

# Aplicações da Rede Neural de Hopfield em Clusterização de Pixels e Roteamento em Grafos

*Thiago Christiano Silva*

# Conteúdo

- *Conceitos Relevantes*
  - *Problemas de Otimização*
  - *Conceitos de Estabilidade e Convergência*
  - *Teoremas de Lyapunov*
  - *Rede Neural de Hopfield*
- *Aplicações do modelo de Hopfield*
  - *Clusterização de Pixels em Imagens Nível de Cinza*
  - *Roteamento em Grafos*
- *Considerações Finais*

# Problemas de Otimização

- Um problema de otimização geralmente pode ser descrito por uma função de energia
  - decomposta em duas partes: a **função de custo associada ao problema** e suas **restrições**

$$E = E_c + \sum_{i=1}^r E_r$$

- Restrições
  - São incorporadas à função de energia descritora do problema com a seguinte ideia: se as restrições forem obedecidas, então a função de energia é minimizada

# Conceitos de Estabilidade e Convergência

## • Estabilidade

- O ponto  $\bar{x}$  é estável se  $\exists \epsilon > 0$ , de tal forma que  $\exists \delta = \delta(\epsilon) > 0$ : se  $\|x(0) - \bar{x}\| < \delta$  implica que  $\|x(t) - \bar{x}\| < \epsilon, \forall t \geq 0$

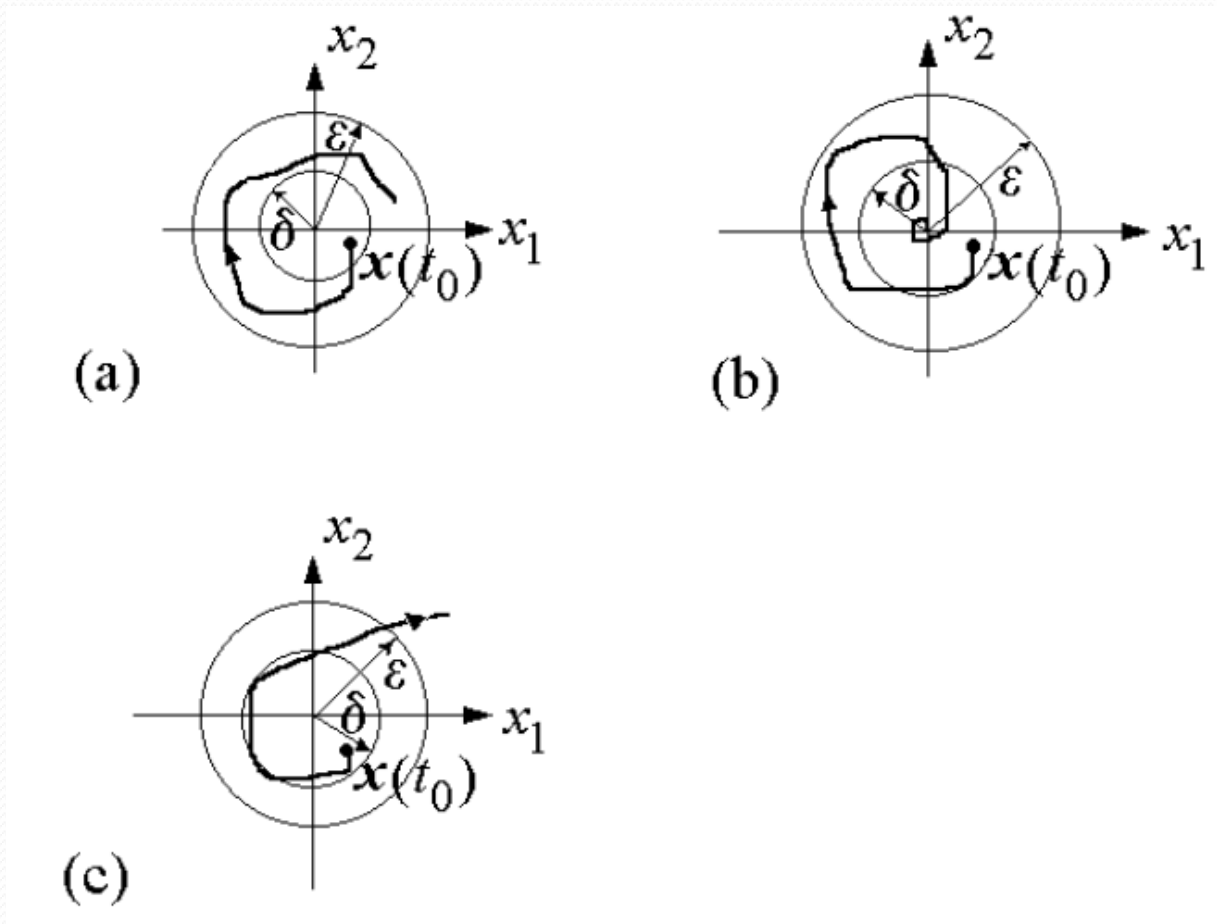
## • Convergência

- O ponto  $\bar{x}$  é convergente se  $\exists \delta > 0$ :  $\|x(0) - \bar{x}\| < \delta$  implica que  $x(t) \rightarrow \bar{x}$ , conforme  $t \rightarrow \infty$

## • Estabilidade Assintótica

- O ponto  $\bar{x}$  é estável assintoticamente se é convergente e estável, não importando a condição inicial

# Ilustração dos conceitos de estabilidade



# Teorema de Lyapunov

- De maneira geral, o **teorema de Lyapunov** apresenta apenas condição suficiente para a **estabilidade** de sistemas dinâmicos
- Uma função de energia  $E(\mathbf{x})$  é Lyapunov se as seguintes propriedades são satisfeitas concomitantemente
  1. A função  $E(\mathbf{x})$  tem derivadas parciais contínuas
  2.  $E(\bar{\mathbf{x}}) = 0$ ,  $\bar{\mathbf{x}}$  sendo um ponto de equilíbrio (estável)
  3.  $E(\mathbf{x}) > 0$ , se  $\mathbf{x}$  não representar um ponto estável
- Ademais,
  - se  $\bar{\mathbf{x}}$  é estável, então  $\frac{dE(\mathbf{x})}{dx} \leq 0$
  - se  $\bar{\mathbf{x}}$  é assintoticamente estável, então  $\frac{dE(\mathbf{x})}{dx} < 0$

# Rede Neural de Hopfield

- É uma rede **recorrente** de uma **única camada**
- É usualmente inicializada com valores  $w_{ij}$  adequados ao problema
- No modelo discreto, o domínio de estados de cada neurônio abrange  $\{-1, 1\}$
- Na sua forma mais simples, a função de ativação utilizada é a função **degrau**
  - **Limitações**: presença de estados espúrios, baixa capacidade de armazenamento quando utilizada para *CAM*
- **É um sistema dinâmico regido por uma Lyapunov**

# Clusterização de *Pixels* em Imagens

- **Clusterização**: consiste em agrupar um conjunto de itens de dados de tal forma que:
  - dados pertencentes a um mesmo grupo compartilham alguma similaridade
  - dados pertencentes a grupos distintos são diferentes de acordo com uma função de similaridade
- Duas abordagens:
  - *Hard-labeling*
  - *Soft-labeling (fuzzy)*



# Clusterização de *Pixels* em Imagens

- Cada *pixel* é representado por um neurônio, o qual é denotado por  $(k, i)$
- A matriz  $U = [\mu_{ki}]$  de dimensões  $N \times c$  representa as partições *fuzzy* (onde  $N$  denota o número de neurônios e  $c$ , o número de clusters desejados)
- Restrições:
  - $\mu_{ki} \in [0,1], 1 \leq k \leq N$  e  $1 \leq i \leq c$
  - $\sum_{i=1}^c \mu_{ki} = 1, 1 \leq k \leq N$
  - $0 \leq \sum_{k=1}^N \mu_{ki} \leq N, 1 \leq i \leq c$

Ex: Matriz  $U$  ( $N = 2, c = 2$ )

	Cluster 1	Cluster 2
Dado 1	0.2	0.8
Dado 2	0.6	0.4

# Clusterização de *Pixels* em Imagens

- O algoritmo *Fuzzy C-Means* é baseado na minimização da seguinte função de custo:

$$J(X, U, V) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ki})^m \|x_k - v_i\|^2$$

onde  $V = [v_1, v_2, \dots, v_c]^T$  indica o vetor de centróides dos clusters,  $x_k$  representa o item de dado e  $m > 1$  denota um parâmetro denominado fator de *fuzzification*

# Clusterização de *Pixels* em Imagens

- Rede de Hopfield *Fuzzy*
  - A principal motivação para o uso do algoritmo *Fuzzy C-Means* é de eliminar a necessidade de encontrar os fatores ponderadores da função de energia
  - A rede é completamente interconectada, **inclusive com autorealimentação**
  - A função de ativação utilizada é a função **logística** (sigmóide)
  - Serão mostradas simplificações da função de energia  $2D$  usualmente utilizada na literatura, até que se consiga uma  $E$  parecida com o algoritmo *Fuzzy C-Means*

# Clusterização de *Pixels* em Imagens

- A função de Lyapunov para o problema pode ser definida por:

$$E = \frac{A}{2} \sum_{k=1}^N \sum_{i=1}^c \mu_{k,i}^m \left[ x_k - \sum_{q=1}^N \frac{x_q \mu_{q,i}^m}{\sum_{h=1}^N \mu_{h,i}^m} \right]^2 + \frac{B}{2} \left[ \left( \sum_{k=1}^N \sum_{i=1}^c \mu_{k,i} \right) - N \right]^2$$

- O primeiro termo modela a distância dos *pixels* ao centro do cluster
- O segundo termo é responsável pela restrição do problema

# Clusterização de *Pixels* em Imagens

- A **qualidade do resultado** de clusterização é muito **sensível** aos **fatores ponderadores  $A$  e  $B$** 
  - Buscar valores ótimos de  $A$  e  $B$  pode constituir uma tarefa lenta, sendo que algumas vezes computacionalmente inviável
- Para amenizar este problema, utiliza-se uma **rede neural de Hopfield**  
*Fuzzy C-Means*
- Simplificações na função de energia:
  - Sabe-se que a soma de cada linha da matriz  $U$  deve valer 1, pois estamos considerando uma classificação *fuzzy*
  - *Por consequência*, a soma de todos os elementos de  $U$  deve valer  $N$ , pois existem  $N$  itens de dados

# Clusterização de *Pixels* em Imagens

- Portanto, à luz das considerações anteriores, o segundo termo resulta em zero e a função de energia associada ao problema fica:

$$E = \frac{A}{2} \sum_{k=1}^N \sum_{i=1}^c \mu_{k,i}^m \left[ x_k - \sum_{q=1}^N \frac{x_q \mu_{q,i}^m}{\sum_{h=1}^N \mu_{h,i}^m} \right]^2$$

- Como há somente um termo, não há a necessidade de contrabalancear os termos da função de energia. Portanto, toma-se  $A = 1$

# Clusterização de *Pixels* em Imagens

- A saída líquida no corpo celular do neurônio  $(k, i)$  é dada por:

$$Net_{k,i} = \left[ x_k - \sum_{q=1}^N \frac{x_q \mu_{q,i}^m}{\sum_{h=1}^N \mu_{h,i}^m} \right]^2 = [x_k - v_i]^2 = D_{k,i}^2$$

- A dominância do cluster  $i$  sobre o vértice  $k$  foi definida como:

$$\mu_{k,i} = \frac{1}{\sum_{j=1}^c \left[ \frac{Net_{k,i}}{Net_{k,j}} \right]^{2/m-1}} = \frac{1}{\sum_{j=1}^c \left[ \frac{D_{k,i}}{D_{k,j}} \right]^{2/m-1}}$$

# Clusterização de *Pixels* em Imagens

- Portanto, a função de energia simplificada, levando em considerações as equações anteriores, é descrita por:

$$J = \sum_{i=1}^c \sum_{k=1}^N (\mu_{k,i})^m D_{k,i}^2$$

- A expressão acima se assemelha muito com a função custo do algoritmo *Fuzzy C-Means*



# Clusterização de *Pixels* em Imagens

- Sumário do Algoritmo

1. Escolha  $c, m > 1$  e  $\epsilon > 0$
2. Normalize os pixels da imagem
3. Determine os centróides primários  $v_i$  (dispostos aleatoriamente)
4. Calcule os valores iniciais dos níveis de dominação  $\mu_{k,i}$
5. Compute os valores dos novos centróides  $v_i$
6. Compute os valores dos novos  $\mu_{k,i}$
7. Compute  $J^{n+1}$
8. Se  $|J^{n+1} - J^n| < \epsilon$ , pare. Caso contrário, volte ao passo (5)

# Simulações Computacionais



(a)



(b)



(c)



(d)

(a) Imagem Original

(b)  $c = 3$

(c)  $c = 5$

(d)  $c = 7$

# Simulações Computacionais



(a)



(b)

(a) Imagem Original

(b)  $c = 3$



(c)

(c)  $c = 5$



(d)

(d)  $c = 7$

# Roteamento em Grafos

- O problema a ser considerado é o de **roteamento com múltiplos destinos**, partindo-se de um vértice-fonte
- Sejam  $s$  o vértice-fonte e  $\mathbf{d} = [d_1, d_2, \dots, d_D]$  os destinos que se desejam atingir, já ordenados
- O problema de otimização é formulado a seguir:

$$\text{Minimize } J^{s \rightarrow \mathbf{d}} = J^{s \rightarrow d_1} + J^{d_1 \rightarrow d_2} + \dots + J^{d_{n-1} \rightarrow d_D}$$

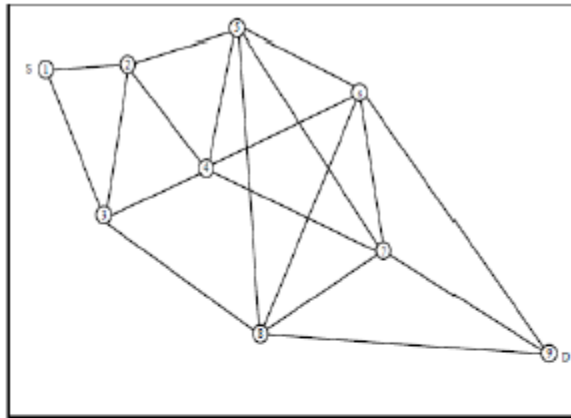
- Técnicas de redução do espaço de vértices da rede serão elaboradas para permitir a aplicação do algoritmo para redes de larga escala

# Roteamento em Grafos

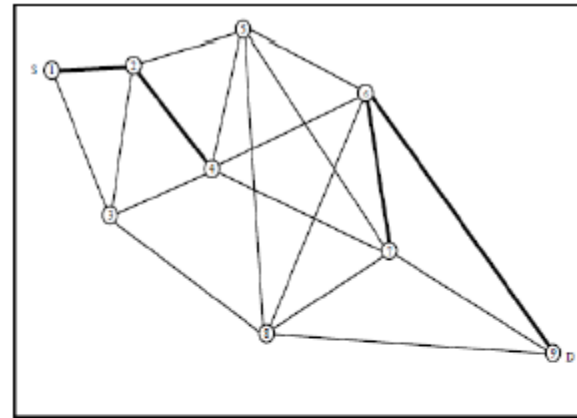
- Método de Redução do Espaço de Vértices (*Screening*)
  - Encontre a aresta de mínimo custo,  $L_{min}$ , e a de maior custo,  $L_{max}$
  - Começando de um valor de *threshold*  $L_{scr} = L_{min}$ , eliminam-se todas as arestas cujo custo exceda o valor  $L_{scr}$
  - Se não existir ainda um caminho da fonte ao destino, incrementa-se o *threshold* até tal caminho existir

$$L_{scr}(n) = L_{min} + (n - 1)\Delta L, \quad n \geq 1$$
$$\Delta L = \frac{1}{n} (L_{max} - L_{min})$$

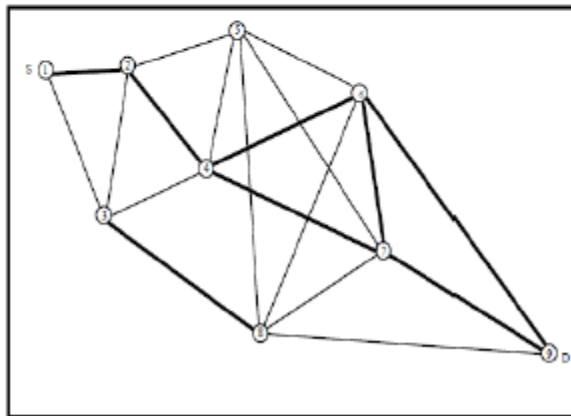
# Procedimento de *Screening*



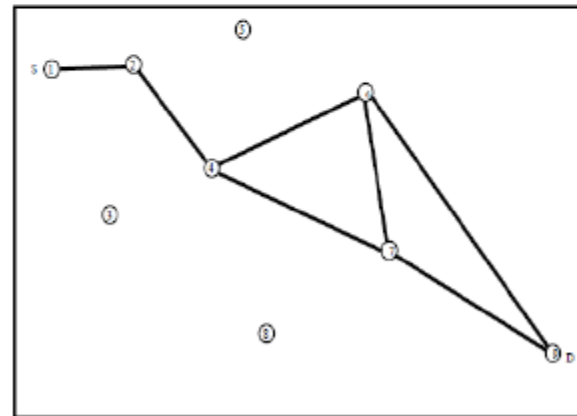
(a) Configuração Inicial



(b) Após o primeiro *screening*



(c) Após o segundo *screening*



(d) Após a aplicação da DFS

# Roteamento em Grafos

- Considere as seguintes variáveis
- Custo:
  - $L_{xi} = \begin{cases} c_{xi}, & \text{se uma conexão entre o vértice } x \text{ e } i \text{ existe} \\ \infty, & \text{caso contrário} \end{cases}$
- Caminho escolhido pela solução:
  - $V_{xi} = \begin{cases} 1, & \text{se a aresta do vértice } x \text{ para o } i \text{ existe na solução final} \\ 0, & \text{caso contrário} \end{cases}$
- Indicador estrutural do grafo :
  - $\gamma_{xi} = \begin{cases} 1, & \text{se NÃO existe uma conexão entre o vértice } x \text{ e } i \\ 0, & \text{caso contrário} \end{cases}$

# Problema de Otimização

$$\begin{aligned} \text{Minimize} \quad & \sum_{x=0}^{N-1} \sum_{x=0}^{N-1} L_{xi} V_{xi} \\ \text{Sujeito a} \quad & \left( \begin{array}{cc} \sum_{\substack{i=0, \\ i \neq x}}^{N-1} V_{xi} & - \sum_{\substack{j=0, \\ j \neq x}}^{N-1} V_{jx} \end{array} \right) \\ & = \begin{cases} 1 & \text{se } x = s, \\ -1 & \text{se } x = d_{xm}, x = \{1, \dots, N\} \text{ e } m = \{1, \dots, D\}, \\ 0 & \text{caso contrário.} \end{cases} \end{aligned}$$



# Função de Energia

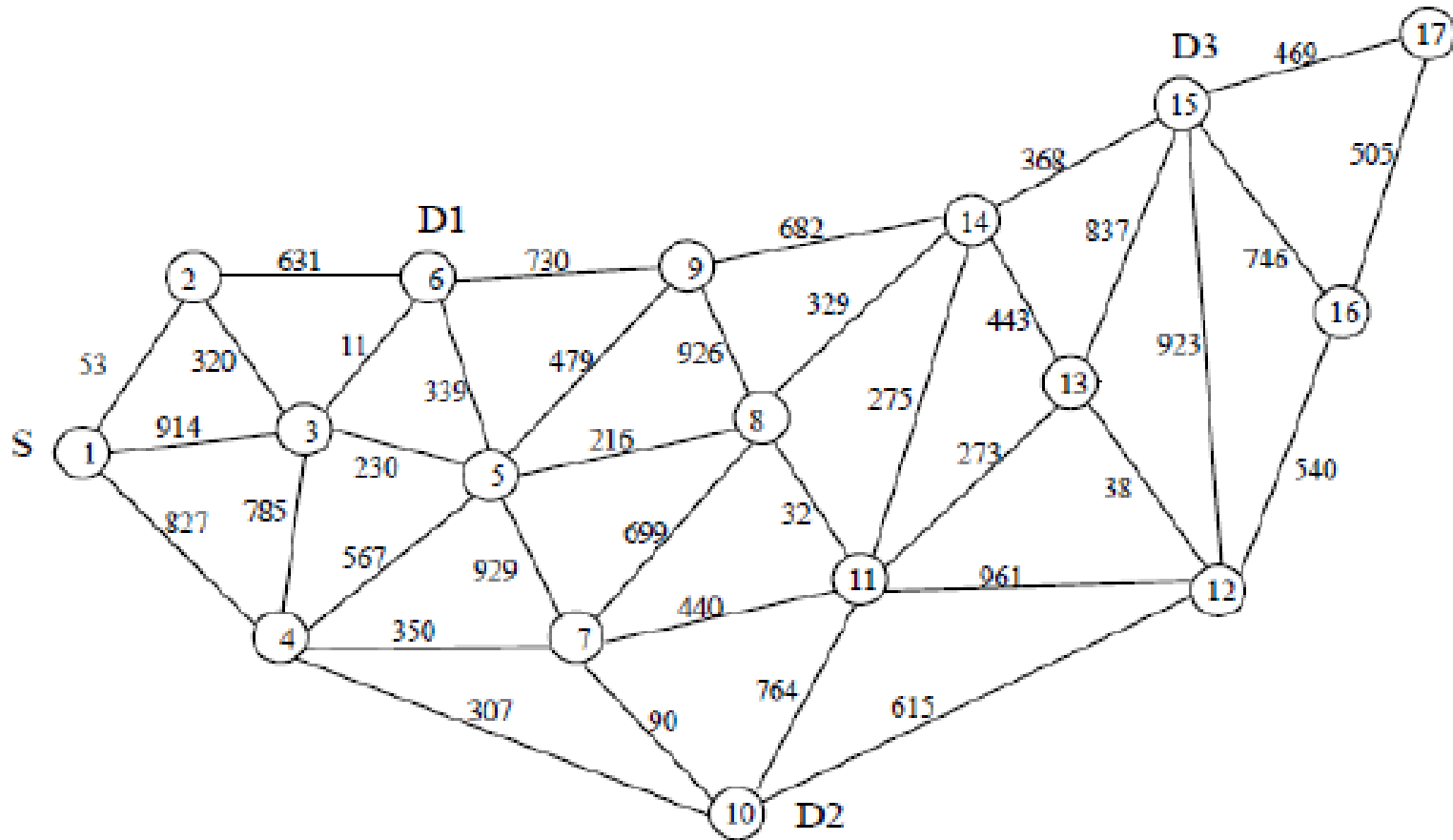
$$\begin{aligned}
 E = & \frac{A}{2} \sum_{x=0}^{N-1} \sum_{\substack{i=0, \\ i \neq x, \\ (x,i) \neq (d,s)}}^{N-1} L_{xi} V_{xi} \\
 & + \frac{B}{2} \sum_{x=0}^{N-1} \sum_{\substack{i=0, \\ i \neq x, \\ (x,i) \neq (d,s)}}^{N-1} \gamma_{xi} V_{xi} \\
 & + \frac{C}{2} \sum_{x=0}^{N-1} \left\{ \sum_{\substack{i=0, \\ i \neq x}}^{N-1} V_{xi} - \sum_{\substack{i=0, \\ i \neq x}}^{N-1} V_{ix} - \phi_x \right\}^2 \\
 & + \frac{D}{2} \sum_{i=0}^{N-1} \sum_{\substack{x=0, \\ x \neq i}}^{N-1} V_{xi}^2
 \end{aligned}$$

$$\phi_x = \begin{cases} 1 & x = s, \\ -1 & x = d_m, \text{ para todo } x = \{1, \dots, N\} \text{ e } m = \{1, \dots, D\}, \\ 0 & \text{caso contrário.} \end{cases}$$

# Roteamento em Grafos

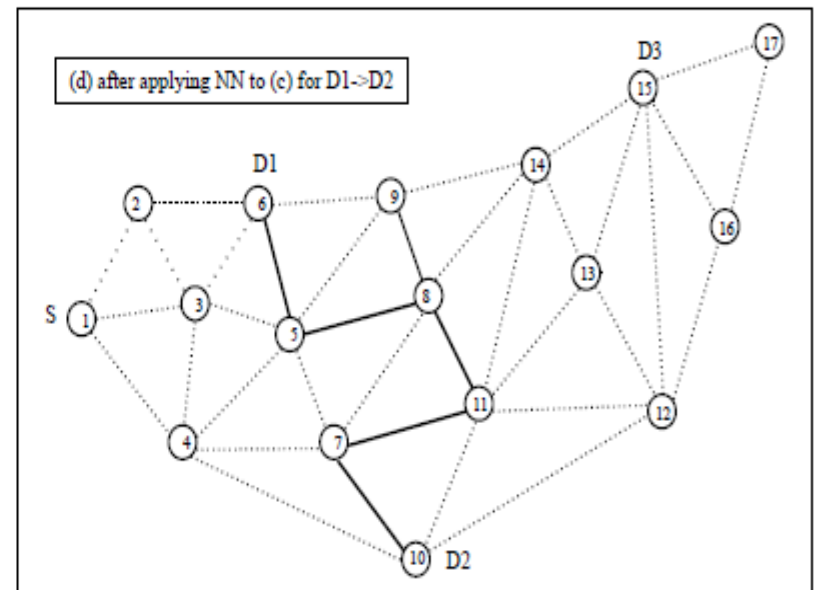
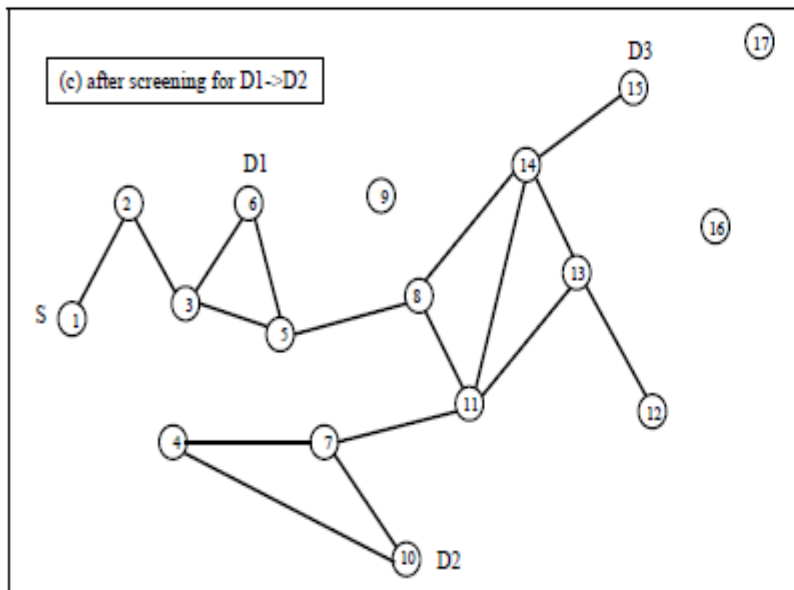
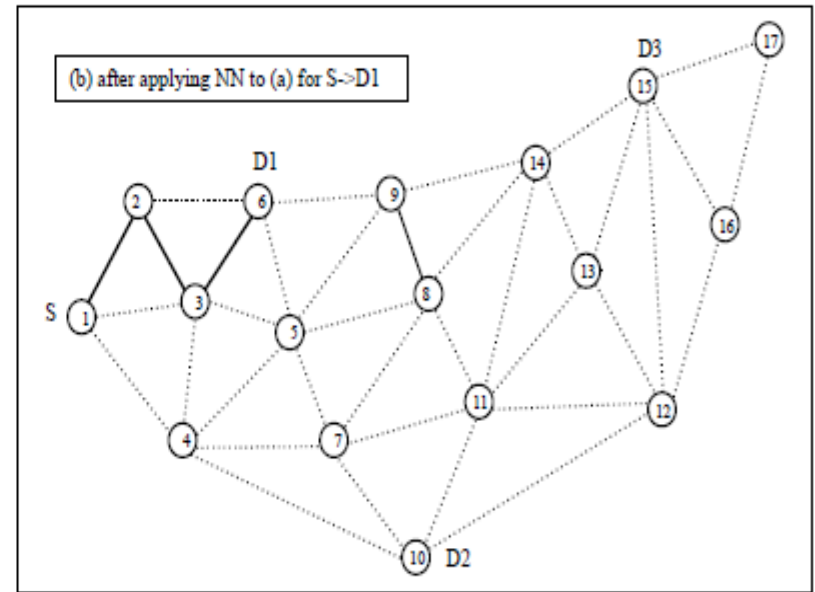
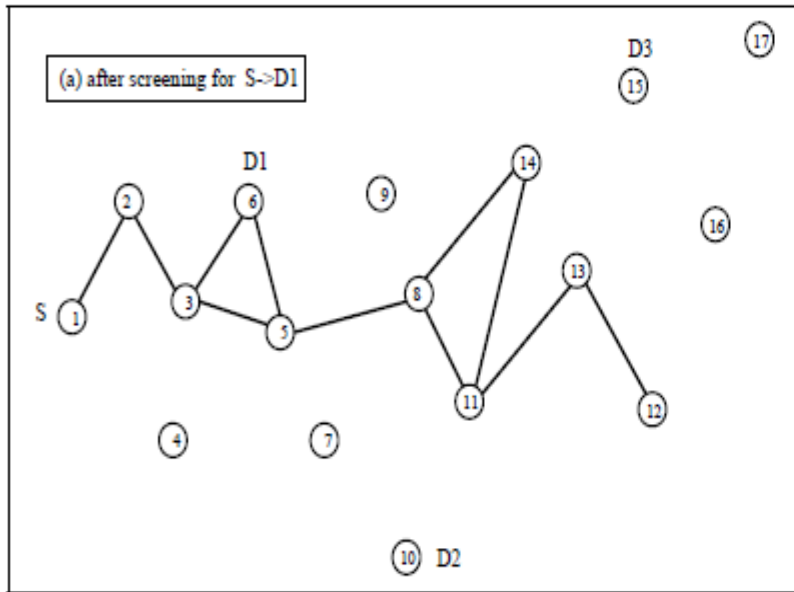
- A minimização da função de energia anterior leva o processo a um estado atrator, o qual é constituído por um mínimo local ou global de energia
- Quando o processo estiver neste estado estável, o mesmo irá corresponder à rota de custo possivelmente ótimo entre a fonte e os destinos

# Simulações Computacionais

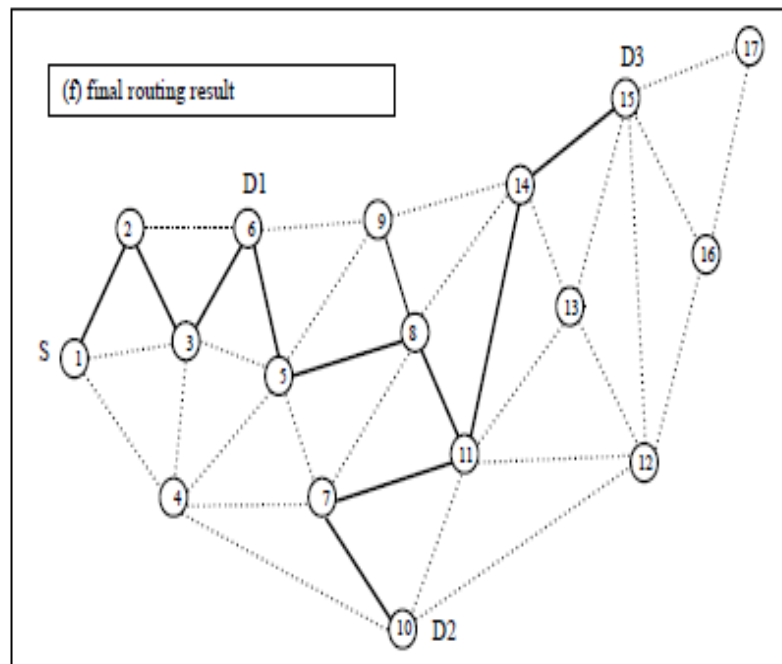
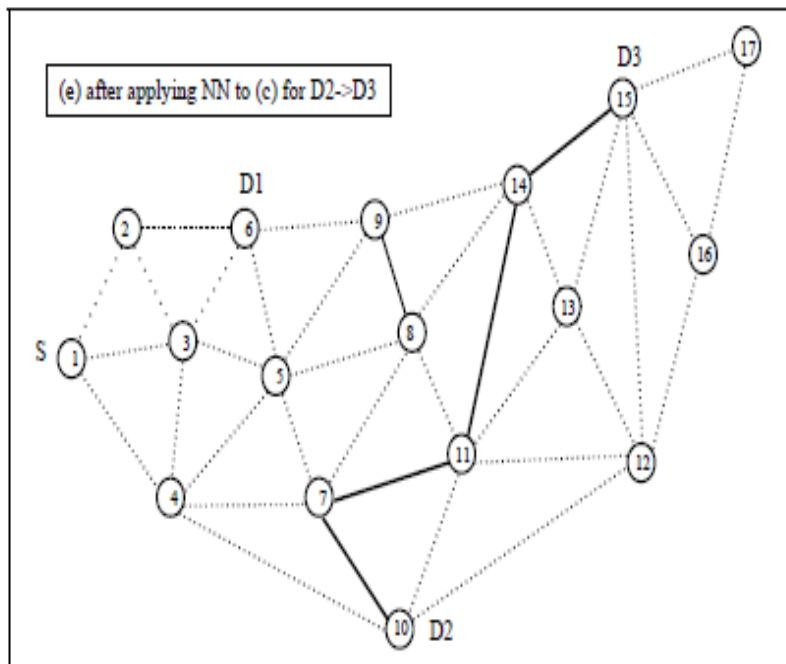


- $L_{min} = 11$
- $L_{max} = 961$

# Simulações Computacionais



# Simulações Computacionais



# Considerações Finais

- As redes neurais apresentam vantagens na resolução de problemas complexos em relação aos processos computacionais convencionais
- A rede neural de Hopfield é um exemplo vivo disso, pois permite a resolução de vários problemas combinatoriais NP-completos com ótimas referências na literatura
- Ademais, o modelo de Hopfield é passível de aplicações em vários ramos da ciência
  - Este trabalho é uma tentativa de provar exatamente isso, uma vez que aplica o modelo de Hopfield no campo de roteamento e clusterização de pixels