

# Uma Análise Semiótica de um Algoritmo de Inteligência Coletiva

Eduardo Ferreira Jucá de Castro, *Member, IEEE*

Fundação CPqD

Rod. Campinas Mogi-Mirim, km 118,5

Campinas – SP

+55 19 37057238

[e.jucah@ieee.org](mailto:e.jucah@ieee.org)

**Abstract** — This paper presents an analysis of a Swarm Intelligence algorithm, considering a Computational Semiotics approach. The Ant Colony System algorithm, applied to the Traveling Salesman Problem, is analysed in order to identify the elementary knowledge operators associated with each step.

**Keywords** — Semiótica Computacional, Swarm Intelligence, Otimização por Colônia de Formigas, Sistema de Colônia de Formigas.

## I. INTRODUÇÃO

A Semiótica Computacional é uma área de estudos dedicada a emular o ciclo de semiose por meio de computadores digitais. A motivação para isso vem da hipótese de que a inteligência está diretamente relacionada com o processo de semiose e um dos pontos chave para esse estudo é determinação de unidades elementares, mínimas, de inteligência e sua relação com a semiótica. Uma proposta de definição para essas unidades elementares de inteligência e seus operadores que seriam responsáveis pela construção de um comportamento inteligente foi apresentada em (Gudwin, 1998).

Uma forma de entender essa proposta é tentar aplicá-la a uma situação concreta. Dentre as diversas alternativas possíveis foi escolhido um algoritmo de Inteligência Coletiva, ou como é mais conhecida, na expressão em inglês, Swarm Intelligence. Especificamente, foi escolhido o algoritmo Sistema de Colônia de Formigas (ACS – do inglês, Ant Colony System), por ser um dos mais representativos trabalhos na área conhecida como Swarm Intelligence (termo em inglês que pode ser traduzido como Inteligência Coletiva). E, para permitir uma análise em um cenário mais concreto, será usada a aplicação do ACS ao Problema do Caixeiro Viajante (TSP – do inglês, Traveling Salesman Problem).

## II. INTELIGÊNCIA COLETIVA

O termo Swarm Intelligence foi introduzido em (Beni e Wang, 1989) e se baseia na observação do comportamento coletivo de algumas espécies de animais, como, formigas, cupins, vespas, bandos de pássaros, etc.

Dentre as aplicações de maior interesse está a otimização de problemas de difícil solução por métodos diretos. E nesse contexto se encaixa a utilização do algoritmo ACS na

solução do Problema do Caixeiro Viajante.

O ACS é um dos algoritmos para otimização, inspirado pelas colônias de formigas (ACO – do inglês, Ant Colony Optimization). Para mais informações e histórico da área ver (Dorigo e Stützle, 2009).

### A. O Problema do Caixeiro Viajante (TSP)

O TSP é um problema clássico da otimização combinatória, muito estudado por pesquisadores de diferentes áreas em função de suas várias aplicações práticas. O TSP pertence a uma classe de problemas não determinísticos de tempo não polinomial, classificado como *NP*-difícil, por isso, sua resolução utilizando métodos heurísticos, ganhou maior importância, principalmente quando é aplicado a problemas de grande porte.

A representação deste problema pode ser feita através de um grafo completo  $G=(V,A)$ , onde  $V$  é o conjunto de vértices representando as cidades e  $A$  o conjunto de arcos ou arestas que conectam cada par de cidades  $i, j \in V$ . A cada aresta é atribuído um valor  $c_{ij}$ , que é a distância entre as cidades  $i$  e  $j$ . (Carvalho, 2007)

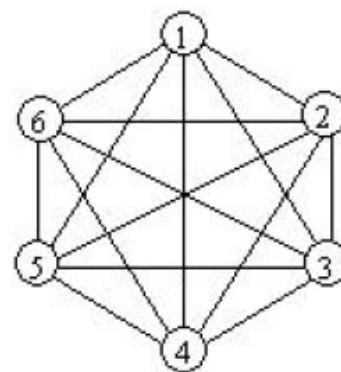


Figura 1: Exemplo de um grafo utilizado no TSP

### B. Sistema de Colônia de Formigas (ACS)

Os algoritmos de formigas, incluindo o ACS, se baseiam na metáfora natural da colônia de formigas. Formigas reais são capazes de encontrar o menor caminho entre uma fonte de alimentos e seu formigueiro, sem o uso de pistas visuais mas apenas com base na informação do feromônio.

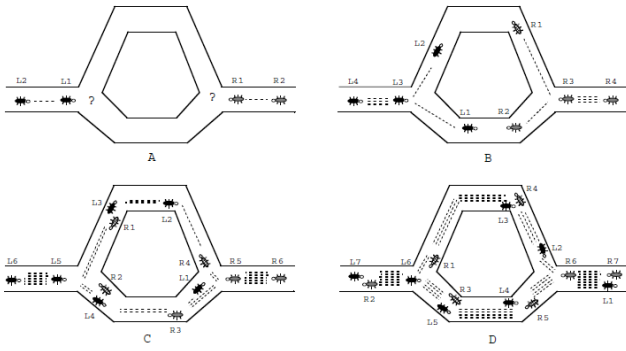


Figura 2: Como formigas reais encontram um caminho mais curto. A) Formigas chegam a um ponto de decisão. B) Algumas formigas escolhem o caminho superior e outras o inferior; a escolha é aleatória. C) Uma vez que a velocidade das formigas é aproximadamente constante, aquelas que escolheram o caminho inferior, mais curto, atingem o outro ponto de decisão antes daquelas que escolheram o caminho superior, mais longo. D) A taxa de acumulação do feromônio é maior no caminho mais curto.

Enquanto caminham, as formigas depositam feromônio no solo e, provavelmente, seguem o feromônio depositado por outras formigas. A ideia é de que o caminho mais curto leva a um retorno mais rápido ao formigueiro, elevando a probabilidade de ser o escolhido por mais formigas, devido a maior concentração de feromônio depositado nele.

É através desse comportamento, depositando feromônio nas trilhas percorridas, que as formigas conseguem cooperar em busca de um objetivo comum, sem a comunicação direta entre cada indivíduo. (Dorigo e Gambardella, 1997)

### C. O algoritmo ACS para o TSP

Informalmente, o funcionamento do ACS é o seguinte:  $m$  formigas artificiais (no restante desse artigo, referenciadas apenas como formigas) são posicionadas em  $n$  cidades escolhidas de acordo com alguma regra (por exemplo, aleatoriamente). Cada formiga constrói um percurso, uma solução para o TSP, aplicando repetidamente uma regra estocástica para a escolha de uma nova cidade, ainda não visitada (regra de transição de estado). Durante a construção do seu percurso, uma formiga também modifica a quantidade de feromônio nas arestas percorridas, aplicando a regra de atualização local. Uma vez que todas as formigas tenham terminado seus percursos, a quantidade de feromônio nas arestas percorridas é novamente atualizado, aplicando a regra de atualização global. As formigas são guiadas, ao construir seus percursos, tanto por uma informação heurística quanto pela informação do feromônio: uma aresta com maior concentração de feromônio é uma escolha muito desejada. As regras de atualização de feromônio são escolhidas de forma a que as arestas que deveriam ser visitadas pelas formigas tendam a receber mais feromônio. (Dorigo e Gambardella, 1997)

O algoritmo ACS é, então:

1. Iniciação:
  - 1.1. Para cada aresta  $(i,j)$  do grafo, estabelece-se um nível inicial de feromônio.
  - 1.2. Para cada formiga  $k$ , escolhe-se um nó aleatório para iniciar o percurso.

## 2. Iterações:

2.1. Para  $t = 1$  até  $n$ , onde  $n$  é o número máximo de iterações faça:

2.1.1. Para  $k = 1$  até  $m$ , onde  $m$  é o número de formigas, faça:

2.1.1.1 Cada formiga  $k$  constrói um caminho selecionando a próxima cidade a ser visitada segundo a regra de transição de estado, determinística ou probabilística influenciada pelo nível de feromônio  $\tau_{ij}$  e da métrica heurística de afinidade  $\eta_{ij}$ , onde  $q$  é uma variável aleatória uniformemente distribuída entre  $[0..1]$ ;

$q_0$  é um parâmetro ajustável ( $0 \leq q_0 \leq 1$ );

$\tau_{ij}$  é a probabilidade da formiga  $k$ , que se encontra na cidade  $i$ , escolher o nó  $j$  como próximo nó a ser visitado;

$\eta_{ij}$  é uma função heurística que representa a atratividade do arco  $(i, j)$  – no caso do problema do caixeiro viajante, adota-se o inverso  $1/d_{ij}$  do valor da distância entre os nós  $i$  e  $j$ ;

$p_{ij}^k$  é a probabilidade da formiga  $k$ , que se encontra na cidade  $i$ , escolher o nó  $j$  como próximo nó a ser visitado;

$J_i^k$  é o conjunto de pontos ainda não visitados pela formiga  $k$ , que se encontra atualmente no ponto  $i$ ;

$\alpha$  é um parâmetro que pondera a importância relativa da trilha de feromônio,  $\tau_{ij}$  na decisão de movimentação da formiga; e

$\beta$  é um valor heurísticamente escolhido, que pondera a influência relativa da distância  $\eta_{ij}$  entre os nós  $i$  e  $j$  no processo de decisão.

$$j = \operatorname{argmax} \left\{ [\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta \right\}, \quad \text{se } q \leq q_0;$$

$$p_{ij}^k = \frac{[\tau_{ij}(t)] \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)] \cdot [\eta_{il}]^\beta}, \quad \text{se } q > q_0;$$

2.1.1.2 Após cada transição da formiga  $k$ , aplique a regra de atualização local, motivada pela evaporação do feromônio,

$$\tau_{ij}(t) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_0$$

2.1.2. Para cada solução, calcule a distância  $L_k(t)$  do caminho descoberto pela formiga  $k$ ,

2.1.3. Se  $L_k(t) < L^*$  então  $S^* \leftarrow S_k(t)$ ;

2.1.4. Para cada aresta  $(i,j)$ , atualize o feromônio  $\tau_{ij}(t)$ ,  $\forall \tau_{ij}(t) \in L^k$ , de acordo o processo de deposição e evaporação de feromônio.

$$\tau_{ij}(t) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta \tau_{ij}(t), \quad \text{onde } \Delta \tau_{ij}(t) = 1/L^k$$

3. Retorne a melhor solução  $S^*$ .

(Carvalho, 2007)

### III. SEMIÓTICA COMPUTACIONAL

Dentre os diversos conceitos definidos no âmbito da Semiótica Computacional, os que são de interesse para esse artigo são as unidades elementares de conhecimento e seus operadores. Há muitas tentativas de definir qual seria a diferença entre “conhecimento” e “informação” e quais seriam as peças elementares de conhecimento, às vezes chamados simplesmente de unidades de conhecimento. Nesse artigo, será considerada a proposta apresentada em (Gudwin, 1998) e resumida a seguir, nos seus aspectos mais relevantes.

#### A. Unidades de Conhecimento

Inicialmente, consideremos a existência de um ambiente, ou mundo real, o qual pode ser definido como um conjunto de fenômenos dinâmicos contínuos ocorrendo em paralelo. Assume-se que nós não somos capazes de conhecer esse ambiente na sua totalidade. A parte desse ambiente que somos capazes de conhecer, em um processo através de nossos sensores, é chamado de nosso *Umwelt*. O *Umwelt*, também chamado de nosso ambiente sensível, é a nossa melhor compreensão possível da realidade. No entanto, é muito importante ressaltar que o *Umwelt* não é a realidade. É apenas aquilo que constitui o nosso melhor entendimento da realidade. Nesse sentido, nossos sensores são a fonte primária de informação que flui para dentro da nossa mente.

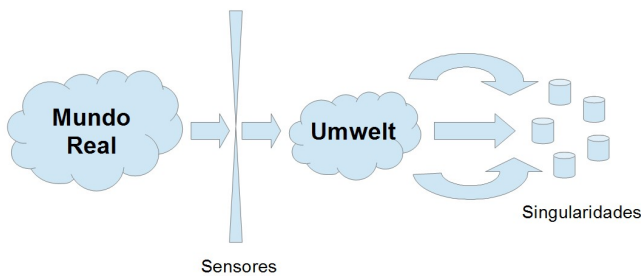


Figura 3: Extração de singularidades

Esses sensores fornecem uma informação parcial e contínua a cerca dos fenômenos ocorrendo no *Umwelt*. A partir dessa fonte contínua de informação, são extraídas as chamadas singularidades, isto é, aglomerados de informação que são agrupados em um conceito único. Essas singularidades são entidades discretas que modelam, em nível específico de resolução, os fenômenos ocorrendo no mundo real. Essas singularidades correspondem às unidades de conhecimento. (Gudwin, 1998)

#### B. Operadores Elementares de Conhecimento

Os operadores de conhecimento propostos, chamados de extração de conhecimento, geração de conhecimento e seleção de conhecimento, podem ser entendidos com generalizações para os métodos de raciocínio clássicos da semiótica, dedução, indução e abdução, respectivamente. Nesse sentido, extração de conhecimento é visto como uma abstração para dedução, geração de conhecimento é uma abstração para indução e seleção de conhecimento é uma abstração para abdução. Suas definições são dadas a seguir.

#### 1. Extração de Conhecimento

Considerando unidades de conhecimento  $b$  e  $a$ , de tal forma que  $b$  é uma abstração de  $a$ , então, uma função  $fke$  que mapeia  $b$  (uma estrutura) em  $a$ ,  $a = fke(b)$ , é chamada de operador de extração de conhecimento.

A figura 4 ilustra uma extração de conhecimento. A partir de um conjunto  $P$  de unidades de conhecimento, chamado de “premissa”, o operador extrai um conjunto  $C$  de unidades de conhecimento, chamado de “conclusão”.

Essa operação é chamada de extração de conhecimento porque as unidades de conhecimento em  $C$  formam um subconjunto das unidades de conhecimento em  $P$ . Então, o operador extrai de  $P$  apenas uma parte do seu conteúdo semântico.

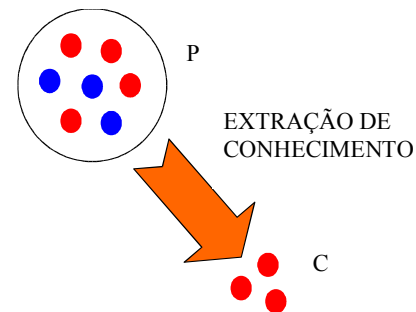


Figura 4: Extração de Conhecimento

#### 2. Geração de Conhecimento

Considere agora os mesmos  $a$  e  $b$  anteriores e também uma função  $fk g$  que mapeia  $a$  em  $b$ , isto é,  $b = fk g(a)$ . Então,  $fk g$  é chamada de operador de geração de conhecimento. Usualmente, esse tipo de operador não apresenta uma única entrada e única saída mas compreende um conjunto de unidades de conhecimento como entrada e um conjunto de unidades de conhecimento como saída correspondente. Assim,  $(b_1, \dots, b_m) = fk g(a_1, \dots, a_n)$ .

A figura 5 ilustra uma geração de conhecimento, sendo a premissa  $P$  uma coleção de unidades de conhecimento  $a_i$  e a conclusão  $C$  é uma coleção de unidades de conhecimento  $b_i$ . Uma das particularidades desse operador é que a definição extensional das unidades de conhecimento em  $C$  necessariamente contém elementos que não estavam originalmente na definição extensional das unidades de conhecimento em  $P$ .

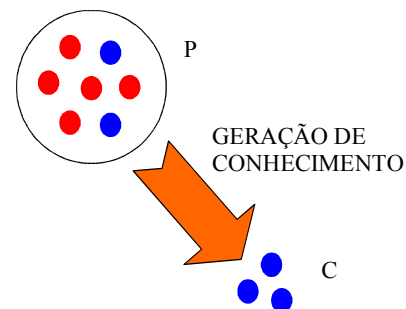


Figura 5: Geração de conhecimento

Eles foram adicionados durante o processo de geração de conhecimento. É isso o que caracteriza a operação de geração de conhecimento.

### 3. Seleção de Conhecimento

Considere agora, um conjunto de unidades de conhecimento como entrada,  $\{a_1, a_2, \dots, a_n\}$ , e um conjunto de candidatos  $\{c_1, c_2, \dots, c_m\}$  para serem a saída. Considere também uma função  $fks$ , que realiza a seleção dentre os candidatos  $b = fks(a_1, a_2, \dots, a_n, c_1, c_2, \dots, c_m)$ , no sentido de que  $b$  é um dos  $c_i$  e os  $a_i$  são usados para avaliar e escolher dentre os  $c_i$ . Então,  $fks$  é chamada um operador de seleção de conhecimento.

A figura 6 ilustra uma seleção de conhecimento. As unidades de conhecimento  $a_i$  pertencem ao conjunto P (Premissa) e os  $c_i$  pertencem ao conjunto H, também chamado de conjunto de hipóteses. As unidades de conhecimento  $b_i$  (mais de uma, nesse caso), são selecionadas dentre os  $c_i$ , e são indicadas por C (Conclusão).

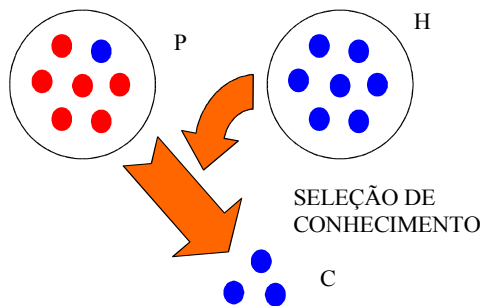


Figura 6: Seleção de conhecimento

Deve ser notado que há um caso particular em que há apenas um  $c_i$  em H, quando a seleção se torna uma “validação”. Nesse caso, as unidades de conhecimento em P são usadas para validar a nova unidade de conhecimento produzida como saída em C. Nesse sentido, se elas não forem capazes de validar  $c_i$ , produzirão uma saída vazia.

### IV. ANÁLISE

Reescrever o algoritmo em termos de processos facilita a identificação de quais operadores elementares de conhecimento estão envolvidos em cada etapa. Para efeito dessa análise, as iterações serão omitidas pois se tratam apenas de repetições das mesmas etapas. Assim, temos as seguintes etapas para o algoritmo ACS aplicado ao problema do caixeiro viajante e sua avaliação correspondente, em termos dos operadores elementares de conhecimento:

1. Atribuir valores iniciais adequados para as estruturas de dados – a partir das informações de quantas cidades e das distâncias entre elas são construídas as listas de cidades a percorrer utilizadas pelas formigas, atribuindo uma cidade inicial para cada formiga. Considerando-se que o universo de escolha para a entrada do problema é maior do que será utilizado, podemos dizer que há uma

seleção de conhecimento.

2. Escolher a próxima cidade a visitar – uma cidade da lista de cidades a visitar é selecionada com base na regra correspondente. Também, trata-se de uma seleção de conhecimento.
3. Adicionar a cidade escolhida na solução parcial. Nesse caso, a lista de caminhos que compõe a solução parcial é aumentada, o que caracteriza uma geração de conhecimento.
4. Aplicar a regra de atualização local, motivada pela evaporação do feromônio – o valor do feromônio para o percurso percorrido é atualizado com base na regra correspondente. Pode-se considerar que é uma extração de conhecimento, na medida em que a informação da distância percorrida é usada para reforçar o caminho com menor distância percorrida.
5. Calcular a distância percorrida na solução de cada formiga, após todas as formigas terem concluído suas soluções. A informação sobre as distâncias é adicionada à solução, sendo, portanto, uma geração de conhecimento.
6. Determinar a melhor solução dentre as obtidas por todas as formigas – que será aquela com menor distância total percorrida. É uma seleção de conhecimento.
7. Aplicar a regra de atualização global, atualizando o feromônio nos percursos da melhor solução. Assim como na etapa 4, trata-se de uma extração de conhecimento.

### V. CONCLUSÃO

Esse artigo apresentou um exercício de análise em que foram utilizados os operadores de conhecimento, conceitos definidos em uma proposta de Semiótica Computacional.

Embora concebidos para a síntese semiótica, espera-se ter sido demonstrada a utilidade da aplicação desses operadores também como ferramenta de análise, de forma a ampliar a sua compreensão e incentivar o seu uso em futuras investidas na direção da construção de uma teoria para sistemas inteligentes.

### REFERÊNCIAS

- Beni, G., Wang, J. (1989) Swarm Intelligence in Cellular Robotic Systems, *Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, Tuscany, Italy, June 26–30.
- Carvalho, M. B. de (2007) Aplicações de meta-heurística genética e fuzzy no sistema de colônia de formigas para o problema do caixeiro viajante. *Tese de Mestrado*, DT-FEEC-UNICAMP.
- Dorigo, M., Gambardella, L.M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem, *Evolutionary Computation, IEEE Transactions on*, 1(1):53-66.
- Dorigo, M., Stützle, T. (2009) Ant Colony Optimization: Overview and Recent Advances. *Technical Report TR/IRIDIA/2009-013*, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium.
- Gudwin, R. R. (1998). On the Generalized Deduction, Induction and Abduction as the Elementary Operators within Computational Semiotics. *Proceedings of ISAS'98, Intelligent Systems and Semiotics, International Conference on*, 14-18/September, Gaithersburg, USA, 1998, pp. 795-800.