

# Inteligência de Enxame\*

1.	Introdução .....	2
2.	Algumas Ideias sobre Insetos Sociais.....	5
2.1.	Curiosidades sobre as formigas.....	9
3.	Colônia de Formigas .....	10
3.1.	Coleta de Alimento pelas Formigas .....	12
3.2.	Otimização por Colônia de Formigas .....	16
3.3.	Uma Simulação de Vida Artificial .....	18
3.4.	Conceitos Básicos sobre Teoria de Grafos .....	19
3.5.	Algoritmo Simples de Otimização por Colônia de Formigas .....	21
3.6.	Algoritmo Genérico de Otimização por Colônia de Formigas .....	24
3.7.	Exemplo de Aplicação.....	26
3.8.	Clusterização de Corpos e Organização de Larvas.....	32
3.9.	Clusterização por Colônia de Formigas.....	34
3.10.	Algoritmo Simples de Clusterização (ACA).....	35
3.11.	Exemplos de Aplicação .....	42
4.	Robótica de Enxame .....	49
4.1.	Coleta de Alimento pelas Formigas .....	54
4.2.	Clusterização de Objetos .....	58
4.3.	Transporte Coletivo de Presas .....	63
5.	Adaptação Social do Conhecimento .....	73
5.1.	Algoritmo de Otimização por Enxame de Partículas .....	75
5.2.	Escopo de Aplicação .....	82
5.3.	De Sistemas Sociais a Enxames de Partículas.....	83
6.	Referências bibliográficas .....	84

---

\* Material baseado nas notas de aula do Prof. Leandro Nunes de Castro. Reprodução de conteúdo autorizada pelo autor.

# 1. Introdução

- Várias espécies se beneficiam da sociabilidade:
  - A vida em grupos sociais aumenta a probabilidade de acasalamento, facilita a caça e coleta de alimentos, reduz a probabilidade de ataque por predadores, permite a divisão de trabalho, etc.
- Comportamentos sociais também inspiraram o desenvolvimento de diversas ferramentas computacionais para a solução de problemas e estratégias de coordenação e controle de robôs.
- O termo *swarm intelligence* foi proposto no fim dos anos de 1980, quando se referia a sistemas robóticos compostos por uma coleção de agentes simples em um ambiente interagindo de acordo com regras locais.
- Algumas definições de *swarm intelligence*:
  - O termo “enxame” (ou coletivo) é utilizado de forma genérica para se referir a qualquer coleção estruturada de agentes capazes de interagir. O exemplo

clássico de um enxame é um enxame de abelhas. Entretanto, a metáfora de um enxame pode ser estendida a outros sistemas com uma arquitetura similar. Uma colônia de formigas pode ser vista como um enxame, onde os agentes são formigas; uma revoada de pássaros é um enxame, onde os agentes são pássaros; um engarrafamento é um enxame, onde os agentes são carros; uma multidão é um enxame de pessoas, um sistema imunológico é um enxame de células e moléculas, e uma economia é um enxame de agentes econômicos. Embora a noção de enxame sugira um aspecto de movimento coletivo no espaço, como em um ‘enxame de pássaros’, estamos interessados em todos os tipos de comportamentos coletivos, não apenas movimento espacial. (FAQ – *Santa Fe*).

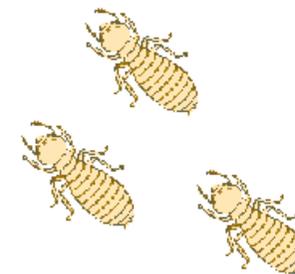
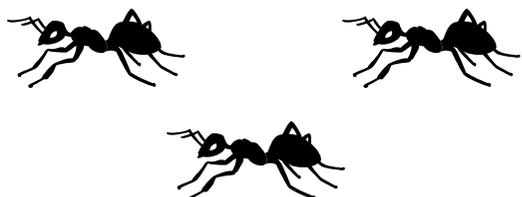
- A inteligência de enxame inclui qualquer tentativa de projetar algoritmos ou dispositivos distribuídos de solução de problemas inspirados no comportamento coletivo de insetos sociais e outras sociedades animais (BONABEAU *et al.*, 1999).

- A inteligência coletiva é uma propriedade de sistemas compostos por agentes não (ou pouco) inteligentes e com capacidade individual limitada, capazes de apresentar comportamentos coletivos inteligentes (WHITE & PAGUREK, 1998).
- Algumas propriedades da inteligência coletiva:
  - *Proximidade*: os agentes devem ser capazes de interagir;
  - *Qualidade*: os agentes devem ser capazes de avaliar seus comportamentos;
  - *Diversidade*: permite ao sistema reagir a situações inesperadas;
  - *Estabilidade*: nem todas as variações ambientais devem afetar o comportamento de um agente;
  - *Adaptabilidade*: capacidade de se adequar a variações ambientais.
- Sendo assim, um *sistema de enxame* é aquele composto por um conjunto de agentes capazes de interagir entre si e com o meio ambiente. A *inteligência de enxame* é uma propriedade emergente de um sistema coletivo que resulta de seus princípios de proximidade, qualidade, diversidade, estabilidade e adaptabilidade.

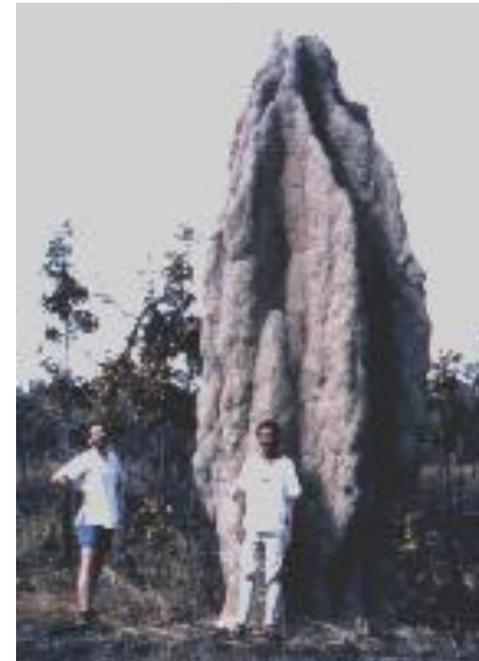
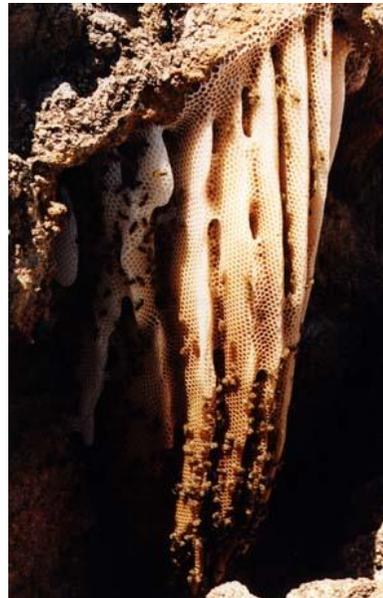
- Duas principais linhas de pesquisa podem ser observadas em inteligência de enxame:
  - Trabalhos inspirados por comportamentos sociais de insetos e outros animais;
  - Trabalhos inspirados na habilidade das sociedades humanas em processar conhecimento.
- Embora existam diferenças entre essas abordagens, elas possuem a seguinte característica importante em comum:
  - População de indivíduos capazes de interagir entre si e com o ambiente.

## 2. Algumas Ideias sobre Insetos Sociais

- Insetos sociais são aqueles que vivem em comunidades ou *colônias*. Exemplos:
  - Formigas, abelhas, vespas e cupins.



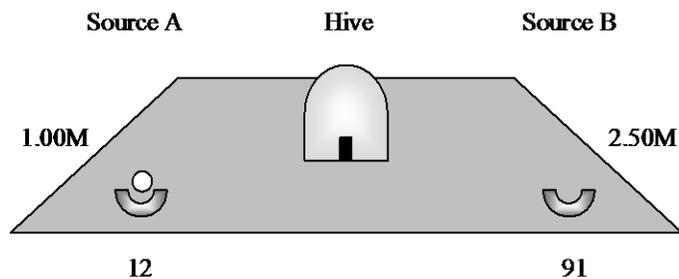
- Uma colônia pode ser definida como uma grande família de insetos (sem hierarquia, na maioria dos casos).
- Dentro de uma colônia existe uma sobreposição entre gerações de pais e filhos.
- Cada inseto parece ter sua própria agenda; mesmo assim, uma colônia parece extremamente bem organizada.
- A integração de todas as atividades individuais não requer supervisão, trata-se de um fenômeno auto-organizado (CAMAZINE *et al.*, 2001).
- Exemplos de ninhos:



- Formigas do tipo *leafcutter* cortam folhas de plantas e árvores para cultivar fungos. Formigas trabalhadoras buscam por alimento a grandes distâncias do ninho, criando literalmente caminhos de e para o ninho.
- Formigas do tipo *weaver* formam correntes com seus próprios corpos permitindo que elas atravessem grandes buracos e carreguem alimento para o ninho.
- Durante sua fase de movimentação e busca por alimento, as formigas do tipo *army* organizam frentes de batalha impressionantes.



- As abelhas constroem uma série de pentes paralelos formando correntes que induzem um aumento local de temperatura. Desta forma, fica mais fácil moldar a colmeia.
- As fontes de alimento são exploradas de acordo com sua qualidade e distância do ninho.



- Exemplos de problemas resolvidos por insetos sociais:
  - Encontrar alimento, construir ou aumentar o ninho, dividir a mão de obra, alimentar a colônia, responder a desafios externos (clima, predadores, etc.), soar alarmes, encontrar um local apropriado para construir o ninho, etc.

## 2.1. Curiosidades sobre as formigas

- As formigas podem levantar até 20 vezes seu próprio peso.
- O cérebro de uma formiga possui aproximadamente  $2,5 \times 10^5$  neurônios, enquanto o cérebro humano possui aproximadamente  $1,0 \times 10^{10}$  neurônios.
  - Portanto, uma colônia de 40.000 formigas possui o mesmo número de neurônios que um cérebro humano.
- Uma formiga vive de 45 a 60 dias.
- As formigas utilizam suas antenas para tocar e sentir cheiro.
- As formigas possuem olhos compostos e seu abdômen possui dois estômagos. O primeiro armazena alimento para a própria formiga e o segundo armazena alimento a ser compartilhado.
- Existem mais de 20.000 espécies conhecidas de formigas, sendo aprox. 2.000 espécies no Brasil.

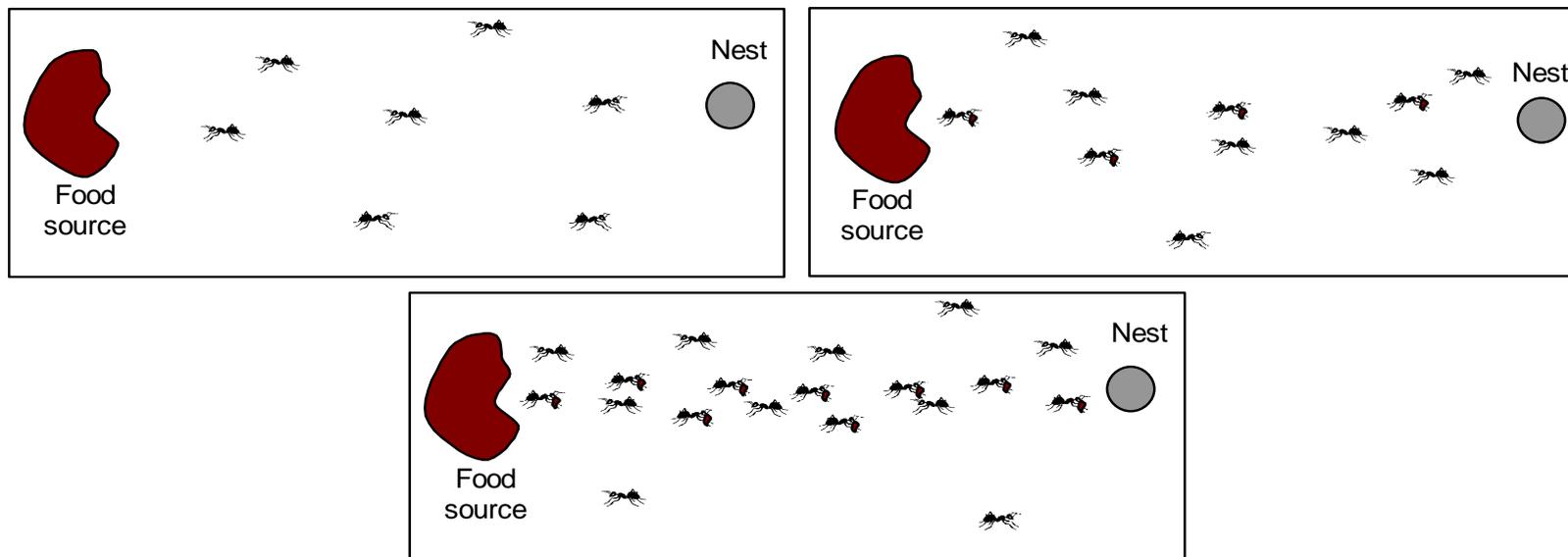
### 3. Colônia de Formigas

- As formigas são os insetos sociais mais amplamente estudados. Exemplos da popularidade das formigas podem ser encontrados em filmes como *Formiguinha Z* e *Vida de Inseto*.
- Considere o seguinte trecho de *Formiguinha Z*, onde uma formiga trabalhadora chamada Z entra no consultório do terapeuta reclamando de sua insignificância:
  - “Eu me sinto insignificante”
  - “Ah, você teve um grande progresso”.
  - “Tive?”
  - “Sim ... você é insignificante!”
- Entretanto, a perspectiva que a maioria das pessoas tem da organização social dos insetos é errônea e absurda. Os filmes acima mostram isso claramente.
- No filme *Formiguinha Z*, por exemplo, existe uma forte hierarquia social, com herdeiros de trono, etc.

- Algumas tarefas que as formigas devem desempenhar:
  - Coletar e distribuir alimento, construir o ninho, cuidar do ninho, dos ovos e das larvas, etc.
- *Alocação de tarefas* é o processo que resulta em alguns trabalhadores realizando tarefas específicas, em intensidade apropriada à situação atual.
- Tratam-se de soluções encontradas para problemas dinâmicos e requerem, portanto, um processo contínuo de adaptação.
- No caso particular das formigas, este processo é auto-organizado. *Nenhuma formiga é capaz de avaliar as necessidades globais do formigueiro e nem de contar a quantidade de trabalhadores envolvidos em cada tarefa de forma a decidir como realocá-los.*
- A capacidade de cada formiga é limitada. Cada trabalhador acaba tomando apenas decisões locais.

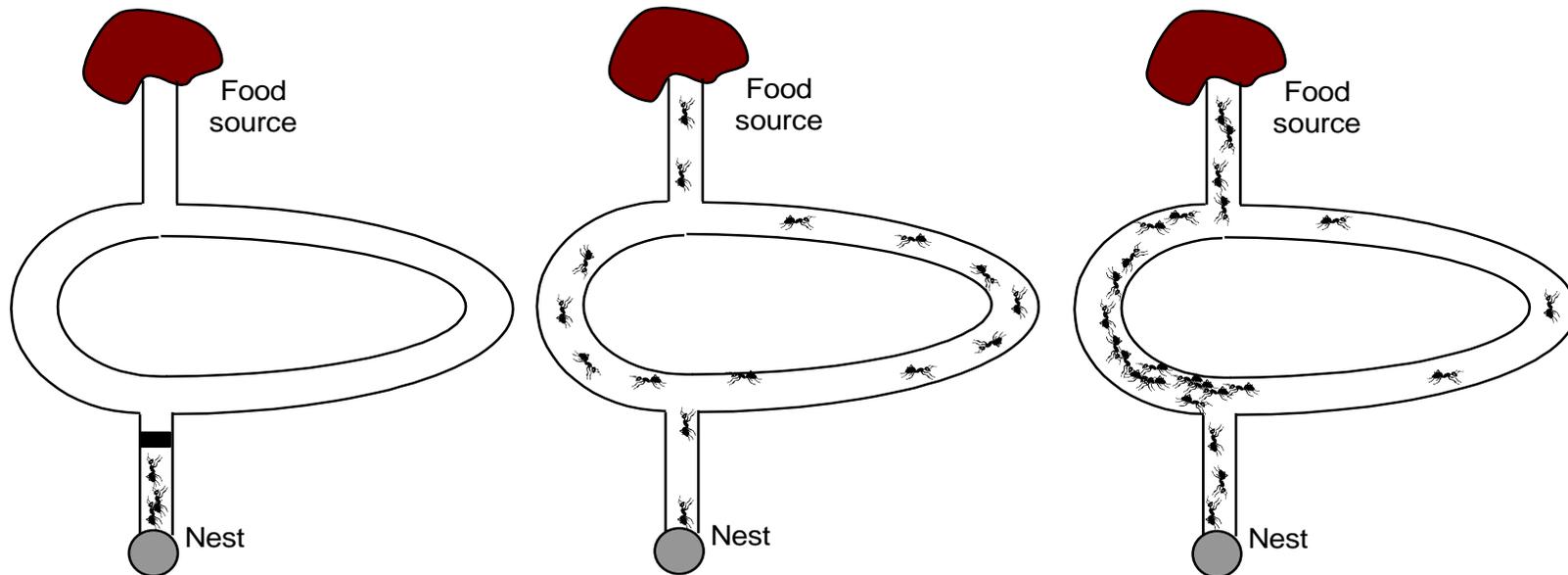
### 3.1. Coleta de Alimento pelas Formigas

- Apesar de existir uma grande variedade de formigas no mundo, boa parte delas possui comportamentos similares de coleta de alimentos.
- Estudos de campo e experimentos de laboratório apontam para uma grande capacidade das formigas em “explorar” ricas fontes de alimentos sem perder a capacidade de explorar o ambiente, assim como em encontrar o menor caminho entre o ninho e a fonte de alimentos.
- Neste sentido, dois comportamentos importantes são observados: *construir uma trilha de feromônio* e *seguir a trilha de feromônio*.
- *Recrutamento* é o nome dado ao mecanismo comportamental que permite que uma colônia de formigas reúna rapidamente uma grande quantidade de coletadoras (*foragers*) em torno de uma determinada fonte de alimento.



- Existem diferentes formas de recrutamento:
  - *Recrutamento em massa*: um explorador descobre a fonte de alimento e retorna ao ninho, liberando uma substância química denominada de *feromônio* e iniciando a formação de uma trilha. Outras formigas detectam a trilha de feromônio, seguem-na e ajudam a reforçá-la.
  - *Recrutamento de grupo*: o explorador guia um grupo de formigas até a fonte de alimento utilizando uma substância química com ação de curto alcance.

- O feromônio possui duas funções importantes:
  - Definir a trilha a ser seguida; e
  - Servir como sinal de orientação para as formigas passeando fora do ninho.
- Exemplo de experimento realizado com formigas para avaliar a importância da trilha de feromônio na coleta de alimentos:



- Algumas observações importantes deste experimento:
  - Os caminhos mais curtos são privilegiados;
  - A probabilidade de um caminho mais curto ser escolhido aumenta com a diferença de comprimento entre os caminhos;
  - Se o caminho mais curto for apresentado (muito) depois do caminho mais longo, ele não será selecionado, a não ser que o feromônio evapore (muito) rapidamente;
  - A quantidade de feromônio que uma formiga libera é diretamente proporcional à qualidade da fonte de alimento (estímulo) encontrada;
  - A aleatoriedade possui um papel importante neste processo. As formigas não seguem as trilhas perfeitamente, elas possuem uma determinada probabilidade de se perderem da trilha ao longo do percurso. Este tipo de comportamento é importante para que seja possível a descoberta de outras fontes de alimento.

- Obs.: Note aqui a presença de realimentação (positiva e negativa), estigmergia e busca baseada em exploração e exploração.

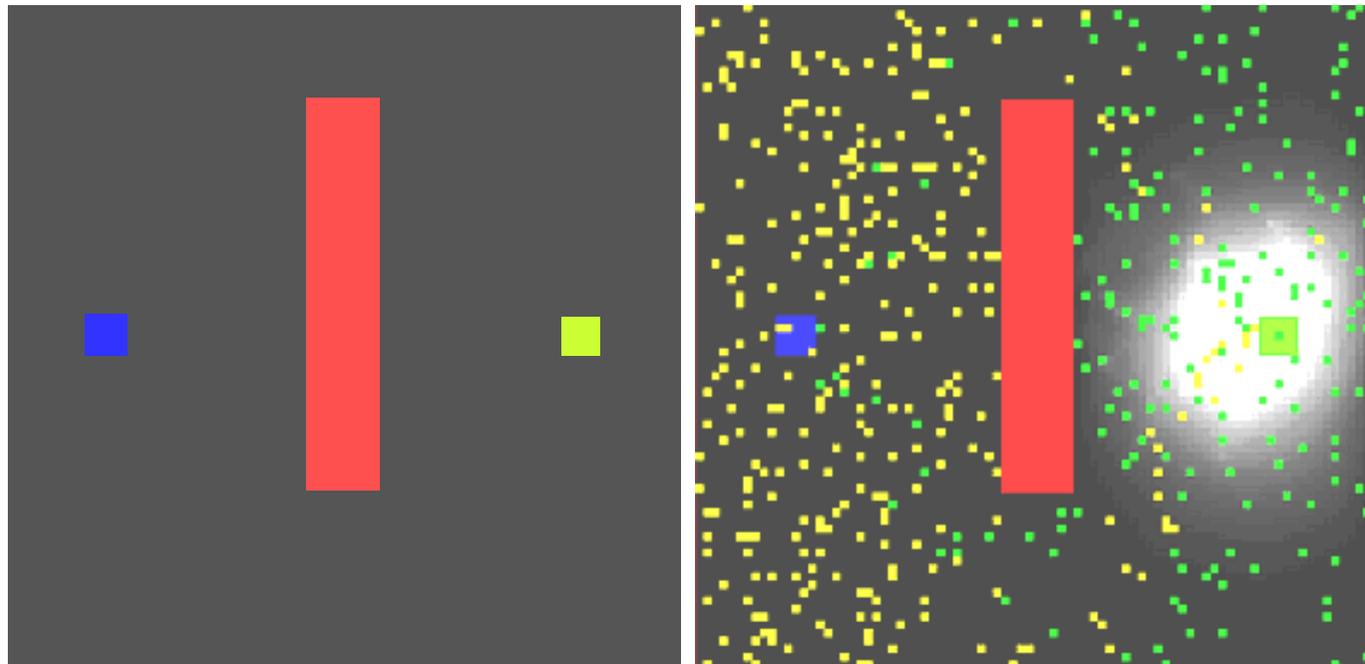
### **3.2. Otimização por Colônia de Formigas**

- A escolha do caminho mais curto entre a fonte de alimento e o ninho permite que as formigas minimizem o tempo gasto nesta viagem.
- O problema de encontrar a menor rota entre o ninho e a fonte de alimento é similar ao problema do caixeiro viajante (TSP).
- Inspirados pelos experimentos de coleta de alimentos por formigas, DORIGO *et al.* (1996) estenderam este modelo para resolver o problema do caixeiro viajante.
- Esta abordagem está baseada em um grupo de “formigas artificiais” que liberam e seguem “trilhas de feromônio artificial”.
- Neste caso, existe uma colônia de formigas (artificiais), cada uma indo de uma cidade a outra de forma independente, favorecendo cidades próximas ou caminhando aleatoriamente (DORIGO & GAMBARDILLA, 1997).

- Após uma formiga artificial completar uma solução válida para o problema do caixeiro viajante, ela libera nas arestas que pertencem à solução proposta uma certa quantidade de feromônio inversamente proporcional ao comprimento total do caminho percorrido: quanto menor o comprimento do caminho percorrido, maior a quantidade de feromônio liberada e vice-versa.
- Depois que todas as formigas tiverem completado suas rotas e liberado feromônio, as arestas que compõem rotas mais curtas terão mais feromônio depositado (acima da média).
- Como o feromônio evapora com o tempo, arestas que participam predominantemente de propostas de solução ruins terão uma queda acentuada na quantidade de feromônio, ao longo das iterações.
- A maior parte dos algoritmos de otimização baseados em colônia de formigas é utilizada para resolver problemas de otimização combinatória representados por grafos (DORIGO *et al.*, 2006).

### 3.3. Uma Simulação de Vida Artificial

- Ninho: lado esquerdo
- Fonte de alimento: lado direito
- $N = 500$  formigas



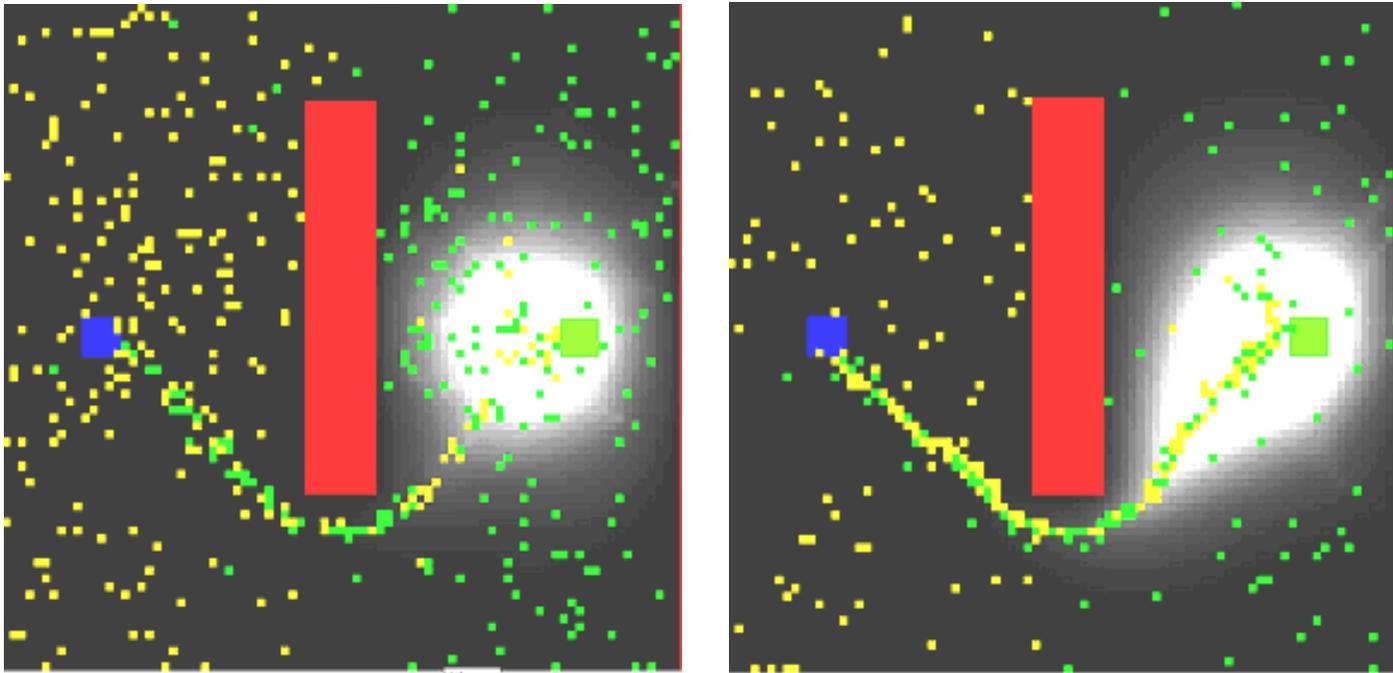
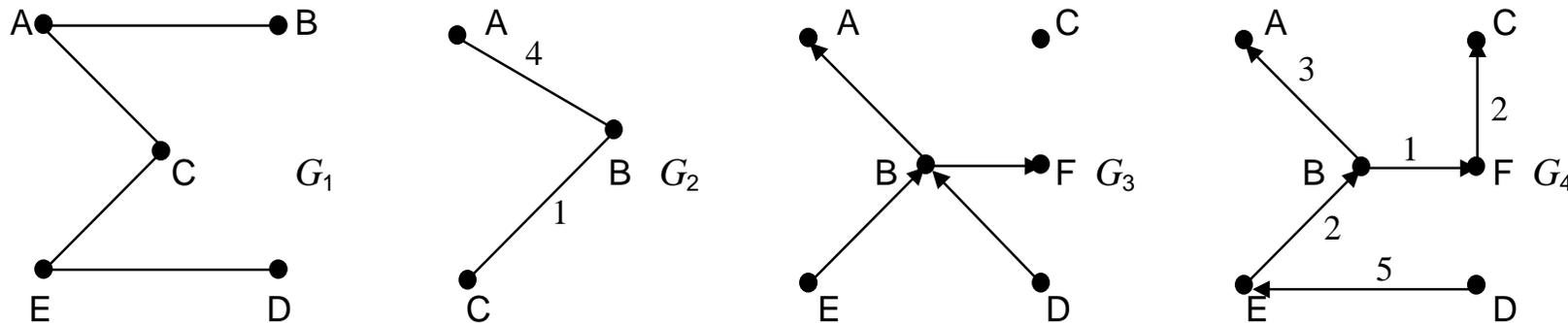


Figura 1 – Simulação de um processo de recrutamento por depósito de feromônio, conduzindo a uma solução de caminho mínimo da fonte de alimento ao ninho.

### 3.4. Conceitos Básicos sobre Teoria de Grafos

- Um grafo pode ser definido como uma 2-upla  $G = (V, E)$  onde  $V$  é um conjunto de vértices ou nós, e  $E$  é um conjunto de arestas ou pares de nós ligando estes vértices:  $V = \{v_0, v_1, \dots, v_N\}$ ,  $E = \{(v_i, v_j) : i \neq j\}$ .

- Um caminho em um grafo consiste em uma sequência de nós e arestas. Quando não existir ambiguidade, um caminho pode ser descrito por uma sequência de nós.
- Um grafo é dito ser:
  - *Conexo*: se existir pelo menos uma aresta ligando cada par de nós.
  - *Direcionado*: se existe uma direção específica de percurso.
  - *Ponderado*: se para cada aresta  $e \in G$  for especificado um número não-negativo  $w(e) \geq 0$  denominado peso ou custo de  $e$ .



### 3.5. Algoritmo Simples de Otimização por Colônia de Formigas

- Algoritmos de otimização por colônia de formigas (ACO) foram inicialmente propostos por DORIGO *et al.* (1991) e DORIGO (1992) como uma abordagem multi-agente para resolver problemas de otimização combinatória.
- Obs: Um problema combinatorial é aquele para o qual existe uma grande quantidade de possíveis soluções em um espaço de busca discreto.
- Tomando um grafo conexo  $G = (V, E)$ , o ACO simples (S-ACO) pode ser utilizado para determinar uma solução (não necessariamente ótima) para o problema do caminho mais curto definido no grafo  $G$ .
- Uma solução é um caminho no grafo conectando um nó inicial  $s$  a um nó destino  $d$ , e o comprimento do caminho é dado pelo número de arestas associadas ao caminho ou pela somatória dos custos das arestas pertencentes ao caminho.
- No S-ACO, existe uma variável  $\tau_{ij}$  denominada de *nível artificial de feromônio* associada a cada aresta  $(i, j)$ .

- Cada formiga artificial é capaz de “liberar feromônio” em uma aresta e avaliar a quantidade de feromônio em uma determinada aresta.
- Cada formiga atravessa uma aresta a cada instante discreto de tempo  $t$  (iteração) e, em cada nó, a informação local sobre a quantidade (nível) de feromônio  $\tau_{ij}$  da aresta é utilizada pela formiga de forma que ela selecione probabilisticamente o próximo nó para o qual ela irá se mover, de acordo com a seguinte regra:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}(t)}{\sum_{j \in N_i} \tau_{ij}(t)} & \text{se } j \in N_i \\ 0 & \text{alhures} \end{cases} \quad (1)$$

onde  $p_{ij}^k(t)$  é a probabilidade de uma formiga  $k$  localizada no nó  $i$  se mover para o nó  $j$ ,  $\tau_{ij}(t)$  é o nível de feromônio da aresta  $(i,j)$ , todos na iteração  $t$ , e  $N_i$  é o conjunto de vizinhos diretos do nó  $i$ .

- Após concluir um percurso válido e ter um retorno da qualidade relativa do mesmo, a formiga  $k$  refaz o percurso depositando feromônio nas arestas:

$$\tau_{ij}(t) \leftarrow \tau_{ij}(t) + \Delta\tau_k, \quad (2)$$

onde  $\Delta\tau_k$  é uma quantidade de feromônio diretamente proporcional à qualidade relativa do percurso realizado pela formiga  $k$ .

- Note que quando uma formiga deposita feromônio numa determinada aresta ela está aumentando a probabilidade de que esta aresta seja selecionada por outra formiga, reforçando uma determinada trilha.
- Para evitar uma convergência prematura do algoritmo, quando aplicado a problemas de caminho mais curto, foi inserido um termo associado à evaporação do feromônio. Considerando as  $N$  formigas, resulta então a fórmula geral de atualização de feromônio:

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) + \sum_{k=1}^N \delta_{ij}^k \Delta\tau_k, \quad (3)$$

onde  $\rho \in [0,1)$  é a taxa de decaimento do feromônio e  $\delta_{ij}^k$  vale 1 se a formiga  $k$  usou a aresta  $(i,j)$  na iteração  $t$  e vale zero caso contrário.

### 3.6. Algoritmo Genérico de Otimização por Colônia de Formigas

- Um algoritmo de otimização por colônia de formigas alterna, por uma quantidade máxima de iterações, a aplicação de dois procedimentos básicos:
  - Um procedimento paralelo de construção/modificação de trilhas no qual um conjunto de  $N$  formigas constroi/modifica  $N$  soluções paralelas para o problema;
  - Uma regra de atualização de feromônio a partir da qual a quantidade de feromônio nas arestas é alterada.
- O processo de construir ou modificar uma solução (caminho) é feito de forma probabilística, e a probabilidade de uma nova aresta ser adicionada à solução sendo construída é função de uma *qualidade heurística*  $\eta$  (*heuristic desirability*) e da quantidade de feromônio  $\tau$  depositada pelas  $N$  formigas.
- A qualidade heurística visa expressar algum atributo (ou conjunto de atributos) que se quer ver presente na solução completa. Exemplo:
  - Quando o caminho mínimo está sendo procurado,  $\eta$  pode ser tomado como sendo inversamente proporcional ao comprimento da aresta.

- A regra de atualização da quantidade de feromônio deve levar em conta a taxa de evaporação de feromônio  $\rho$  e a qualidade das soluções produzidas.
- Seja *best* a melhor solução encontrada até a iteração atual, *max\_it* a quantidade máxima de iterações que o algoritmo irá executar, e *e* a quantidade de nós (ou vértices) no grafo:

**procedure** [*best*] = ACO(*max\_it*)

Initialize  $\tau_{ij}$  //usually every edge is initialized with the same  $\tau_0$

Place each ant *k* on a randomly selected edge

$t \leftarrow 1$

**while**  $t < \text{max\_it}$  **do**,

**for**  $i = 1$  to  $N$  **do**, //for each ant

    Build a solution using a probabilistic transition rule ( $e-1$ ) times. The rule is function of  $\tau$  and  $\eta$

    // $e$  is the number of edges on the graph  $G$

**end for**

  Evaluate the cost of every solution built

**if** an improved solution is found,

**then** update the *best* solution found

**end if**

  Update pheromone trails

$t \leftarrow t + 1$

**end while**

**end procedure**

### 3.7. Exemplo de Aplicação

- Considere o problema do caixeiro viajante (TSP), o qual pode ser diretamente representado em um grafo.
- Neste problema, as formigas constroem as soluções movendo-se de um nó para outro do grafo.
- A cada iteração, uma formiga  $k$ ,  $k = 1, \dots, N$ , constroi um caminho (rota) aplicando uma regra de transição probabilística  $(e-1)$  vezes.
- A transição de uma formiga da cidade  $i$  para a cidade  $j$  na iteração  $t$  irá depender de três fatores:
  - do fato da cidade já ter sido visitada ou não;
  - do inverso da distância  $d_{ij}$  entre as cidades  $i$  e  $j$ , denominado de *visibilidade*  $\eta_{ij} = 1/d_{ij}$ ; e
  - da quantidade de feromônio  $\tau_{ij}$  na aresta ligando as cidades  $i$  e  $j$ .
- Como no caso do TSP cada cidade não deve ser visitada mais do que uma vez, é preciso armazenar informação sobre as cidades que já foram visitadas. Isso pode

ser feito empregando-se, por exemplo, uma lista que irá definir o conjunto de cidades  $J_i^k$  que a formiga  $k$  ainda deve visitar, estando na cidade  $i$ .

- A probabilidade de uma formiga  $k$  ir de uma cidade  $i$  para uma cidade  $j$  na iteração  $t$  é dada pela seguinte regra de transição:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} & \text{se } j \in J_i^k \\ 0 & \text{alhores} \end{cases} \quad (4)$$

onde  $\tau_{ij}(t)$  é o nível de feromônio na aresta  $(i,j)$ , e  $\eta_{ij}$  é a visibilidade da cidade  $j$  quando na cidade  $i$ . Os parâmetros  $\alpha$  e  $\beta$  são definidos pelo usuário e controlam o peso relativo da intensidade da trilha (feromônio) e da visibilidade. Por exemplo, se  $\alpha = 0$ , então cidades mais próximas tenderão a ser escolhidas, enquanto que se  $\beta = 0$ , cidades associadas a arestas com maior quantidade de feromônio tenderão a ser escolhidas.

- De forma similar ao algoritmo simples (S-ACO), a liberação de feromônio nas arestas é proporcional à qualidade do percurso. Neste caso, a quantidade de feromônio liberada em cada aresta  $(i,j)$  pela formiga  $k$ ,  $\Delta\tau_{ij}^k(t)$ , depende de seu desempenho:

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L^k(t) & \text{se } (i,j) \in T^k(t) \\ 0 & \text{alhures} \end{cases} \quad (5)$$

onde  $L^k(t)$  é o comprimento da rota  $T^k(t)$ , percorrida pela formiga  $k$  na iteração  $t$ , e  $Q$  é um parâmetro definido pelo usuário.

- A regra de atualização de feromônio é a mesma do caso simples:

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t),$$

onde  $\rho \in [0,1)$  é a taxa de decaimento de feromônio,  $\Delta\tau_{ij}(t) = \sum_k \Delta\tau_{ij}^k(t)$ , e  $k = 1, \dots, N$  é o índice das formigas.

- Os proponentes do algoritmo sugerem a utilização de  $N = e$ , ou seja, a quantidade de formigas é igual à quantidade de cidades do grafo.
- Os autores também introduziram o conceito de “formigas elitistas”, responsáveis por reforçar a melhor rota encontrada até o momento, adicionando  $b \cdot Q / L_{best}$  ao seu valor de feromônio, onde  $b$  é a quantidade de formigas elitistas, e  $L_{best}$  é o comprimento da melhor rota encontrada até o momento.
- Alguns parâmetros sugeridos:  $\alpha = 1$ ,  $\beta = 5$ ,  $\rho = 0,5$ ,  $N = e$ ,  $Q = 100$ ,  $\tau_0 = 10^{-6}$ , e  $b = 5$ .

### **Escopo de aplicação**

- Problemas de otimização combinatória em geral. Alguns autores sugerem que trata-se da melhor heurística para os problemas de *sequential ordering*, *quadratic assignment* e está entre as melhores alternativas para os problemas de roteamento de veículos e de redes de comunicação de dados (DORIGO & STÜTZLE, 2004).
- Outras aplicações: coloração de grafos, *scheduling*, *multiple knapsack* e *frequency assignment*.

```

procedure [best] = AS-TSP(max_it)
  Initialize  $\tau_{ij}$  //usually every edge is initialized with the same  $\tau_0$ 
  Place each ant  $k$  on a randomly selected city
  Let  $best$  be the best tour found and  $L_{best}$  its length and set  $t \leftarrow 1$ 
  while  $t < \text{max\_it}$  do,
    for  $i = 1$  to  $N$  do, //for every ant
      Build tour  $T^k(t)$  by applying  $(e-1)$  times the following step:
      At city  $i$ , choose the next city  $j$  with probability given by
      Equation (4) //e is the number of cities on the graph
    end for
    Evaluate the length of the tour performed by each ant
    if a shorter tour is found, then update  $best$  and  $L_{best}$ 
    end if
    for every city  $e$  do,
      Update pheromone trails by applying the rule:
       $\tau_{ij}(t+1) \leftarrow (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) + b.\Delta\tau_{ij}^b(t)$ , where
       $\Delta\tau_{ij}(t) = \sum_k \Delta\tau_{ij}^k(t)$ ,  $k = 1, \dots, N$ ;
      
$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L^k(t) & \text{if } (i, j) \in T^k(t) \\ 0 & \text{otherwise} \end{cases}, \text{ and}$$

      
$$\Delta\tau_{ij}^b(t) = \begin{cases} Q/L_{best} & \text{if } (i, j) \in best \\ 0 & \text{otherwise} \end{cases}.$$

    end for
     $t \leftarrow t + 1$ 
  end while
end procedure

```

---

## Da Biologia para a Computação

---

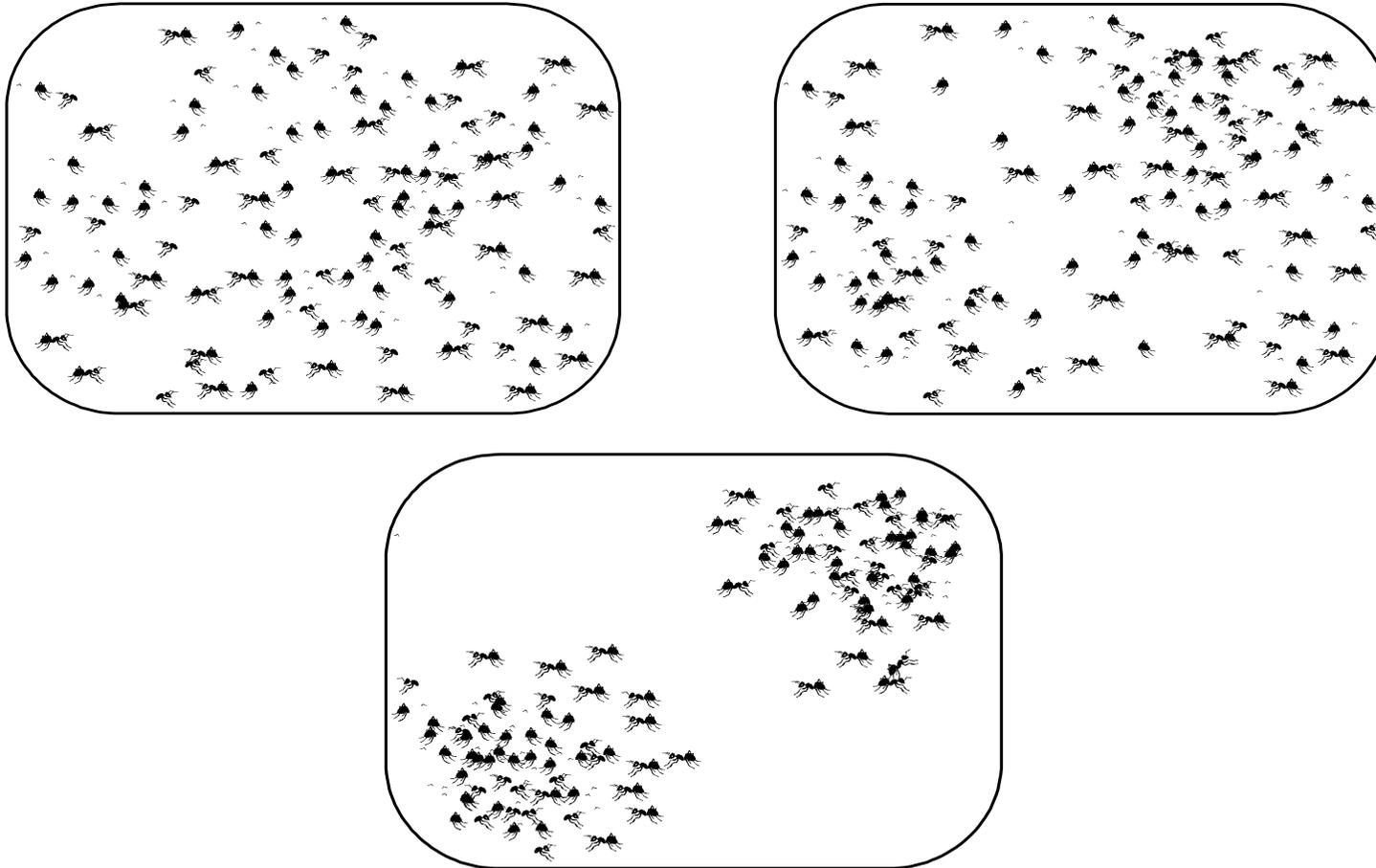
<b>Biologia</b>	<b>Algoritmos ACO</b>
Formiga	Agente usado para construir soluções para o problema
Colônia de formigas	População (colônia) de indivíduos cooperativos conhecidos como formigas artificiais
Trilha de feromônio	Modificação do ambiente promovida pelas formigas artificiais com o objetivo de fornecer uma comunicação indireta com outras formigas da colônia
Evaporação do feromônio	Redução do nível de feromônio de um dado ramo com o passar do tempo

---

### 3.8. Clusterização de Corpos e Organização de Larvas

- Para limpar seus formigueiros, algumas espécies de formigas juntam corpos e partes de corpos de formigas mortas em regiões específicas do formigueiro.
- O mecanismo básico por trás deste processo é uma atração entre os itens mortos mediada pelas formigas. Pequenos amontoados se formam e vão crescendo atraindo uma maior quantidade de corpos naquela região do espaço.
- Este comportamento pode ser modelado utilizando-se duas regras simples:
  - *Regra para pegar um item:* se uma formiga encontra um item morto ela o pega e passeia pelo ambiente até encontrar outro item morto. A probabilidade desta formiga pegar o item morto é inversamente proporcional à quantidade de itens mortos naquela região do espaço.
  - *Regra para largar um item:* carregando um item a formiga eventualmente encontra mais itens no caminho. A probabilidade desta formiga deixar este item junto aos outros é proporcional à quantidade de itens mortos naquela região.

- Como resultado dessas regras comportamentais simples, todos os itens mortos irão, eventualmente, ser agrupados em um número arbitrário de localidades.



### 3.9. Clusterização por Colônia de Formigas

- A análise de cluster ou clusterização de dados pode ser definida como a organização ou separação de um conjunto de dados ou padrões em grupos denominados de clusters. Essa organização é feita baseada em algum critério de similaridade.
- Os dados são geralmente representados por um vetor de medidas ou atributos que corresponde a um ponto em um espaço multidimensional.
- Intuitivamente, dados em um mesmo cluster são mais semelhantes do que dados que não pertencem ao mesmo cluster.
- O problema de clusterização de dados pode ser definido como a seguir:
  - Seja um conjunto  $\mathbf{X}$  de  $P$  amostras (dados),  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_P\}$ , cada qual de dimensão  $L$ . Encontre um esquema de discriminação para agrupar (clusterizar) os dados em  $c$  grupos denominados de clusters. O número de clusters pode ser determinado automaticamente ou não.

- Para desenvolver um esquema de discriminação de forma a clusterizar os dados, é necessário definir uma métrica, geralmente uma medida de distância, que quantifica o grau de similaridade (ou dissimilaridade) entre dois pontos em um determinado espaço métrico.
- A métrica mais comumente utilizada é a distância euclidiana:  
$$D_2(\mathbf{x}_i, \mathbf{x}_j) = (\sum_k (x_{i,k} - x_{j,k})^2)^{1/2} = \|\mathbf{x}_i - \mathbf{x}_j\|_2.$$
- É importante salientar que clusterização envolve o agrupamento de dados não-rotulados, ou seja, dados cujas classes não são pré-conhecidas.

### **3.10. Algoritmo Simples de Clusterização (ACA)**

- Neste algoritmo, uma colônia de “agentes-formigas” movendo-se aleatoriamente em uma grade bidimensional tem a capacidade de pegar itens dentro da grade e movê-los para outras posições da grade.
- A ideia geral é de que itens isolados devem ser pegos e movidos para locais da grade em que se encontram mais itens daquele mesmo tipo.

- Note, entretanto, que o grupo ao qual cada item pertence é desconhecido a priori.
- DENEUBOURG *et al.* (1991) propuseram um modelo teórico para estudar (modelar) a organização de cemitérios em algumas espécies de formigas.
- Suponha que exista um único tipo de item no ambiente, e que uma determinada quantidade de agentes-formiga, cuja função é carregar itens de uma posição a outra da grade, está disponível.
- A probabilidade  $p_p$  de que uma formiga “descarregada” se movendo aleatoriamente pela grade pegue um determinado item é:

$$p_p = \left( \frac{k_1}{k_1 + f} \right)^2$$

onde  $f$  é a fração de itens percebidos na vizinhança da formiga, e  $k_1$  é uma constante (*threshold* ou limiar). Para  $f \ll k_1$ ,  $p_p \approx 1$ , ou seja, a probabilidade de uma formiga pegar um item quando há poucos itens em sua vizinhança é grande.

- A probabilidade  $p_d$  de uma formiga “carregada” movendo-se aleatoriamente pelo ambiente deixar este item em uma determinada posição da grade é dada por:

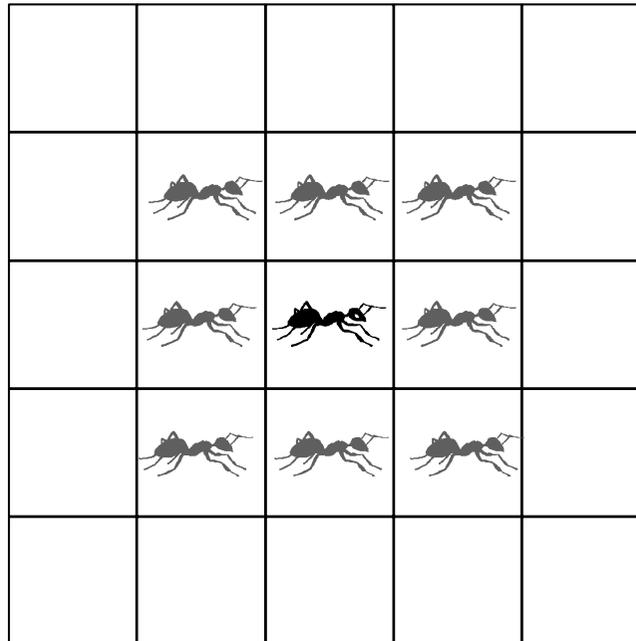
$$p_d = \left( \frac{f}{k_2 + f} \right)^2$$

onde  $f$  é a fração de itens percebidos na vizinhança da formiga, e  $k_2$  é outra constante (*threshold* ou limiar). Para  $f \ll k_2$ ,  $p_d \approx 0$ , ou seja, a probabilidade de uma formiga deixar um item quando há poucos itens em sua vizinhança é pequena.

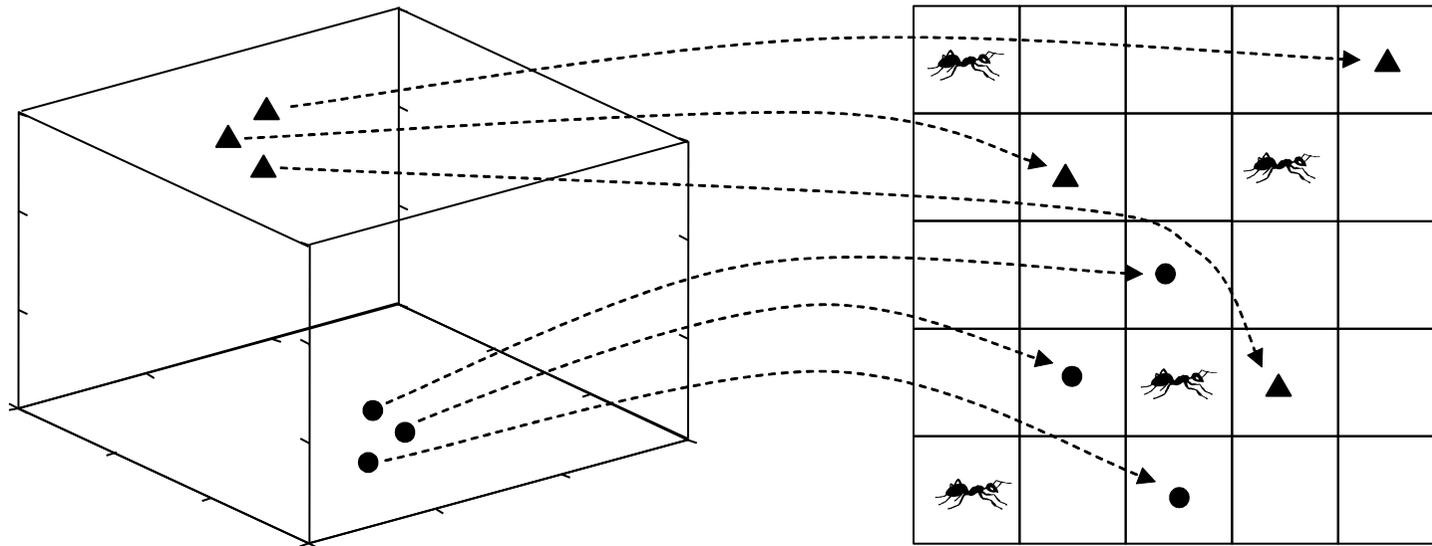
- Para utilizar este modelo teórico como uma ferramenta de clusterização, ainda é necessário definir dois aspectos importantes:
  - Qual o tipo de ambiente no qual as formigas vão se movimentar?
  - Como definir a função  $f$ ?

## Definição do Ambiente

- No algoritmo padrão, as formigas movem-se em uma grade bidimensional contendo  $m \times m$  células, e possuem a capacidade de perceber o ambiente em uma vizinhança de sua posição atual  $\text{Neigh}_{(s \times s)}$ .



- Neste caso, os padrões de entrada são projetados em regiões aleatórias da grade e devem posteriormente ser reposicionados de forma a preservar as relações de vizinhança entre itens “vizinhos” no espaço original de atributos.



### Definição da Fração de Itens Percebidos $f$

- Note que  $f$  pode ser entendida como sendo a “visibilidade” de cada formiga.

- Assim como no caso da função de fitness em algoritmos evolutivos,  $f$  será uma função do problema a ser tratado. Por exemplo, em um contexto de sistemas robóticos,  $f$  pode ser definida como sendo o quociente entre a quantidade  $Q$  de itens encontrados nas últimas  $T$  iterações do algoritmo e a maior quantidade possível de itens que poderia ser encontrada neste período.
- Supondo que as formigas se movem em uma grade bidimensional, o algoritmo padrão de clusterização baseado em colônia de formigas pode ser descrito como a seguir:

```
procedure [] = ACA(max_it, N, k1, k2)
  Place every item  $i$  on a random cell of the grid
  Place every ant  $k$  on a random cell of the grid unoccupied by ants
   $t \leftarrow 1$ 
  while  $t < \text{max\_it}$  do,
    for  $i = 1$  to  $N$  do,           //for every ant
      if unladen ant AND cell occupied by item  $\mathbf{x}_i$ , then
        Compute  $f(\mathbf{x}_i)$  and  $p_p(\mathbf{x}_i)$ 
        Pick up item  $\mathbf{x}_i$  with probability  $p_p(\mathbf{x}_i)$ 
      else if ant carrying item  $\mathbf{x}_i$  AND cell empty, then
        Compute  $f(\mathbf{x}_i)$  and  $p_d(\mathbf{x}_i)$ 
        Deposit (drop) item  $\mathbf{x}_i$  with probability  $p_d(\mathbf{x}_i)$ 
      end if
      Move to a randomly selected neighboring and unoccupied cell
    end for
     $t \leftarrow t + 1$ 
  end while
  print location of items
end procedure
```

### 3.11. Exemplos de Aplicação

- LUMER & FAIETA (1994) aplicaram o algoritmo padrão ao problema de análise exploratória de dados, onde o objetivo era encontrar clusters em dados não-rotulados.
- Os dados foram tomados em um espaço euclidiano de dimensão  $L$ ,  $\mathcal{R}^L$ , e foi utilizada uma grade bidimensional com vizinhança unitária.
- A função  $f$  é dada por:

$$f(\mathbf{x}_i) = \begin{cases} \frac{1}{s^2} \sum_{\mathbf{x}_j \in \text{Neigh}_{(s \times s)}(r)} \left[ 1 - \frac{d(\mathbf{x}_i, \mathbf{x}_j)}{\alpha} \right] & \text{se } f > 0 \\ 0 & \text{alhures} \end{cases} \quad (6)$$

onde  $f(\mathbf{x}_i)$  é uma medida da similaridade média do item  $\mathbf{x}_i$  em relação a outro item  $\mathbf{x}_j$  na vizinhança de  $\mathbf{x}_i$ ,  $\alpha$  é um fator que define a escala de dissimilaridade, e  $d(\mathbf{x}_i, \mathbf{x}_j)$  é a distância euclidiana entre os dados  $\mathbf{x}_i$  e  $\mathbf{x}_j$  em seus espaços originais.

- As probabilidades de pegar e deixar um item foram dadas por:

$$p_p(\mathbf{x}_i) = \left( \frac{k_1}{k_1 + f(\mathbf{x}_i)} \right)^2 \quad (7)$$

$$p_d(\mathbf{x}_i) = \begin{cases} 2f(\mathbf{x}_i) & \text{se } f(\mathbf{x}_i) < k_2 \\ 1 & \text{se } f(\mathbf{x}_i) \geq k_2 \end{cases} \quad (8)$$

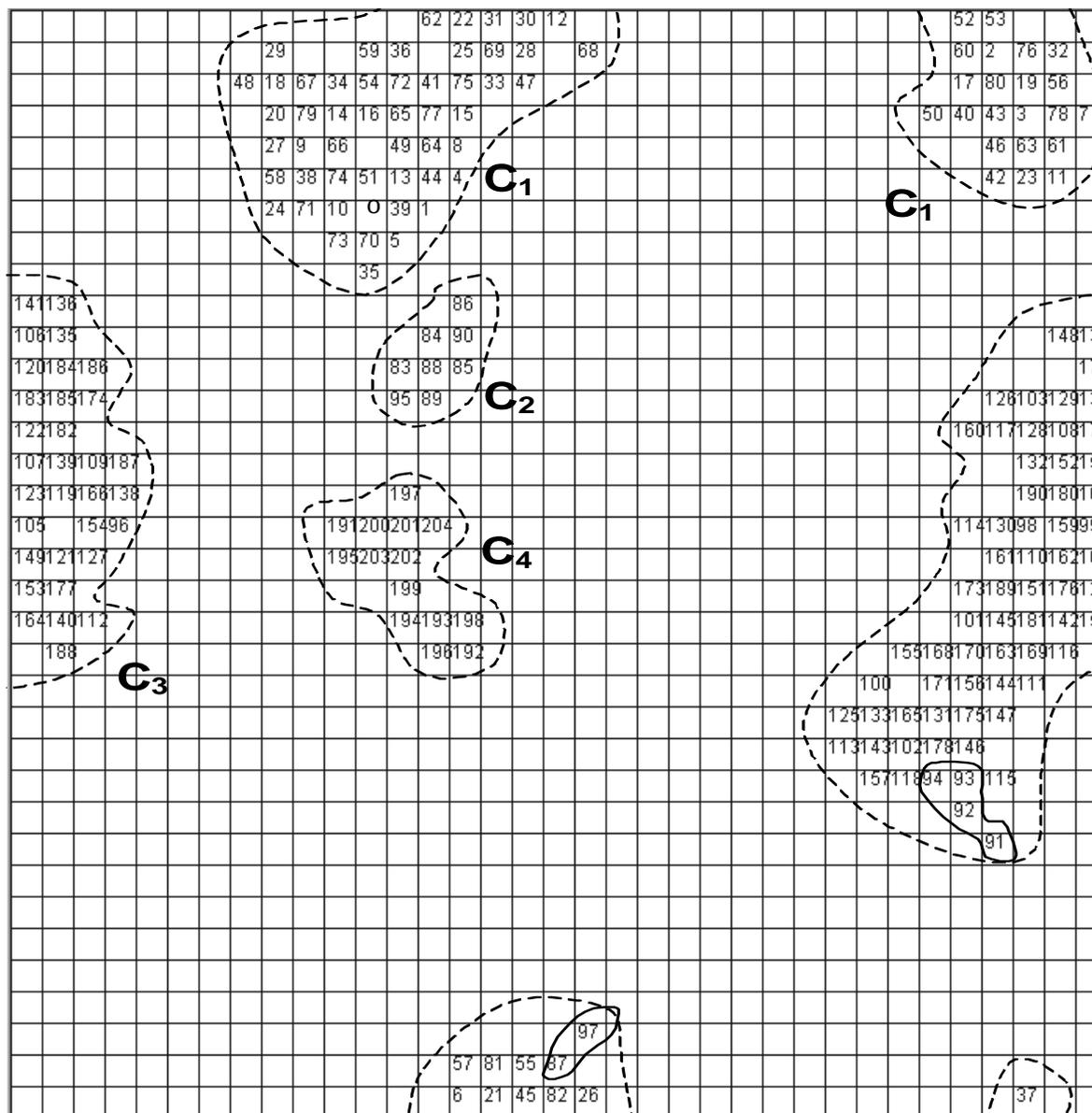
- Embora o algoritmo ACA seja capaz de agrupar os dados, ele geralmente encontra uma quantidade de grupos maior do que aquela associada aos grupos naturais. Além disso, o algoritmo padrão não estabiliza em uma dada solução, ele fica construindo e reconstruindo grupos constantemente.
- Para aliviar estes problemas, VIZINE *et al.* (2005) propuseram três modificações no algoritmo original:
  - Um decaimento para o parâmetro  $k_1$ ;
  - Um campo de visão progressivo que permite uma visão mais abrangente para as formigas; e

- A adição de feromônio aos itens carregados pelas formigas e possibilidade de transferência de feromônio para a grade.
- Decaimento de  $k_1$ :
  - A cada ciclo (10.000 passos de formiga)  $k_1$  sofre um decaimento geométrico:  
$$k_1 \leftarrow 0,98 \times k_1,$$
$$k_{1\min} = 0,001.$$
  - Quando uma formiga percebe um grupo grande ela aumenta seu campo de visão:  
If  $f(\mathbf{x}_i) > \theta$  and  $s^2 \leq s^2_{\max}$ ,  
then  $s^2 \leftarrow s^2 + n_s$   
Sugestão dos autores:  $s^2_{\max} = 7 \times 7$  e  $\theta = 0,6$
  - Inspirados no processo de realimentação positiva via feromônio na construção de ninhos pelos cupins, os autores propuseram a adição de um nível de feromônio à grade  $\phi(i)$ , onde  $i$  é o índice da célula da grade:

$$P_p(i) = \frac{1}{f(i)\phi(i)} \left( \frac{k_1}{k_1 + f(i)} \right)^2 \quad (9)$$

$$P_d(i) = f(i)\phi(i) \left( \frac{f(i)}{k_2 + f(i)} \right)^2 \quad (10)$$

- Uma aplicação investigada: *Yeast galactose data* (bioinformática: expressão de genes).
  - 205 amostras com 20 atributos divididas em 4 grupos distintos (Obs.: dados não-rotulados)
  - Parâmetros do algoritmo:  $n_{ants} = 10$ , tamanho da grade  $35 \times 35$ ,  $\alpha = 1,05$ ,  $\theta = 0,6$ ,  $k_1 = 0,20$ , e  $k_2 = 0,05$ .



## Escopo de aplicação

- Os algoritmos de clusterização baseados em colônia de formigas são aplicáveis a problemas de análise exploratória de dados, onde um conjunto de dados não rotulados está disponível e alguma informação (grau de similaridade entre itens, inferência sobre a pertinência de novos itens, etc.) deve ser extraída (inferida) desses dados.
- Aspecto importante do algoritmo: redução da dimensionalidade e, portanto, a capacidade de visualizar relações de vizinhança entre dados de elevada dimensão.
- Uma análise de sensibilidade a alguns parâmetros de algoritmos ACA foram realizadas em SHERAFAT *et al.* (2005).

## Da Biologia para a Computação

<b>Biologia (Ant Clustering)</b>	<b>Algoritmo ACA</b>
Ambiente (Arena)	Grade bidimensional na qual os itens são projetados e as formigas se movem
Formiga	Agente capaz de se mover no ambiente, pegar e largar itens
Colônia de formigas	População (colônia) de agentes cooperativos conhecidos como formigas artificiais
Corpos e larvas de formigas	Itens (p.ex. dados de entrada)
Pilha (grupos) de corpos	Clusters de itens
Visibilidade de uma formiga	Fração de itens percebidos: $f$

## 4. Robótica de Enxame

- Em *robótica autônoma*, a chamada *robótica coletiva* ou *robótica de enxame* é baseada em metáforas e inspiração tomada de sistemas biológicos, em particular de insetos sociais, para o projeto de sistemas de controle distribuído ou estratégias de coordenação para grupos de robôs.
- Comportamentos coletivos de insetos sociais fornecem fortes evidências de que sistemas compostos por agentes simples são capazes de realizar tarefas complexas específicas (MARTINOLI, 2001).
- Sabe-se, entretanto, que as capacidades cognitivas destes insetos são muito restritas.
  - Sendo assim, os comportamentos complexos que surgem devem ser propriedades emergentes resultantes das interações dos agentes e deles com seu ambiente, onde cada agente geralmente segue regras comportamentais simples.

- Portanto, a robótica coletiva inspirada em insetos sociais é diferente das abordagens de inteligência artificial clássica, no sentido de que a robótica coletiva é do tipo *bottom-up*: grupos de agentes simples seguindo regras comportamentais simples (sistemas auto-organizados).
- Grupos de robôs móveis são projetados e construídos com o objetivo principal de estudar características como arquitetura de grupo, origem de cooperação, aprendizagem, resolução de conflitos, etc.
- O crescente interesse pela robótica coletiva nos últimos anos deve-se a vários fatores:
  - Algumas tarefas são inerentemente muito complexas (ou impossíveis) de serem resolvidas por um único robô;
  - Melhorias de desempenho podem ser conseguidas utilizando-se múltiplos robôs;

- A construção e utilização de robôs simples é geralmente muito mais barata, flexível e tolerante a falhas do que projetar um único robô com alta capacidade de processamento de informação e sensores complicados;
- A queda nos preços de robôs comerciais simples, como os robôs Khepera<sup>®</sup>;
- O progresso da robótica móvel facilitou o estudo com grupos de robôs;
- Estudos em Vida Artificial contribuíram para um maior entendimento e formalização de processos auto-organizados e fenômenos emergentes; e
- As características construtivas e sintéticas da robótica coletiva contribuem para uma melhor compreensão de diversos fenômenos biológicos e sociais.
- Uma das características construtivas marcantes da robótica coletiva é a utilização de vários robôs com regras comportamentais simples e individuais.
- Sendo assim, o comportamento coletivo será uma propriedade emergente do grupo.

- Essa característica gera então uma dúvida importante: como poderemos prever que o comportamento do grupo será apropriado para a realização de uma determinada tarefa?
- Outra dificuldade da robótica coletiva é que, devido à falta de conhecimento global do sistema, o sistema pode estagnar em algum ponto de operação.
- Existe uma grande quantidade de trabalhos em robótica coletiva, e descreveremos quatro deles inspirados nos seguintes comportamentos biológicos das formigas:
  - Coleta de alimento;
  - Clusterização de corpos;
  - Agrupamento em torno da fonte de alimento (recrutamento); e
  - Transporte coletivo de presas.
- O enfoque desta parte do curso será nas regras comportamentais de robôs individuais que levam a comportamentos emergentes. Não serão apresentadas discussões sobre os aspectos construtivos e detalhes de implementação dos robôs.

- Há uma grande quantidade de vídeos didáticos e informativos dedicados à inteligência de enxame, robótica coletiva e morfogênese em robótica. Sugere-se que o leitor assista aos seguintes vídeos:

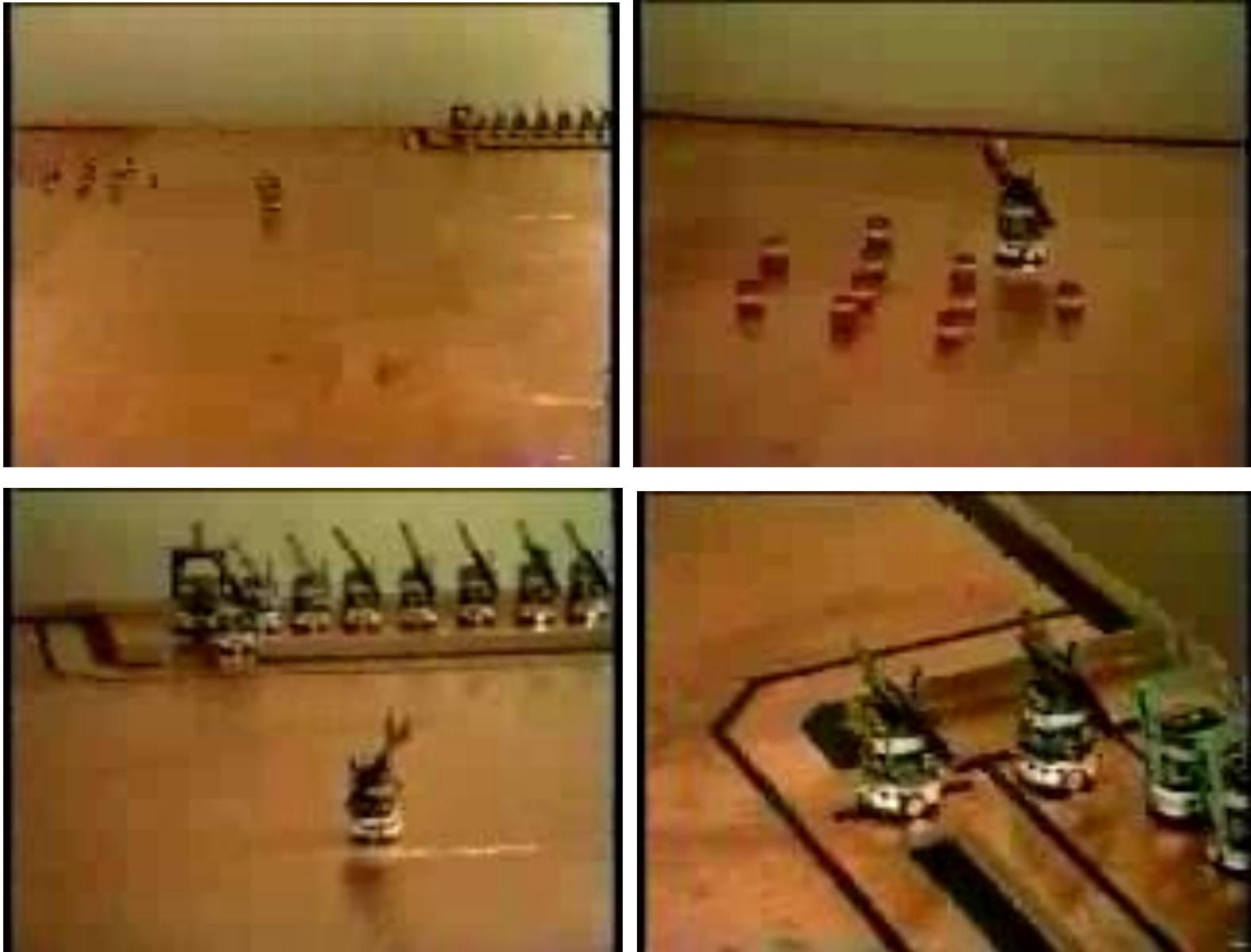
- ✓ [http://www.youtube.com/watch?v=-G66iL\\_VdA](http://www.youtube.com/watch?v=-G66iL_VdA)
- ✓ <http://www.youtube.com/watch?v=seGqyO32pv4>
- ✓ <http://www.youtube.com/watch?v=CJOubyiITsE>
- ✓ <http://www.youtube.com/watch?v=v6W-sEpJEqY>
- ✓ <http://www.youtube.com/watch?v=M2nn1X9Xlps>
- ✓ [http://www.youtube.com/watch?v=uJ\\_0T\\_UnhJI](http://www.youtube.com/watch?v=uJ_0T_UnhJI)
- ✓ <http://www.youtube.com/watch?v=HJ5vzh9qQJ4>
- ✓ <http://www.youtube.com/watch?v=LFwk303p0zY>
- ✓ [http://www.youtube.com/watch?v=dhVZkt\\_145o](http://www.youtube.com/watch?v=dhVZkt_145o)

## 4.1. Coleta de Alimento pelas Formigas

- KRIEGER *et al.* (2000) demonstraram que um sistema de alocação de tarefas inspirado no comportamento de formigas coletando alimento fornece um mecanismo simples, robusto e eficiente para regular as atividades de um grupo de robôs.
- O objetivo é coletar “itens de alimento” sem que os robôs tenham informação alguma sobre o ambiente e a quantidade de robôs na colônia. Uma quantidade mínima de itens de alimento deve ser mantida no ninho para que a energia da colônia seja mantida acima de um determinado limiar.
- A escolha deste mecanismo biológico deve-se a vários fatores:
  - A coleta de alimentos é um dos problemas para o qual se possui um maior grau de conhecimento sobre como se dá a divisão de tarefas;
  - A eficiência da coleta de alimentos é um fator-chave para a produtividade da colônia; e

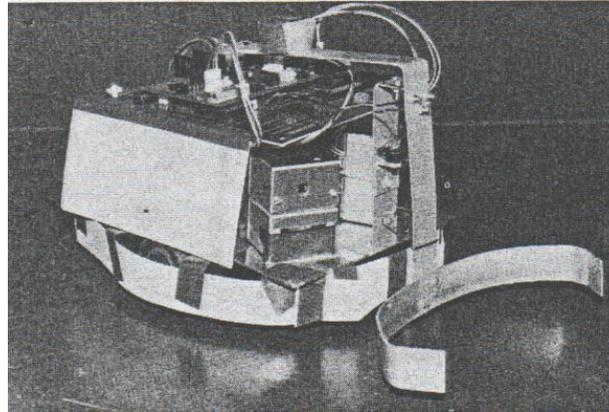
- A coleta de alimentos é uma das tarefas que tem recebido maior atenção pela comunidade de robótica coletiva.
- Seja um grupo de robôs em um “ninho” central e um conjunto de “itens de alimentação”, todos dentro de uma arena finita.
- O experimento e as regras comportamentais podem ser resumidos como a seguir:
  - Cada robô possui a capacidade de avaliar e alterar a energia da colônia;
  - Enquanto no ninho, os robôs recebem informação sobre o nível de energia da colônia através de mensagens de rádio;
  - Os robôs saem do ninho sequencialmente apenas quando a energia da colônia cai abaixo de um determinado limiar;
  - Cada robô é programado para evitar colisões com obstáculos;
  - Ao retornar ao ninho, cada robô renova sua energia decrementando a energia da colônia, e descarrega o item de alimento coletado em uma cesta, aumentando a energia da colônia; e

- Em alguns experimentos, os robôs foram programados para recrutar outros robôs assim que eles encontram uma região da arena rica em alimento, imitando o tipo de recrutamento observado em algumas espécies de formigas.
- Em resumo, os robôs possuem informações gerais sobre o nível de energia da colônia, evitam colisões e, em alguns casos, podem recrutar outros robôs para coletar alimento.
- Os experimentos realizados mostraram, dentre outras coisas, que o recrutamento em questão é uma forma eficiente de “explorar” itens agrupados (clusterizados) no ambiente. Além disso, o sistema também se mostrou capaz de promover a manutenção de um nível de energia mínima no ninho.
- Vídeo disponível em:
  - <http://www.nature.com/nature/journal/v406/n6799/extref/406992ai1.mov>



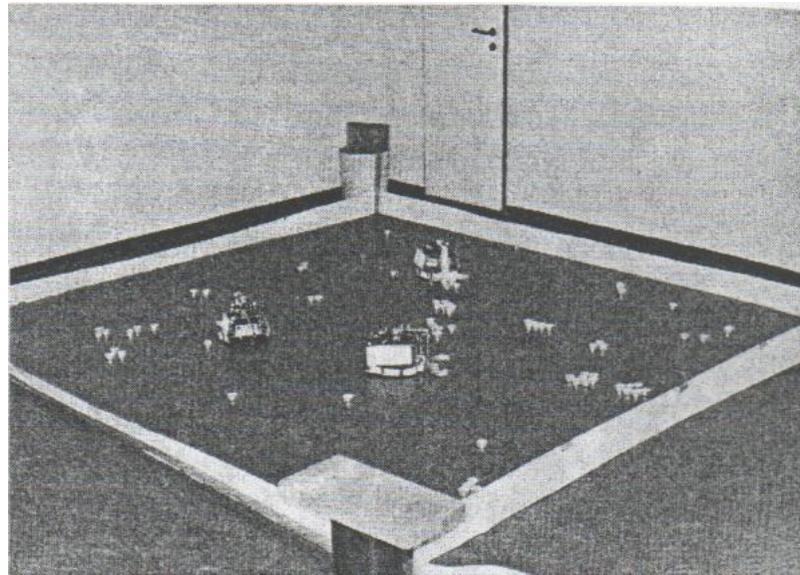
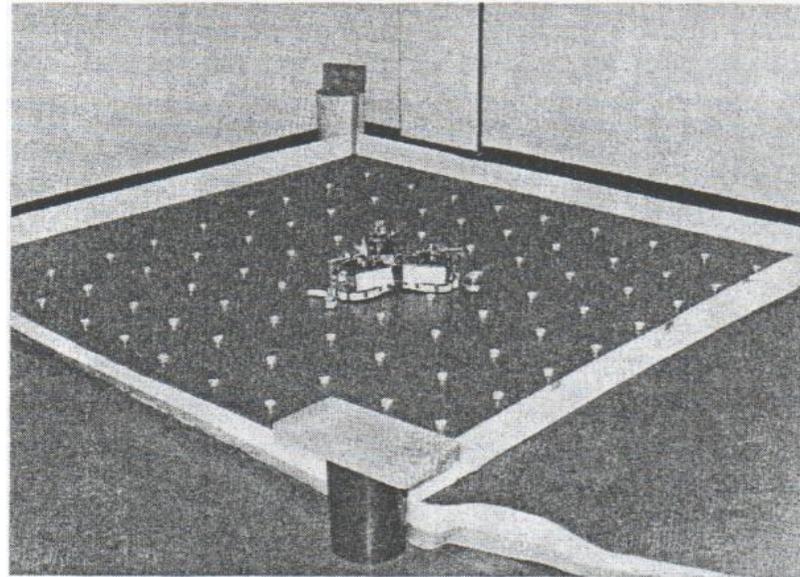
## 4.2. Clusterização de Objetos

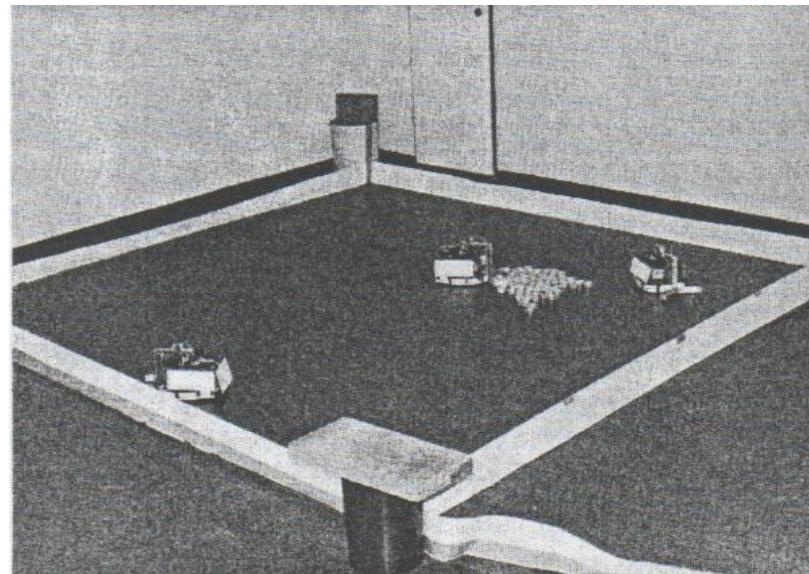
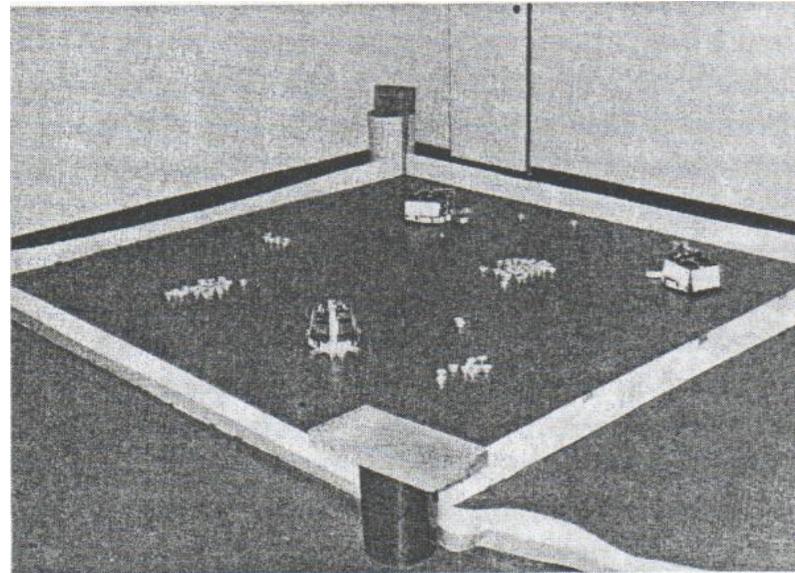
- BECKERS *et al.* (1994) desenvolveram um sistema de robótica coletiva para juntar itens dispersos em uma arena em um único cluster, simulando o comportamento de clusterização de itens mortos em colônia de formigas.
- Os robôs foram projetados de forma que eles pudessem mover uma certa quantidade de itens, e tal que a probabilidade de um item ser depositado em uma determinada região da arena fosse diretamente proporcional à quantidade de itens naquela região.
- Isso foi feito através da percepção de uma densidade local de itens utilizando um mecanismo simples de disparo.
- Nessas simulações, os robôs foram equipados com uma garra em forma de C com um sensor capaz de detectar a presença de três ou mais itens em seu interior.



- Neste caso, os robôs possuem apenas três regras comportamentais:
  - Quando nenhum sensor é ativado, o robô executa seu comportamento padrão, que corresponde a uma movimentação em linha reta até que um obstáculo seja detectado ou um sensor (*switch*) seja ativado.
  - Ao detectar um obstáculo, o robô executa um comportamento para evitar colisão do tipo girar em torno do eixo com um ângulo aleatório, e o comportamento padrão volta a ser executado. Se o robô está empurrando alguns itens quando ele encontra um obstáculo, estes itens são retidos pela garra enquanto o robô faz a manobra.

- Quando a garra empurra três ou mais itens, um sensor é ativado resultando na liberação dos itens, que consiste na abertura da garra seguida de uma marcha a ré. Feito isso, o robô gira com um ângulo aleatório em torno de seu eixo e volta a seu comportamento padrão.
- Não existe nenhum tipo de comunicação explícita entre os robôs. Eles são autônomos com todos os sensores, motores, e circuitos de controle independentes.
- Portanto, o comportamento resultante desta colônia de robôs é conseguido através de um processo auto-organizado baseado na reação dos robôs à configuração ambiental corrente (estigmergia).
- Os resultados apresentados mostraram que estas regras comportamentais simples permitem controle e coordenação de um grupo de robôs sem comunicação direta.
- Trata-se, portanto, de uma verificação de mundo real, em que comportamentos similares a insetos sociais podem ser implementados em robôs para que eles realizem uma determinada tarefa.





### 4.3. Transporte Coletivo de Presas

- Várias espécies de formigas são capazes de transportar coletivamente presas tão grandes que estas jamais poderiam ser coletadas por uma única formiga.
- Se uma única formiga encontrar uma presa e for capaz de transportá-la sozinha para o ninho, ela o fará.
- Entretanto, se ela não for capaz, esta formiga poderá recrutar outras formigas via comunicação direta ou através da criação de trilhas de feromônio (recrutamento em massa).
- Quando mesmo um grupo grande de formigas não é capaz de mover uma presa, formigas trabalhadoras especializadas com grandes mandíbulas poderão ser recrutadas de forma a cortar a presa em pedaços mais facilmente transportáveis.
- Apesar de muito estudados, estes fenômenos de transporte coletivo de presas ainda não possuem um modelo formal.
- Existem várias características interessantes do transporte coletivo por formigas:

- Ele é mais eficiente do que o transporte individual, e a capacidade coletiva das formigas cresce de forma não-linear com a quantidade de formigas envolvidas na tarefa;
- A resistência ao transporte parece ser o fator determinante na forma como a presa será transportada. Por exemplo: se ela será puxada ou arrastada, e se haverá mais do que uma formiga envolvida nesta tarefa.
- Se uma única formiga tenta carregar uma presa e falha, mais formigas poderão ser recrutadas. A formiga perde algum tempo tentando *realinhar* seu corpo sem soltar a presa na esperança de quebrar a inércia e movê-la. Caso o realinhamento corporal não seja suficiente, a formiga também tenta se *reposicionar* em torno da presa.
- Acredita-se que a quantidade de formigas envolvidas na coleta de presa é uma função da dificuldade em movê-la, e não apenas do peso da presa.
- A resistência ao movimento estimula o recrutamento de mais formigas.

- O sucesso em empurrar uma presa em uma determinada direção é seguido por outras tentativas na mesma direção.
- O recrutamento é cessado quando as formigas envolvidas no transporte são capazes de transportar a presa em uma direção bem definida.
- Em resumo:
  - Se uma determinada presa não pode ser movida, as formigas iniciam tentando se realinhar e se reposicionar em torno dela.
  - Somente quando estas duas estratégias falham é que mais formigas são recrutadas.
- É importante notar que a coordenação no transporte coletivo parece ser mediada pela presa, ou seja, o resultado da ação de uma formiga irá influenciar o estímulo percebido pelas outras formigas.

## Cooperative Box Pushing

- Inspirados por essas observações, KUBE & ZANG (1992) estudaram o problema de transporte coletivo de presas visando uma implementação em robôs.
- O problema por eles abordado – *collective box pushing* – é, em essência, equivalente ao transporte coletivo de presas (KUBE *et al.*, 2005).
- Três tarefas foram estudadas:
  - Empurrar uma caixa em qualquer direção;
  - Empurrar a caixa em uma direção pré-especificada;
  - Transportar a caixa para vários destinos pré-especificados.
- Os robôs possuem cinco comportamentos hierárquicos:
  - Evitar colisões;
  - Realizar a tarefa (objetivo);
  - Reduzir a velocidade;
  - Seguir o vizinho mais próximo;

- Encontrar a presa.
- Apenas três regras comportamentais foram implementadas:
  - Um único robô se esforça para puxar ou empurrar a presa tentando diversos realinhamentos e reposicionamentos;
  - Se ele não for bem sucedido, ele recruta outros robôs;
  - O grupo de robôs irá tentar de forma coletiva e cooperativa mover a presa (objeto) tentando se realinhar e reposicionar independentemente, até que uma configuração satisfatória dos robôs resulte em uma movimentação do objeto.
- Assim como no exemplo anterior, nenhum mecanismo de comunicação direta é utilizado no transporte do objeto.
- Verificou-se que, na ausência de realinhamento e reposicionamento, o sistema sofria estagnação.
- Este sistema é considerado o único modelo existente para o respectivo fenômeno biológico envolvido.

- Vídeos disponíveis em:
  - <http://webdocs.cs.ualberta.ca/~kube/research.html>

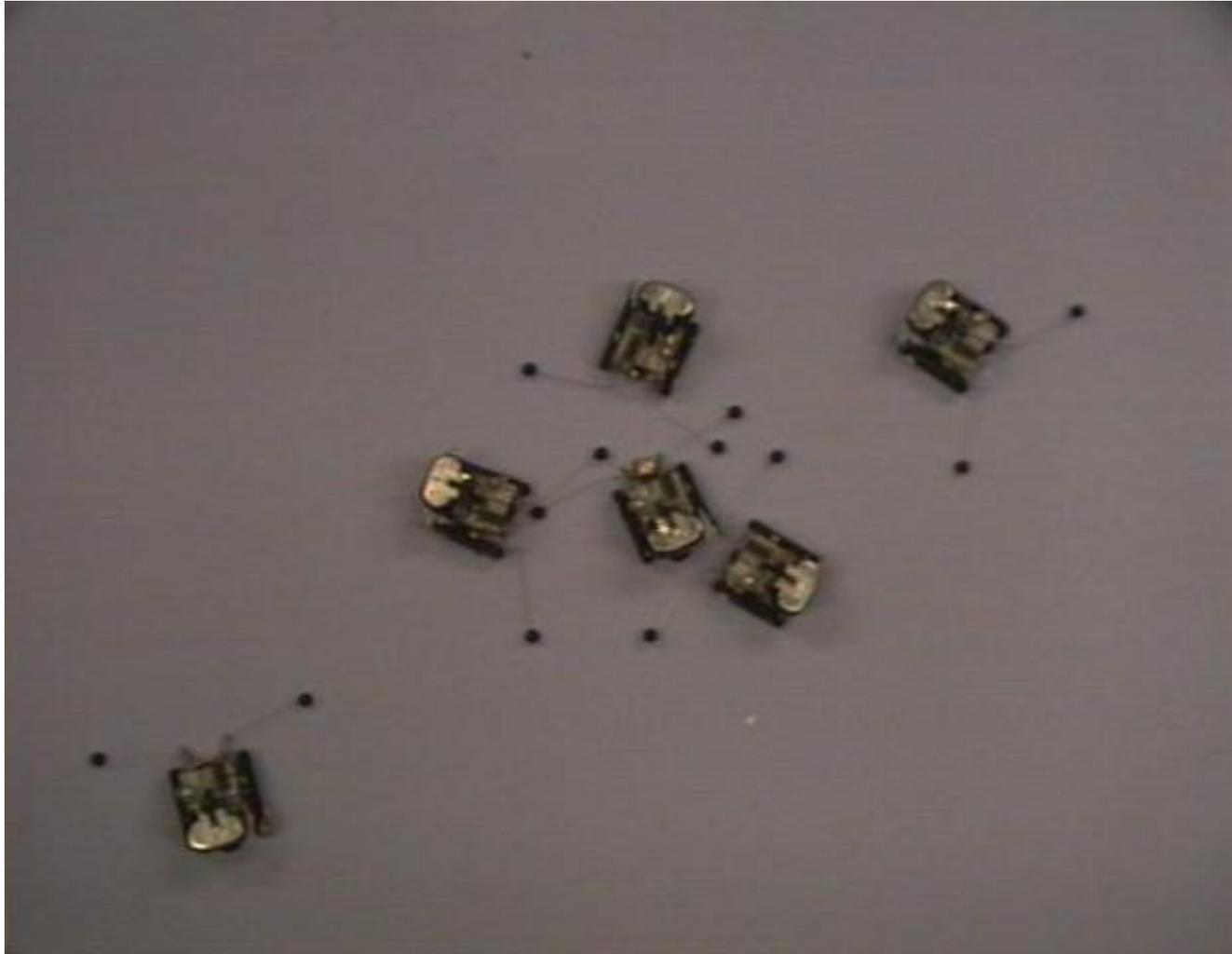


## **Recrutamento**

- Um dos problemas envolvidos no transporte coletivo de presas grandes é o recrutamento de formigas em torno da presa.
- Estudos teóricos sugerem dois tipos básicos de recrutamento baseado na emissão de substâncias químicas: recrutamento de curto alcance via liberação de químicos no ar (recrutamento de grupo) e recrutamento de longo alcance via trilhas de feromônio (recrutamento em massa).
- Inspirados nesses (e outros) comportamentos, pesquisadores do MIT AI Lab desenvolveram uma comunidade de micro-robôs para simular o que eles denominaram de AntFarm.
- Este projeto possuía dois objetivos principais:
  - Integrar diversos sensores e atuadores em micro-robôs; e

- Estudar o comportamento de uma comunidade robótica estruturada a partir das interações de vários robôs simples, cuja comunicação se dá através de sensores infravermelhos.
- Um dos comportamentos estudados foi o recrutamento de robôs em torno de regiões do espaço ricas em alimento.
- Os robôs eram inicialmente espalhados de forma aleatória em uma arena e um item de alimento colocado nesta arena.
- Três regras comportamentais simples foram definidas:
  - Assim que um robô detecta alimento ele emite um sinal infravermelho associado ao evento “Encontrei alimento”. Qualquer robô a um raio de 12” deste recrutador detectará o sinal;
  - Quando um robô detecta o sinal ele vai em direção ao chamado e propaga outro sinal associado ao evento “Recebi um chamado de alimento”;
  - Quando outro robô recebe a mensagem do segundo robô, ele recruta mais robôs.

- Vídeos disponíveis em:
  - <http://www.ai.mit.edu/projects/ants/>



## Escopo da Robótica Coletiva

- Diversos pesquisadores em robótica coletiva sugerem que suas aplicações em potencial requerem a miniaturização desses robôs.
- Micro- e nano-robôs que, por construção, possuem capacidades sensoriais e cognitivas restritas, deverão operar em grandes grupos na realização de tarefas.
- Xavier Défago, um pesquisador com intensa atuação em robótica coletiva, identifica quatro classes principais de aplicação para eles:
  - *Exploração*: trabalhos em ambientes hostis, inacessíveis ou de difícil comunicação (p. ex. superfície de outros planetas);
  - *Indústrias*: grupos ou times de robôs deverão trabalhar em linhas de montagem;
  - *Medicina e cirurgia*: existem diversas aplicações médicas que utilizam nano-robôs, como micro-cirurgia e controle, e local de aplicação de drogas dentro do organismo;
  - *Materiais inteligentes*: são materiais feitos a partir de diversos módulos pequenos cada qual formando uma grande estrutura capaz de alterar sua forma e propriedades físicas dinamicamente.
- Link de referência para pesquisas mais avançadas: <http://www.swarm-bots.org/>

## 5. Adaptação Social do Conhecimento

- A técnica de otimização baseada em partículas (*Particle Swarm Optimization* – PSO) possui como uma de suas principais motivações criar uma simulação do comportamento social humano, particularmente a capacidade humana de processar conhecimento (KENNEDY, 2005).
- Assim como todas as outras abordagens de inteligência de enxame, ela está baseada em uma população de indivíduos capazes de interagir entre si e com o meio ambiente.
- Comportamentos globais serão, portanto, resultados emergentes dessas interações.

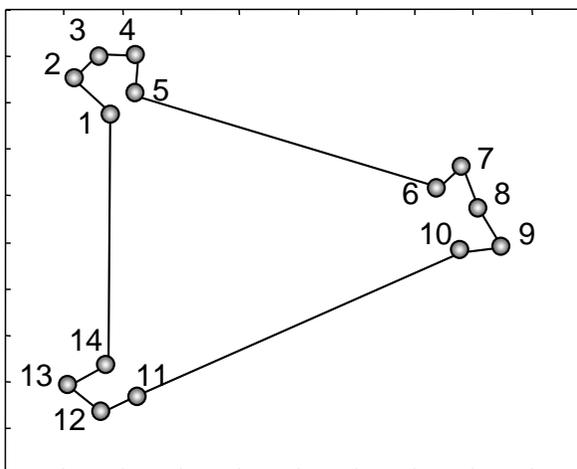
- Uma teoria sociocognitiva muito simples está por trás do PSO:
  - Cada indivíduo de uma população possui sua própria experiência e é capaz de avaliar a qualidade desta experiência; e
  - Como os indivíduos são sociais, eles também possuem conhecimentos sobre como seus vizinhos se comportaram (desempenharam).
- Estes dois tipos de informação correspondem à aprendizagem individual (cognitiva) e à transmissão cultural (social), respectivamente.
- Portanto, a probabilidade de um determinado indivíduo tomar uma certa decisão será uma função de seu desempenho no passado e do desempenho de alguns de seus vizinhos.
- KENNEDY *et al.* (2001) utilizaram três princípios para resumir o processo de adaptação cultural:
  - *Avalie*: os indivíduos possuem a capacidade de sentir o ambiente de forma a avaliar seu próprio comportamento;

- *Compare*: os indivíduos usam uns aos outros como material comparativo;
- *Imite*: a imitação é central em organizações sociais humanas e é importante para a aquisição e manutenção das habilidades mentais.

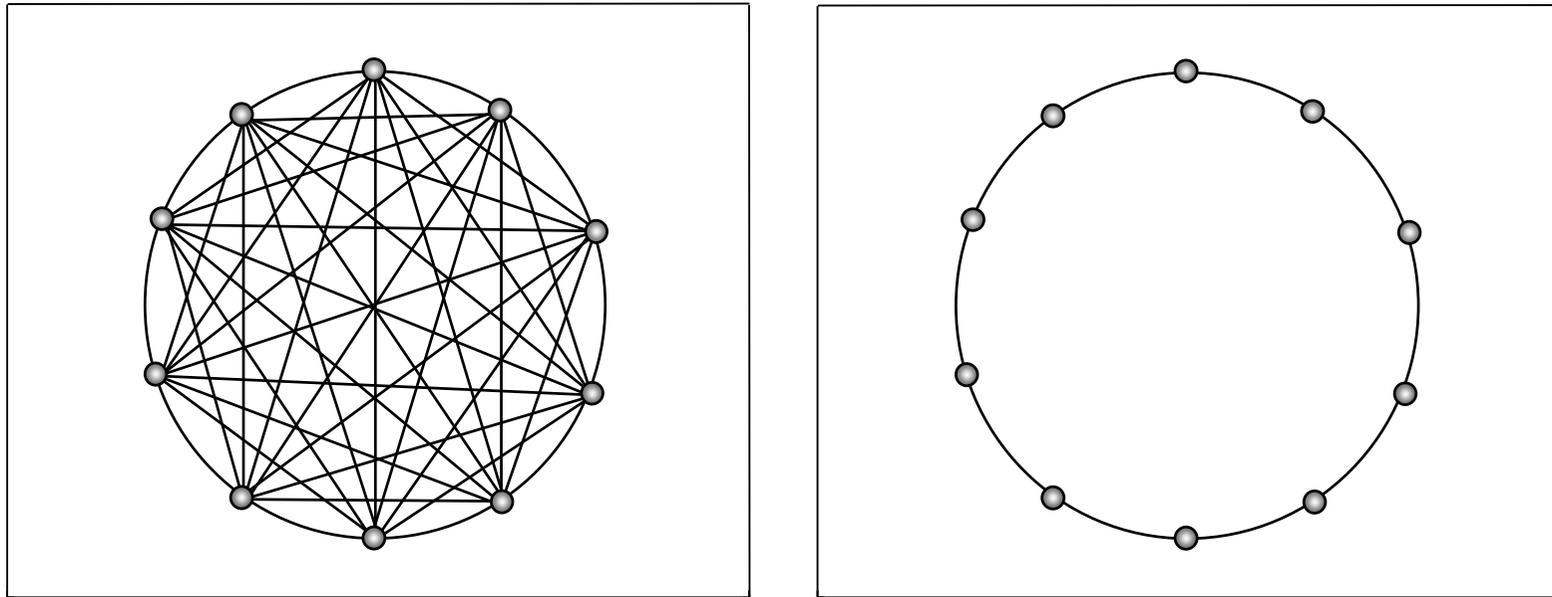
### **5.1. Algoritmo de Otimização por Enxame de Partículas**

- No algoritmo PSO, os indivíduos que são candidatos à solução de um determinado problema aprendem a partir de suas próprias experiências e das experiências de outros.
  - Eles se avaliam, comparam seus desempenhos com os de seus vizinhos e imitam somente aqueles que são melhores do que eles.
- Os indivíduos da população são representados por pontos, denominados de partículas, em um espaço  $\mathcal{R}^L$ .
- As variações nos atributos destes pontos levam a novos pontos no espaço, ou seja, correspondem a movimentações no espaço.

- Uma ideia inspirada em sistemas cognitivos é a de que estas partículas tenderão a se mover em direção umas das outras e irão influenciar umas às outras.
  - Em termos matemáticos, os principais componentes de um algoritmo PSO podem ser representados como a seguir:
    - A posição de uma partícula  $i$  é dada por  $\mathbf{x}_i$  ( $\mathbf{x}_i \in \mathfrak{R}^L$ );
    - Essa partícula irá se mover com uma velocidade vetorial  $\mathbf{v}_i$ ;
- $$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1). \quad (11)$$
- A inspiração tomada nas ciências sociais e na psicologia sugere que os indivíduos sejam influenciados por suas próprias experiências prévias e pela experiência de alguns de seus vizinhos.
  - Entretanto, a vizinhança aqui corresponde à vizinhança topológica e não à vizinhança no espaço de atributos de cada indivíduo (partícula).
  - Sendo assim, a vizinhança de cada indivíduo é definida a partir de um arranjo topológico.



- Existem diversas formas de se definir a vizinhança de um indivíduo (KENNEDY, 1999).
- A maior parte dos algoritmos de PSO emprega dois princípios sociométricos:
  - O primeiro, denominado de *gbest* ( $g$  = global), conecta conceitualmente todos os membros de uma população entre si. Como consequência, o comportamento de cada partícula é influenciado pelo comportamento de todas as outras partículas.
  - O segundo, denominado de *lbest* ( $l$  = local), cria uma vizinhança para cada indivíduo composta por ele próprio e seus  $k$  vizinhos mais próximos.



Dois exemplos de vizinhança para PSO (à esquerda: global; à direita: em anel)

- Uma partícula irá se mover em uma determinada direção que é função da posição atual da partícula  $\mathbf{x}_i(t)$ , de uma velocidade  $\mathbf{v}_i(t)$ , da posição  $\mathbf{p}_i$  que levou ao melhor desempenho da partícula até o momento, e da posição  $\mathbf{p}_g$  do melhor desempenho de sua vizinhança até o momento:

$$\mathbf{x}_i(t+1) = f(\mathbf{x}_i(t), \mathbf{v}_i(t), \mathbf{p}_i, \mathbf{p}_g).$$

- A velocidade da partícula será dada por:

$$\mathbf{v}_i(t+1) = \omega \mathbf{v}_i(t) + \varphi_1(\mathbf{p}_i - \mathbf{x}_i(t)) + \varphi_2(\mathbf{p}_g - \mathbf{x}_i(t)), \quad (12)$$

onde  $\omega$  (fator de inércia),  $\varphi_1$  (fator cognitivo) e  $\varphi_2$  (fator social) são constantes limitadas a um intervalo finito, geralmente  $[0,+1]$ .

Obs.: Primeiro atualiza-se a velocidade e depois a posição:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1).$$

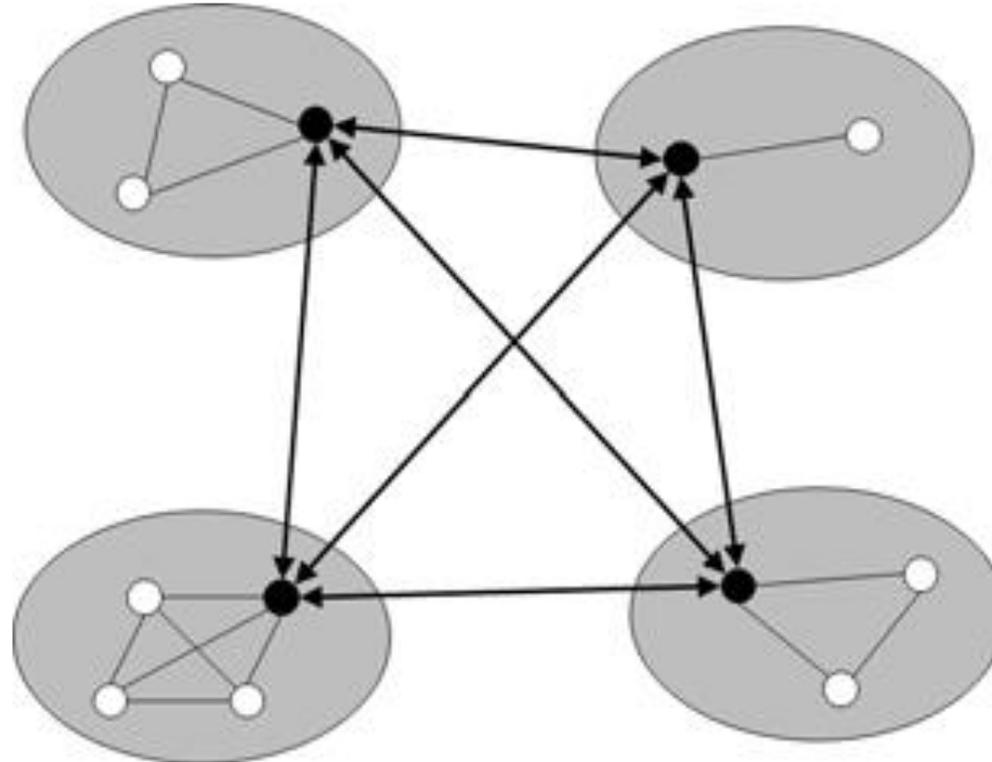
- Em KENNEDY (1997), o autor denomina o penúltimo termo da Equação (12) como sendo o termo “cognitivo” e o último como sendo o termo “social”.
- Para limitar a velocidade de uma partícula de forma que o sistema não se instabilize, são impostos limites para seus valores:

If  $v_{id} > v_{\max}$ , then  $v_{id} = v_{\max}$ ,

Else if  $v_{id} < -v_{\max}$  then  $v_{id} = -v_{\max}$

**(Versão para Maximização)**

```
procedure [X] = PSO(max_it, ω, φ1, φ2, Vmax)
  initialize X //usually every particle xi is initialized at random
  initialize vi //at random, vi ∈ [-Vmax, Vmax]
  t ← 1
  while t < max_it do,
    for i = 1 to N do,           //for each particle
      if f(xi) > f(pi),
        then pi = xi,           //best individual performance so far
      end if
      for j = indexes of neighbors
        if f(pj) > f(pg),
          then g = j,           // best neighbor performance so far
        end if
      end for
      vi(t+1) = ωvi(t) + φ1(pi - xi(t)) + φ2(pg - xi(t))
      vi ∈ [-Vmax, Vmax]
      xi(t+1) = xi(t) + vi(t+1)
    end for
    t ← t + 1
  end while
end procedure
```



Mais um exemplo de vizinhança para PSO, desta vez dinâmica: TRIBES

## 5.2. Escopo de Aplicação

- As primeiras aplicações do algoritmo PSO foram na determinação de pesos e arquitetura de redes neurais artificiais.
- Esta técnica tem sido aplicada a problemas de otimização em espaços contínuos em geral (POLI *et al.*, 2007), incluindo otimização dinâmica e multi-objetivo.

T. Blackwell and J. Branke.

Multi-swarm optimization in dynamic environments.

*Lecture Notes in Computer Science*, 3005:489–500, 2004.

M. Reyes-Sierra and C. A. C. Coello.

Multi-objective particle swarm optimizers: A survey of the state-of-the-art.

*International Journal of Computational Intelligence Research*  
2(3):287–308, 2006.

### 5.3. De Sistemas Sociais a Enxames de Partículas

<b>Sociocognição</b>	<b>Algoritmo PSO</b>
Indivíduo	Partícula
População de indivíduos	Enxame de partículas
Esquecimento e aprendizagem	Incremento ou decremento nos valores de alguns atributos das partículas
Experiência própria de um indivíduo	Cada partícula possui algum conhecimento de sua história (desempenho) e emprega este conhecimento para direcionar seus próximos movimentos no espaço
Interações sociais	Cada partícula também possui conhecimento sobre a vida (desempenho) de outras partículas e emprega este conhecimento para direcionar seus próximos movimentos no espaço

## 6. Referências bibliográficas

- BECKERS, R.; HOLLAND, O.E.; DENEUBOURG, J.-L. “From Local Actions to Global Tasks: Stigmergy and Collective Robotics”, in Brooks, R.A. and Maes, P. (eds.) Proceedings of the 4th International Workshop on the Synthesis and Simulation of Life, Artificial Life IV, MIT Press, pp. 181-189, 1994.
- BONABEAU, E.; THERAULAZ, G.; DORIGO, M. “Swarm Intelligence: From Natural to Artificial Systems”, Oxford University Press, 1999.
- CAMAZINE, S.; DENEUBOURG, J.-L.; FRANKS, N.R.; SNEYD, J.; THÉRAULAZ, G.; BONABEAU, E. “Self-Organization in Biological Systems”, Princeton University Press, 2001.
- DE CASTRO, L.N. “Fundamentals of Natural Computing”, Chapter 5 – Swarm Intelligence, Chapman & Hall / CRC, 2006.
- DENEUBOURG, J.-L.; GOSS, S.; FRANKS, N.; SENDOVA-FRANKS, A.; DETRAIN, C.; CHRÉTIEN, L. “The Dynamics of Collective Sorting: Robot-like Ant and Ant-like Robot”, in Meyer, J.A. and Wilson, S.W. (eds.) Simulation of Adaptive Behavior: From Animals to Animats, pp. 356-365, MIT Press / Bradford Books, 1991.

- DORIGO, M. “Optimization, Learning and Natural Algorithms”, Ph.D. Thesis, Dipartimento di Elettronica, Politecnico di Milano, 1992. (in Italian)
- DORIGO, M.; BIRATTARI, M.; STÜTZLE, T. “Ant Colony Optimization – Artificial Ants as a Computational Intelligence Technique”, IEEE Comp. Intel. Magazine, pp. 28-39, 2006.
- DORIGO, M.; GAMBARDELLA, L.M. “Ant Colonies for the Traveling Salesman Problem”, BioSystems, vol. 43, pp. 73-81, 1997.
- DORIGO, M.; MANIEZZO, V.; COLORNI, A. “Positive Feedback as a Search Strategy”, Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, 1991.
- DORIGO, M.; MANIEZZO, V.; COLORNI, A. “The Ant System: Optimization by a Colony of Cooperating Agents”, IEEE Transactions on Systems, Man, and Cybernetics – Part B, vol. 26, no. 1, pp. 29-41, 1996.
- DORIGO, M.; STÜTZLE, T. “Ant Colony Optimization”, MIT Press, 2004.
- FAQ Santa Fe (Swarm\_Development\_Group): [<http://www.swarm.org/>]
- KENNEDY, J. “Particle Swarms: Optimization Based on Sociocognition”, in de Castro, L.N. and Von Zuben, F.J. (eds.) Recent Developments in Biologically Inspired Computing, Idea Group Publishing, Chapter X, pp. 235-268, 2005.

KENNEDY, J. “Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance”, Proceedings of the Congress on Evolutionary Computation, vol. 3, pp. 1931-1938, 1999.

KENNEDY, J. “The Particle Swarm: Social Adaptation of Knowledge”, Proceedings of the IEEE International Conference on Evolutionary Computation (CEC’1997), pp. 303-308, 1997.

KENNEDY, J.; EBERHART, R.; SHI, Y. “Swarm Intelligence”, Morgan Kaufmann Publishers, 2001.

KRIEGER, M.J.B.; BILLETER, J.-B.; KELLER, L. “Ant-Task Allocation and Recruitment in Cooperative Robots”, Nature, vol. 406, no. 31, pp. 992-995, 2000.

K-TEAM WEBSITE: <http://www.k-team.com/>

KUBE, C.R.; PARKER, C.A.C.; WANG, T.; ZHANG, H. “Biologically Inspired Collective Robotics”, in de Castro, L.N. and Von Zuben, F.J. (eds.) Recent Developments in Biologically Inspired Computing, Idea Group Publishing, Chapter XV, pp. 367-397, 2005.

KUBE, C.R.; ZHANG, H. “Collective Robotic Intelligence”, Simulation of Adaptive Behavior: From Animals to Animats 2, pp. 460-468, MIT Press, 1992.

- LUMER, E.D.; FAIETA, B. “Diversity and Adaptation in Populations of Clustering Ants”, in Cliff, D., Husbands, P., Meyer, J.A. and Wilson, S.W. (eds.) *Simulation of Adaptive Behavior: From Animals to Animats 3*, pp. 499-508, MIT Press, 1994.
- MARTINOLI, A. “Collective Complexity out of Individual Simplicity”, *Artificial Life*, vol. 7, no. 3, pp. 315-319, 2001.
- POLI, R.; KENNEDY, J.; BLACKWELL, T. “Particle swarm optimization”, *Swarm Intelligence*, vol. 1, no. 1, pp. 33-57, 2007.
- SHERAFAT, V.; DE CASTRO, L.N.; HRUSCHKA, E.R. “The Influence of Pheromone and Adaptive Vision in the Standard Ant Clustering Algorithm”, in de Castro, L.N. and Von Zuben, F.J. (eds.) *Recent Developments in Biologically Inspired Computing*, Idea Group Publishing, Chapter IX, pp. 207-234, 2005.
- VIZINE, A.L.; DE CASTRO, L.N.; HRUSCHKA, E.R.; GUDWIN, R.R. “Towards Improving Clustering Ants: An Adaptive Ant Clustering Algorithm”, *Informatica*, vol. 29, pp. 143-154, 2005.
- WHITE, T.; PAGUREK, B. “Towards Multi-Swarm Problem Solving in Networks” *Proceedings of the Third Int. Conf. on Multi-Agent Systems (ICMAS’1998)*, pp. 333-340, 1998.