

Inteligência Coletiva

Profa. Dra. Gina Maira B. de Oliveira



Universidade Federal de Uberlândia

Inteligência Coletiva



Inteligência Coletiva

Na natureza, as espécies se beneficiam por se organizarem em sociedades: aumenta a probabilidade de acasalamento, facilita a caça e coleta de alimentos, reduz a probabilidade de ataque por predadores, etc.

O comportamento coletivo de diversas espécies tem inspirado a criação de novos modelos computacionais baseados em sistemas auto-organizados, distribuídos, autônomos, flexíveis e dinâmicos .

Inteligência Coletiva

O termo *swarm intelligence*, ou inteligência de enxame, foi proposto no fim da década de 1980, para se referir a sistemas robóticos compostos por uma coleção de agentes simples em um ambiente interagindo de acordo com regras locais.

Inteligência Coletiva se refere ao estudo de algoritmos ou sistemas distribuídos de solução de problemas inspirados no comportamento coletivo de insetos sociais e outras sociedades animais.

Inteligência Coletiva

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).

Inteligência Coletiva

As propriedades principais de um sistema de inteligência de enxame são (Millonas, 1994):

- **Proximidade:** agentes devem ser capazes de interagir;
- **Qualidade:** 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 adequação a variações ambientais.

Inteligência Coletiva

- **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.**

Inteligência Coletiva

Duas principais linhas de pesquisa que emergem dessas propriedades podem ser observadas na inteligência de enxames:

- **Sistemas inspirados no estudo do comportamento de insetos sociais, como formigas, abelhas, cupins e vespas.**
- **Sistemas inspirados na habilidade das sociedades humanas em processar conhecimento.**

Inteligência Coletiva

As técnicas mais conhecidos são: a otimização por colônia de formigas (*Ant Colony Optimization* – ACO) e a otimização por enxame de partículas (*Particle Swarm Optimization* – PSO).

Outros algoritmos: algoritmos de colônia de abelhas, algoritmo de colônia de vagalumes; algoritmo shuffled frog-leaping, algoritmos de coleta de alimentos por bactérias, algoritmos baseados em cardumes de peixes, etc.

Otimização por colônia de formigas (ACO)



Inspiração: colônia de formigas

- Algoritmos de formigas: propostos inicialmente por Marco Dorigo para solução de problemas de otimização combinatória.
- Inspirado na observação de formigas reais.
- **Formiga: inseto social, ou seja, vive em colônias e tem comportamento direcionado para a sobrevivência da colônia ao invés de um único indivíduo.**
- Formigas, que são quase cegas, conseguem achar o caminho mais curto entre sua colônia a uma fonte de alimento. Como?

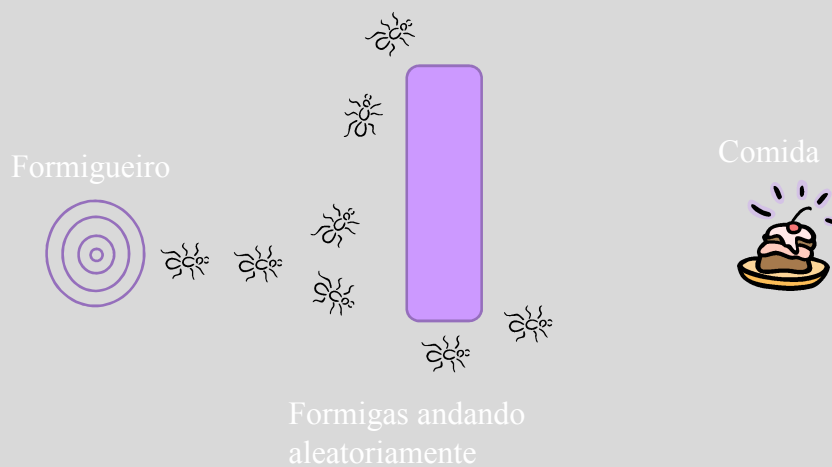
Inspiração: colônia de formigas

- **Formigas se comunicam através de marcas químicas deixadas no chão chamadas feromônio.**
- **Ao passar por um local uma formiga deixa uma certa quantidade de feromônio no chão formando assim trilhas.**
- **Outras formigas podem detectar tais trilhas com feromônio, e tendem a escolher seu caminho por trilhas com mais feromônio.**

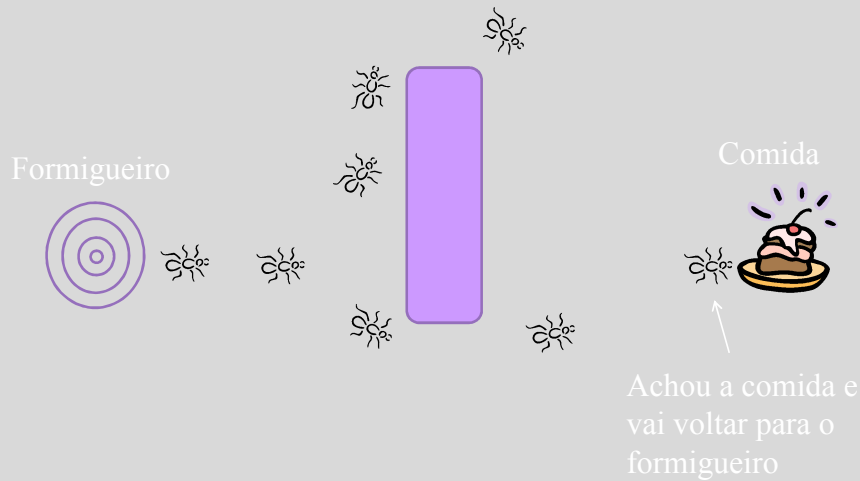
Inspiração: colônia de formigas

- Quando mais formigas passarem por uma trilha, mais feromônio esta terá, formando assim uma realimentação positiva.
- Este comportamento simples de seguir trilhas faz emergir um comportamento mais complexo de encontrar trilhas mais curtas entre dois pontos.

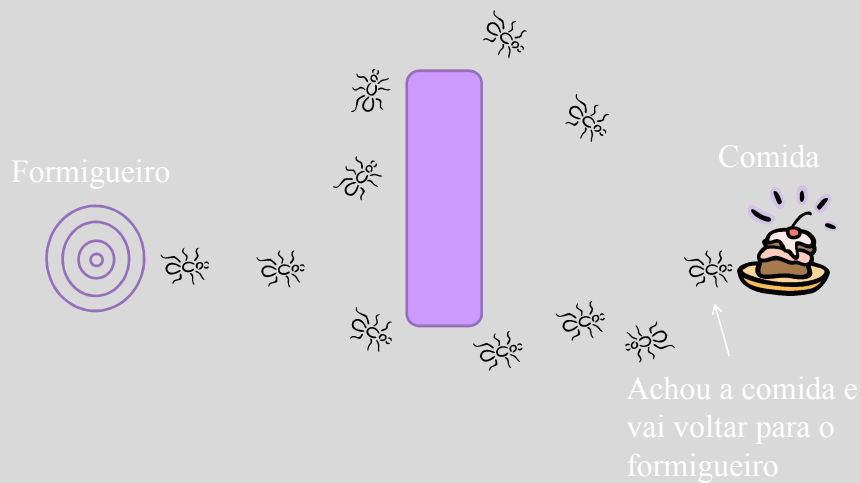
Inspiração: colônia de formigas



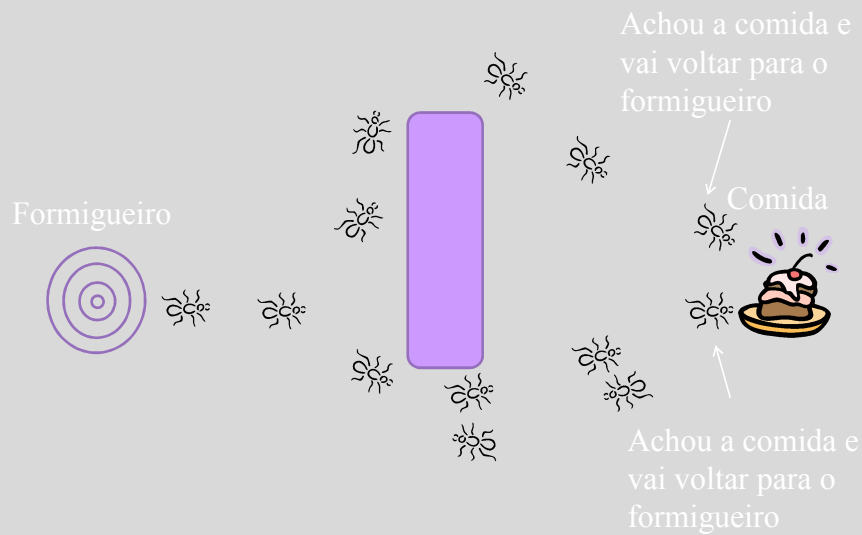
Inspiração: colônia de formigas



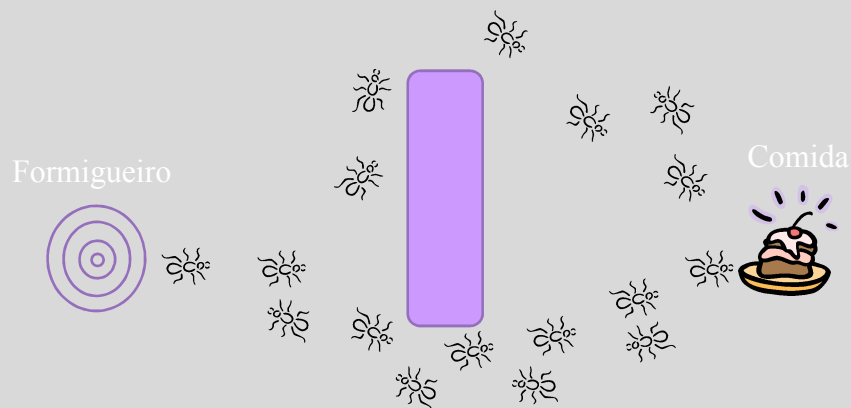
Inspiração: colônia de formigas



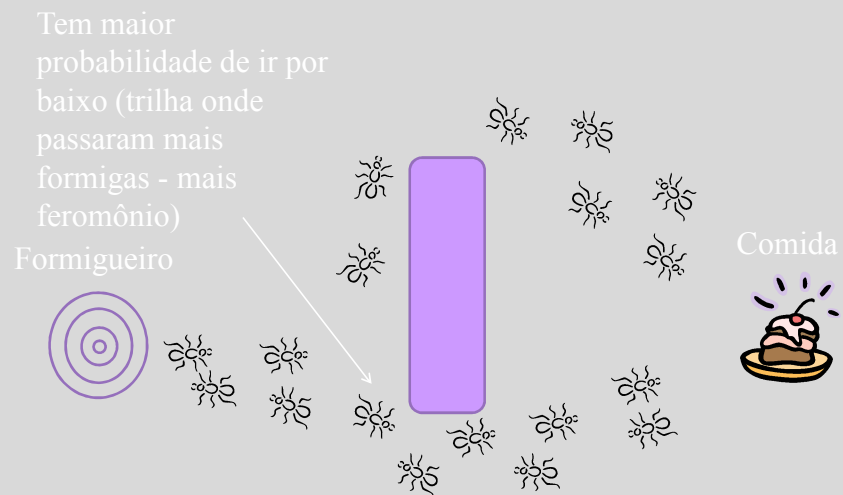
Inspiração: colônia de formigas



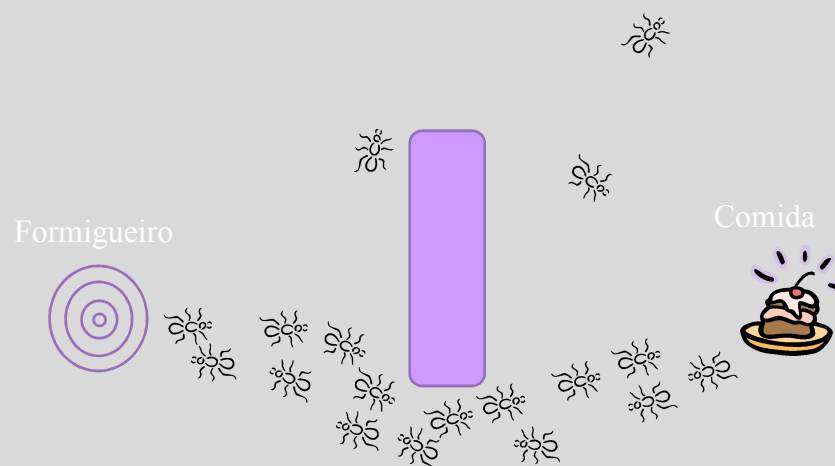
Inspiração: colônia de formigas



Inspiração: colônia de formigas

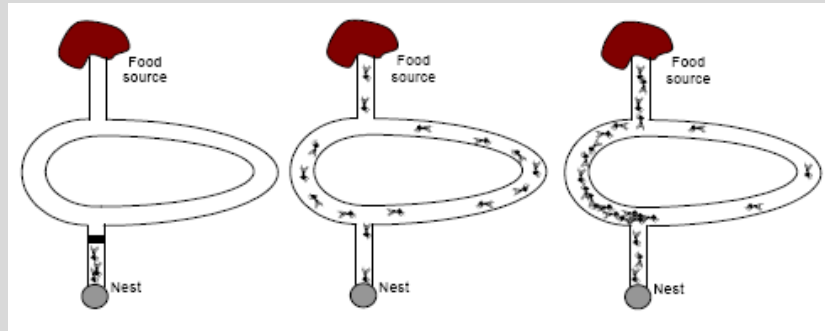


Inspiração: colônia de formigas



Inspiração: colônia de formigas

Exemplo de experimento realizado com formigas para avaliar a importância da trilha de feromônio na coleta de alimentos:



Inspiração: colônia de formigas

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;
- 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. As formigas não seguem as trilhas perfeitamente, elas possuem uma determinada probabilidade de se perderem da trilha: comportamento importante para que seja possível a descoberta de outras fontes de alimento.

Otimização por colônia de formigas (ACO)

O termo algoritmo de otimização por colônia de formigas (Ant Colony Optimization – ACO) é um termo genérico que designa um procedimento geral de uma classe de metaheurísticas baseadas no comportamento de formigas.

Ant System (AS) foi o primeiro algoritmo baseado no comportamento de formigas desenvolvido por Dorigo et al. (1991).

Otimização por colônia de formigas (ACO)

Inicialmente três diferentes variantes foram propostas: AS-density, AS-quantity e AS-cycle, diferindo-se pela maneira na qual as trilhas de feromônio eram atualizadas (Dorigo et al., 1996).

A última variante mostrou-se mais eficiente e a maior parte dos algoritmos de otimização por colônia de formigas atuais derivam-se dela.

Otimização por colônia de formigas (ACO)

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 constrói/modifica N soluções paralelas;
- Uma regra de atualização de feromônio a partir da qual a quantidade de feromônio nas arestas é alterada.

Otimização por colônia de formigas (ACO)

O processo de construir ou modificar uma solução (caminho) é feito de forma probabilística.

A probabilidade de uma nova aresta ser adicionada à solução sendo construída é função de uma qualidade heurística η (heuristic desirability ou visibilidade) e da quantidade de feromônio τ depositada pelas N formigas.

A visibilidade η visa expressar algum atributo (ou conjunto de atributos) que se quer ver presente na solução completa.

Otimização por colônia de formigas (ACO)

Exemplo: quando o caminho está sendo construído para solução do Prob. C. Viajante, η 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 ρ e a qualidade das soluções produzidas.

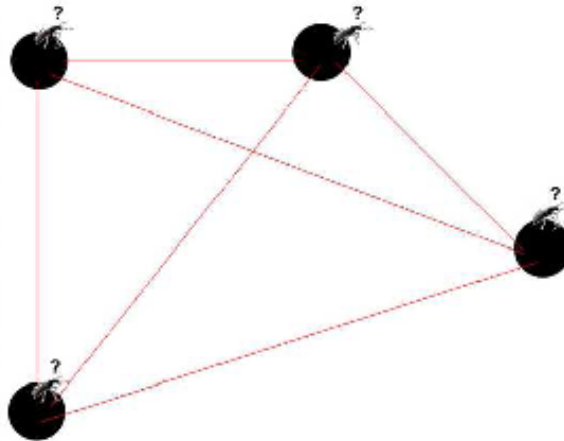
Otimização por colônia de formigas (ACO)

O modelo matemático do ACO é dado a partir de um grafo G com V vértices representando um problema.

Uma formiga artificial é colocada em cada um dos vértices.

Otimização por colônia de formigas (ACO)

- Dado um grafo com n vértices, colocar uma formiga artificial em cada um destes



Otimização por colônia de formigas (ACO)

- Soluções parciais do problema são chamadas de estados.
- Cada formiga muda de um estado anterior para o próximo, que corresponde a uma solução parcial mais completa, com o objetivo de chegar ao estado final que é a solução total do problema.
- Em cada passo da construção da solução a formiga visualiza o conjunto de expansões possíveis para a solução atual, isto é, identifica o conj. de estados viáveis para o qual se pode passar a partir do atual.

Otimização por colônia de formigas (ACO)

Cada formiga percorre um caminho seguindo uma fórmula probabilística em função do feromônio “depositado” em cada aresta do grafo para chegar ao destino.

Para construir a solução cada formiga (k) utiliza iterativamente uma regra de transição de estado ($p_{i,j}$) (função probabilística) para decidir se incluirá ou não determinada aresta na solução:

Otimização por colônia de formigas (ACO)

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

onde:

τ_{ij} é a trilha de feromônio da aresta (i, j),

η_{ij} é a heurística local (visibilidade) da aresta (i, j),

α é a importância relativa da trilha de feromônio,

β é a importância relativa da heurística local,

J_i^k conjunto de estados acessíveis para a formiga k a partir do estado atual i

Otimização por colônia de formigas (ACO)

Cada formiga percorre um caminho seguindo essa probabilidade de transição em função do feromônio depositado em cada aresta do grafo e da visibilidade dessas arestas.

Após a construção de todos os caminhos a intensidade de feromônio em cada aresta (trilha) é acrescida de forma proporcional à qualidade da solução gerada.

Otimização por colônia de formigas (ACO)

A liberação de feromônio nas arestas é proporcional à qualidade do percurso (solução total).

Neste caso, a quantidade de feromônio liberada em cada aresta (i,j) pela formiga k , dado por $\Delta\tau_{ij}^k$, depende do desempenho total do percurso T^k :

$$\Delta\tau_{ij}^k(t) = \begin{cases} f(T^k(t)) & \text{se } (i,j) \in T^k(t) \\ 0 & \text{caso contrário} \end{cases}$$

Onde $f(T^k(t))$ é uma medida da qualidade da rota $T^k(t)$ percorrida pela formiga k na iteração t .

Otimização por colônia de formigas (ACO)

Para evitar uma convergência prematura do algoritmo, foi inserido um termo associado à evaporação do feromônio.

Considerando-se as N formigas, resulta então a fórmula geral de atualização de feromônio:

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

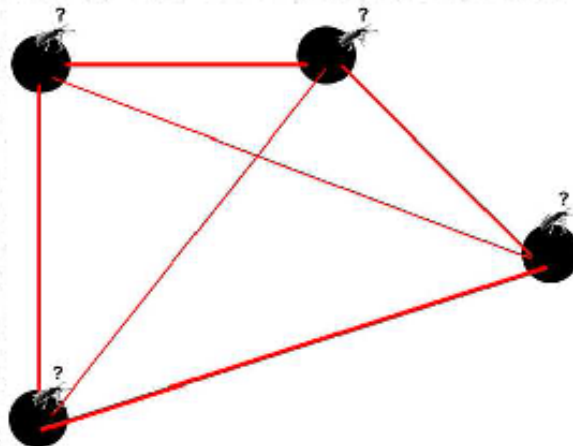
onde $\rho \in [0,1)$ é a taxa de decaimento de feromônio,

$$\Delta\tau_{ij}(t) = \sum_k \Delta\tau_{ij}^k(t),$$

e $k = 1, \dots, N$ é o índice das formigas.

Otimização por colônia de formigas (ACO)

- Após a construção de todos os caminhos a intensidade de feromônio em cada aresta é acrescida *proporcionalmente à qualidade da solução gerada*:

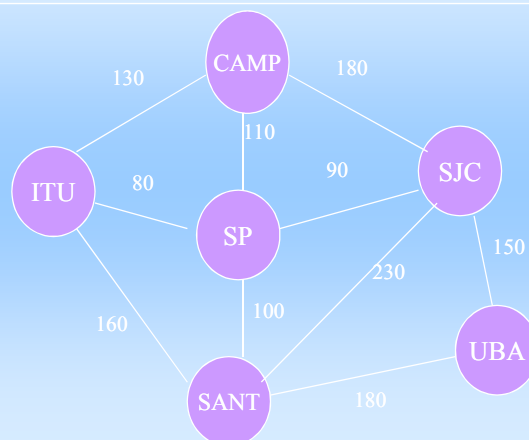


Procedimento Geral de um ACO

1. Inicialize os parâmetros k , ρ , α , β e as trilhas de feromônio τ_{ij} com o mesmo valor inicial τ_0 .
2. Coloque cada formiga k em uma aresta aleatoriamente selecionada do grafo $G(V,E)$, onde V é o conjunto de vértices e E é o conjunto de arestas de G .
3. Para cada formiga k , em cada aresta até o destino, construa soluções baseadas na *regra de transição de estado* ($p_{i,j}^k$).
4. Avalie o custo de todas as soluções (calcule $f(S)$).
5. Guarde a melhor solução até o momento.
6. Para cada aresta do grafo, aplique a regra de atualização da trilha de feromônio (calcule $\Delta\tau_{i,j}$ e $\tau_{i,j}$).
7. Se condição de término não for alcançada, retorne ao passo 2.

Exemplo de Aplicação

Problema do Carteiro Viajante



Otimização por colônia de formigas (ACO)

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

J_i^k : conjunto de cidades conectadas à cidade i e ainda não visitadas pela formiga k na solução parcial;

$\eta_{i,j}$: inversamente proporcional ao comprimento da aresta (i,j)

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

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.

Otimização por colônia de formigas (ACO)

O algoritmo ACO original para solução de problemas combinatoriais é conhecido como Ant System (AS) (Dorigo et al., 1991).

Desde então vários outros algoritmos ACO foram introduzidos compartilhando dos mesmos princípios.

O algoritmo ANT-Q (Gambardella & Dorigo, 1995) é uma extensão do AS. No ANT-Q a atualização da trilha de feromônio usando-se um valor que é uma previsão do valor do próximo estado. Mostrou desempenho equivalente ao AS (mas é mais complexo).

Otimização por colônia de formigas (ACO)

Ant Colony System (ACS) (Gambardella & Dorigo, 1996): atualização do feromônio local (atualização online) além da atualização do feromônio realizada no fim do processo de construção do caminho (atualização offline).

AS Elitista (Dorigo et al., 1996): apenas a melhor solução gerada até o momento tem a trilha de feromônio atualizada.

MAX-MIN AS (MMAS) (Stutzle & Hoos, 1996): apenas a melhor formiga a cada passo atualiza as trilhas de feromônio e a quantidade do feromônio é limitada por limites mínimo e máximo.

Otimização por colônia de formigas (ACO)

Rank-Based AS (Bullheimer et al., 1997): feromônio atualizado de acordo com uma função de classificação ponderada do desempenho de cada formiga.

Approximate Nondeterministic Tree-Search (ANTS) (Maniezzo, 1999) modificou o modo de cálculo da atualização de feromônio e combinou ACO com técnicas de limites inferiores de programação matemática (similar em estrutura aos algoritmos de busca em árvore)

Hyper-Cube AS (Blum et al., 2001) atualização do feromônio a partir de uma regra que define o peso de cada solução, escalonando os valores da função objetivo.

Otimização por colônia de formigas (ACO)

Best Worst Ant System (BWAS) (Cordón et al., 2000): incorpora conceitos de computação evolutiva (Goldberg, 1989). O algoritmo reforça as trilhas da melhor solução global e penaliza cada trilha da pior solução gerada na iteração atual. Uma mutação na trilha de feromônio é introduzida para gerar diversidade no processo de busca. Além disso, esse modelo também considera a reinicialização da trilha de feromônio quando essa fica estagnada.

Otimização por colônia de formigas (ACO)

Os algoritmos ACOs são bem adequados para problemas NP difíceis e de otimização estocástica em geral, tais como:

- caminho mínimo (Dorigo & Gambardella, 1997),
- roteamento de veículos (Aloise et al., 2002; Bell & McMullen, 2004; Lopes et al., 2007)
- roteamento de redes (Schoonderwoerd et al., 1996)
- escalonamento (Merkle & Middendorf, 2003),
- coloração de grafos (Costa & Hertz, 1997),
- otimização multiobjetivo (Chaharsooghi & Kermani, 2008).

Otimização por colônia de formigas (ACO)

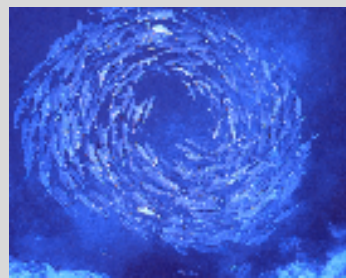
Diversas aplicações, destacando-se o ajuste de parâmetros de controladores PID (Tan et al., 2005), problemas de manufatura (Solimanpur et al., 2004), de linhas de montagem (Ying & Liao, 2004) e de layout industrial (Hani et al., 2007), processamento de imagens (Nezamabadipour et al., 2006), roteamento em redes de telecomunicações (Di Caro & Dorigo, 1998), redes de sensores sem fio (Silva et al., 2007), despacho econômico de carga (Coelho et al., 2005, 2008), planejamento de rotas (Tavares Neto & Coelho, 2003, 2004, 2005), dentre vários outros. Áreas como bioinformática (Chan & Freitas, 2006) e composição de músicas (Geis & Middendorf, 2007) também têm sido exploradas com as técnicas de ACO.

Otimização baseada em Enxame de Partículas (PSO)



Inspiração: comportamento social

O algoritmo de otimização por enxame de partículas (*Particle Swarm Optimization – PSO*) emergiu de experiências com algoritmos que modelam o “comportamento social” observado em muitas espécies de pássaros e cardumes de peixes, e até mesmo do comportamento social humano.



Inspiração: comportamento social

Uma teoria sócio-cognitiva muito simples é subjacente à PSO:

- Cada indivíduo de uma população possui sua própria experiência e é capaz de estimar a qualidade dessa experiência.
- Como os indivíduos são sociais, eles também possuem conhecimentos sobre como seus vizinhos comportam-se.

Esses dois tipos de informação correspondem à aprendizagem individual (cognitiva) e à transmissão cultural (social), respectivamente.

Inspiração: comportamento social

A probabilidade de que um determinado indivíduo tome uma certa decisão será uma função de seu desempenho no passado e do desempenho de alguns de seus vizinhos.

Inspiração: comportamento social

Kennedy et al. (2001) utilizaram três princípios para resumir o processo de adaptação cultural:

- Avaliar – os indivíduos possuem a capacidade de sentir o ambiente de forma a estimar seu próprio comportamento;**
- Comparar – os indivíduos usam uns aos outros como referência comparativa;**
- Imitar – a imitação é central em organizações sociais humanas e é importante para a aquisição e manutenção das habilidades mentais.-**

Inspiração: comportamento social

Assim como as outras abordagens de inteligência coletiva, PSO está baseado em uma população de indivíduos capazes de interagir entre si e com o meio ambiente.

Com base nas propriedades de autoavaliação, comparação e imitação, os indivíduos são capazes de lidar com um número de possíveis situações que o ambiente lhes apresenta.

Os comportamentos globais serão, portanto, resultados emergentes dessas interações.

Otimização baseada em Enxame de Partículas (PSO)

O algoritmo de otimização por enxame de partículas (PSO) foi introduzido por Kennedy e Elberhart em 1995 para tratar problemas no domínio contínuo.

Técnica inspirada no comportamento social de revoadas de pássaros. Os resultados obtidos com a simulação das revoadas originou o PSO.



Otimização baseada em Enxame de Partículas (PSO)

Os indivíduos da população são representados por pontos, denominados de partículas, que voam em um espaço de busca \mathcal{R}^d (d : dimensão do espaço).

Os indivíduos que são candidatos à solução de um determinado problema aprendem a partir de suas próprias experiências e de outros.

Eles se avaliam, comparam seus desempenhos com os de seus vizinhos e imitam somente aqueles que são melhores do que eles.

Otimização baseada em Enxame de Partículas (PSO)

As variações nos atributos desses pontos levam a novos pontos no espaço: correspondem a movimentações.

Uma ideia inspirada em sistemas cognitivos é a de que essas partículas tenderão a mover-se em direção umas às outras e irão influenciar umas às outras.

O processo de atualização de uma partícula possui dois componentes:

- Melhor experiência (posição visitada) pessoal
- Melhor experiência (posição visitada) do grupo

Otimização baseada em Enxame de Partículas (PSO)

A maior parte dos algoritmos de PSO empregam dois princípios sóciométricos: g_{Best} e p_{Best}

g_{Best} : 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.

p_{Best} : cria uma vizinhança para cada indivíduo composta por ele próprio e seus vizinhos mais próximos.

Ambas as métricas são medidas por uma função de avaliação, também chamada função objetivo ou de aptidão (*fitness*).

Otimização baseada em Enxame de Partículas (PSO)

Uma partícula p_i irá se mover em uma determinada direção que é função da posição atual da partícula $x_i(t)$, de uma velocidade $v_i(t+1)$, da posição da partícula que levou ao seu melhor desempenho até o momento (p_{Best}), e do melhor desempenho global do sistema até o momento (g_{Best}).

A velocidade da partícula será dada por:

$$v_i(t+1) = v_i(t) + \varphi_1 \times (p_B - x_i(t)) + \varphi_2 \times (g_B - x_i(t))$$

onde: φ_1 e φ_2 são constantes limitadas a um intervalo finito, e denominados como sendo respectivamente os componentes “cognitivo” e “social”.

Otimização baseada em Enxame de Partículas (PSO)

Para limitar a velocidade de uma partícula para que o sistema não extrapole o espaço de busca, são impostos limites $(-v_{max}, +v_{max})$ para seus valores em cada dimensão (d) do espaço de busca.

Uma vez que a velocidade da partícula é calculada, a posição da partícula p_i na próxima iteração é estabelecida como uma influência aditiva da posição antiga e da velocidade calculada, sendo expressa por:

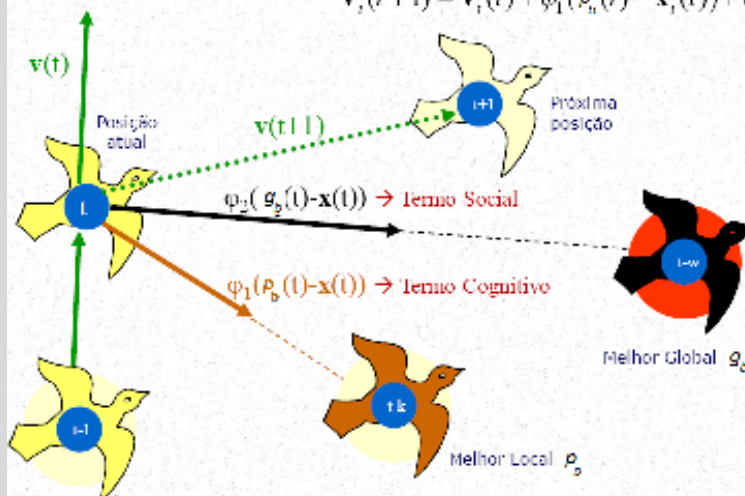
$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Otimização baseada em Enxame de Partículas (PSO)

Deslocamento de uma partícula!

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

$$v_i(t+1) = v_i(t) + \phi_1(p_b(t) - x_i(t)) + \phi_2(g_b(t) - x_i(t))$$



Otimização baseada em Enxame de Partículas (PSO)

O algoritmo PSO é repetido até que um critério de terminação é atingido ou as mudanças nas velocidades das partículas estejam perto de zero.

Otimização baseada em Enxame de Partículas (PSO)

1. Determine o número de partículas P da população.
2. Inicialize aleatoriamente a posição inicial (x) de cada partícula p de P .
3. Atribua uma velocidade inicial (v) igual para todas as partículas.
4. Para cada partícula p em P faça:
 - (a) Calcule sua aptidão $f(p)$.
 - (b) Calcule e melhor posição da partícula p até o momento (p_{Best}).
5. Descubra a partícula com melhor aptidão de toda a população (g_{Best}).
6. Para cada partícula p em P faça:
 - (a) Atualize a velocidade da partícula pela fórmula:
$$v = v + \phi 1 \times (p_{Best} - x) + \phi 2 \times (g_{Best} - x).$$
 - (b) Atualize a posição da partícula pela fórmula:
$$x = x + v.$$
7. Se a condição de término não for alcançada, retorne ao passo 4.

Otimização baseada em Enxame de Partículas (PSO)

Desde sua proposição original, muitos tipos de esquemas foram introduzidos para refinar o PSO:

- peso de inércia da velocidade (ω) (1998);
- fator de constrição (χ), limitando a velocidade (1999);
- operação de cruzamento baseado em AGs (2001);
- EPSO: distrib. gaussiana dos parâms ω , ϕ_1 e ϕ_2 (2002);
- inclusão de um operador de autoadaptação (2004);
- CSV-PSO: que ajusta dinamicamente ω , o limite da velocidade e o espaço de vôo (2005);
- INPSO: vizinhança independente com subenxames independentes (2005);

Otimização baseada em Enxame de Partículas (PSO)

Aplicações da técnica de PSO têm sido amplamente investigadas na literatura:

- melhoria de desempenho em outros métodos híbridos com AGs, RNAs, sistemas nebulosos, evolução diferencial, etc.
- sintonização de **controladores PID**; despacho econômico de carga (Wang & Singh, 2009), projeto de circuitos lógicos, robótica, projeto de antenas, otimização de redes de energia elétrica, despacho de carga (Abido, 2003), jogos, **processamento de imagens**, mineração de dados, etc.

Outros métodos

Métodos de inteligência coletiva:

- algoritmos de colônia de abelhas,
- algoritmo de colônia de vagalumes,
- algoritmo shuffled frog-leaping,
- algoritmos de coleta de alimentos por bactérias,
- algoritmos baseados em cardumes de peixes,
- sistemas imunológicos artificiais,
- bat algorithms,
- cuckoo search