

## Otimização com base no comportamento das Formigas



Importante: Apresentação produzida a partir de material disponível na Internet.

## Introdução

- ◆ Em alguns seres vivos o comportamento de grupo é um importante fator para a sobrevivência. As abelhas, por exemplo, após explorarem e definirem a localização de uma boa fonte de alimento oferecem o produto final as suas companheiras. Caso forem solicitadas, avisam sobre a posição do recurso através de uma forma de dança que indica a distância da colméia até o alimento e o ângulo de direção. (Free, 1977; Maeterlinck, 1980).

## Introdução

- ◆ As formigas são insetos sociais que possuem um sistema complexo de organização e divisão de tarefas, cuja principal função é garantir a sobrevivência do formigueiro.
- ◆ Chama atenção a complexidade das tarefas executadas pelo formigueiro quando comparada com a simplicidade de cada formiga individualmente.
- ◆ Dependendo da espécie, algumas formigas podem enxergar até muitas centenas de metros, enquanto outras são quase cegas.
- ◆ Para a obtenção do alimento, a colônia resolve um interessante problema de otimização. Inicialmente, as formigas percorrem de modo aleatório a região próxima ao formigueiro em busca do alimento. Cada formiga, enquanto percorre o seu caminho, deposita sobre o solo uma substância chamada feromônio, formando um caminho ou rastro de feromônio.

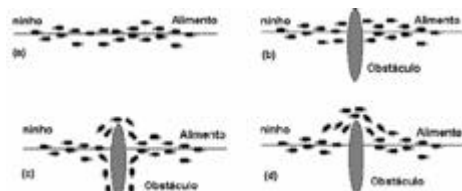
## Introdução

- ◆ As formigas subsequentes detectam a presença desta substância e tendem a escolher o caminho marcado com a maior concentração de feromônio. O feromônio portanto, além de possibilitar a formação de um caminho de volta para a formiga, também tem a função de informar as outras formigas sobre quais os melhores caminhos até o alimento.
- ◆ Depois de algum tempo, os caminhos mais eficientes – ou de menor distância percorrida até o alimento – acumulam uma quantidade maior de feromônio.
- ◆ Inversamente, os caminhos menos eficientes – ou de maior distância percorrida até o alimento – apresentam uma pequena concentração de feromônio, devido ao menor número de formigas que passaram por ele e ao processo de evaporação natural do feromônio.

## Introdução

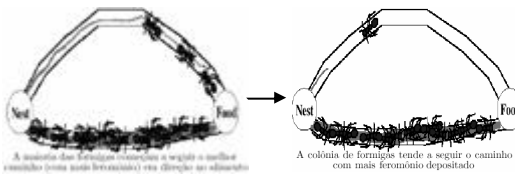
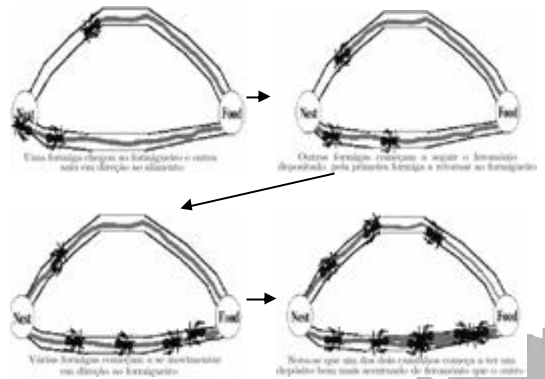
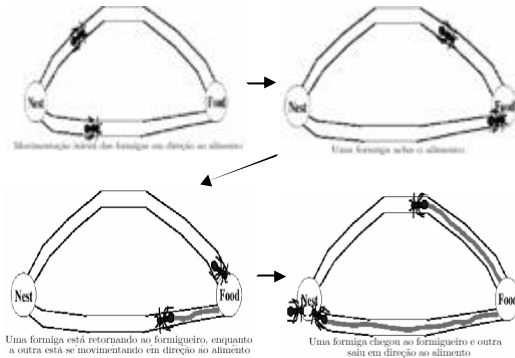
- ◆ No problema de otimização que o formigueiro se defronta, cada formiga é capaz de construir uma solução completa do problema; contudo, a melhor solução só é obtida mediante cruzamento das diversas soluções encontradas.
- ◆ A meta-heurística da colônia de formigas foi inspirada na observação das colônias de formigas reais, em particular como elas encontram o menor caminho entre a fonte de alimentos e o formigueiro.
- ◆ Algoritmo de formigas proposto por Dorigo (1991, 1992) para a solução de problemas de otimização combinatoria (ex.: problema do caixeiro viajante).

### UM EXEMPLO COM FORMIGAS REAIS

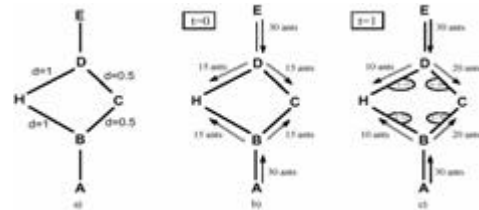


Transposição de obstáculos com a construção de uma trilha de feromônio alternativa. Em (a) a trilha do ninho até a fonte de comida está desocupada. A colocação de um obstáculo em (b) obstrui a trilha e faz com que aproximadamente 50% das formigas se dirijam para cada lado, procurando refazer a parte bloqueada (c). Como o feromônio depositado no lado mais curto do obstáculo se acumula mais rapidamente, por haver uma maior concentração de formigas as novas formigas que chegam tendem cada vez mais a seguir este caminho, até que a nova trilha é formada (d).

### UM OUTRO EXEMPLO COM FORMIGAS REAIS

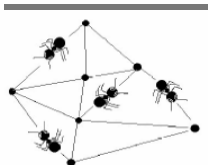


### UM EXEMPLO COM FORMIGAS ARTIFICIAIS



### Vamos apresentar:

- ◆ Ant System (AS)
- ◆ Ant Colony System (ACS)
- ◆ AS e GA (Genetic Algorithms)
- ◆ Savings based Ant System (SbAS)
- ◆ Problema de otimização de uma função real num espaço contínuo (Amarildo de Vicente)



### Aplicações do algoritmo ACO

Problem name	Authors	Algorithm name	Year
Traveling salesman	Dorigo, Maniczo & Colomi	AS	1991
	Gambardella & Dorigo	Ant-Q	1995
	Dorigo & Gambardella	ACS & ACS-3-opt	1996
	Stützle & Hoos	MMAS	1997
	Bullnheimer, Hartl & Strauss	AS, ...	1997
Quadratic assignment	Colomi, Dorigo & Maniczo	BWAS	2000
	Maniczo, Colomi & Dorigo	AS-QAP	1994
	Gambardella, Taillard & Dorigo	IAS-QAP*	1997
	Stützle & Hoos	MMAS-QAP	1997
	Maniczo	ANTS-QAP	1998
Scheduling problem	Maniczo & Colomi	AS-QAP*	1999
	Colomi, Dorigo & Maniczo	AS-JSP	1994
	Stützle	AS-FSP	1997
	Bauer et al.	ACS-SMTTP	1999
	den Boer, Stützle & Dorigo	ANTS-SMTTP	1999
	Mohile, Middendorf & Schneck	ACO-RCPSP	2000

Fonte: (Dorigo; Stützle, 2000)

## Aplicações do algoritmo ACO

Vehicle routing	Bullhölzer, Hartl & Strauss Gambardella, Taillard & Agazzi	AS-VRP HAS-VRP	1997 1999
Connection-oriented network routing	Schoonderwoerd <i>et al.</i> White, Pagnsek & Oppacher Di Caro & Dorigo Bonabeau <i>et al.</i>	ABC ASGA AntNet-PS ABC-rms ants	1996 1998 1998 1998
Connection-less network routing	Di Caro & Dorigo Subramanian, Druschel & Chen Bertone <i>et al.</i> van der Pol & Rothkrantz	AntNet & AntNet-FA Regular ants CAF ABC-backward	1997 1997 1998 1998
Sequential ordering	Gambardella & Dorigo	HAS-SOP	1997
Graph coloring	Costa & Hertz	ANTCOL	1997

Fonte: (Dorigo; Stützle, 2000)

## Aplicações do algoritmo ACO

Shortest common supersequence	Michel & Middendorf	AS-SCS	1998
Frequency assignment	Maniezzo & Carbonaro	ANTS-FAP	1998
Generalized assignment	Ramalhinho Lourenço & Serra	MMAS-GAP	1998
Multiple knapsack	Legatramón & Michalewicz	AS-MKP	1999
Optical networks routing	Navarro Varela & Sinclair	ACO-VWP	1999
Redundancy allocation	Liang & Smith	ACO-RAP	1999
Constraint satisfaction	Schon	Ant-P-solver	2000

Fonte: (Dorigo; Stützle, 2000)

## Ant System (AS) (Dorigo, 1991, 1996)

Ant System, foi proposto inicialmente por Colnari, Dorigo e Maniezzo (1991, 1992)

O algoritmo *Ant System* foi o primeiro algoritmo baseado em colônias de formigas, e quando foi introduzido pela primeira vez, ele foi aplicado ao problema do caixeiro viajante (*TSP – Traveling salesman problem*).

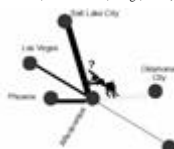
O Sistema de Formigas, primeira meta-heurística de otimização de colônia de formigas, quando aplicado ao problema do caixeiro viajante, inicia-se com cada formiga construindo uma solução a partir de um dos nós da rede do problema. Cada formiga  $k$  constrói o seu caminho movendo-se através de uma sequência de locais vizinhos, onde os movimentos são selecionados segundo uma distribuição de probabilidades.

## Ant System (AS) (Dorigo, 1991, 1996)

**Regra de transição:** Define-se a probabilidade de transição de uma cidade  $i$  para uma cidade  $j$  para a  $k$ -th formiga.

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{u \in J_k(i)} [\tau_{iu}]^{\alpha} \cdot [\eta_{iu}]^{\beta}} & \text{se } j \in J_k(i) \\ 0 & \text{caso contrário} \end{cases}$$

Lembrete: expressão soma Maniezzo, Gambardella, Luigi, 2002)



Denomina-se por  $\eta_{ij}$  (atratividade do arco  $(i,j)$ ),  $1/D_{ij}$  ( $D_{ij}$  é o custo da aresta). Esta quantidade não é alterada durante o AS.  $\tau_{ij}$  é quantidade de feromônio existente no arco  $(i,j)$ .

$J_k(i)$  é o conjunto de cidades ainda não visitadas, e  $\alpha$  e  $\beta$  são parâmetros que controlam a importância relativa ao feromônio e a função que representa a atratividade.

## Ant System (AS) (Dorigo, 1991, 1996)

**Atualização Global:** Após todas as formigas completarem a rota (tour), realiza-se a atualização global, onde ocorre a evaporação e inserção do feromônio nas arestas visitadas. Esta operação é realizada através da seguinte expressão:

$$\tau_{ij} = \rho \tau_{ij} + \Delta \tau_{ij}$$

onde  $\rho$  é um coeficiente tal que  $(1-\rho)$  representa a evaporação de feromônio.

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k$$

$m$  representa o número de formigas.

## Ant System (AS) (Dorigo, 1991, 1996)

onde  $\tau_{ij}^k$  é a quantidade por unidade de comprimento, do feromônio depositado na aresta  $(i,j)$  pela  $k$ -th formiga, e é dada por

$$\Delta \tau_{ij}^k = \begin{cases} \frac{1}{L_k} & \text{se a } k\text{-th formiga usar a aresta } (i,j) \text{ em sua rota} \\ 0 & \text{caso contrário} \end{cases}$$

onde  $L_k$  é o comprimento da rota da  $k$ -th formiga.

Para garantir que uma formiga não retorne a uma cidade já visitada, associa-se a cada formiga uma lista tabu.

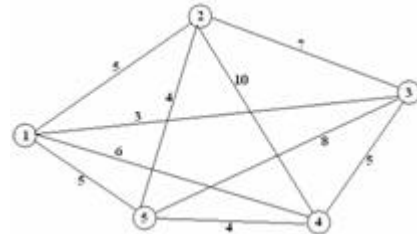
Quando a rota é completada a lista tabu é utilizada para computar a solução corrente da formiga. (caminho percorrido pela formiga) (a cada ciclo completo a lista tabu é esvaziada)

### Algoritmo Ant System (AS)

(1)	<b>Inicialização</b>
(2)	<b>Loop</b>
(3)	Aplicar a função de transição para escolher o próximo vértice $j$
(4)	Mover-se para o vértice $j$
(5)	<b>Until</b> todas as formigas completarem o <i>tour</i>
(6)	Atualização global do feromônio
(7)	<b>Until</b> Condição final

De acordo com a figura acima, temos: (1) As formigas são colocadas nos vértices do grafo seguindo alguma regra ou função; outra regra é utilizada para inicializar os valores do feromônio nas arestas. (2) cada *loop* corresponde a um passo da simulação e deve ser executado por todas as formigas. (4) *tour* definido como uma solução possível para o Problema do Caixeiro. (5) é feita a atualização global de feromônio nas arestas.

### Algoritmo Ant System (AS) – Exemplo



### Algoritmo Ant System (AS) – Exemplo

3 formigas: uma no nó 2, uma no nó 4 e outra no nó 5. Parâmetros inicialização:  $\alpha=1.5$ ;  $\beta=2$ ;  $\tau_0=0.99$ ;  $\tau_0=1$ ;  $k?$  [0,1] (dist. uniforme)

Lista  $tabu_1=[2]$

Lista  $tabu_2=[4]$

Lista  $tabu_3=[5]$

formiga no nó 2	formiga no nó 4	formiga no nó 5
$p_{21}=0.3010$ $p_{23}=0.1536$ $p_{24}=0.0752$ $p_{25}=0.4702$	$p_{41}=0.1980$ $p_{42}=0.0713$ $p_{43}=0.2851$ $p_{45}=0.4455$	$p_{51}=0.2215$ $p_{52}=0.3460$ $p_{53}=0.0865$ $p_{54}=0.3460$
nó 1 0.3010 nó 3 0.4545 nó 4 0.5298 nó 5 1	nó 1 0.1980 nó 2 0.2693 nó 3 0.5545 nó 5 1	nó 1 0.2215 nó 2 0.5675 nó 3 0.6540 nó 4 1
$k=0.9901$	$k=0.4860$	$k=0.4565$

### Algoritmo Ant System (AS) – Exemplo

Lista  $tabu_1=[2, 5]$

Lista  $tabu_2=[4, 3]$

Lista  $tabu_3=[5, 2]$

formiga no nó 5	formiga no nó 3	formiga no nó 2
$p_{51}=0.3386$ $p_{53}=0.1323$ $p_{54}=0.5291$	$p_{31}=0.7551$ $p_{32}=0.1387$ $p_{34}=0.1062$	$p_{21}=0.5681$ $p_{23}=0.2899$ $p_{24}=0.1420$
nó 1 0.3386 nó 3 0.4709 nó 4 1	nó 1 0.7551 nó 2 0.8938 nó 5 1	nó 1 0.5681 nó 3 0.8580 nó 4 1
$k=0.2311$	$k=0.8913$	$k=0.0185$

### Algoritmo Ant System (AS) – Exemplo

Lista  $tabu_1=[2, 5, 1]$

Lista  $tabu_2=[4, 3, 2]$

Lista  $tabu_3=[5, 2, 1]$

formiga no nó 1	formiga no nó 2	formiga no nó 3
$p_{13}=0.80$ $p_{14}=0.20$	$p_{21}=0.3902$ $p_{25}=0.6098$	$p_{32}=0.80$ $p_{34}=0.20$
nó 3 0.80 nó 4 1	nó 1 0.3902 nó 5 1	nó 3 0.80 nó 4 1
$k=0.6068$	$k=0.7621$	$k=0.8214$

### Algoritmo Ant System (AS) – Exemplo

Lista  $tabu_1=[2, 5, 1, 3, 4, 2]$

$L_1=27$

Lista  $tabu_2=[4, 3, 2, 5, 1, 4]$

$L_2=27$

Lista  $tabu_3=[5, 2, 1, 4, 3, 5]$

$L_3=28$

atualização de feromônio:

$$\tau_{25} = 0.99 * 1 + (1/27 + 1/27 + 1/28) = 1.0998$$

$$\tau_{43} = 0.99 * 1 + 0 = 0.99$$

$$\vdots$$

Obs.: com  $\alpha=1$  e  $\beta=5$ , uma formiga foi testada e a seguinte rota foi obtida: 2 – 5 – 4 – 3 – 1 – 2 com  $L=21$

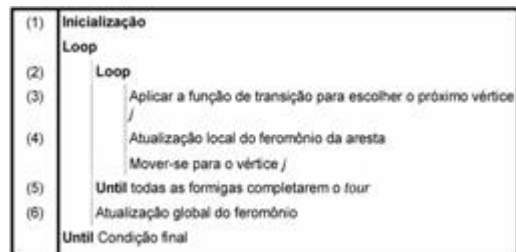
## Ant Colony System (ACS) (Dorigo, 1997)

O modelo Ant Colony System (Dorigo, 1997) utiliza, como base, o modelo Ant System.

As principais diferenças são:

- a introdução de uma nova equação, denominada *pseudo-random proportional* (Stützle; Holger, 1996) na regra de transição;
- a modificação da regra de atualização global, de forma que apenas a formiga que percorreu o menor caminho global possa atualizar o feromônio;
- a inserção da atualização local de feromônio, com atualização de feromônio durante a realização do *tour*.

## Algoritmo Ant Colony System (ACS)



## Algoritmo Ant Colony System (ACS)

De acordo com a figura anterior, temos:

(1) As formigas são colocadas nos vértices do grafo seguindo alguma regra ou função; outra regra é utilizada para inicializar os valores de feromônio nas arestas. Ant Colony System (Dorigo, 1997) procura distribuir as formigas randomicamente nos diversos vértices, com pelo menos uma em cada vértice. Isto refere-se a testes anteriores em Ant System (Dorigo, 1996) que mostraram que, para a resolução do Problema do Caixeiro Viajante, a distribuição randômica da posição inicial das formigas obtêm melhores resultados, se comparado com simulações onde todas as formigas são posicionadas num único vértice inicial. Em relação a quantidade de feromônio nas arestas, o algoritmo do modelo Ant Colony System utiliza um valor inicial igual para todas as arestas do grafo. Este valor deve ser diferente de zero.

(2) cada *loop* corresponde a um passo da simulação e deve ser executado por todas as formigas.

(5) *tour* definido como uma solução possível para o Problema do Caixeiro Viajante.

## Algoritmo Ant Colony System (ACS)

### Regra de transição:

Acréscimo de um fator  $q_0 \in [0,1]$  (fixo)

Sorteio de um valor  $q \in [0,1]$  (uniformemente distribuídos) sempre que uma formiga for mudar de cidade

$$j = \begin{cases} \arg \max_{u \in J_k(i)} \{ [\tau_{iu}]^a \cdot [\eta_{iu}]^b \} & \text{se } q \leq q_0 \text{ (exploitation)} \\ S & \text{caso contrário (biased exploration)} \end{cases}$$

S é uma variável aleatória selecionada de acordo com a distribuição de probabilidade dada por

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^a \cdot [\eta_{ij}]^b}{\sum_{u \in J_k(i)} [\tau_{iu}]^a \cdot [\eta_{iu}]^b} & \text{se } j \in J_k(i) \\ 0 & \text{caso contrário} \end{cases}$$

## Algoritmo Ant Colony System (ACS)

**Atualização Global:** A atualização global do feromônio em Ant Colony System difere daquela em Ant System porque somente a formiga com o menor *tour* global é autorizada a atualizar o feromônio na sua trilha.

$$\tau_{ij} = r \tau_{ij} + d \Delta \tau_{ij}$$

onde  $d \in [0,1]$  é um parâmetro de evaporação e inserção global de feromônio.

$$\Delta \tau_{ij} = \begin{cases} \frac{1}{L_{\text{melhor}}} & \text{se a aresta } (i,j) \in \text{a melhor tour} \\ 0 & \text{caso contrário} \end{cases}$$

$L_{\text{melhor}}$  é o comprimento do menor tour.

## Algoritmo Ant Colony System (ACS)

**Atualização Local:** O modelo original Ant System atualiza o feromônio nas arestas apenas após todas as formigas completarem o *tour*. Ant Colony System introduz a equação de atualização local, que possibilita a evaporação e reinserção de feromônio nas arestas após cada formiga realizar a regra de transição. A atualização local é definida pela equação:

$$\tau_{ij} = r \tau_{ij} + (1-r) \Delta \tau_{ij}$$

sendo

$$\Delta \tau_{ij} = \begin{cases} 0 & \text{(a)} \\ \tau_{\text{inicial}} & \text{(b)} \\ g \max_{z \in J_k(i)} \tau_{jz} & \text{(c)} \end{cases}$$

- em (a) a reinserção é nula.
- em (b) proporcional ao valor inicial de feromônio nas arestas do caminho.
- em (c) com base na fórmula utilizada por Q-learning (Dorigo, 1997).
- Algoritmo que utilizam (c) são chamados de **Ant-Q** (Dorigo; Gambardella, 1996)

## Algoritmo Ant System e Genetic Algorithm (ASGA)

Trabalho desenvolvido por Boechel(2003) (PPGCC)

O algoritmo genético é aplicado na tentativa de melhorar o desempenho do AS. (falta de citação do trabalho de White, Pagurek, Oppacher (1998) e de Reimann, Shtovba, Nepomuceno (2001) – aplicação no VRP)

Uma operação de *crossover* (cruzamento) é realizada sobre algumas soluções selecionadas a partir do algoritmo das formigas, resultando novas soluções, as quais são incorporadas ao conjunto de soluções.

Essas novas soluções podem ou não serem aceitas em função da qualidade da solução encontrada.

## Algoritmo Ant System e Genetic Algorithm (ASGA)

Como é realizado o cruzamento:

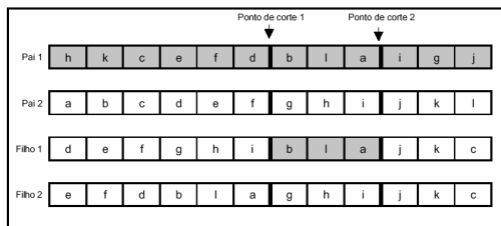
Escolhe-se (aleatoriamente) duas soluções encontradas pelo AS.

Escolha aleatória de dois pontos de corte em cada um dos elementos selecionados.

A seção definida entre estes dois pontos é copiada integralmente no descendente.

Os lugares restantes são preenchidos usando as informações não repetidas na seção de cruzamento, começando do segundo ponto de corte.

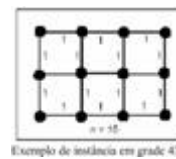
## Algoritmo Ant System e Genetic Algorithm (ASGA)



Exemplo de cruzamento com o operador OX (Order Crossover)

## Algoritmo Ant System e Genetic Algorithm (ASGA)

Alguns resultados destes testes para o problema do TSP



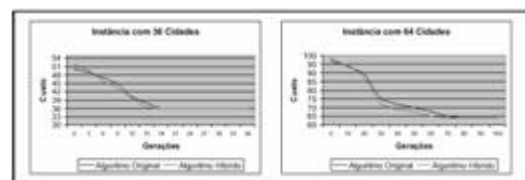
Exemplo do problema

## Algoritmo Ant System e Genetic Algorithm (ASGA)

Problema	Custo Ótimo	Algoritmo Original		Algoritmo Híbrido	
		Média Gerações	Tempo (mm:ss)	Média Gerações	Tempo (mm:ss)
Q4x4 16 Cidades	16	12	0:01:10	9	0:01:03
Q5x5 25 Cidades	25,4	18	0:02:06	15,5	0:01:56
Q6x6 36 Cidades	36	30	0:03:57	27	0:03:11
Q8x8 64 Cidades	64	66	0:08:32	72	0:07:56
Q10x10 100 Cidades	100	134	0:14:49	123	0:13:38

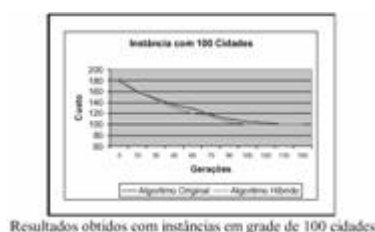
Comparação do desempenho dos algoritmos original e híbrido.

## Algoritmo Ant System e Genetic Algorithm (ASGA)



Resultados obtidos com instâncias em grade de 49 e 64 cidades.

### Algoritmo Ant System e Genetic Algorithm (ASGA)



### Algoritmo Savings based Ant System (SbAS)

Doerner, Gronalt, Hartl, Reimann, Strauss and Stummer (2002)

Aplicação no VRP

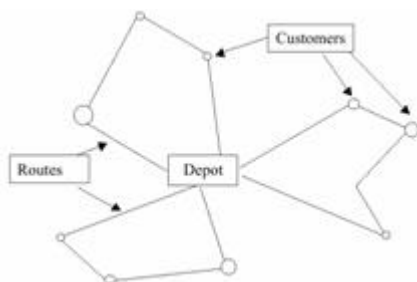
Objetivo do VRP é encontrar um conjunto de rotas com custos mínimos, iniciando e finalizando num depósito e servindo uma demanda conhecida de um número de clientes.

Cada cliente deve ser servido exatamente uma única vez.

A capacidade do veículo e o tempo de viagem são restrições que não podem ser violadas.

São problemas NP-hard

### Algoritmo Savings based Ant System (SbAS)



### Algoritmo Savings based Ant System (SbAS)

Algoritmo Savings (Clarke; Wright, 1964)

Este algoritmo é a base da maioria das ferramentas comerciais para resolver VRPs em aplicações industriais.

O processo é inicializado, atribuindo a cada cliente um tour em separado.

Em seguida, para cada cliente  $i$  e  $j$  as seguintes medidas de ganhos (savings) são calculadas:

$$s(i, j) = d(i, 0) + d(0, j) - d(i, j)$$

onde  $d(i, j)$  é a distância entre  $i$  e  $j$  e o índice 0 denota o depósito.

### Algoritmo Savings based Ant System (SbAS)

Assim, os valores  $s(i, j)$  contêm os savings da combinação de 2 clientes  $i$  e  $j$  em um tour.

No processo iterativo, clientes ou tour parciais são combinadas de acordo com estes ganhos, iniciando com aquele que determina o maior ganho, até não existirem mais combinações factíveis.

O resultado deste algoritmo é um conjunto sub-ótimo de tours passando por todos os clientes.

### Algoritmo Savings based Ant System (SbAS)

Exemplo:

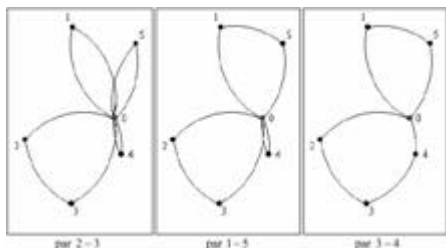


Ordenação dos pares em ordem decrescente de ganhos

Arco	Ganho
2 - 3	67.69
1 - 5	57.34
3 - 4	52.44
1 - 2	51.54
2 - 4	24.64
1 - 3	7.09
2 - 5	6.37
4 - 5	5.63
3 - 5	0.26
1 - 4	0.04

## Algoritmo Savings based Ant System (SbAS)

### Exemplo:



## Algoritmo Savings based Ant System (SbAS)

O SbAS funciona da seguinte maneira:

Solução inicial: viagens isoladas para cada cliente.

Gera-se a lista de savings  $s(i,j)$  ou  $s_{ij}$

Regra de transição:

Combinações viáveis são escolhidas da lista de acordo com a regra (modificada) do AS:

$$p_{ij} = \begin{cases} \frac{[s_{ij}]^a \cdot [t_{ij}]^b}{\sum_{h \in \Omega_i} [s_h]^a \cdot [t_h]^b} & \text{se } h \in \Omega_i \\ 0 & \text{caso contrário} \end{cases} \quad \text{onde } t_{ij} = [s(i,j)]^2 \cdot c_{ij}^m \text{ é a atratividade.}$$

O processo é finalizado quando não houver mais uma combinação factível que seja possível.

## Algoritmo Savings based Ant System (SbAS)

O algoritmo 2-opt é aplicado em todas as tours construídas pelas formigas, antes da atualização das informações de feromônio.

Atualização global:

$$\tau_{ij}^{new} = \rho \tau_{ij}^{old} + \sum_{k=1}^m \Delta \tau_{ij}^k + \sigma \Delta \tau_{ij}^0$$

A atratividade sofre aumento com as novas informações de feromônio.

Esta variação nos valores da atratividade a cada novo ciclo é uma contribuição importante para este processo.

## OBTENÇÃO DO ÓTIMO GLOBAL DE FUNÇÕES PELO PROCESSO ANT SYSTEM (Amarildo de Vicente)

Pela própria característica, da forma como foi apresentado originalmente o *Ant System* se aplica apenas a problemas que permitem uma representação por meio de grafos, o que não é o caso de um problema de otimização de uma função real num espaço contínuo.

Para este tipo de problema, foi desenvolvido um trabalho (ver MONMARCHÉ et al., 1999) onde se segue a estratégia de um tipo especial de formiga (*Pachycondyla apicalis*) na busca de alimento.

O interesse sobre estas formigas para otimização se deve ao fato de que elas usam princípios relativamente simples para capturarem suas presas, tanto de um ponto de vista local quanto global.

Sob um ponto de vista global o comportamento destas formigas é o seguinte: Partindo de seu ninho, elas cobrem globalmente uma dada superfície dividindo-a em regiões de caça individuais.

## OBTENÇÃO DO ÓTIMO GLOBAL DE FUNÇÕES PELO PROCESSO ANT SYSTEM (Amarildo de Vicente)

Estas regiões de caça são criadas uniformemente em torno do ninho e têm seus centros localizados a uma distância máxima (em média) de aproximadamente 10 metros do mesmo.

Possuem uma amplitude de 2.5 metros aproximadamente e são distribuídas em todas as direções em torno do ninho. Com isto, as formigas cobrem com um mosaico de pequenas regiões uma grande superfície ao redor de sua moradia.

Periodicamente se observam trocas de local do ninho. Estas trocas podem ocorrer, por exemplo, por dilapidação da atual moradia ou por diminuição de presas.

A mudança do ninho é um processo complexo que usa formigas especializadas na busca de um novo local, bem como em mecanismos de recrutamento para fazer toda a colônia mudar para o novo ninho.

## Proposta de trabalho

Procedimento híbrido,  
combinado SbAS com AG



(Maniezzo, Gambardella, Luigi, 2002)

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha + \eta_{ij}^\beta}{\sum_{(i,j) \in \text{tabu}_k} (\tau_{ij}^\alpha + \eta_{ij}^\beta)} & \text{if } (i,j) \notin \text{tabu}_k \\ 0 & \text{otherwise} \end{cases}$$

defined parameters ( $0 \leq \alpha, \beta \leq 1$ )

## Ant System (AS) (Dorigo, 1991, 1996)

O comprimento da aresta entre duas cidades  $i$  e  $j$ , é representado por  $D_{ij}$ , onde  $D_{ij}$  é também chamado de distância euclidiana entre  $i$  e  $j$ .

$$D_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2}$$

Define-se como  $\text{tabu}_k$  um vetor dinamicamente crescente que contém a lista  $\text{tabu}$  da  $k$ -th formiga e  $\text{tabu}_k(s)$  é o  $s$ -th elemento da lista, isto é, a  $s$ -th cidade visitada pela  $k$ -th formiga na rota corrente.

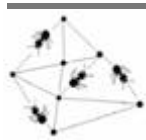
Denomina-se  $\tau_{ij}$  (atratividade do arco  $(i,j)$ ) a qtidade  $1/D_{ij}$ . Esta qtidade não é alterada durante o AS.

## Ant System (AS) (Dorigo, 1991, 1996)

Uma instância do problema do caixeiro viajante é definida por um grafo  $(N,E)$ , onde  $N$  é o conjunto de todas as cidades e  $E$  o conjunto de arestas que podem ser percorridas (todas as ligações entre todas as cidades).

- Problema de  $n$  cidades tem espaço de busca dado por:  $S = (n-1)!/2$
- Problema com 8 cidades, corresponde a um total de 2520 rotas

No algoritmo *Ant System* formigas artificiais constroem soluções (rotas) para o caixeiro viajante movendo-se no grafo do problema de uma cidade para outra.



## Heurística e meta-heurística

- ◆ **Heurística:** (do grego *heuriskein* = descobrir, encontrar) É qualquer método ou técnica criada, desenvolvida para resolver um determinado tipo de problema, através de um enfoque intuitivo, para se obter uma solução razoável. Técnica que procura boas (ou próximas às ótimas) soluções a um razoável custo computacional, sem ser capaz de garantir a viabilidade do ótimo.
- ◆ **Meta-Heurística:** Deriva do grego *heuriskein* = descobrir, encontrar e o sufixo "*meta*" = além de, em um nível superior. São técnicas que, quando aplicadas a métodos de busca local, permitem a superação da otimalidade local com vistas à obtenção de soluções de qualidade superior. São heurísticas de busca no espaço de soluções.