

Universidade Federal de São Carlos  
Bacharelado em Ciência da Computação  
Disciplina de Inteligência Artificial

## Aplicação de Estratégias de Busca na Resolução de jogos de Sudoku

Professor: Ricardo Cerri  
Otavio Cesar Toma da Silva 726576  
Rafael Sales Pavarina 726583  
Rene Ferrante Neto 726587

São Carlos  
16 de Novembro de 2018

# SudokuAutoSolver

Solucionador de Sudoku usando estratégias de busca vistas na aula de Inteligência Artificial.

O projeto foi desenvolvido em javascript com o auxílio da biblioteca para desenhos [p5.js](#).

Para executá-lo basta entrar na pasta *src* e abrir o arquivo *sudoku.html* no seu navegador.

É possível preencher a matriz do sudoku inserindo números pelo teclado, e mover-se a célula selecionada com as setinhas. Nota-se que não é possível preencher o sudoku de forma a quebrar sua regra de colocação de números.

Há dois botões, um para resolver o Sudoku por meio de busca cega, e outro por meio de informada. A partir do momento em que qualquer um é clicado não é possível mais inserir valores na matriz e os valores inseridos (não modificáveis) tem sua célula com fundo cinza.

Em ambas as buscas se contabiliza o tempo que decorreu desde seu início até o seu término, podendo este ser visualizado ao final da execução no log do console do navegador; se utiliza uma matriz fixa para ser desenhada na tela (matrizDesenho); e se separa as matrizes possíveis em categorias, sendo estas pertencentes ao array *explorados* caso já tenha-se passado por elas na busca e *fronteira*, para as próximas a serem exploradas.

## Busca Cega - Busca em Profundidade

A busca cega implementada é a de DFS (Depth First Search), ou Busca em Profundidade.

A busca em profundidade implementada busca a primeira célula em branco e coloca em uma pilha todas as matrizes possíveis de acordo com os números que podem ser colocados nesta célula (fronteira), prosseguindo-se então para um próximo estado com uma dessas possíveis matrizes e voltando-se a buscar o primeiro espaço em branco. Ao não encontrar possibilidades de números e um espaço em branco ocorre o bracktracking, pois se retira uma das matrizes acumuladas na fronteira e a coloca como estado atual.

## Busca Informada - A\*

A busca A\* implementada se inicia inserindo o estado inicial na fronteira e, a cada passo, é escolhido o melhor estado da fronteira para ser expandido (por isso a fronteira é sempre reordenada em cada iteração). Essa ordenação é feita baseando-se em duas restrições: a primeira é o custo de cada estado e a segunda é uma heurística definida para detectar estados para os quais não compensa continuar a busca.

A função de custo utilizada para cada estado foi definida como sendo a quantidade de valores possíveis que podem ser colocados na célula da matriz que possui a menor quantidade de valores possíveis para serem inseridos nela, enquanto que a heurística definida faz com que estados “impossíveis” não sejam inseridos na fronteira. Um estado é definido como impossível se a sua matriz contém alguma célula vazia que não consiga receber nenhum valor em um movimento válido do sudoku.

A busca A\* implementada obteve resultados positivos, encontrando a meta (matriz com 81 elementos) mais rapidamente do que a busca cega em todos os casos testados.