

Estudo das abordagens do algoritmo de colônia de formigas para o problema do caixeiro viajante.

Julia Manfrin Dias¹, Christiane R. S. Brasil¹

¹Faculdade de Computação – Universidade Federal de Uberlândia (UFU)
CP 38408-100 – Uberlândia – MG – Brazil

{juliamanfrindias, christiane.ufu}@gmail.com

***Abstract.** Among the various algorithms in the Bioinformatics area, the Ants Colony Optimization Algorithm can be noted for its efficiency and simplicity. The main objective of this work is to apply this algorithm for the travelling salesman problem_ a problem classified as NP-hard _ comparing with other results obtained in the literature in order to analyze the performance of this method in two different approaches: AS and ACS.*

***Resumo.** Dentre os diversos algoritmos da área de Bioinformática, o Algoritmo de Colônia de Formigas pode ser destacado por sua eficiência e simplicidade. O objetivo principal deste trabalho é aplicar o algoritmo ACO para o problema do caixeiro viajante_ um problema classificado como NP-difícil _ comparando com outros resultados já obtidos na literatura, a fim de analisar o desempenho deste método em duas abordagens diferentes: AS e ACS.*

1. Introdução

A área de Bioinformática vem crescendo notavelmente no âmbito mundial, uma vez que diversos pesquisadores dos mais diferentes lugares têm desenvolvido métodos computacionais inspirados na fenômenos naturais para auxiliar na busca de soluções para problemas complexos.

Dentre esses métodos, pode-se destacar os algoritmos que mimetizam o comportamento social de colônias de insetos, que seguem uma organização dentro da sociedade em que vivem, tais como as abelhas e as formigas. Esses métodos são definidos por [Dorigo e Stutzle, 2004] como Inteligência de Exame (*Swarm Intelligence*).

Neste sentido, o comportamento social de formigas pode ser enfatizado por sua simplicidade e eficiência: as formigas encontram o menor caminho entre seu ninho e a fonte de alimento por meio do mecanismo natural das trilhas de feromônio espalhadas ao longo do percurso percorrido.

Deste modo, foram desenvolvidos algoritmos que simulam o comportamento das formigas, denominado Algoritmo de Colônia de Formigas (ACO, do inglês *Ant Colony Algorithm*) [Dorigo et al, 1996] com o objetivo de buscar soluções ou aproximações para problemas complexos em que não consegue encontrar a solução ótima em tempo polinomial. Tais problemas são conhecidos como problemas não-polinomiais (NP).

Neste contexto, um problema bastante citado na literatura como não-polinomial é o Problema do Caixeiro Viajante (PCV, também conhecido como TSP, do inglês *Travelling Salesman Problem*) [Dorigo e Gambardella, 1997], que pode ser descrito da seguinte maneira: dado um número N de cidades que devem ser visitadas por um caixeiro, deve-se encontrar a rota com o menor comprimento possível do percurso, considerando o início e o término na mesma cidade. Vale lembrar que todas as cidades são interligadas umas as outras e que cada cidade deve ser visitada apenas uma vez.

Portanto, pode-se afirmar que o problema PCV é uma aplicação de otimização combinatória, cujo custo computacional cresce exponencialmente de acordo com o aumento do número de cidades. Deste modo, o PCV pode ser classificado como NP-difícil.

É importante ressaltar que o problema PCV simula várias situações reais, como problemas de roteamento de veículos, redes de distribuição postal, ou diversos outros problemas que podem ser modelados com grafos, em termos de ciclos Hamiltonianos¹ em grafos completos².

2. Objetivo

O objetivo principal deste trabalho é aplicar o algoritmo ACO para o problema do caixeiro viajante, comparando com outros resultados já obtidos na literatura, a fim de ratificar a eficiência deste método, considerando duas abordagens diferentes de implementação efetuadas: *Ant System* (AS) e *Ant Colony System* (ACS).

3. Metodologia

Otimização por Colônia de Formigas (do inglês, *Ant Colony Optimization - ACO*) é um método de otimização de problemas baseado no comportamento de formigas na busca de alimentos. Essa heurística foi desenvolvida por Marco Dorigo na década de 1990 [Dorigo et al., 1996]. As formigas confeccionam vários caminhos da colônia até o local onde há o alimento e vão depositando feromônio ao longo dele, formando trilhas. As demais formigas da colônia seguem as trilhas de feromônios e localizam o caminho mais curto.

Os algoritmos de ACO simulam este processo que ocorre na natureza, utilizando formigas artificiais. O algoritmo, por meio de iterações, gera uma solução com base na informação heurística e na informação do feromônio encontrado nas trilhas. A cada iteração, a informação do feromônio é atualizada, considerando-se a evaporação e o depósito da substância. Ao construir a solução, a formiga faz um cálculo de probabilidade, o qual determina o caminho que deve seguir.

Existem duas abordagens diferentes implementadas neste trabalho: *Ant Colony System* (ACS) e *Ant System* (AS). O ACS difere do AS nas fórmulas usadas e na atualização do feromônio. O ACS atualiza o feromônio toda vez que uma formiga completa seu caminho, diferentemente do AS que atualiza depois que todas as formigas percorrem o

¹ Ciclo hamiltoniano é aquele que contém todos os vértices do grafo exatamente uma vez, com exceção dos vértices inicial e final, que têm que ser o mesmo.

² Grafo completo é aquele que todo vértice é adjacente a todos os outros que compõem este grafo.

caminho. Portanto, o ACS considera a atualização do feromônio a cada iteração, tornando-o mais robusto e com melhor desempenho.

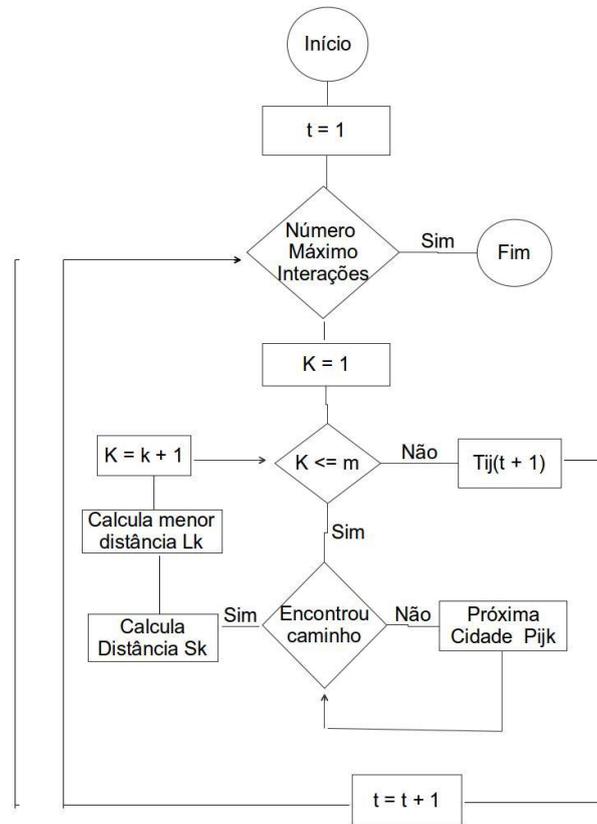


Figura 1. Fluxograma do algoritmo ACS.

Este fluxograma (Figura 1) representa um algoritmo ACS, onde i é o contador que definirá o número máximo de interações do algoritmo. A quantidade de formigas é representada por m , k é o contador de formigas, que varia de 1 a m . A formiga k deve encontrar um caminho até seu destino fazendo um cálculo probabilístico, que determina qual cidade será visitada. Assim que a formiga encontra o caminho, ela calcula a distância e armazena o resultado na variável Sk e a variável Lk armazena a menor distância encontrada até o momento. Depois que cada uma das m formigas encontram seus caminhos, o feromônio é atualizado.

Tabela 1. Comparativo entre as formulas de atualização de feromônio do AS e ACS.

Alg.	Expressão	Realizado
AS	$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k$	por todas as formigas ao completar o caminho
ACS	$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0$	por todas as formigas em cada passo da construção

No *Ant System* (AS) os valores para a importância da informação heurística β é 2 e para α é 0.1. Estes valores são indicados por Dorigo em seus estudos. A quantidade de feromônio inicia com valor 1.0 para todas as arestas.

O *Ant Colony System* (ACS) usa parâmetros iguais para α e β , e inicia o feromônio em todas as arestas com valor 1.0.

Note que α e β são parâmetros para indicar respectivamente a importância do feromônio e da informação heurística. τ_{ij} é a quantidade de feromônio na aresta ij , η_{ij} é o inverso de d_{ij} e N_i^k o conjunto das cidades ainda não visitadas pela formiga k .

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}$$

Figura 2. Fórmula da probabilidade de visitar uma cidade.

Deste modo, calcula-se a probabilidade de todas as cidades que a formiga ainda não visitou e escolhe ir para aquela de maior probabilidade.

4. Resultados

Usando um conjunto de 42 cidades obtido da base TSPLIB (<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/dantzig42.tsp>) e aplicando duas abordagens do ACO, observa-se que o algoritmo apresenta uma solução satisfatória.

Foram implementadas duas versões do ACO: o *Ant System* (AS) e o *Ant Colony System* (ACS).

Segundo o artigo, a solução ótima é 699 e foi obtida usando outra abordagem de ACO denominada MMAS (Carvalho e Ramos, 2007) e o ACS apresenta a solução em torno de 815. Considerando que a abordagem MMAS é mais elaborada em relação a atualização do feromônio. MMAS usa preferencialmente, ou na maioria das iterações, a melhor formiga da iteração, obtendo, deste modo, um efeito de intensificação e de diversificação, uma vez que a cada iteração, tem-se uma diferente melhor solução.

Portanto, pode-se notar que este fator de atualização do feromônio é fundamental para melhores aproximações da solução ótima. Observe que AS apresenta resultados inferiores a todos, uma vez que sua atualização é a menos elaborada.

Tabela 1. Resultados obtidos com abordagens diferentes de ACO para a base dantzig42.tsp usando 100 iterações e 42 formigas.

Alg.	Solução aproximada
AS	1199.993896
ACS	821.765747
MMAS	699

Tabela 2. Resultados obtidos com abordagens diferentes de ACO para a base burma14.tsp usando 40 iterações e 14 formigas.

Alg.	Solução aproximada
AS	40.011627
ACS	31.882530
Solução ótima	30.879

Observe que para um conjunto menor de cidades (<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/burma14.tsp>), a aproximação alcançada foi bem mais próxima da ótima usando o método ACS.

6. Conclusão

Neste trabalho pode-se obter resultados preliminares que indicam que por meio de técnicas mais elaboradas de atualização de feromônio é possível conseguir melhores resultados para o problema do caixeiro viajante. Isto pode ser observado com as abordagens AS e ACS, a primeira sendo menos elaborada que a segunda. Outro detalhe importante que foi analisado é que para uma base pequenas com 14 cidades, como no caso de burma14.tsp, os métodos não diferem tanto no desempenho, uma vez que a complexidade do grafo tratado é relativamente simples, se aproximando bem da solução ótima. Em contrapartida, para um grafo mais complexo, com 42 cidades no caso de dantzig42.tsp, pode-se perceber uma diferença notável no desempenho entre AS e ACS, sendo que até o ACS sendo mais elaborado ainda obtém um resultado distante da solução ótima.

Portanto, para trabalhos futuros pretende-se desenvolver a abordagem MMAS, a técnica mais eficiente mencionada neste trabalho, visando atingir a solução ótima, a fim de aplicar esta abordagem em problemas mais complexos inspirados em situações reais.

7. Referências

- Carvalho, E. R e Ramos, G. S. (2007) “Otimização por Colônia de Formigas (*Ant Colony Optimization - ACO*)”.
- http://www.inf.ufpr.br/aurora/disciplinas/topicosia2/downloads/trabalhos/ACO_TSP.pdf
- Dorigo, M. e Gambardella, L.M. (1997) “Ant colony system: A cooperative learning approach to the traveling salesman problem”. *IEEE Transactions on Evolutionary Computation*, Vol.1, No.1, pages 1–24.
- Dorigo, M., Maniezzo, V., e Colomi, A. (1996) “Ant system: Optimization by a colony of cooperatings agents”. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1):1–13.
- Dorigo, M. e Stutzle, T. (2004) “Ant Colony Optimization”. MIT Press Cambridge.