Análise de Heurísticas para a Resolução do jogo Sudoku

Cícero Alan Leite Cruz; Emanuel Dantas Filho;
Fábio Jorge Almeida Morais; Tiago Lucas Pereira Clementino

Departamento de Sistemas e Computação – Universidade Federal de Campina Grande (UFCG) – Campina Grande – PB – Brasil – Disciplina: Inteligência Artificial: Professora: Joseana Fechine – Período 2007.1
{alanlcruz, emanueldan, fabio.jorge, tiago.luks} @gmail.com

Abstract *This article has for objective to describe the project developed for disciplines Artificial Intelligence, of the department of Systems and Computation of the UFCG. The project uses heuristical to decide the problem of the Sudoku game. The implementation was made in Java, using heuristical, as the based ones on problems of satisfaction of restrictions, CSP. With the purpose to find solutions for the problem considered for the game. A study was carried through, esteem which heuristic to be found solution would be more attractive it, after that it was implemented some resulted observed techniques and, information these contained in this article.*

Resumo. *Este artigo tem por objetivo descrever o projeto desenvolvido para a disciplina Inteligência Artificial, do departamento de Sistemas e Computação da UFCG. O projeto utiliza heurísticas para resolver o problema do jogo Sudoku. A implementação foi feita em Java, utilizando heurísticas, como as baseadas em problemas de satisfação de restrições, CSP. Com a finalidade de encontrar soluções para o problema proposto pelo jogo. Foi realizado um estudo, para estimar qual heurística seria mais atrativa para a solução a ser encontrada, em seguida foi implementada algumas técnicas e observados resultados, informações estas contidas neste artigo.*

1. Introdução

O projeto desenvolvido se propõe a resolver o problema do Sudoku, um jogo de raciocínio e lógica. O jogo é composto por um tabuleiro, de dimensões m x m, onde em uma configuração inicial alguns números, com valores de 1 a m são inseridos, segundo as seguintes regras: Não podem haver números repetidos nas linhas horizontais e verticais, assim como em cada quadrante (quadrado com dimensões n x n, sendo n a raiz quadrada de

m), o número de valores iniciados em cada quadrante é constante. O objetivo do jogo é completar todos os quadrados do tabuleiro, seguindo estas mesmas regras.

A proposta do projeto consiste em demonstrar a configuração de uma solução para o problema utilizando uma técnica de Inteligência Artificial. A solução do Sudoku pode ser abordada de várias maneiras, como por exemplo, através de busca cega, busca heurística ou problemas de satisfação de restrições.

Este artigo está organizado da seguinte forma: na seção 2 é descrito mais suncitamente o jogo Sudoku; na seção seguinte é apresentada a metodologia adotada na busca pela solução, quais as técnicas de IA utilizadas para a resolução deste problema e os motivos da escolha destas; na seção 4 são descritos os resultados encontrados para as técnicas escolhidas; na seção seguinte é feita análise destes resultados; e, por fim, na seção 6 conclusões e possíveis trabalhos futuros são descritos.

2. Jogo Sudoku

Sudoku é um problema cripto-aritmético japonês. Dada uma grade de 25 x 25 quadrados, onde 317 deles são totalmente preenchidos com um número entre 1 e 25, a idéia é preencher o restante dos quadrados tal que cada linha e coluna seja uma permutação dos números de 1 a 25. Entretanto, cada uma das 25 grades de 5 x 5 quadrados começa nas colunas (linhas) 1, 6, 11, 16, 21 e deve ser uma permutação de números de 1 a 25. [2]

Um exemplo de Sudoku, em tamanho menor (9 x 9), é o alvo de nosso projeto, apresentado na Figura 1. Há um certo número de células a serem preenchidas, que correspondem às variáveis do problema. As restrições são as seguintes: os números a serem colocados nos quadrados vazios devem ser de 1 a 9 e diferentes dos números já preenchidos na mesma linha e coluna do tabuleiro e do quadrante 3 x 3 deste.

V_0	6	7	V_1	5	4	9	V_2	V_3
V_4	4	9	6	V_5	V_6	5	V_7	7
8	V_8	V_9	1	V_{10}	7	V_{11}	6	2
V_{12}	V_{13}	1	3	7	V_{14}	V_{15}	2	4
4	7	V_{16}	V_{17}	V_{18}	6	3	5	V_{19}
6	V_{20}	5	2	4	V_{21}	V_{22}	V_{23}	9
V_{24}	2	V_{25}	V_{26}	6	8	1	V_{27}	5
7	V_{28}	6	V_{29}	1	V_{30}	2	4	V_{31}
5	1	V_{32}	4	V_{33}	9	V_{34}	3	V_{35}

Figura 1. Exemplo de uma configuração inicial para o Sudoku 9 x 9

3. Metodologia

Para o desenvolvimento de uma solução para o jogo, há a necessidade de uma demanda mais refinada do que uma simples busca clássica. O primeiro passo foi buscar uma solução por meio de heurísticas baseadas em problemas de satisfação de restrições (*CSP – Constraint Satisfaction Problems*), são tipos de problemas que impõe propriedades estruturais adicionais à solução a ser encontrada. O que torna bem mais eficiente que uma busca cega.

3.1 Variável mais restringida

O problema a ser resolvido pode ser formalizado como segue: tem-se um conjunto de variáveis (os quadrados ainda não preenchidos) que podem assumir valores dentro de um dado domínio (valores de 1 a 9), sendo obedecido um conjunto de restrições definidos pelas regras do jogo – definem os valores que essas variáveis podem assumir.

Dentre as possibilidades propostas por CSP, foi escolhida a heurística da variável mais restringida, ou seja, escolher as variáveis que possuem menor número possível de valores a jogar. O sistema armazena para cada variável uma lista com os valores que esta pode assumir no momento da jogada, a idéia consiste em escolher as variáveis com a menor quantidade de valores nesta lista. Em caso de empate um escolha aleatória deve ser feita entre as variáveis com listas de mesmo tamanho.

3.2 Outras heurísticas utilizadas

Foi observado com a aplicação da heurística da variável mais restringida, que acontecia uma concentração de valores em determinadas regiões do tabuleiro, depois de verificar os resultados de experimentos sobre diversas amostras, chegou-se a conclusão que esta heurística não seria suficiente para encontrar a solução do problema com eficiência.

Uma outra técnica foi acrescentada a resolução do problema, no momento em que duas variáveis empatassem em número de valores possíveis a jogar, o sistema deve escolher a variável com maior distância da jogada anterior, como forma de tentar diminuir esta concentração de valores. Persistindo o empate, a escolha deve ser feita de forma aleatória.

A técnica descrita acima melhorou a eficiência na busca por uma solução do jogo, foi observado que o tempo para encontrar a solução diminuiu e a freqüência de descoberta de um solução havia crescido, porem ainda não acontecia de forma eficiente.

Uma última técnica foi adicionada ao sistema, no caso de empate ao verificar qual variável era a mais restringida, deve-se escolher a que estivesse em um quadrante menos denso, também como forma de diminuir esta concentração. Em seguida persistindo o empate, escolheria a que estivesse a uma maior distância da ultima jogada. Portanto, uma combinacao de três heurísticas para encontrar a solução. Depois de implementada esta combinacao, houve uma melhoria significativa nos resultados obtidos.

3.3 Forward Checking e Backtracking

A técnica de *Forward Checking* consiste em checar previamente que valores (números) podem ser aplicados a que variáveis (células). Isto é realizado adicionando a cada variável não instanciada (vazia) uma lista com os possíveis valores que podem ser instanciados nela. Esta lista é revisada em todas as variáveis não instanciadas numa mesma linha, coluna ou quadrante de uma variável que acaba de ser instanciada.

Ao atribuir um valor a uma nova variável, se o algoritmo constatar que alguma das variáveis só pode ser instanciada com um valor (só existe ele na sua lista), ela deve ser instanciada com este valor imediatamente, e se repete o processo. Caso ela não possa ser instanciada por mais nenhum dos valores (está com a lista de valores possíveis vazia) deverá ser realizado um procedimento de *Backtracking*.

Backtracking é uma técnica baseada na análise por retrocesso. Como já foi dito, o *Backtracking* é executado quando uma das variáveis não tiver mais opções de instância, o que leva o algoritmo a revisar todas as suas ultimas alterações de baixo para cima, desfazendo-as e refazendo-as com outros valores possíveis, tomando decisões diferentes e testando a cada nova decisão. Aprofundando-se, assim, gradualmente até onde for necessário para que ele siga em frente com o processo de preenchimento do tabuleiro.

4. Resultados

A busca com as três heurísticas, terá uma solução mais rápida, portanto o espaço de estados gerado é menor que o espaço gerado por uma busca cega, ou na solução com apenas uma destas heurísticas. O percentual de solução para o problema utilizando estas três heurísticas é bem maior.

As heurísticas utilizadas são todas admissíveis, porem apenas as técnicas de variável mais restringida e do quadrante menos denso são monotônicas, logo elas garantem uma solução ótima na maioria dos casos. Uma heurística é admissível quando o seu valor nunca superestima o custo de atingir a meta. E uma heurística é monotônica quando o seu valor sempre cresce ou sempre decresce.

Todos os objetivos levantados foram cumpridos. Foi desenvolvido o jogo envolvendo os escopos de várias heurísticas já citadas. Para projetos futuros podem ser implementadas outras buscas usando diferentes estratégias.

6. Considerações Finais

O desenvolvimento deste projeto foi válido para que pudéssemos por em prática algumas técnicas que aprendemos na disciplina. O projeto realizado ainda é passível de melhorias, pode ser implementado outras heurísticas na busca da freqüência de 100% de descoberta da solução dado uma configuração inicial qualquer para o jogo Sudoku. Em virtude da falta de tempo e de não ser conhecido nenhuma outra heurística não foi implementada outras formas de busca.

O trabalho e dedicação da equipe no desenvolvimento do projeto foram satisfatórios para conclusão com êxito. O projeto foi concluído no tempo determinado. Porém, observou-se que uma dificuldade no processo de desenvolvimento, a questão da equipe não reunir-se integralmente dificultou o andamento do projeto. O desenvolvimento aconteceu de forma distribuída, com apoio de uma ferramenta para gerenciar o que estava sendo produzido, o SVN, o que possibilitou o controle de versão do projeto entre os desenvolvedores. Foi um risco alto, podendo ter acarretado o não cumprimento do projeto no tempo estabelecido.

7. Referências Bibliográficas

[1] FECHINE, Joseana M. (2007) "*Disciplina:Inteligência Artificial I*" On-line. Disponível em: http://www.dsc.ufcg.edu.br/~joseana/IA1-20071.html. Acesso em: 05 de agosto de 2007.

[2] PEREIRA, Marluce Rodrigues Pereira. *Particionamento automático de restrições.* On-line. Disponível em: http://www.lam.ufrj.br/index.php?option=com_docman&task=doc_view&gid=21. Acesso em: 10 de setembro de 2007.

[3] PINTO, M. J.; YANASSE, H. H. (2003) *"Uma Heurística para a Resolução do Problema Integrado de Corte e Seqüenciamento baseada em uma Partição do Problema"*. Disponível em: http://hermes2.dpi.inpe.br:1905/col/lac.inpe.br/worcap/2003/10.24.09.54/doc/Artigo_worcap_2003_versaofinal.pdf