

Problema 8-Puzzle: Análise da solução usando Algoritmos Genéticos e Backtracking

Nelson Florêncio Junior, Frederico Gadelha Guimarães
PPGCC - Programa de Pós-Graduação em Ciência da Computação
UFOP - Universidade Federal de Ouro Preto
Ouro Preto, Minas Gerais, Brasil
email: sidnelson21@gmail.com, fredericoguimaraes@ufmg.br

Resumo—Com jogos de tabuleiro, como o 8-Puzzle, onde existem regras bem definidas podemos gerar um espaço de busca bem definido, porém algumas soluções podem ser custosas para o desempenho de um determinado computador. Dado o problema podemos resolvê-lo com uma busca de estados em um grafo, onde cada nó do grafo representa um estado (configuração) do puzzle. Deve-se então encontrar um caminho num grafo que leve ao estado final, no caso um puzzle ordenado. Os Algoritmos Genéticos, pertencentes a Computação Evolutiva, proporcionam uma busca inteligente, evoluindo baseando-se nos resultados obtidos de testes anteriores, diferentemente do algoritmo baseado no paradigma *Backtracking* (Tentativa e Erro), onde testamos todos os caminhos possíveis num grafo até encontrar a solução. Desenvolvendo os algoritmos e comparando-os seus resultados através de uma análise de complexidade podemos chegar a uma conclusão de qual paradigma se encaixa melhor ao problema ou em que situação cada um se encaixa melhor.

Keywords-8-Puzzle, Grafos, Algoritmos Genéticos e Backtracking.

I. INTRODUÇÃO

O jogo do 8-Puzzle é um jogo de tabuleiro de blocos deslizáveis. Existem diversas maneiras de solucioná-lo, tais como algoritmos de combinação, busca em largura, profundidade, técnicas de busca direcional, entre diversas outras. O problema é bastante abordado na disciplina de Inteligência Artificial, segundo [1] além do seu apelo intelectual inerente, os jogos de tabuleiro tem certas propriedades que os tornaram objetos de estudo ideal para trabalhos iniciais.

Dentre as técnicas de resolução do 8-puzzle pode-se citar a linha de pesquisa em Computação Evolutiva que é dividida em três subáreas: Algoritmos Genéticos, Estratégias de Evolução e Programação Genética. Segundo [2] Algoritmos Genéticos são algoritmos de otimização global, baseados nos mecanismos de seleção natural e da genética. Eles empregam uma estratégia de busca paralela e estruturada, embora aleatória, direcionada à busca de pontos de “alta aptidão”, ou seja, pontos os quais a função pode ser maximizada ou minimizada.

Segundo [1], a teoria da busca em espaço de estados é a nossa ferramenta primordial para responder a estas questões, representando este problema como um grafo de espaço de estados, podemos usar a teoria dos grafos para analisar a estrutura e a complexidade tanto do problema como do procedimento de busca que empregamos para resolvê-lo.

II. JUSTIFICATIVAS

Os jogos de tabuleiros facilitam o aprendizado em Inteligência Artificial, isso devido a suas regras bem estabelecidas possibilitando melhor entendimento do espaço de busca se compararmos com problemas mais complexos da vida real. Sendo assim para o início de uma pesquisa em Computação Evolutiva seria interessante desenvolver uma aplicação envolvendo Algoritmos Genéticos e comparar com algum método convencional, no caso o *Backtracking*. Segundo [2] as técnicas convencionais de busca possuem limitação de sua natureza serial. Com a expansão do processamento paralelo, técnicas de otimização paralelas serão largamente beneficiadas, o que é o caso dos Algoritmos Genéticos.

Numa pesquisa semelhante realizada nos Estados Unidos, segundo [3] os resultados encontrados com esta aplicação foram bastante satisfatórios. No Brasil existem poucas aplicações para este problema com características evolutivas, por isso devido a característica do trabalho ele se mostra bem interessante.

III. OBJETIVOS

O objetivo deste trabalho é desenvolver dois algoritmos que resolvam o problema do 8-Puzzle, um baseado no paradigma de *Backtracking* e outro em Computação Evolutiva, especificamente Algoritmos Genéticos. Fazer uma análise de complexidade entre os dois algoritmos e apresentar os resultados.

IV. METODOLOGIA

A metodologia a ser adotada para realização deste trabalho é a seguinte:

- Conceito de busca de espaço de estados, para do problema do 8-Puzzle. Para representarmos os estados usaremos uma estrutura de grafos, percorrendo seus vértices até encontrar uma solução ou ponto de parada. A Figura 1 mostra o estado inicial, com os números embaralhados, e o estado final com os números ordenados. Cada estado (configuração do puzzle) representa um nó no grafo, que dependendo da posição da peça de valor zero (posição vazia) pode gerar de um até quatro novos estados. Lembrando que cada peça só pode movimentar-se em para vertical ou horizontal e

para onde está localizada a peça de valor zero (posição vazia).

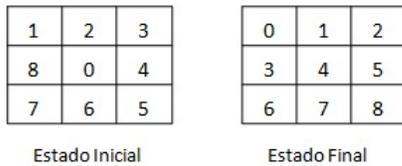


Figura 1. Estados do Puzzle

- Usando uma busca em profundidade e controlando seus caminhos, criando um limite profundidade, criaremos uma solução em *Backtracking* (tentativa e erro), testando a cada estado encontrado se o objetivo foi alcançado. A Figura 2 demonstra um estado do puzzle que gera novos nós de acordo com seus possíveis movimentos. Este método servirá como base para comparação do Algoritmo Genético.

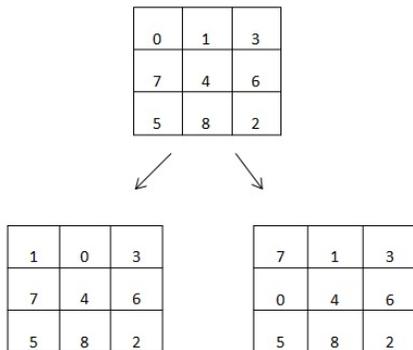


Figura 2. Expansão do grafo, a cada possível movimento

- Usando uma simulação de processos naturais de sobrevivência e reprodução pelos Algoritmos Genéticos, encontrar uma sequência de movimentos que leve a solução do 8-puzzle. Nela cada indivíduo representará uma solução, ou seja, uma sequência de movimentos realizados a cada estado do puzzle. A função de aptidão funciona como uma heurística que classificará os melhores candidatos à solução do problema e/ou geração de indivíduos mais aptos. Os operadores de crossover (cruzamento) e mutação que irão alterar os movimentos possibilitando a criação de novos indivíduos baseados nos melhores da geração anterior. A meta é que a cada geração possa encontrar um indivíduo com uma sequência de movimentos melhor.
- Desenvolver uma análise de complexidade com os resultados obtidos. A análise de algoritmos, de acordo

com [4], deve prever as necessidades de recursos dos algoritmos. Como recurso entende-se memória, processamento, *etc.* Com base nos recursos necessários é possível medir o tempo computacional do algoritmo, para que, a partir dessa informação, possa se escolher entre dois algoritmos que chegam ao resultado desejado.

- A princípio os algoritmos serão desenvolvidos na linguagem C++, no ambiente de desenvolvimento Code-Blocks 10.05.

V. RESULTADOS ESPERADOS

Algoritmos que resolvam o problema do 8-puzzle. E uma análise bem estruturada sobre ambos. Espera-se que, através da análise de complexidade, o problema proposto seja resolvido em menor tempo com Algoritmo Genético do que com algoritmo baseado em Tentativa e Erro (*Backtracking*).

REFERÊNCIAS

- [1] G. F. Luger, *Inteligência Artificial: estruturas e estratégias para a solução de problemas complexos*, 4th ed. Bookman, 2004, ISBN 85-363-0396-4.
- [2] S. O. Rezende, *Sistemas Inteligentes: Fundamentos e Aplicações*, 1st ed. RECOPE-IA Rede Cooperativa de Pesquisa em Inteligência Artificial, 2005, ISBN: 9788520416839.
- [3] T. Qian, "Using genetic algorithm to solve sliding tile puzzles," 2007, <http://www.cs.oswego.edu/~qian/csc466/Ting%20-%20GA.pdf> visitado em 01/06/2011.
- [4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009, ISBN-13: 978-0-262-53305-8.