

IV Semana de Ciência e Tecnologia do IFMG campus Bambuí
IV Jornada Científica
06 a 09 de Dezembro de 2011
Aplicação de princípios do algoritmo genético ao cubo de Rubik

Álef Willis Magno MIRANDA¹; Larissa Evangelista de OLIVEIRA²; Armando José PINTO³; Mário Luiz Viana ALVARENGA⁴

¹Aluno do curso Técnico em Informática e bolsista do Programa Institucional de Bolsas de Iniciação Científica Júnior do IFMG campus Bambuí

²Aluna do curso Licenciatura em Física e bolsista do Programa Institucional de Bolsas de Iniciação Científica, FAPEMIG

³Aluno do curso Licenciatura em Física

⁴Professor e coordenador do curso Licenciatura em Física

RESUMO

Este artigo faz parte do projeto “Desenvolvimento e aplicação de métodos de otimização global à determinação estrutural de superfícies sólidas via técnicas de difração de elétrons”. No intuito de aprender a programar em Java e lidar com Algoritmos Genéticos, o primeiro passo é a criação de algoritmos que representem e realizem rotações no cubo de Rubik (também conhecido como cubo mágico). A linguagem de programação Java permite a portabilidade do código (o algoritmo é multiplataforma), além disso, é orientada a objetos, o que garante a manipulação dos dados como forma de objetos reais. O objetivo inicial é conseguir soluções do cubo de Rubik, partindo-se de uma configuração inicial aleatória com um número de fixo de rotações também aleatórias. O algoritmo usa padrões de repetição para efetuar as rotações no cubo com maior eficiência. Também faz-se uso de outras estruturas para aumentar o desempenho do algoritmo, tais como o uso de *threads* (estrutura/objeto que permite a execução simultânea e independente de algoritmos) e execução do algoritmo em mais de um computador por meio das redes de computadores. Ainda não temos resultados quanto às soluções do cubo, porém quanto à velocidade de execução do algoritmo, o desempenho é excelente, este realiza mais de cem mil rotações no cubo por segundo (resultado obtido em um notebook com o processador *AMD Athlon X2* e *3GB de memória RAM DDR2 de 800Mhz*, sem o uso do paralelismo). Este conhecimento adquirido garante a próxima etapa do projeto, que consiste em paralelizar algoritmos referentes à determinação estrutural de superfícies sólidas, que têm alto custo de processamento.

Palavras-chave: Cubo de Rubik, Algoritmo Genético, programação paralela

INTRODUÇÃO

Algoritmos Genéticos são uma classe de algoritmos evolucionários, os quais desenvolvem a busca pela melhor configuração de variáveis em um determinado sistema usando populações de modelos aleatórios, que se baseia nos mecanismos de evolução natural. Algoritmos Genéticos têm sido aplicados como ferramenta de busca global em sistemas que necessitam de ajuste de muitas variáveis, e não possuem solução analítica, em diversas áreas, como Engenharias, Física, Telecomunicações, Geoprocessamento, dentre outras (Viana, 2009). Este projeto tem como objetivo dar continuidade à aplicação do Algoritmo Genético na determinação estrutural de superfícies

IV Semana de Ciência e Tecnologia do IFMG campus Bambuí
IV Jornada Científica
06 a 09 de Dezembro de 2011

sólidas através de técnicas de difração de elétrons (Viana, 2008). No intuito de desenvolver e testar a nossa implementação, que vem sendo desenvolvida em Java, aplicamos, nesta etapa, o Algoritmo Genético na solução do Cubo de Rubik.

A escolha do Java dá-se pelo fato de ser orientada a objetos e, principalmente, por ser portátil, ou seja, os algoritmos nela compilados podem ser executados em diferentes sistemas operacionais. Escolhemos o problema do cubo de Rubik por dois motivos: primeiro porque a implementação dos sistemas de rotação mostram similaridade com sistemas atômicos de superfícies e segundo pelo fato de que sua solução constitui uma sequência de tentativas e erros, adequado para o teste de métodos de otimização.

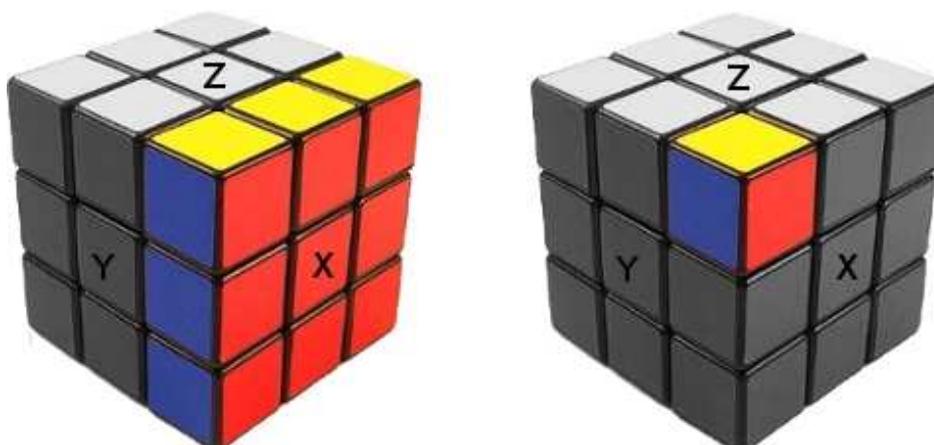
MATERIAIS E MÉTODOS

O projeto é desenvolvido no Núcleo de Física do IFMG campus Bambuí através do uso de computadores do laboratório do curso de Física.

O cubo de Rubik recebeu o nome de classe Java *MasterCube*. Este é um objeto formado por um tensor tridimensional de 27 cubos menores (estes são também objetos e possuem o nome *SmallerCube*). O *MasterCube* possui métodos de rotação setorial, sendo cada setor um conjunto de nove *SmallerCubes* em determinada dimensão (x, y ou z) e posição do setor (0, 1 ou 2). Uma rotação setorial é um conjunto de rotações aplicadas aos *SmallerCubes* localizados no setor a ser rotacionado, o que mostrou-se vantajoso com relação ao desempenho. Estas rotações são realizadas de acordo com o eixo de rotação, em múltiplos de 90°, não são comutativas e utiliza-se de uma classe que possui constantes de orientação, cores, rotações, etc.;

O *SmallerCube* possui um vetor de dez posições que o descrevem, sendo as três primeiras as posições x, y e z; as seis seguintes, a orientação das faces e das cores; e a décima o número de faces visíveis do mesmo. Esta última é utilizada na hora das rotações com o objetivo de se obter maior desempenho.

Exemplo:



Setor:

Eixo: X
Posição: 0

SmallerCube:

Posição(x, y, z): (0, 0, 0)

Imagem 1: Mostra um setor e um SmallerCube

IV Semana de Ciência e Tecnologia do IFMG campus Bambuí
IV Jornada Científica
06 a 09 de Dezembro de 2011

Quando instanciado o objeto *MasterCube*, cada face apresenta uma cor diferente, ou seja, o cubo é construído já na posição da solução que se busca. Para se conseguir uma configuração aleatória, basta inicializar o método (nome do método: *scramble(int numOfRotations)*) que realiza um número predefinido de rotações aleatórias. Daí então, podemos testar o código partindo de várias configurações aleatórias diferentes.

A ideia de paralelização deste algoritmo é criar um algoritmo administrador que envia mensagens por rede a outros que são responsáveis por controlar as instâncias dos MasterCubes. Faz-se uso de objetos do tipo *Thread* e objetos do tipo *Socket*.

- *Thread*: Objeto que permite a execução de um bloco de código simultaneamente ao algoritmo principal.
- *Socket*: Objeto que permite o envio de informações de um aplicativo a outro utilizando-se pacotes de bytes, sendo necessário apenas endereço IP ou nome da máquina (por exemplo: localhost, que representa o IP de loopback, ou seja, a própria máquina), e a porta que identifica o aplicativo, esta é gerada automaticamente durante a conexão.

Cada *Thread* é responsável por registrar todas suas ações em arquivos de texto, especificando data e hora da ação.

RESULTADOS E DISCUSSÃO

Quanto às sequências de rotações que solucionam o cubo de Rubik em determinada configuração inicial, não existem resultados significativos, porém, quanto ao desempenho, testes feitos mostram que a velocidade com que o algoritmo realiza suas ações será suficiente para que tenhamos soluções de maneira bastante rápida.

O teste de rotações por segundo realizado em um notebook da marca Acer, modelo Aspire 4535 (processador *AMD Athlon X2* e *3GB de memória RAM DDR2 de 800Mhz*, sem o uso do paralelismo), mostrou uma taxa de cem mil a cento e cinquenta mil rotações por segundo.

Esse resultado incentiva ainda mais o uso do paralelismo em algoritmos que possuem alto custo computacional, pois fazendo isso, o tempo para se conseguir os resultados desejados decresce com o número de processadores usados. No caso do Algoritmo Genético a paralelização é altamente viável, pois cada indivíduo pode ser avaliado independentemente dos demais.

CONCLUSÃO

A criação do algoritmo para as rotações do cubo de Rubik, proporcionou a aquisição de conhecimento de várias áreas, o que torna agora possível a realização da segunda etapa do projeto, que é implementar o algoritmo genético na sua solução.

O uso de paralelismo pode acelerar os cálculos de maneira bastante satisfatória, o que é fundamental quando se analisa sistemas complexos que requerem números muito grandes de tentativas e erros.

AGRADECIMENTOS

IV Semana de Ciência e Tecnologia do IFMG campus Bambuí
IV Jornada Científica
06 a 09 de Dezembro de 2011

Ao incentivo do IFMG campus Bambuí pela cessão da bolsa para a execução do projeto e ao orientador Mário Luiz Viana Alvarenga, que pelo projeto apresentado tem-nos proporcionado grande oportunidade de obtenção de conhecimentos.

REFERÊNCIAS BIBLIOGRÁFICAS

VIANA, M. L.; SOARES, E. A.; DE CARVALHO, V. E.; Estudo e aplicação de algoritmo genético e determinação estrutural de superfícies via técnicas de difração de elétrons. TESE DE DOUTORADO, UFMG, 2009.

VIANA, M. L.; SILVA, W. S.; SOARES, E. A.; DE CARVALHO, V. E.; CASTILHO, C. M. C.; VAN HOVE, M. A.; Scaling behavior of genetic algorithms applied to surface structural determination by LEED. SURFACE SCIENCE, 602 (2008) 3395-2402.