

PLANEJAMENTO DE ROTAS PARA ROBÔS MÓVEIS AUTÔNOMOS UTILIZANDO COLÔNIA DE FORMIGAS

Matheus Gomes Araujo¹; Matheus Giovanni Pires²

1. Bolsista FAPESB, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: matheus.araujo.ecomp@gmail.com
2. Orientador, Departamento de Ciências Exatas, Universidade Estadual de Feira de Santana, e-mail: mgpires@ecompu.uefs.br

PALAVRAS-CHAVE: Planejamento de Rotas, Robôs Móveis Autônomos, Colônia de Formigas.

INTRODUÇÃO

Sistemas capazes de automatizar processos com o objetivo de preservar a integridade humana apresentam um apelo muito interessante do ponto de vista científico. Existem várias tarefas do mundo real que oferecem perigo a integridade humana, tais como, inspeção interna de dutos de transporte de petróleo, limpeza de sistemas de ar-condicionado, a pulverização de inseticidas em estufas (Mandow, 1996), detecção e salvamento de pessoas soterradas em escombros de estruturas danificadas (Murphy, 2000), entre outras. Neste contexto, os robôs móveis autônomos podem realizar tarefas que seriam impossíveis e/ou perigosas para os seres humanos (Bay, 1995).

A abordagem baseada em Colônia de Formigas foi apresentada inicialmente por Dorigo *et al.* (1996), propondo um paradigma computacional para a resolução de problemas combinatórios baseado no comportamento encontrado ao observar formigas se movendo entre a colônia e a fonte de alimento. As formigas reais são capazes de encontrar o caminho mais curto para uma fonte de alimento do formigueiro sem a utilização de dados visuais. Enquanto caminham, as formigas depositam no solo uma substância denominada feromônio (designação genérica de substâncias secretadas pelas formigas que servem de meio de comunicação entre elas), e tem seu deslocamento baseado em trilhas de feromônios previamente depositados por outras formigas. Estas trilhas de feromônios podem ser observadas por outras formigas e motivá-las em seguir determinado caminho, isto é, um movimento aleatório das formigas segue com maior probabilidade uma trilha de feromônio. Esta é uma maneira de como as trilhas são reforçadas e mais e mais formigas tendem a seguir aquela trilha (Tavares Neto & Coelho, 2004).

Este trabalho tem o objetivo de desenvolver uma abordagem baseada em colônia de formigas para o planejamento de rotas para robôs móveis autônomos. O planejamento de rotas consiste em definir o melhor caminho entre dois pontos de um mapa, do qual o robô deverá percorrer para cumprir o objetivo em questão, como por exemplo, a entrega de medicamentos em um hospital ou a inspeção de um sistema de encanamentos.

METODOLOGIA

O problema de planejamento de rotas consiste em definir o melhor caminho entre dois pontos de um mapa, do qual o robô deverá percorrer para cumprir uma determinada tarefa. Dependendo do tipo da aplicação, o melhor caminho pode significar o menor ou o mais rápido, ou até mesmo uma combinação de vários critérios. Neste trabalho, com o intuito de facilitar a realização dos testes e também gerar resultados que pudessem ser comparados com outras abordagens encontradas na literatura, optou-se por utilizar dados do problema do caixeiro viajante (*Traveling Salesman Problem – TSP*) para testar a abordagem proposta.

O problema do caixeiro viajante consiste em encontrar o caminho de menor custo (distância) que se inicie e termine em uma cidade, passando por todas as demais cidades, sem repetição. Este problema pode ser facilmente modelado em um grafo, conforme ilustrado na Figura 1, onde cada nó do grafo representa uma cidade e os pesos das arestas representam a distância entre as cidades.

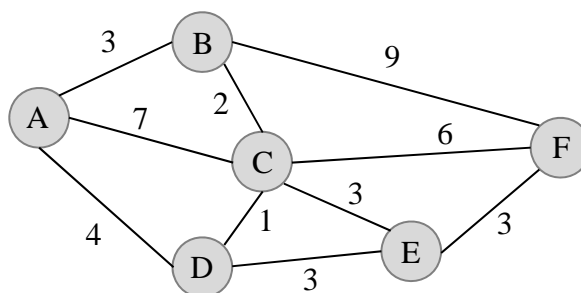


Figura 1: Exemplo de modelagem do problema do caixeiro viajante com 6 cidades.

Formalmente, o problema do caixeiro viajante pode ser definido como um grafo (N, E) , onde N é o conjunto de cidades e E o conjunto de arestas entre as cidades. A distância entre uma cidade i e j é chamada de d_{ij} .

Neste trabalho foram implementadas duas versões de algoritmos para otimização baseado em colônia de formigas – o *Ant System (AS)* e o *Ant Colony System (ACS)*. O *Ant System* foi o primeiro exemplo de algoritmo proposto por Dorigo *et al.* (1996) e o *Ant Colony System* foi proposto como uma melhoria do *AS* por Dorigo & Gambardella (1997). Embora o desempenho do *AS* não seja suficiente para competir com os melhores algoritmos propostos para resolver o *TSP*, ele é usado como ponto de partida para a formulação de outros algoritmos de otimização por colônia de formigas (Dorigo *et al.*, 1996).

A resolução do *TSP* pelo *AS* ocorre da seguinte maneira. Na fase de construção da solução, cada formiga escolhe para qual cidade ir com uma probabilidade que é uma função da distância da cidade e da quantidade de feromônio presente na aresta que conecta a essa cidade. Para não permitir passos inválidos, a formiga tem uma memória de quais cidades ela já visitou. Assim, enquanto ela constrói uma solução, ela é forçada a não visitar uma mesma cidade duas vezes até que a solução esteja completa. Quando a formiga termina um caminho completo, ela deposita certa quantidade de feromônio em cada aresta (i, j) que ela visitou (Dorigo *et al.*, 1996). No *AS*, não há depósito de feromônio durante a fase de construção da solução. Ao dar um passo, a formiga calcula a probabilidade de todas as cidades que ela ainda não visitou e escolhe ir para aquela de maior probabilidade. Após a construção da solução, a atualização do feromônio de cada aresta é realizada, a qual envolve tanto o incremento do feromônio quanto a sua evaporação.

O *ACS* introduziu a atualização do feromônio local (atualização *online*) além da atualização do feromônio realizada no fim do processo de construção do caminho (atualização *offline*). A atualização *online* é realizada em cada passo da construção da solução, ou seja, as formigas atualizam o feromônio da aresta pela qual acabaram de atravessar. Além da atualização local, o *ACS* compreende uma atualização global de feromônio. Esta atualização é feita somente pela formiga que construiu a melhor solução até o momento.

ANÁLISE E DISCUSSÃO DOS RESULTADOS

Os dados utilizados nos experimentos foram obtidos do repositório TSPLIB (disponível em <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95>) e foram considerados apenas os problemas simétricos, ou seja, a distância entre duas cidades i e j é igual à distância entre as cidades j e i ($d_{ij} = d_{ji}$). A escolha dos conjuntos de dados foi determinada pela disponibilidade de resultados obtidos por outros trabalhos encontrados na literatura, quando estes se utilizaram de um determinado conjunto de dados.

Primeiramente, o algoritmo *Ant System* foi testado com o conjunto Oliver30. Este conjunto se refere a um problema do caixeiro viajante simétrico com 30 cidades. A Tabela 1 mostra a comparação dos resultados obtidos pelo *AS* e por outros dois algoritmos, Busca Tabu

(BT) e Recozimento Simulado (RS). Os resultados descritos na Tabela 1 foram obtidos a partir de dez execuções de cada algoritmo. As colunas *Melhor*, *Média* e *Desvio Padrão* se referem ao melhor, a média e o desvio padrão dos dez resultados obtidos, respectivamente.

Tabela 1: Resultados obtidos pelo AS, BT e RS para o conjunto Oliver30.

	<i>Melhor</i>	<i>Média</i>	<i>Desvio padrão</i>
AS	423,74	425,82	2,5
BT	420	420,6	1,5
RS	422	459,8	25,1

A partir dos resultados ilustrados na Tabela 1, pode-se constatar que o melhor resultado para o problema Oliver30 foi obtido pela Busca Tabu. O segundo melhor resultado foi obtido pelo Recozimento Simulado, porém, o desvio padrão apresentado foi o maior entre todas as abordagens. Isto significa que a variação das respostas é muito alta, o que demonstra uma fragilidade da abordagem.

O *Ant Colony System* foi comparado com três abordagens: Algoritmos Genéticos (AG), Recozimento Simulado (RS) e Programação Evolutiva (PE). Nesta comparação foram utilizados três conjuntos de dados: Eil50, Eil75 e KroA100. O Eil50, Eil75 e KroA100 caracterizam um problema do caixeiro viajante simétrico com 50, 75 e 100 cidades, respectivamente.

A Tabela 2 ilustra para cada abordagem duas colunas. A primeira mostra o melhor resultado encontrado entre 10 execuções e a segunda coluna mostra o número de iterações executadas pelo algoritmo. Os resultados que não estão disponíveis na literatura estão descritos como N/A.

Tabela 2: Resultados obtidos pelo ACS, AG, PE e RS para os conjuntos Eil50, Eil75 e KroA100.

	<i>ACS</i>	<i>ACS</i>	<i>AG</i>	<i>AG</i>	<i>PE</i>	<i>PE</i>	<i>RS</i>	<i>RS</i>
	<i>Melhor</i>	<i>Iteração</i>	<i>Melhor</i>	<i>Iteração</i>	<i>Melhor</i>	<i>Iteração</i>	<i>Melhor</i>	<i>Iteração</i>
Eil50	427	2000	428	25000	426	100000	443	68512
Eil75	542	4000	545	80000	542	325000	580	173250
KroA100	21320	6000	21761	103000	N/A	N/A	N/A	N/A

A partir dos resultados ilustrados na Tabela 2, pode-se constatar que os melhores resultados encontrados são do ACS, exceto para o conjunto Eil50, onde a abordagem PE obteve melhor desempenho. Para o conjunto Eil75, as abordagens ACS e PE obtiveram a mesma solução, porém, o ACS encontrou a solução em um número de iterações muito menor em relação à abordagem PE. Isto acontece não somente para este conjunto, mas também para os conjuntos Eil50 e KroA100.

CONCLUSÃO

Este trabalho teve como objetivo apresentar uma abordagem baseada em colônia de formigas para a resolução do problema de planejamento de rotas para robôs móveis autônomos. O planejamento de rotas consiste em definir o melhor caminho entre dois pontos de um mapa, do qual o robô deverá percorrer para cumprir uma determinada tarefa, como por exemplo, a entrega de medicamentos em um hospital ou a inspeção de um sistema de encanamentos.

Neste trabalho foram implementadas duas versões de algoritmos para otimização baseado em colônia de formigas – o *Ant System (AS)* e o *Ant Colony System (ACS)*. O *Ant System* foi o primeiro exemplo de algoritmo proposto por Dorigo *et al.* (1996) e o *Ant Colony System* foi proposto como uma melhoria do *AS* por Dorigo & Gambardella (1997).

A partir da análise dos resultados constatou-se que o *Ant System* não possui um bom desempenho em comparação com o algoritmo Busca Tabu. Por outro lado, os seus resultados são mais estáveis quando comparado com os resultados do algoritmo Recozimento Simulado. Em relação ao *Ant Colony Systems*, a maioria dos resultados obtidos foram melhores aos encontrados pelas abordagens Algoritmos Genéticos, Recozimento Simulado e Programação Evolutiva. Além disso, o *ACS* foi a abordagem que encontrou a solução em menor tempo. Portanto, o *ACS* se mostrou uma abordagem eficiente para o problema do planejamento de rotas.

REFERÊNCIAS BIBLIOGRÁFICAS

- MANDOW, A.; GOMEZ-DE-GABRIEL, J.M.; MARTINEZ, J.L.; MUNOZ, V.F.; OLLERO, A.; GARCIA-CEREZO, A. 1996. The autonomous mobile robot Aurora for greenhouse operation. *IEEE Robotics & Automation Magazine*. 3(4): 18-28.
- MURPHY, R.R. 2000. Marsupial and Shape-Shifting Robots for Urban Search and Rescue. *IEEE Intelligent Systems*. 15(2): 14-19.
- BAY, J. S. 1995. Design of the army-ant cooperative lifting robot. *IEEE Robotics & Automation Magazine*. 2(1): 36-43.
- DORIGO, M.; MANIEZZO, V.; COLORNI, A. 1996. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*. 26(1): 29-41.
- TAVARES NETO, R.F.; COELHO, L.S. 2004. Colônia de Formigas: Uma abordagem Promissora para Aplicações de Atribuição Quadrática e Projeto de Layout. In: XXIV ENEGEP - Encontro Nacional de Engenharia de Produção.
- DORIGO, M.; GAMBARDELLA, L. M. 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*. 1(1): 1-24.