

CC7711

Inteligência Artificial e Robótica

Prof. Dr. Flavio Tonidandel



The background of the slide features a complex network diagram. It consists of numerous circular nodes of varying sizes, connected by thin, light blue lines. The nodes are distributed across the entire frame, with a higher density on the left side where they form a more solid-looking blue area. The overall effect is one of a global or interconnected system.

Inteligência Coletiva

CC7711 - Inteligência Artificial e Robótica

Inteligência Coletiva(*Swarm Intelligence*)

Swarm Intelligence: “Qualquer tentativa de se desenvolver algoritmos ou dispositivos distribuídos de resolução de problemas inspirados no comportamento coletivo de colônias de insetos sociais ou outras sociedades animais” *[Bonabeau, Dorigo, and Theraulaz, 1999]*

- **Colônias Sociais de Insetos**

- Podem desenvolver sistemas complexos, inteligentes,,flexíveis e confiáveis
- Cada indivíduo de uma sociedade
 - Segue regras simples, com comunicação local (trilhas de odores, sons, toques), de baixa demanda computacional
- Estrutura Global do Sistema
 - Emerge a partir dos comportamentos individuais gerando um comportamento global inteligente

Colônias de Insetos

- **Colônias Sociais de Insetos (Formigas, Cupins, Abelhas, etc.)**
 - Comportamentos complexos
 - Flexíveis e robustos
 - Comportamento do Inseto depende do de outros iguais a ele
- **Como surge esse comportamento inteligente de grupo ?**
 - **Auto-organização**
 - **Interações Locais**
 - Inseto a inseto
 - Inseto com o ambiente
 - **Especialização**
 - Cada inseto tem o seu objetivo/tarefa a cumprir

An abstract blue geometric pattern consisting of interconnected lines and dots, resembling a network or molecular structure, is located on the left side of the slide.

Colônia de Formigas

(Ant Colony Optimization – ACO)

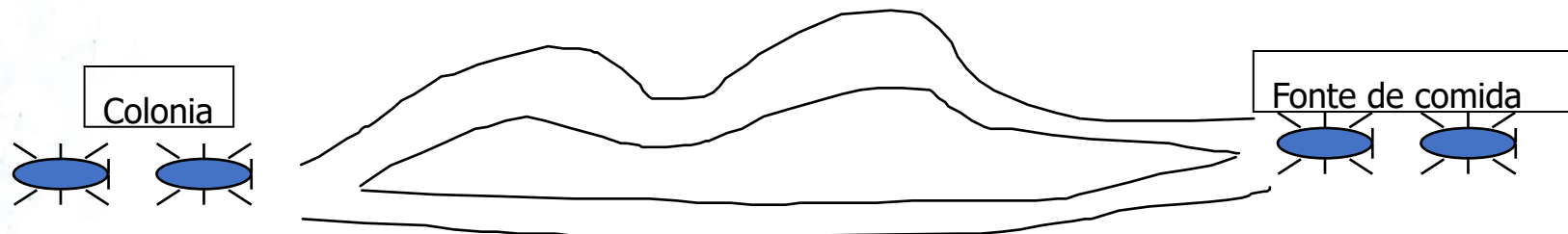
Colônia de Formigas

- Formigas não possuem comportamentos individuais sofisticados, porém realizam tarefas complexas coletivamente
- Formigas possuem uma sofisticada sinergia baseada em sinais:
 - comunicação usando feromônios
 - Trilhas são definidas de modo que outras formigas podem seguir



Trilhas de Feromônios

- Espécies liberam trilhas de feromônios quando estão caminhando:
 - a partir da colônia
 - Para a colônia
 - Ambas as direções
- Feromônios evaporam.
- Feromônios se acumulam a medida que multiplas formigas usam a trilha



Colônia de Formigas: Problema de Rotas

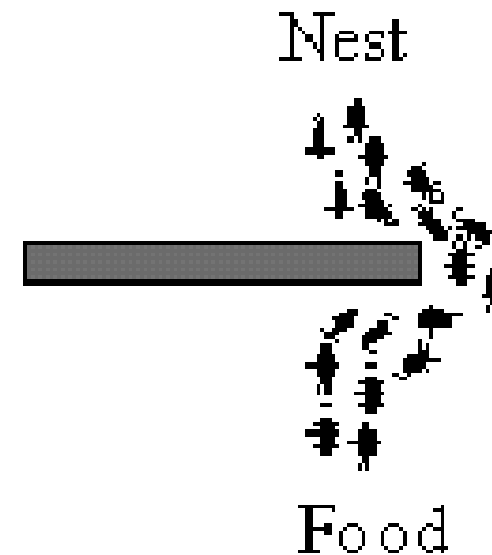
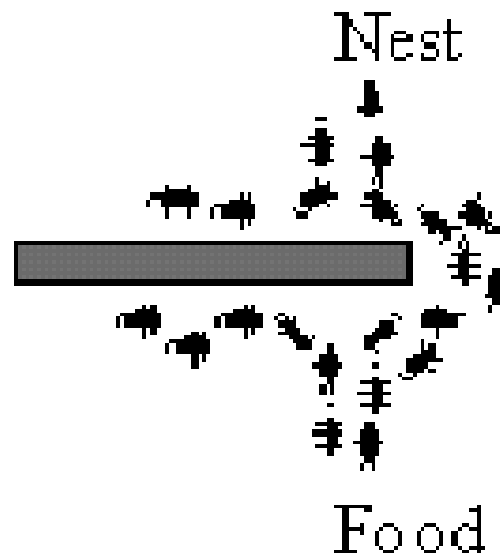
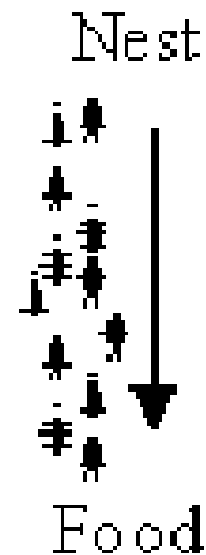
- Idéia
 - Formigas soltam diferentes feromônios usados para computar o menor caminho entre a fonte e o destino
 - Sistema adaptativo e flexível contra falhas ou congestionamento da rede
 - Usa somente conhecimento local para as rotas e assim evita verificar todos os nós da rede



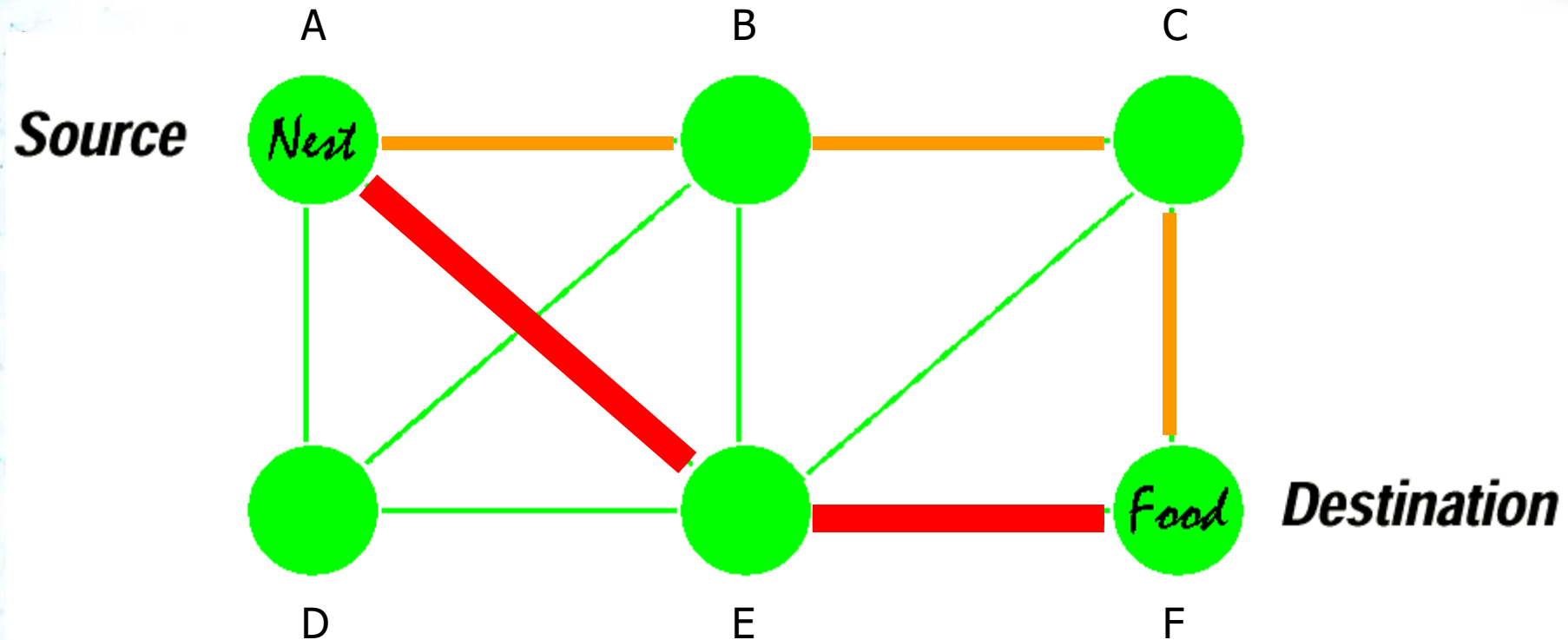
Por que roteamento ?

Roteamento convencional depende de:

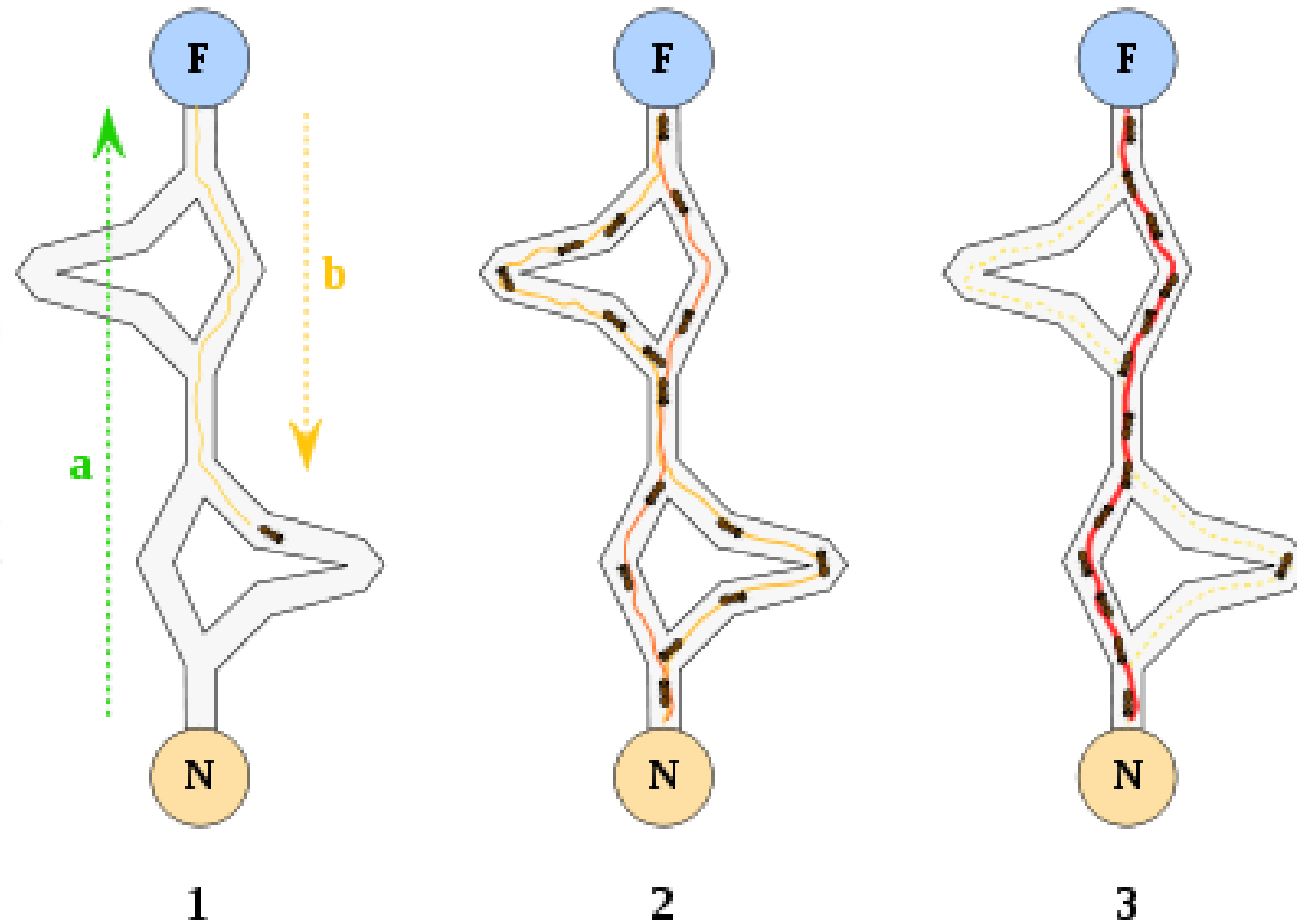
- Situação global de todos os nós
- controle centralizado
- Menor caminho fixado (algoritmo de Dijkstra);
- Habilidade limitada para lidar com congestionamento ou falhas.



Colônia de Formigas



Colônia de Formigas



Como escolher o caminho a seguir ?

Probabilidade da formiga ir de i para j:

$$p^{ij}(t) = [T_{ij}(t)]^{\alpha} [\eta]^{\beta} / \mathbb{N}$$

$$\mathbb{N} = \sum_k [T_k(t)]^{\alpha} [\eta]^{\beta}$$

T_{ij} = Quantidade de feromônios no trajeto entre i e j no instante t

α , β são parâmetros de controle ajustáveis que determinam a sensibilidade do algoritmo quanto ao feromônio e preferência

η é uma função constante que permite atribuir preferência de caminhos, geralmente se usa 1



Atualização de Feromônios

- Atualização de feromônios :

$$\Delta\tau_{ij}^k = Q / L^k(t) \quad \text{if } (i, j) \in T^k(t) \text{ else } 0.$$



- T é o tour de i para j feito pela formiga k no tempo t, L é o tamanho desse tour e Q é uma heurística que determina a quantidade de feromonios.
- Evaporação de Feromônios em função do tempo

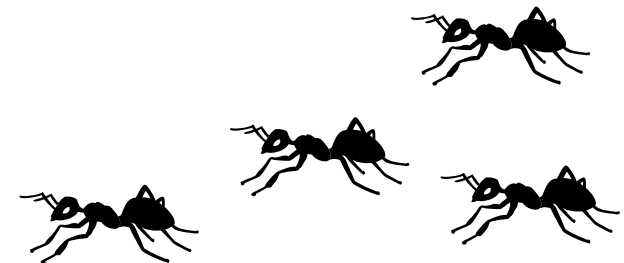
$$\tau_{ij}(t) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

- ρ = taxa de evaporação

Colônias de Formigas

Alguns resultados práticos

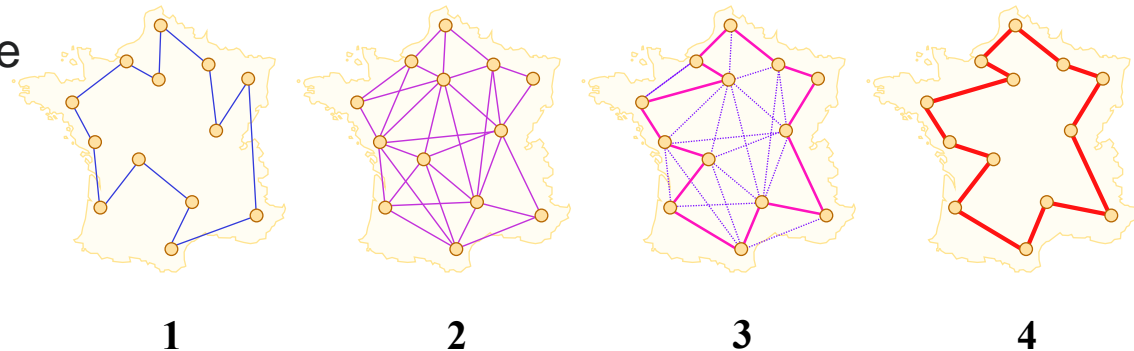
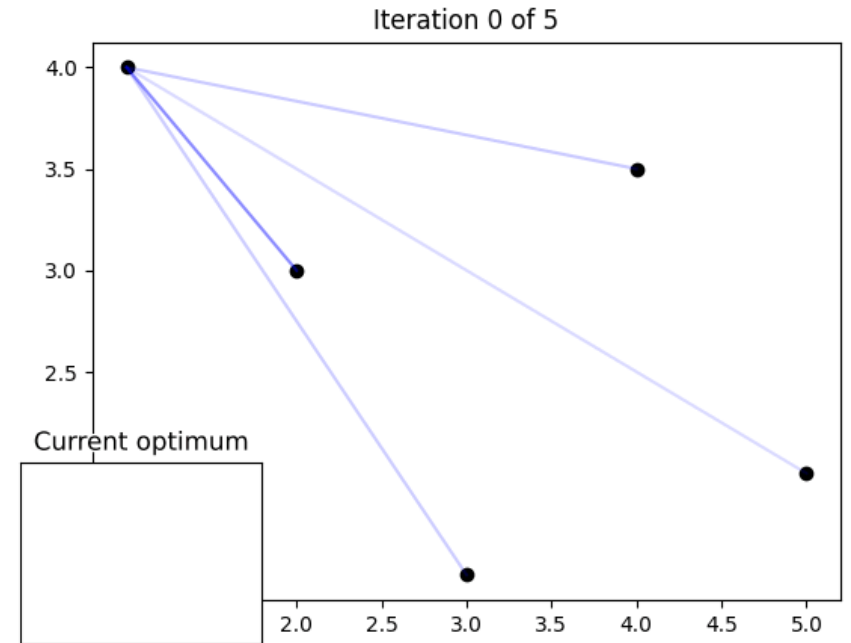
- Algoritmos de Otimização por Colônias de Formigas
 - Extremamente bem-sucedidos na otimização de uma grande variedade de problemas combinatoriais de otimização de grande dificuldade
 - Algumas das melhores soluções para alguns problemas conhecidos (Caxeiro Viajante)
- Controle de Sistemas de Telecomunicações Baseados em Formigas
 - Soluções extremamente flexíveis
 - Melhor solução para alguns problemas



ACO aplicada no Caixeiro Viajante

O objetivo é encontrar a viagem de ida e volta mais curta para ligar todas as cidades sem repetição. O algoritmo geral é relativamente simples e baseado em um conjunto de formigas, cada uma fazendo uma possível viagem de ida e volta pelas cidades. Em cada etapa, a formiga escolhe se muda de uma cidade para outra de acordo com algumas regras:

1. Deve visitar cada cidade exatamente uma vez;
2. Uma cidade distante tem menos chance de ser escolhida (a visibilidade);
3. Quanto mais intensa for a trilha de feromônio disposta em uma borda entre duas cidades, maior a probabilidade de que essa borda seja escolhida;
4. Tendo completado sua jornada, a formiga deposita feromônios em todas as arestas que percorreu;
5. Após cada iteração, rastros de feromônios evaporam



Outras Aplicações de CF

Problemas de roteamento de veículos:

- roteamento de veículos de vários depósitos**
- roteamento de veículo de entrega dividida**
- roteamento de veículos com coleta e entrega**
- roteamento de veículos com janelas de tempo**
- roteamento de veículos com vários trabalhadores em serviço**

Problema de conjunto

- Problema de conjunto de cobertura (set cover)**
- Problema de partição (partition problem)**
- Problema de partição de árvore de grafos com restrição de peso**
- Problema de mochila múltipla (Multiple Knapsack problem)**
- Problema máximo de conjunto independente**

Problemas de Escalonamento:

- Escalonamento de oficinas de trabalho**
- Escalonamento de loja aberta**
- Escalonamento de agendamento de projetos com restrição de recursos**
- Escalonamento de sequência de montagem**

Outros problemas:

- Projeto Físico de Nanoeletrônica**
- Previsão de falência**
- Roteamento de rede orientado à conexão**
- Mineração de dados**
- Concepção de redes de energia e eletricidade**
- Projeto de peptídeos inibitórios para interações proteicas**
- Projeto de circuitos eletrônicos de potência**
- Enovelamento de proteínas**

An abstract blue geometric pattern consisting of interconnected lines and dots, resembling a network or molecular structure, is located on the left side of the slide.

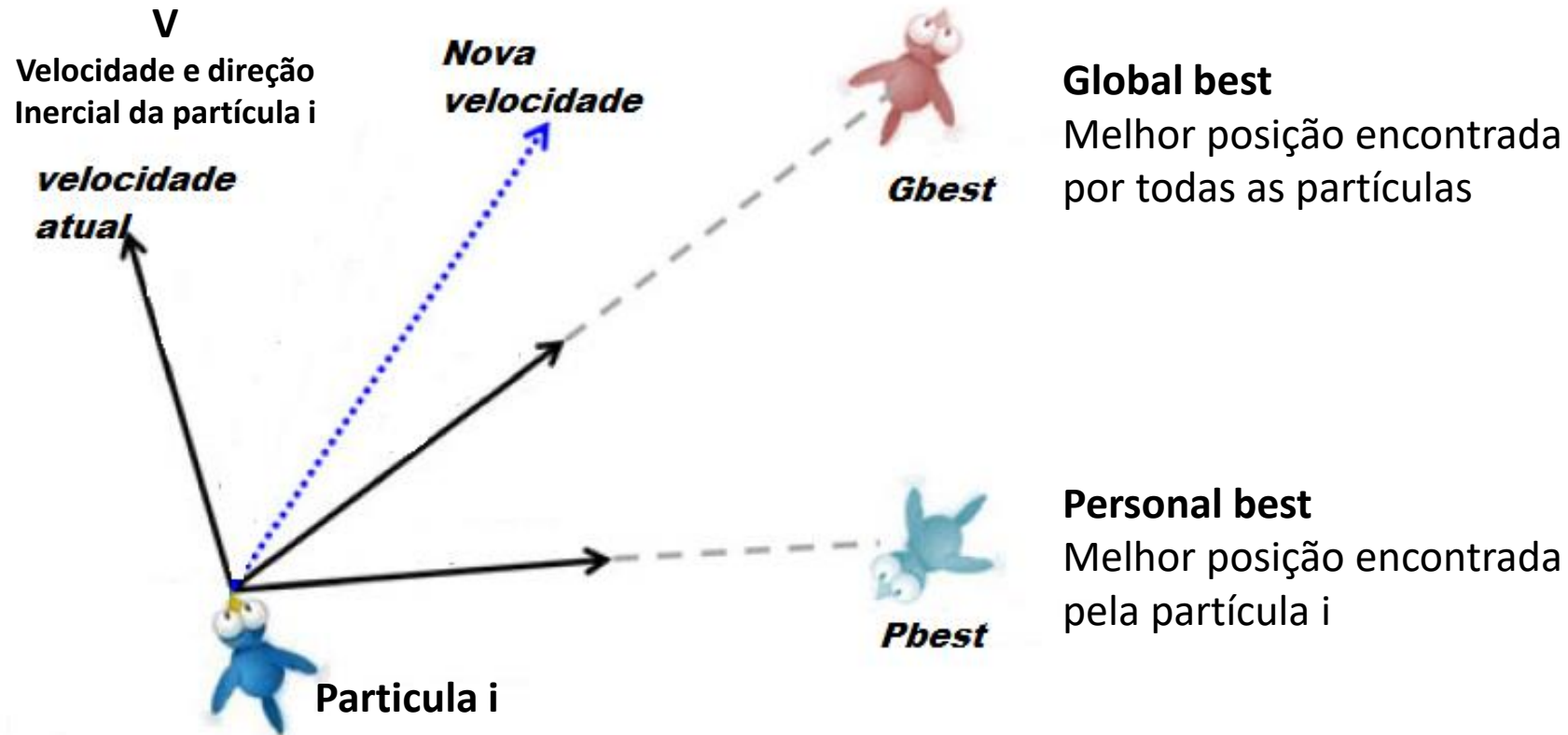
PSO – Particle Swarm Optimization

Particle Swarm Optimization (PSO)

Simple Algoritmo

- Método computacional iterativo para minimização de função
- Se esta função representar um problema, ela otimiza esta solução
- A função que deve ser minimizada é uma função que mede a distância de uma ou mais partículas à uma possível solução de melhor qualidade.
- Criado por Russell C. Eberhart and James Kennedy em 1995
- Inspirado no comportamento social de acasalamento das aves e cardumes de peixes

Particle Swarm Optimization (PSO)



Particle Swarm Optimization (PSO)

Para cada partícula p_i , sua nova velocidade vetorial v_i será:

$$v_i = wv_i + c_1.rand().(pbest_i - x_i) + c_2.rand().(gbest_i - x_i)$$

$$x_i = x_i + v_i \text{ (atualiza a posição da partícula)}$$

onde:

$pbest_i$ é a melhor posição em que a partícula a_i já esteve

$gbest_i$ é a melhor posição em que algum vizinho de a_i já esteve.

w é o peso de inércia ; c_1 é o peso dado a $pbest$ e c_2 é o peso dado a $gbest$

x_i é a posição atual da partícula i

Rand() são números aleatórios entre 0 e 1, que permitem explorar um pouco regiões ao redor.

Particle Swarm Optimization (PSO)

$$v_i = \boxed{wv_i} + \boxed{c_1.rand().(pbest_i - x_i)} + \boxed{c_2.rand().(gbest_i - x_i)}$$

Termo de Inércia

Força a partícula
a manter sua
velocidade e
direção atuais

Termo Cognitivo

Força a partícula
a voltar para uma posição
anterior melhor

Tendência Conservativa

Termo Aprendizado Social

Força a partícula
a aprender e seguir a melhor
posição já alcançada por uma
partícula

Influência Social

Particle Swarm Optimization (PSO)

Parâmetros a serem definidos:

1 – Número de partículas

Estima-se que entre 10 e 50 seja um bom número de partículas iniciais. Elas devem ser bem espalhadas para evitar mínimos locais.

2 – Constantes C1, C2 e W

Usualmente $C1+C2 = 4$.

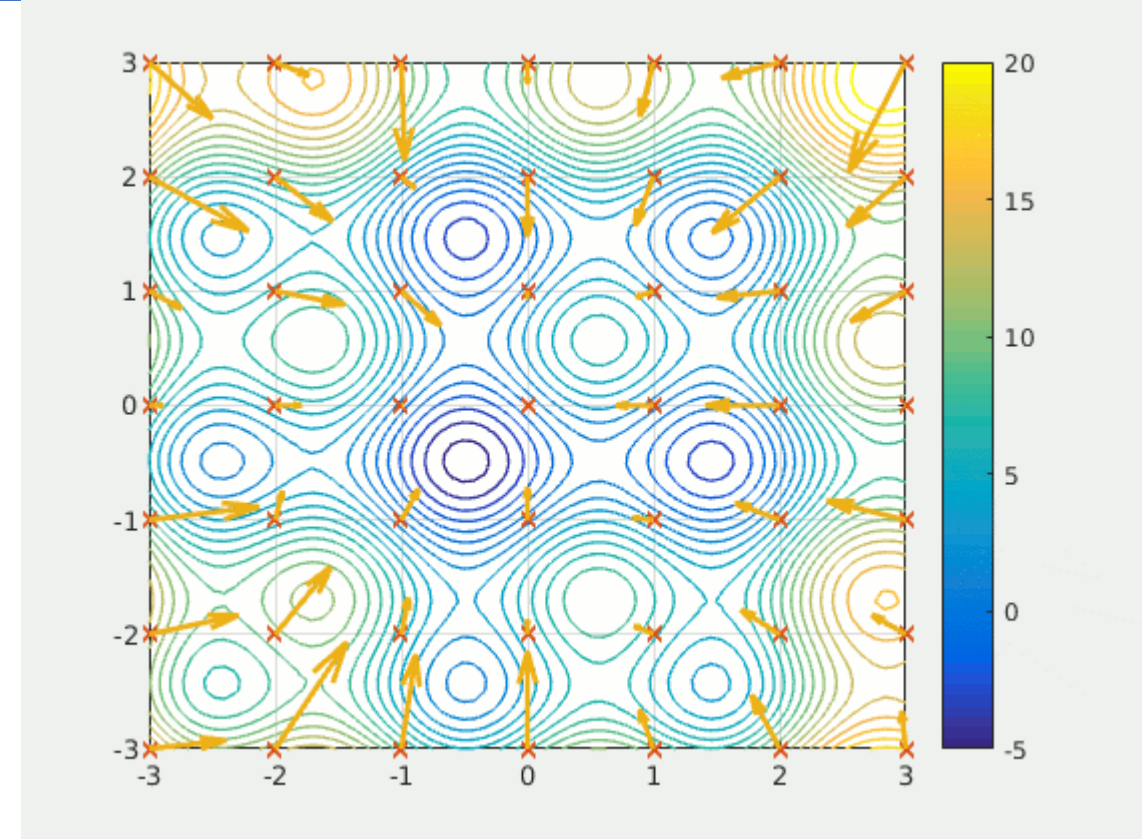
W foi originalmente definido como 1, mas tem-se explorado valores diferentes para W

3 – Velocidade máxima da partícula

*Vmax – baixa velocidade = demora em convergir
alta velocidade = muita instabilidade*

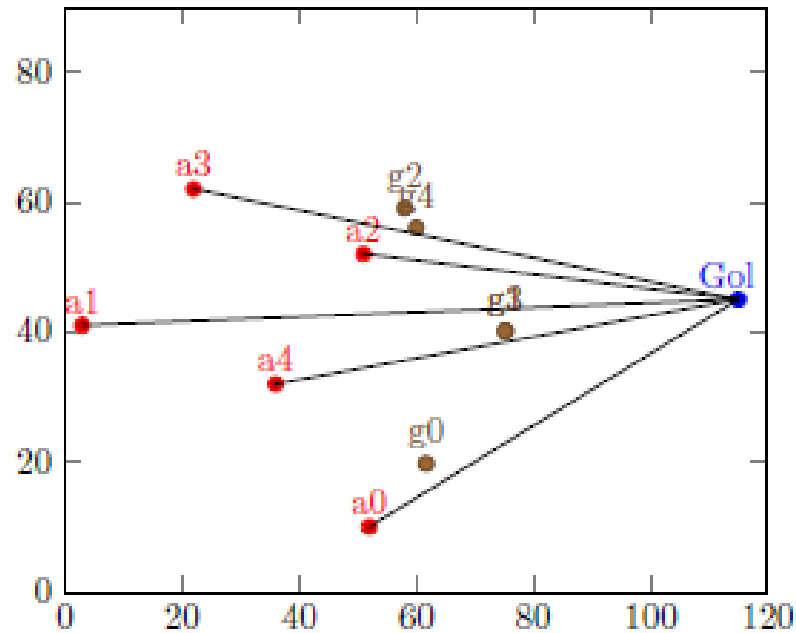
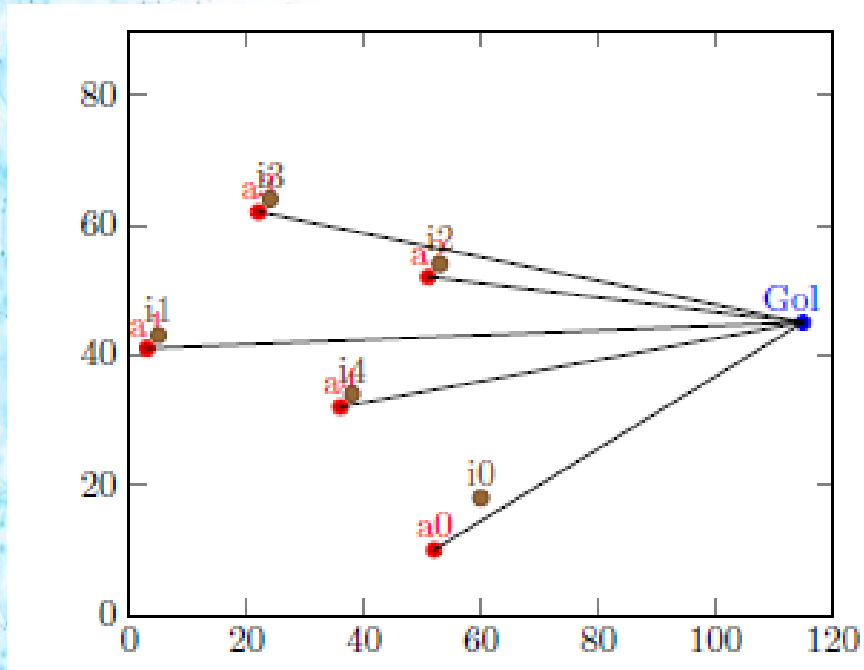
4 – Critério de parada

*Solução alcançada; velocidade baixa das partículas;
pouca movimentação; etc*



Aplicações de Particle Swarm Optimization (PSO)

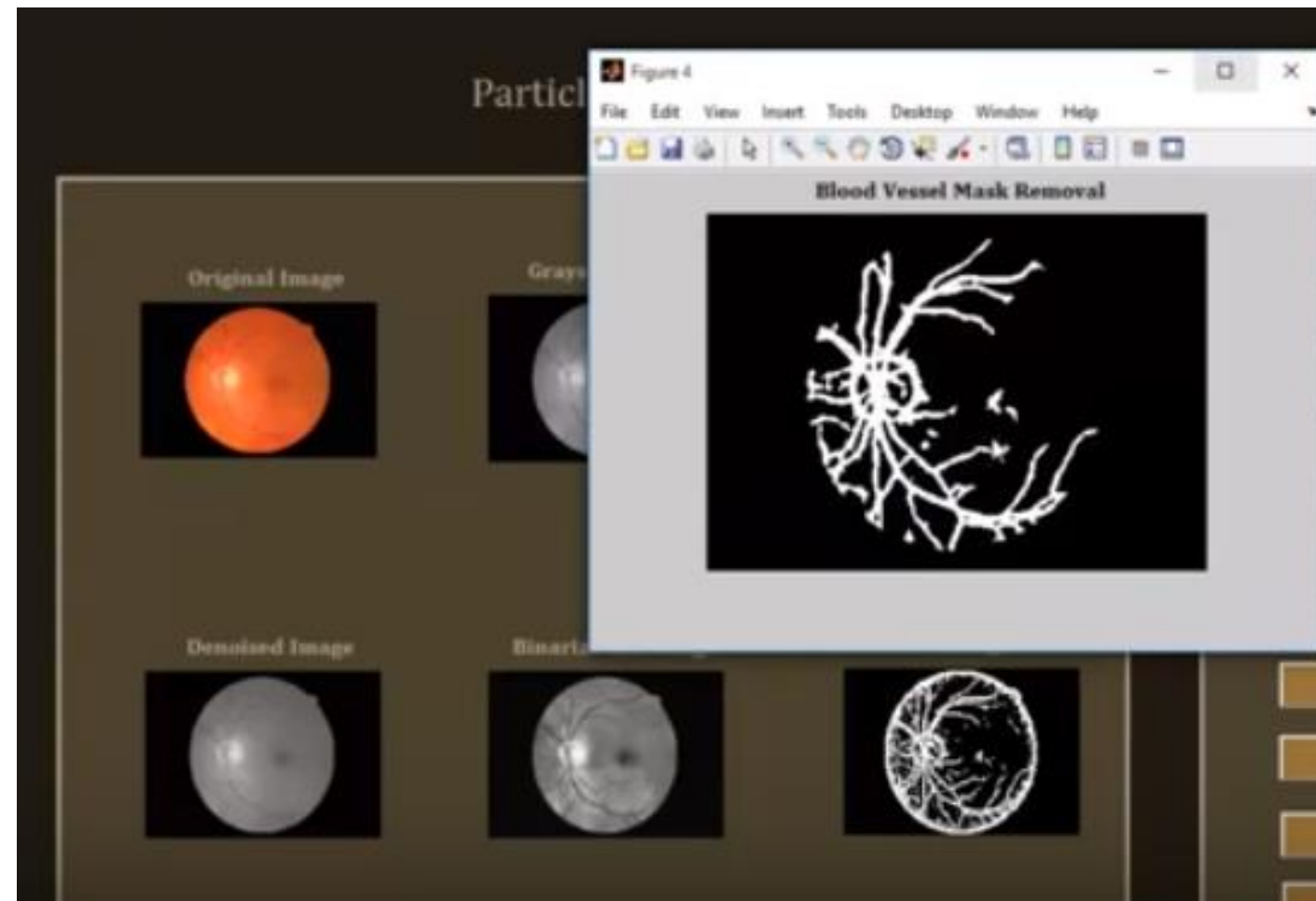
Posicionamento de robôs em um jogo de futebol:



Cada partícula é um cenário que envolve a posição de 5 robôs

Aplicações de Particle Swarm Optimization (PSO)

Segmentação de imagem:



Robótica baseada em *Swarms*

Múltiplos Robôs

- **Vantagens significativas em relação a robôs individuais**
 - Sensoreamento e ação em múltiplos lugares
 - Reconfiguração dependente da tarefa
 - Divisão de trabalho eficiente
 - Robustez devido à redundância e adaptabilidade
 - Auto-organização de comportamento



Bibliografia desta aula

Para aprofundamento da Aula de Inteligência Coletiva (Colônia Formiga):

Recomendação de Leitura

- M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
- J. Kennedy and R. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, Inc, San Francisco, CA, 2001
- J. Kennedy and R. Eberhart. Particle Swarm Optimization, in: Proc. IEEE Int'l. Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV:1942-1948.
- C.A. Coello Coello, G. Toscano, M.S. Lechuga. Handling Multiple Objectives with Particle Swarm Optimization. *IEEE Transactions on Evol. Computation*, Vol. 8, No. 3, June 2004.