

Avaliação de Algoritmos de Busca aplicados a um Simulador de Braço Robótico

Bruno Brandoli Machado, Jonathan de Andrade Silva, Wesley Nunes Gonçalves

¹ Universidade Católica Dom Bosco – UCDB,
Av. Tamandaré, 6000, Jardim Seminário,
Campo Grande, MS, Brasil, 79117-900

{bmachado, jsilva, wnunes}@acad.ucdb.br

Abstract. *In this article, a library for the evaluation of the search algorithms was implemented. The search algorithms are capable automatically to find a solution for data problem.*

Resumo. *Nesse artigo, foi implementada uma biblioteca para a avaliação dos algoritmos de busca. Os algoritmos de busca são capazes automaticamente encontrar uma solução para um dado problema.*

1. Introdução

Uma das mais notáveis e impressionantes habilidades do nosso cérebro é a capacidade de tomar decisões. Este processo, na maioria das vezes, é desenvolvido por regras expandidas das assimilações de suas experiências. Estas experiências vão sendo adquiridas ao longo do tempo, sendo que o grau de maior desenvolvimento do cérebro humano ocorre durante os primeiros anos de vida [Haykin 1999].

A implementação de um modelo computacional capaz de tomar decisões de forma eficiente é uma tarefa complexa. O cérebro humano tem seus próprios processos para tomada de decisão que podem apenas serem desempenhados aproximadamente em modelos de inteligência artificial. Um dos modelos utilizados são os algoritmos de busca. Esses algoritmos encontram uma sequência de ações que levam o agente a um estado meta.

O homem tem criado máquinas que desempenham tarefas de forma autônoma. Uma aplicação dessas máquinas é a simulação de um braço robótico em ambientes conhecidos. O braço robótico possui a capacidade de movimento similar ao braço humano, sendo amplamente utilizado na indústria (e.g. indústria automotiva e portuária). A ação motora do braço pode substituir a mão-de-obra humana em tarefas complexas, inseguras ou repetitivas (e.g. combate ao fogo, exploração espacial).

O objetivo deste trabalho é a implementação de um simulador de braço robótico autônomo. Esse braço robótico é formado por duas articulações com movimentos livres (ombro e cotovelo). Este presente estudo utiliza técnicas de inteligência artificial visando encontrar o caminho ótimo até o alvo. O alvo é um objeto que se pretende alcançar em uma posição qualquer no espaço tridimensional. O caminho do braço até o alvo é determinado pela busca em um espaço de estados, sendo avaliada a confiabilidade e o custo das soluções dos algoritmos de busca.

Este artigo está estruturado em sete seções. Na seção 2, são apresentados os principais trabalhos correlatos na área. A seção 3 define os conceitos e algumas técnicas de inteligência artificial. Na próxima seção, são descritas as implementações e as ferramentas utilizadas para realização dos experimentos. Na seção 5, são realizados os experimentos aplicados ao problema do braço robótico. A seção 6 apresenta os resultados obtidos da avaliação dos algoritmos de busca. Finalmente, na seção 7 são avaliadas as limitações e a eficiência de cada algoritmo de busca.

2. Trabalhos Correlatos

Os algoritmos de busca têm sido aplicados em diversos problemas de inteligência artificial. Os passos para solução desses problemas, na maioria das vezes, não são conhecidos, mas freqüentemente a solução é determinada pela exploração de alternativas através das buscas [Korf 1999]. Em [Korf 1999] são descritos os principais algoritmos de busca.

As buscas tabu e têmpera simulada foram aplicadas na solução do problema de alocação de salas [Souza et al. 2002]. A solução inicial é alcançada por um procedimento construtivo parcialmente guloso. Em seguida, essa solução é aplicada à têmpera simulada. Por fim, a solução é refinada pela busca tabu. A função heurística utilizada é dividida em requisitos essenciais que são alocações que devem ser satisfeitas e requisitos não-essenciais cujo atendimento é desejável. A função heurística é dada como a soma dessas duas características.

Formulações paralelas para o algoritmo A* foram apresentadas em [Kumar et al. 1988]. Diversos processadores são utilizados para resolução do problema de busca, o que aumenta o desempenho no processamento da solução final. Essas formulações do algoritmo A* são aplicadas aos problemas do caixeiro viajante e cobertura de vértices.

Em [Taylor and Korf 1993] a busca em profundidade é avaliada. Apesar dessa busca necessitar de memória constante, muitas vezes a solução não é encontrada devido a ciclos gerados pelos operadores, implicando em um *loop*. Taylor apresenta uma nova técnica para eliminação desses nós duplicados baseados em uma máquina de estados finitos e detecção de ciclos. Esta nova busca em profundidade é testada com os problemas do quadrado e cubo mágico.

Sam [Thangiah et al. 1994] define têmpera simulada (*simulated annealing*) como uma estratégia de busca que permite realizar movimentos ruins com base em uma probabilidade. Com isso, a busca permite escapar de máximos ou mínimos locais, inspirados no processo físico termodinâmico (e.g. produção de cristais). Neste artigo, o algoritmo foi modelado para definir rotas de veículos, diminuindo o tempo de deslocamento dos veículos ao seu destino.

A combinação dos algoritmos de busca em largura e profundidade, permite percorrer vários estados utilizando poucos recursos de memória [Ravi and Somenzi 1995]. Estes algoritmos foram utilizados para um grande conjunto de entradas no processamento de imagens, cuja heurística é baseada no limiar que define o subconjunto de estados gerados por um diagrama de decisão (*Binary Diagram Decision*).

O algoritmo de descida de gradiente para encontrar o mínimo global é proposto em [Thrun et al. 2000]. Neste artigo é descrito um método para localização e mapeamento

de robôs em tempo real. Este algoritmo não armazena o histórico dos estados utilizando somente o estado atual. Logo possui um custo de armazenamento constante.

3. Algoritmos de Busca

Os algoritmos de busca atuam em um ambiente de espaço de estados. A modelagem de um problema implica em definir diversas características. Segundo Korf [Korf 1999], essas características consistem em um conjunto de estados e operadores de um problema, além do estado inicial e final.

Os algoritmos de busca modelados em um espaço de estados, geralmente, são divididos em buscas exaustivas, heurísticas e locais.

3.1. Buscas Exaustivas

As buscas exaustivas não requerem nenhum conhecimento específico sobre o problema. Apenas é necessário que o problema seja modelado em um espaço de busca, um estado inicial, um conjunto de operadores válidos e um ou mais estados finais.

Diversas buscas foram propostas nessa categoria. As mais importantes, segundo Korf [Korf 1999] são busca em largura, busca em profundidade, busca em aprofundamento iterativo e busca bidirecional.

3.2. Buscas Heurísticas

Para solução de problemas mais complexos, geralmente é necessário um conhecimento adicional. Este conhecimento é estimado através de uma função heurística. A heurística consiste em medir a probabilidade de um nó convergir para uma solução.

A expansão dos nós será guiada por uma heurística estabelecida. Diversos algoritmos utilizam essa estratégia. Os algoritmos mais conhecidos são as buscas gulosa e A*, e suas variações.

3.3. Buscas Locais

As buscas locais se baseiam na idéia de melhorar iterativamente o estado atual. Essa melhora é concentrada na heurística, ou seja, o algoritmo escolhe o próximo estado que seja melhor que o estado atual. Alguns algoritmos aceitam movimentos de piora para tentar escapar dos máximos e mínimos locais.

Esses algoritmos armazenam apenas uma quantidade constante de estados, consumindo assim um espaço de memória constante. Apesar de não preservarem todos os estados não visitados, na maioria das vezes, são os melhores métodos para lidar com problemas reais e complexos. Os algoritmos mais conhecidos são descida de gradiente ou subida de encosta, têmpera simulada, busca em feixe local e busca tabu.

4. Implementação

Para a implementação deste trabalho foi utilizada a linguagem Java, por ser portátil entre os sistemas operacionais e por fornecer mecanismos de programação orientado a objetos. Para construção do braço robótico, onde as buscas serão aplicadas, foi utilizada a API Java3D. O Java3D é uma extensão da linguagem Java, que oferece uma aplicação de alto nível para controle gráfico e descrição de cenas em espaços tridimensionais.

O principal resultado deste trabalho é a implementação de uma biblioteca para algoritmos de busca. A biblioteca oferece suporte aos algoritmos de busca em Largura, Profundidade, Profundidade Limitada, Aprofundamento Iterativo, Busca Gulosa, Subida de Encosta, Descida de Gradiente e Têmpera Simulada. Para a utilização da biblioteca é necessário modelar o problema em uma classe. Nessa classe são necessários os métodos que descrevem o conjunto de operadores e o custo de geração dos estados. Além disso, a classe deve conter o estado inicial, final e os atributos do problema.

O braço robótico foi construído com dois objetos cilíndricos capazes de rotacionar sob duas articulações representadas por esferas, uma delas simulando o ombro e a outra o cotovelo, como mostra a Figura 1.a. Esses objetos são nativos de um conjunto de classes da API Java3D. Os movimentos do braço até o alvo são realizados através de transformações de rotação e translação nos objetos que constituem o braço. Com eventos do *mouse* é possível visualizar o braço de diferentes ângulos. O controle motor do braço pode ser guiado pelos algoritmos de busca ou manualmente através de botões presentes na *interface*. O alvo será representado por uma esfera e seu posicionamento deve ser informado pelos campos das coordenadas do ambiente virtual.

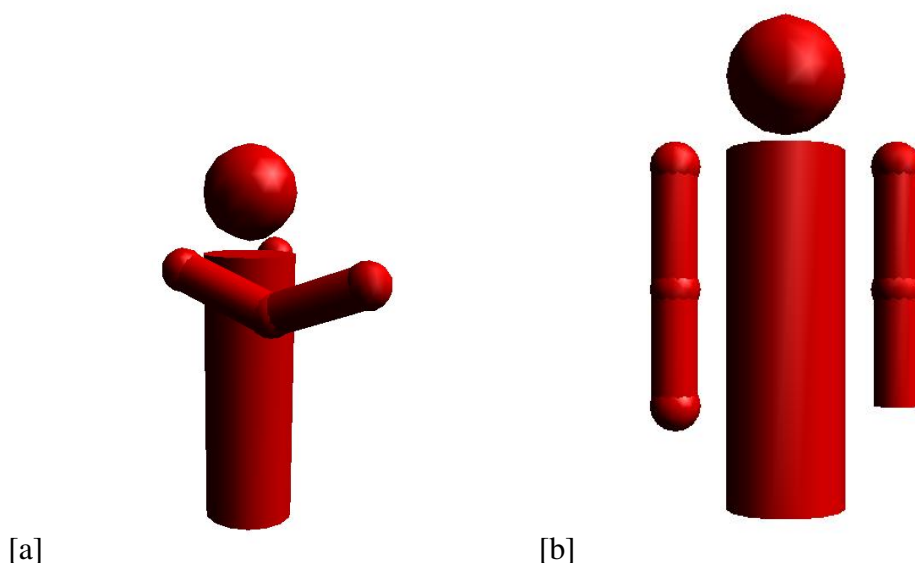


Figura 1. *Movimento do Braço Robótico.*

5. Experimentos

Nesta seção, são descritos os experimentos, cujo objetivo é locomover o braço até o alvo estabelecido pelo usuário. O braço sempre será iniciado na mesma posição mostrado na Figura 1.b. Os testes foram executados em ambiente Intel Pentium 4, 2.8 *GHertz* e 256 *MBytes* de memória com sistema operacional GNU/Linux Fedora Core 5 (*kernel* 2.6.16-1.2122.FC5). A versão da Máquina Virtual Java foi a 1.4.2_09 e a versão da API Java3D foi a 1.3.1.

Para a realização dos experimentos o alvo foi posicionado em uma coordenada gerada de forma manual. A posição está ao alcance do braço. Essa posição gerada foi aplicada todos algoritmos de busca implementados na biblioteca. Os operadores do espaço de

estados são representados pelos movimentos verticais, horizontais e rotações simulando o braço e o ante-braço.

6. Resultados e Análise

Nesta Seção, encontram-se avaliações do desempenho dos algoritmos de busca dado um posicionamento do alvo no espaço tridimensional. Na Tabela 1 são listados os algoritmos da biblioteca. Ainda na Tabela 1 são apresentados os resultados, quantificados em profundidade atingida pela busca executada, os nós visitados do conjunto de todos os estados acessíveis e tempo de execução para atingir o alvo partindo da posição inicial do braço.

Nota-se que o desempenho dos algoritmos de busca não informada, com movimentos complexos, não obtiveram sucesso devido ao custo de memória. A Tabela 1 mostra o resultado de um movimento complexo. O conjunto de movimentos avaliado foi dois movimentos do braço para esquerda, dois movimentos para cima, dois movimentos do ante-braço para cima, duas rotações do braço para esquerda e dois movimentos para cima.

Entretanto, no resultado obtido existe uma margem de erro associado. Essa margem é aceitável pois a posição do braço é próxima ao valor do alvo, como pode ser vista na Tabela 1.

Algoritmos	Profundidade	Nós Visitados	Tempo	Erro
Busca em Largura	*	*	*	*
Busca em Profundidade	**	**	**	**
Busca com Profundidade Limitada	**	**	**	**
Aprofundamento Iterativo	**	**	**	**
Busca Gulosa	8	32	150 ms	0.0333
Descida de Gradiente	8	32	180 ms	0.0333
Têmpera Simulada	***	***	***	***

Tabela 1. Avaliação dos Algoritmos de Busca.

* Representa falta de memória. ** Representa loop. *** Representa um mínimo local.

7. Conclusão e Trabalhos Futuros

A grande contribuição deste trabalho é a integração dos algoritmos de busca à problemas de simulação robótica em ambiente virtuais. Outra contribuição é a comparação entre os algoritmos de busca.

Um trabalho futuro é a realização em um conjunto maior e mais significativo de testes para avaliação dos algoritmos de busca. Além disso os testes podem ser realizados com diferentes ângulos de rotação do braço.

Referências

- Haykin, S. (1999). *Neural Networks. A Comprehensive Foundation*. Prentice Hall, New Jersey, USA.
- Korf, R. (1999). Artificial intelligence search algorithms. In *Algorithms and Theory of Computation Handbook*.
- Kumar, V., Ramesh, K., and Rao, V. N. (1988). Parallel best-first search of state-space graphs: A summary of results. In *National Conference on Artificial Intelligence*, pages 122–127.
- Ravi, K. and Somenzi, F. (1995). High-density reachability analysis. In *Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design*, pages 154–158, Washington, DC, USA. IEEE Computer Society.
- Souza, M. J. F., Martins, A. X., and de Araújo, C. R. (2002). Experiências com simulated annealing e busca tabu na resolução do problema de alocação de salas. In *XXXIV Simpósio Brasileiro de Pesquisa Operacional*.
- Taylor, L. A. and Korf, R. E. (1993). Pruning duplicate nodes in depth-first search. In *National Conference on Artificial Intelligence*, pages 756–761.
- Thangiah, S. R., Osman, I. H., and Sun, T. (1994). Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows. Technical report, Computer Science Department, Slippery Rock University.
- Thrun, S., Burgard, W., and Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *The IEEE International Conference on Robotics & Automation*.