temos a seguinte questão:

### Determinação de um Conjunto de Índices que Definem um Casamento com Perturbação em Duas Dimensões

**Instância:** Um conjunto  $P$  de  $k$  pontos em  $\mathbb{R}^2$  (padrão) e um conjunto  $S$  de  $n$  pontos em  $\mathbb{R}^2$  (amostra), com  $k \leq n$ .

**Questão:** Determinar uma coleção de pares de conjuntos  $(\{m_1, m_2, \dots, m_k\}, \mathcal{R})$ , tais que existe pelo menos um ponto de  $S$  em cada uma das regiões  $\mathcal{L}_i$ , formada pelos pontos que estão a uma distância máxima  $\rho$  da região  $\mathcal{R}$  quando ancorada em  $\Psi(p_1)$ .

Vamos escrever a seguinte proposição:

**Proposição 13** *É possível determinar a coleção de pares de conjuntos descrita acima para os conjuntos  $P$  e  $S$ .*

**Demonstração:**

Por indução em  $k$ :

**Caso Base:** Por estarmos trabalhando em duas dimensões, temos o seguinte caso base: Se  $k = 2$ , devemos fornecer os pares de valores  $(\{i, j\}, \{\mathcal{R}\})$ , para cada um dos valores  $1 \leq i \leq n, 1 \leq j \leq n, i \neq j$ , se a seguinte inequação for verdadeira:

$$|d(p_1, p_2) - d(s_i, s_j)| \leq 2\rho$$

Neste caso, a região  $\mathcal{R}$  será formada pela união de todos os vetores  $\overrightarrow{s_\ell - p_2}$ , tal que o ponto  $s_\ell$  esteja dentro do círculo de raio  $2\rho$ , centrado no ponto  $p_2$ .

**Hipótese:** Podemos determinar todas as duplas  $(M, \mathcal{R})$  que definem todos os casamentos possíveis de  $P'$  em  $S$ , onde  $P'$  tem  $k - 1$  pontos e  $S$  tem  $n$  pontos.

**Passo:** Se  $k > 2$ , pela hipótese de indução, é possível resolver o problema para conjuntos  $P'$  de  $k - 1$  pontos e  $S$  de  $n$  pontos.

Se o conjunto obtido pela aplicação da hipótese de indução for vazio, podemos afirmar que não existe casamento com perturbação de  $P$  em  $S$ . Se, com a aplicação da hipótese de indução aos  $k - 1$  primeiros pontos de  $P$  e  $S$  obtivermos um conjunto de duplas  $(M = \{m_1, m_2, \dots, m_{k-1}\}, \mathcal{R})$ , devemos procurar estender pelo menos uma destas duplas para o último ponto de  $P$ .

Para isto, basta verificar se existe algum ponto de  $S$  na região  $\mathcal{R}$ , ancorada em  $\Psi(p_k)$  e atualizar os conjuntos  $M$  e  $\mathcal{R}$  de forma a refletir os pontos que estão nesta região, caso haja algum. O algoritmo apresentado abaixo detalha a forma como estes conjuntos serão atualizados.  $\diamond$

## 6.8 Algoritmo para Casamento com Perturbação em Duas Dimensões

O algoritmo para casamento com perturbação é baseado no algoritmo para casamento exato. Utilizamos as notações abaixo, além daquelas apresentados para o casamento exato:

1.  $(p_1, p_2, p_i) \approx_{\mathcal{R}_{i-1}} (s_{m_1}, s_{m_1, m_2}, s_{m_1, m_i})$  se o ponto  $s_{m_1, m_i}$  estiver dentro da região de liberdade  $\mathcal{L}_{i-1}$ , definida a partir da região  $\mathcal{R}_{i-1}$ , quando esta é ancorada em  $\Psi(p_i)$ ;
2.  $(p_1, p_2, p_i) \approx_\rho (s_{m_1}, s_{m_1, m_2}, s_{m_1, m_i})$  se o ponto  $s_{m_1, m_i}$  estiver no cone definido pelo ângulo  $\sphericalangle(\Psi(p_1), p_i^c, p_i^d)$ , onde  $p_i^c$  e  $p_i^d$  são os pontos de interseção do círculo, centrado em  $\Psi(p_i)$  de raio  $2\rho$  e as (duas) retas que passam por  $\Psi(p_1)$  e tangenciam este círculo. (ver figura 6.5).
3.  $(s_i, s_j, s_\ell) - (p_1, p_2, p_k)$  denota o vetor  $\overrightarrow{s_\ell - \Psi(p_k)}$ .

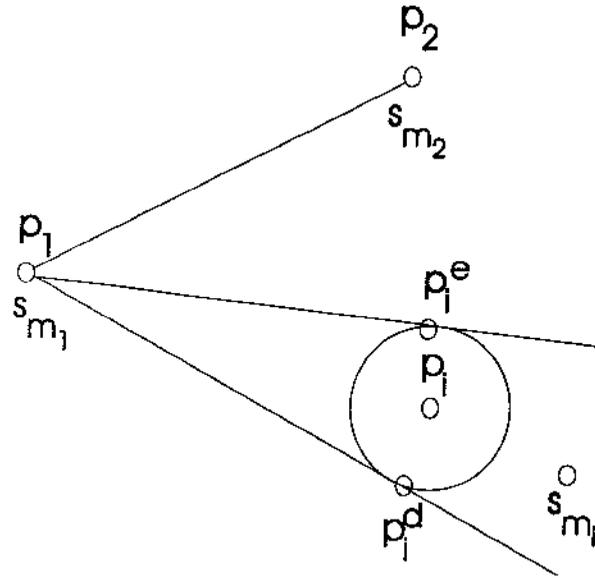


Figura 6.5: Uma situação em que  $(p_1, p_2, p_i) \approx_\rho (s_{m_1}, s_{m_1, m_2}, s_{m_1, m_i})$

Usando as notações acima, podemos enunciar o algoritmo para casamento com perturbação em duas dimensões. O algoritmo é basicamente o mesmo do caso exato. A única alteração é que o procedimento *EncontraÍndice* retorna agora um conjunto de duplas de valores  $(M_k, \mathcal{R}_k)$ , onde  $M_k$  representa o conjunto de índices e  $\mathcal{R}_k$  a região de perturbação.

**Algoritmo** *Casamento Com Perturbação<sub>2</sub>*

1. Ordene circularmente os conjunto  $S$  e  $P$ ;
2. [Determina o valor de  $\rho$  ]  
**Seja**  $\rho = \text{Determina}\rho (P)$ ;
3. [Chama o procedimento que devolve o conjunto de índices ]  
**Seja**  $\Gamma = \text{EncontraÍndice} (P, S, \rho)$ ;
4. [Verifica se a resposta é um conjunto vazio ]  
**se**  $\Gamma \neq \{\}$ , **então**
  - 4.1 [Podemos tomar qualquer conjunto ]  
**Seja**  $(M_k, \mathcal{R}_k) \in \Gamma$ ;
  - 4.2 [Retorna o conjunto alterado  $M_k$  ]  
**retorne**  $\text{Busca} (M_k, \mathcal{R}_k)$ ;
5. **senão retorne**  $(\{\})$ . [Um casamento com perturbação não foi encontrado.]

**procedimento** *EncontraÍndice*  $(P, S, \rho)$

1. [Caso base: verifica se cada dupla de pontos de  $S$  apresenta um casamento com  $p_1$  e  $p_2$  ]  
se  $k = 2$  então retorne *CasoBase* ( $P, S, \rho$ )
  2. senão
    - 2.1 [Obtém todos os casamentos para  $k - 1$  pontos ]  
Seja  $\Gamma = \text{EncontraÍndice} (\{p_1, \dots, p_{k-1}, \{s_1, \dots, s_{n-1}\}, \rho)$ ;
    - 2.2 [Inicializa o valor do conjunto que será retornado ]  
Seja  $\Gamma' = \{\}$ ;
    - 2.3 [Itera em todos os valores retornados pela recursão ]  
para cada  $(\{m_1, m_2, \dots, m_{k-1}\}, \mathcal{R}_{k-1}) \in \Gamma$  faça
      - 2.3.1 [Inicializa o valor de  $m_k$  ]  
Faça  $m_k = m_{k-1}$ ;
      - 2.3.2 [O procedimento *Casa* verifica se os pontos  $p_k$  e  $s_{m_k}$  casam ]  
enquanto não *Casa* ( $p_1, p_2, p_k, s_{m_1}, s_{m_1, m_2}, s_{m_1, m_k}$ ) e  
 $((p_1, p_2, p_k) \prec (s_{m_1}, s_{m_1, m_2}, s_{m_1, m_k}))$  faça  
[Incrementa o valor de  $m_k$  ]  
Faça  $m_k = (m_k + 1) \bmod n$ ;
      - 2.3.3 [Verifica se o ponto  $p_k$  se casa com  $s_{m_k}$  ]  
se *Casa* ( $p_1, p_2, p_k, s_{m_1}, s_{m_1, m_2}, s_{m_1, m_k}$ ) então  
[Inicializa o valor de  $\mathcal{R}$  ]  
Seja  $\mathcal{R} = \{(s_{m_1}, s_{m_1, m_2}, s_{m_1, m_k}) - (p_1, p_2, p_k)\}$ ;  
[Incrementa o valor de  $m_k$  ]  
Faça  $m_k = (m_k + 1) \bmod n$ ;  
[Enquanto o ponto  $s_{m_k}$  se casa com  $p_k$  ]  
enquanto  $(p_1, p_2, p_k) \approx_\rho (s_{m_1}, s_{m_1, m_2}, s_{m_1, m_k})$  faça
        - [Acrescenta o vetor correspondente à região  $\mathcal{R}$  ]  
Faça  $\mathcal{R} = \mathcal{R} \cup \{(s_{m_1}, s_{m_1, m_2}, s_{m_1, m_k}) - (p_1, p_2, p_k)\}$ ;
        - [Incrementa o valor de  $m_k$  ]  
Faça  $m_k = (m_k + 1) \bmod n$ ;
3. retorne  $\Gamma'$ .

Finalmente, apresentamos os procedimentos *Casa*, *Busca*, *Determina* e *CasoBase*:

**procedimento** *Casa* ( $p_1, p_2, p_i, s_1, s_2, s_j, \mathcal{R}$ )

retorne  $(p_1, p_2, p_i) \approx_{\mathcal{R}} (s_1, s_2, s_j)$ .

**procedimento** *Busca* ( $\{m_1, m_2, \dots, m_k\}, R = \{\mathcal{R}_2, \dots, \mathcal{R}_k\}$ )

1. [O ponto  $p_1$  é associado a  $s_{m_1}$  ]  
 [Procuramos um ponto a ser associado a  $p_2$  ]  
 Seja  $\ell_2 = m_2$ ;
2. [Itera em  $\ell_2$  até encontrar um ponto que se case com  $p_2$ , em relação à região  $\mathcal{R}$  ]  
**enquanto** (não *Casa* ( $p_1, p_2, p_2, s_{m_1}, s_{m_2}, s_{\ell_2}, \mathcal{R}$ )) **faça**  
     [Decrementa o valor de  $\ell_2$  ]  
     **Faça**  $\ell_2 = (\ell_2 - 1) \bmod n$ ;
3. [Procuramos pontos de  $S$  que se casam com os demais pontos de  $P$  ]  
**para**  $i = 3$  até  $k$  **faça**  
     [Itera procurando ponto de  $S$  que se case com  $p_i$  ]  
     **enquanto** (não *Casa* ( $p_1, p_2, p_i, s_{m_1}, s_{m_1, m_2}, s_{m_1, m_i}, \mathcal{R}$ )) **faça**  
         [Decrementa o valor de  $m_i$  ]  
          $m_i = (m_i - 1) \bmod n$ ;
4. [Retorna o conjunto de índices ]  
 retorne  $\{m_1, \dots, m_k\}$ .

**procedimento** *Determinar* ( $P$ )

1. [Inicializa  $\rho$  com um valor suficientemente grande ]  
 Seja  $\rho = d(p_1, p_2)$ ;
2. [Para cada par de pontos  $(p_i, p_{i,j})$ , faça ]  
**para**  $i = 1$  até  $k$  **faça**  
     2.1 **para**  $j = 2$  até  $k - 1$  **faça**  
         2.1.1 [Verifica se a altura do triângulo formado pelos pontos  $p_i, p_{i,j}$  e  $p_{i,j+1}$  é menor que  $\rho$  ]  
         **Faça**  $\rho = \min\{\rho, d(p_{i,j}, r(p_i, p_{i,j+1})), d(p_{i,j+1}, r(p_i, p_{i,j}))\}$ ;
- 2.2 [Realiza a mesma verificação para o primeiro e último pontos da lista  $P_i$  ]  
     **Faça**  $\rho = \min\{\rho, d(p_{i,2}, r(p_i, p_{i,k})), d(p_{i,k}, r(p_i, p_{i,2}))\}$ ;
3. retorne  $\rho/2$ .

**procedimento** *CasoBase* ( $P, S, \rho$ )

1. [Inicializa a coleção de valores que definem um casamento ]  
Seja  $\Gamma = \{\}$ ;
2. [Itera em cada par de pontos  $(s_i, s_{i,j})$  de  $S$  ]  
para  $i = 1$  até  $n$  faça
  - 2.1 para  $j = 2$  até  $n$ , faça
    - 2.1.1 [Se  $s_i$  e  $s_{i,j}$  casam com  $p_1$  e  $p_2$  ]  
se *Casa*  $(p_1, p_2, p_2, s_i, s_{i,j}, s_{i,j}, \{\})$  então
      - [Inicializa  $\mathcal{R}$  com o vetor correspondente ]  
Seja  $\mathcal{R} = \{(s_i, s_{i,j}, s_{i,j}) - (p_1, p_2, p_2)\}$ ;
      - [Inicializa o valor de  $\ell$  ]  
Seja  $\ell = (j - 1) \bmod n$ ;
      - [Verifica se  $s_\ell$  pode se casar também com  $p_2$  ]  
enquanto  $(p_1, p_2, p_2) \approx_\rho (s_i, s_{i,j}, s_{i,\ell})$  então
        - [Se  $s_\ell$  se casa com  $p_2$  ]  
se *Casa*  $(p_1, p_2, p_2, s_i, s_{i,j}, s_{i,\ell}, \{\})$  então
          - [Adiciona-o à região  $\mathcal{R}$  ]  
Faça  $\mathcal{R} = \mathcal{R} \cup \{d(s_{i,j}, s_i) - \rho\}$ ;
          - [Decrementa o valor de  $\ell$  ]  
Faça  $\ell = (\ell - 1) \bmod n$ ;
      - [Faz o mesmo procedimento para os pontos seguintes a  $s_{i,j}$  ]  
Faça  $\ell = (j + 1) \bmod n$   
enquanto  $(p_1, p_2, p_2) \approx_\rho (s_i, s_{i,j}, s_{i,\ell})$  então
        - se *Casa*  $(p_1, p_2, p_2, s_i, s_{i,j}, s_{i,\ell}, \{\})$  então
          - Faça  $\mathcal{R} = \mathcal{R} \cup \{d(s_i, s_{i,j}) - \rho\}$ ;
          - Faça  $\ell = (\ell + 1) \bmod n$ ;
      - [Acrescenta  $\mathcal{R}$  à  $\Gamma$  ]  
Faça  $\Gamma = \Gamma \cup (\{i, j\}, \{\mathcal{R}\})$ ;
3. retorne  $\Gamma$ ;

A corretude do algoritmo *CasamentoComPerturbação<sub>2</sub>* segue imediatamente da seção 6.7.

Devemos observar que, para a obtenção de todos os casamentos, bastaria retornar todos os elementos de  $\Gamma$  ao invés de apenas um, depois que cada um deles for devidamente atualizado pelo procedimento *Busca*.

## 6.9 Análise de Complexidade do Algoritmo

O algoritmo *CasamentoComPerturbação*<sub>2</sub> que enunciamos acima é basicamente uma modificação do algoritmo *CasamentoExato*<sub>2</sub>. A única alteração existente em termos de complexidade é no tempo de verificação da relação  $\approx_{\mathcal{R}_i}$  que toma tempo proporcional ao número de elementos da região  $\mathcal{R}$ . Antes, este teste era feito em tempo constante.

Os testes destas relações são feitos no máximo  $O(kn^2)$  vezes, e, uma vez que fixamos o casamento de  $p_1$  com um ponto  $s_{m_1} \in S$ ,  $s_{m_i}$ , o candidato a casamento com  $p_i$ , pode assumir todos os demais valores, porém nunca volta atrás (passo 2.4.1). Daí, a complexidade é dada por  $O(kn^3)$ , devido às  $n$  possíveis associações de  $p_i$  em  $S$ .

Com isso, estabelecemos:

**Teorema 14** *Dado um conjunto  $P$  de  $k$  pontos e um conjunto  $S$  de  $n \geq k$  pontos no plano, o problema de Casamento com Perturbação em duas dimensões pode ser resolvido em tempo  $O(kn^3)$ .*

# Capítulo 7

## Casamento de Pontos em Dimensões Maiores

### 7.1 Definição do Casamento com Perturbação em Dimensões Maiores

Neste capítulo, descrevemos um algoritmo para casamento com perturbação em dimensões maiores, através de uma generalização natural do algoritmo para casamento com perturbação em duas dimensões. Não apresentaremos o algoritmo para casamento exato pois, primeiramente, tal algoritmo é apresentado em [3], e o algoritmo para casamento com perturbação em  $d$  dimensões vai advir da generalização do algoritmo para casamento com perturbação em duas dimensões, e não do algoritmo para o caso exato em  $d$  dimensões.

O problema de casamento com perturbação em  $d$  dimensões pode ser generalizado imediatamente da definição 3 da seção 6.1, conforme se vê abaixo:

**Definição 4 Casamento com Perturbação em  $d$  Dimensões** *Sejam  $P$  um conjunto de  $k$  pontos.  $S$  um conjunto de  $n \geq k$  pontos em  $\mathbb{R}^d$  e  $\rho > 0$ . Dizemos que existe um casamento com perturbação  $\rho$  em  $d$  dimensões de  $P$  em  $S$  se existem funções*

1.  $f$  injetora do conjunto  $P$  no conjunto  $S$ ; e
2.  $\phi$  combinação de translação e rotação do conjunto  $P$ ,

tais que estas duas funções satisfaçam à seguinte relação:

$$d(f(p), \phi(p)) \leq \rho, \forall p \in P$$

### Casamento com Perturbação em $d$ Dimensões

**Instância:** Um conjunto  $P$  de  $k$  pontos em  $\mathbb{R}^d$  (padrão) e um conjunto  $S$  de  $n$  pontos em  $\mathbb{R}^d$  (amostra), com  $k \geq n$ .

**Questão:** Determinar uma coleção de pares de conjuntos  $(\{m_1, m_2, \dots, m_k\}, \mathcal{R})$ , tal que existe pelo menos um ponto de  $S$  em cada uma das regiões formada pelos pontos que estão a uma distância máxima  $\rho$  da região  $\mathcal{R}$  quando trasladamos o conjunto  $P$  de forma que os pontos  $p_1$  e  $s_{m_1}$  estejam coincidentes e rotacionamos este conjunto de forma que o simplexo  $p_1, p_2, \dots, p_k$  seja congruente ao simplexo  $s_{m_1}, s_{m_2}, \dots, s_{m_k}$ .

A seguir, usaremos algumas notações já introduzidas na capítulo 6, sem explicitar sua definição, pois sua generalização de duas dimensões para  $n$  dimensões é imediata.

## 7.2 Determinação de um Raio $\rho$ Conveniente

Enunciamos alguns resultados que nos auxiliarão a descrever o algoritmo para casamento com perturbação em  $d$  dimensões. Sejam  $p, q_i \in P, i = 1, \dots, d$ ,  $B(p, \rho)$  a bola centrada em  $p$  de raio  $\rho$  e  $\langle q_1, \dots, q_d \rangle$  o hiperplano definido pelos pontos  $q_1, \dots, q_d$ , onde o conjunto destes pontos tem independência afim e  $p \neq q_i, i = 1, \dots, d$ .

**Teorema 15** *O valor máximo permitido para o raio das bolas de perturbação dos pontos de  $P$  de modo que a ordem circular destes pontos seja mantida é:*

$$\rho = \frac{\min_{p, q_i \in P} d(p, \langle q_1, \dots, q_d \rangle)}{2}$$

### Demonstração:

Mostremos que este valor é válido. Sejam  $p' \in B(p, \rho)$  e  $q'_i \in B(q_i, \rho)$ . Para se alterar a ordem circular entre os pontos  $p$  e  $q_i, i = 1, \dots, d$ , é necessário que o hiperplano  $\langle q_1, \dots, q_d \rangle$  intersecte o interior da bola  $B(p, \rho)$ . Porém, como estamos tomando  $\rho = d(p, \langle q_1, \dots, q_d \rangle)/2$ , temos que qualquer ponto no interior da bola  $B(p, \rho)$  distará de  $p$  no máximo  $d(p, \langle q_1, \dots, q_d \rangle)/2$ . Isto segue do fato de que, no pior caso,  $p'$  pode estar contido no hiperplano  $\langle q'_1, \dots, q'_d \rangle$ . Então, as ordens circulares não serão alteradas.

Provemos que este valor é máximo. Sejam  $\bar{p}, \bar{q}_i \in P, \bar{p} \neq \bar{q}_i, i = 1, \dots, d$ , tais que  $\rho = d(\bar{p}, \langle \bar{q}_1, \dots, \bar{q}_d \rangle)/2$  e seja  $p_\perp$  a projeção de  $\bar{p}$  perpendicularmente sobre  $\langle \bar{q}_1, \dots, \bar{q}_d \rangle$ . Temos que  $d(\bar{p}, p_\perp) = d(\bar{p}, \langle \bar{q}_1, \dots, \bar{q}_d \rangle) = 2\rho$ . Assuma, sem perda de generalidade que  $\bar{p}$  esteja do lado esquerdo do hiperplano  $\langle \bar{q}_1, \dots, \bar{q}_d \rangle$ .

Se tomarmos os raios das bolas como sendo  $\rho' > \rho$ , teremos pontos  $p' \in B(\bar{p}, \rho)$ ,  $q'_i \in B(\bar{q}_i, \rho)$  tais que  $p'$  esteja à direita do hiperplano  $\langle q'_1, \dots, q'_d \rangle$ , alterando a ordem original de  $\bar{p}$  e  $\bar{q}_i, i = 1, \dots, d$ .  $\diamond$

A princípio, para calcular  $\rho$ , teríamos que olhar para todas as  $(d+1)$ -tuplas de pontos de  $P$ , o que gastaria tempo  $O(k^{d+1})$ . No entanto, não é difícil ver que podem ser generalizados os lemas 10 e 11 da seção 6.4 de modo que temos uma maneira de calcular  $\rho$  em tempo  $O(k^d)$ .

Seja  $P_i = \{p_{i,1}, p_{i,2}, \dots, p_{i,d-1}, \dots, p_{i,k}\}$  a lista circular ao redor do hiperplano  $\langle p_{i,1}, \dots, p_{i,d-1} \rangle$ . Omitiremos a prova do seguinte lema:

**Lema 16** *A  $(d+1)$ -tupla de pontos que define  $\rho$  é dada por um ponto  $p_i$  e  $d$  pontos consecutivos na lista circular  $P_i$ .*

Similarmente, é consequência imediata de [5] o seguinte lema:

**Lema 17** *O problema do cálculo de  $\rho$  em  $d$  dimensões possui cota inferior  $\Omega(k^d)$ .*

Quanto à região de perturbação, podemos representá-la da mesma maneira como fizemos para duas dimensões: uma coleção de listas de vetores. A única diferença é que os vetores têm  $d$  coordenadas, o que implica que a verificação de que um ponto está dentro da região de casamento pode ser feita em tempo  $O(nd)$ .

## 7.3 Análise do Problema de $d$ Dimensões

De maneira similar ao que foi feito de duas dimensões, iniciaremos a região de perturbação com uma bola de raio  $\rho$  e, a cada passo iremos colocar esta região em torno de um ponto de  $P$ , procurando pontos de  $S$ .

A translação do conjunto  $P$  é determinada pela associação do ponto  $p_1$  a um ponto de  $S$ . A associação dos pontos  $p_2, \dots, p_d$  com pontos de  $S$  determinam a rotação do conjunto  $P$ . Devemos verificar se o conjunto de associações feitas acima pode ser estendido aos demais pontos de  $P$ .

A cada passo, podemos encolher a região de perturbação, mediante sua interseção com as bolas  $B(s_{i_j}, \rho)$ , onde os pontos  $s_{i_j}$  são os pontos de  $S$  interiores à região de perturbação, colocada no ponto  $p_i \in P$ .

Prosseguindo desta maneira, verificamos se encontramos pontos de  $S$  nas regiões de casamento. Se chegarmos até o último ponto de  $P$ , podemos afirmar que existe um casamento de  $P$  em  $S$ , embora não possamos precisar que pontos definem este casamento. Para isso, fazemos uma segunda varredura, determinando quais pontos candidatos realmente definem um casamento.

## 7.4 Construção de Solução por Indução

Iremos, mais uma vez, dividir o problema em duas partes: na primeira, determinaremos uma coleção de pares de valores  $(\{m_1, m_2, \dots, m_k\}, \mathcal{R})$  onde  $m_i$  para  $i > d$  representa o último ponto de  $S_i$  candidato a se casar com  $p_i$ . Na segunda parte, iremos fazer uma varredura para determinar que índice em cada um destes conjuntos determina um ponto a ser associado a  $p_i$ . Seja  $\Psi$  a composição da translação que leva  $p_1$  a  $s_{m_1}$  com a rotação que leva os pontos  $p_2, \dots, p_d$  serem associados com os pontos  $s_{m_1, m_2}, \dots, s_{m_1, m_d}$ .

Temos o seguinte problema:

### Determinação de um Conjunto de Índices que Definem um Casamento com Perturbação em Duas Dimensões

**Instância:** Um conjunto  $P$  de  $k$  pontos em  $\mathbb{R}^2$  (padrão) e um conjunto  $S$  de  $n$  pontos em  $\mathbb{R}^d$  (amostra), com  $k \leq n$ .

**Questão:** Determinar uma coleção de pares de conjuntos  $(\{m_1, m_2, \dots, m_k\}, \mathcal{R})$ , tais que existe pelo menos um ponto de  $S$  em cada uma das regiões  $\mathcal{L}_i$ , formada pelos pontos que estão a uma distância máxima  $\rho$  da região  $\mathcal{R}$  quando ancorada em  $\Psi(p_1)$ .

Vamos demonstrar a seguinte proposição:

**Proposição 18** *É possível determinar a coleção de pares de conjuntos descrita acima para um conjunto  $P$  de  $k$  pontos e  $S$  de  $n$  pontos em  $\mathbb{R}^d$ .*

#### Demonstração:

Por indução em  $k$ :

**Caso Base:** Como estamos trabalhando em  $d$  dimensões, temos o seguinte caso base: se  $k = d$ , devemos fornecer os pares de valores  $(\{m_1, m_2, \dots, m_d\}, \{\mathcal{R}\})$ , se o ponto  $s_{m_i}$  dista, no máximo,  $2\rho$  de  $p_i$ , para todo  $1 \leq i \leq d$  após a translação e rotação do conjunto  $P$ . Neste caso, a região  $\mathcal{R}$  será formada pela interseção de diversas regiões  $\mathcal{R}_i$ , onde  $\mathcal{R}_i$  é formado pela união de todos os vetores  $\overrightarrow{s_\ell - p_i}$ , tal que o ponto  $s_\ell$  esteja dentro do círculo de raio  $2\rho$ , centrado no ponto  $p_i$ .

**Hipótese:** Podemos determinar todas as duplas  $(M, \mathcal{R})$  que definem todos os casamentos possíveis de  $P'$  em  $S$ , onde  $P'$  tem  $k - 1$  pontos e  $S$  possui  $n$  pontos.

**Passo:** Se  $k > d$ , por indução, sabemos resolver o problema para conjuntos  $P'$  de  $k - 1$  pontos e  $S$  de  $n$  pontos.

Se o conjunto obtido pela aplicação da hipótese de indução for vazio, podemos afirmar que não existe casamento com perturbação de  $P$  em  $S$ . Se com a aplicação

da hipótese de indução aos  $k - 1$  primeiros pontos de  $P$  e  $n$  pontos de  $S$  obtivermos um conjunto de duplas  $(M = \{m_1, m_2, \dots, \{m_{k-1,1}, \dots, m_{k-1,k-1}\}, \mathcal{R})$ , devemos procurar estendê-lo para o último ponto de  $P$ .

Para isto, para cada uma destas duplas, devemos verificar se existe algum ponto de  $S$  na região  $\mathcal{R}$ , colocada no ponto  $\Psi(p_k)$ . Basta agora atualizar os conjuntos  $M$  e  $\mathcal{R}$  de forma a refletir os pontos que estão nesta região.  $\diamond$

## 7.5 Algoritmo para Casamento com Perturbação em Dimensão $d$

O algoritmo para casamento com perturbação em dimensão  $d$  é semelhante ao algoritmo para casamento com perturbação em duas dimensões. As únicas diferenças na sua descrição são a adaptação das notações utilizadas para  $d$ -dimensões e o caso base.

No caso de duas dimensões, verificamos, no caso base, se cada dupla de pontos de  $S$  apresenta um casamento com  $p_1$  e  $p_2$ ; temos, então,  $n(n - 1)$  pares de pontos a serem considerados. No caso de  $d$  dimensões, temos que considerar todos os arranjos de  $d$  elementos dos pontos de  $S$ ; temos  $n!/(n - d)! \in O(n^d)$  possibilidades.

Os procedimentos *EncontraÍndice*, *Casa*, *Busca* e *Determina $\rho$*  devem ser alterados a fim de utilizarem os operadores  $\prec$ ,  $\approx_{\mathcal{R}}$  e  $\approx_{\rho}$  com  $d + 1$  argumentos à sua esquerda e direita.

A natural semelhança do caso de duas dimensões para  $d$  dimensões decorre do fato de que do mesmo modo que em duas dimensões consideramos a lista circular de  $P$  contra cada um das  $n$  listas circulares de  $S$ , no caso de  $d$  dimensões, faz-se o mesmo contra as  $\binom{n}{d-1} \in O(n^{d-1})$  listas circulares de  $S$ .

Temos, então  $O(n^{d-1})$  possíveis candidatos obtidos pelo caso base. Para cada um deles, verifica-se, no máximo,  $kn$  vezes se um ponto está dentro da região de perturbação. Como gasta-se tempo  $O(nd)$  para se saber se um dado ponto está nesta região, podemos estabelecer o seguinte resultado:

**Teorema 19** *Dado um conjunto  $P$  de  $k$  pontos e um conjunto  $S$  de  $n \geq k$  pontos em  $\mathbb{R}^d$ , o problema de Casamento com Perturbação em  $d$  dimensões pode ser resolvido em tempo  $O(dkn^{d+1})$ .*

# Capítulo 8

## Conclusão

### 8.1 Contribuições deste Trabalho

Embora o problema de casamento de pontos já tenha sido estudado anteriormente, as abordagens encontradas na literatura incluem basicamente duas formas: considerando-se apenas o caso exato, com conjuntos de diferentes cardinalidades; considerando-se o caso aproximado, com conjuntos de mesma cardinalidade.

Em [3], os autores apresentam algoritmos eficientes para resolver o problema exato em uma, duas ou mais dimensões.

No presente trabalho enfocamos a obtenção de soluções para os problemas aqui estudados através de construção baseada em provas por indução. A partir da demonstração de que é possível resolver cada problema, extraímos algoritmos para os casos exatos e com perturbação, cuja corretude advém diretamente da prova apresentada.

Em [2], os autores estudam o caso de mapeamento isométrico entre dois conjuntos de pontos de *mesma* cardinalidade, tal que a distância entre os pontos que estão sendo associados seja menor que um valor  $\varepsilon$  dado e apresentam um algoritmo  $O(n^8)$  que resolve este problema.

Nesta tese, apresentamos um algoritmo  $O(kn^3)$  para conjuntos de pontos em duas dimensões (de cardinalidades  $k$  e  $n$ ), e um algoritmo  $O(kn^{d+1})$  para pontos em dimensão  $d$ , ambos para casamentos que associem pontos a uma distância menor que  $\rho$ , valor este determinado pelo algoritmo.

Evidentemente, este resultado pode ser estendido para considerar associações de pontos a uma distância menor ou igual a  $\rho$ . A eficiência do algoritmo é garantida pela manutenção da ordem circular (que determina o valor de  $\rho$ ).

No capítulo 2, descrevemos um algoritmo para casamento exato de pontos em uma dimensão de mesma complexidade do algoritmo apresentado em [3]. Este algoritmo possui complexidade  $O(n \log n + (n - k + 1)k)$ . No capítulo 3, consideramos o caso com per-

turbação e descrevemos um algoritmo eficiente, que possui a mesma complexidade do algoritmo do caso exato.

No capítulo 5, estudamos o problema de casamento exato em duas dimensões e descrevemos um algoritmo  $O(nk^2)$  para este problema, baseado no algoritmo apresentado em [3].

No capítulo 6, passamos ao caso com perturbação. Deparamo-nos com o problema do cálculo do valor do raio máximo de perturbação ( $\rho$ ) que preserve a ordem circular dos pontos casados e verificamos que ele pode ser obtido, resolvendo-se um problema de minimização (não linear). Porém, apresentamos um lema que soluciona eficientemente este problema de forma exata baseado em uma propriedade geométrica. Além disso, definimos uma representação para as regiões de perturbação que nos permite determinar eficientemente se um ponto está na região de casamento (que envolve a região de perturbação). Com isso, apresentamos um algoritmo de complexidade  $O(kn^3)$  que resolve este problema em duas dimensões.

No capítulo 7, estendemos os algoritmos do caso de dimensão 2 para dimensões maiores, de modo que em tempo  $O(kn^{d+1})$  podemos resolver o problema de casamento com perturbação eficientemente em dimensões arbitrárias.

## 8.2 Tratamento de Escala

O tratamento de escala no caso de casamento exato é muito simples. Se estamos em duas dimensões, dado que os pontos  $p_1$  e  $s_{m_1}$  são associados, basta assumir que o ponto  $p_2$  pode se casar com qualquer ponto  $s_j \neq s_{m_1}$ , e esta associação nos produzirá o fator de escala.

Porém, no caso com perturbação, o tratamento não é tão trivial. Embora possamos utilizar o casamento do segundo par de pontos para determinar o fator de escala, não podemos simplesmente fazer uma escala do conjunto  $P$  de forma que, após sua translação e rotação, os pontos  $p_2$  e  $s_j$  fiquem coincidentes, por que seria necessário considerar também escalas provenientes do casamento de  $p_2$  com qualquer ponto  $s_j$  que esteja dentro da região de casamento colocada sobre  $p_2$ .

Desta forma, teríamos um *intervalo* de escalas permitidas, e os valores de  $\varepsilon_e$  e  $\varepsilon_d$ , no caso de uma dimensão e da região de perturbação  $\mathcal{R}$  serão dependentes da escala tomada. O tratamento de escala requer então um tratamento bem mais complexo que o do caso exato.

## 8.3 Determinação de Melhor Casamento

Os algoritmos para casamento com perturbação apresentados aqui encontram *um* casamento de  $P$  em  $S$ . Com uma simples alteração, é possível determinar todos os casamen-

tos. Porém, poderíamos estar interessados em determinar o *melhor* casamento. Para isto, precisamos definir um critério que indique quando um casamento é melhor que outro. Por exemplo, um dos critérios que se pode adotar é o seguinte: dizemos que um casamento  $W_i$  é melhor que outro casamento  $W_j$  se a maior das distâncias entre os pontos que estão sendo associados em  $W_i$  for menor que a maior das distâncias em  $W_j$ .

Para determinar o melhor casamento segundo o critério acima, podemos fazer a seguinte alteração: o procedimento *Busca* poderia devolver a maior distância entre os pares associados, além de atualizar convenientemente o conjunto. Chamamos, então, este procedimento para todos os elementos de  $\Gamma$  e retornamos aquele que apresenta o menor valor retornado pelo procedimento acima.

É uma questão também relevante de se estudar a de determinar um melhor casamento utilizando outros critérios de qualidade de solução.

# Bibliografia

- [1] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity and symmetries of geometric objects. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 308–315, 1987.
- [2] E. M. Arkin, K. Kedem, J. S. B. Mitchell, J. Sprinzak, and M. Werman. Matching points into noise regions: combinatorial bounds and algorithms. In *Proc. 2nd ACM-SIAM Sympos. Discrete Algorithms*, pages 42–51, 1991.
- [3] P. J. de Rezende and D. T. Lee. Point set pattern matching in  $d$ -dimensions. *Algorithmica*, 13:387–404, 1995.
- [4] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. In *Proc. 24th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 93–91, 1983.
- [5] J. Erickson and R. Seidel. Better lower bounds on detecting affine and spherical degeneracies. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci. (FOCS 93)*, pages 528–536, 1993.
- [6] J. E. Goodman and R. Pollack. Multidimensional sorting. *SIAM J. Comput.*, pages 484–507, 1983.
- [7] P. J. Heffernan. Generalized approximate algorithms for point set congruence. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes in Computer Science*, pages 373–384, 1993.
- [8] S. Iwanowski. Linear time algorithms for testing approximate congruence in the plane. *Pre-print*, 1988.
- [9] S. Ray. A heuristic noise reduction algorithm applied to handwritten numeric characters. *Pattern Recognition Letters*, pages 9–12, 1988.
- [10] G. T. Toussaint. Pattern recognition and geometrical complexity. In *Proc. 5th IEEE Internat. Conf. Pattern Recogn.*, pages 1324–1347, 1980.