

UNIVERSIDADE DE SÃO PAULO – USP
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE
COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIAS DA COMPUTAÇÃO

RELATÓRIO
[SCC5900] Projeto de algoritmos
Projeto 01: Backtracking

Professor Gustavo Batista

Aluno:

Misael Costa Junior

NUSP° 9254469

Este relatório resume as etapas realizadas para implementação e execução do projeto.

1. Execução do algoritmo

O algoritmo foi implementado na classe Sudoku.java dentro do diretório (projeto Java) ProblemaSudoku. O projeto foi “escrito” na linguagem de programação Java e executado por intermédio da IDE Eclipse. A execução é realizada com a retirada do comentário entre as linhas 272 e 274, correspondente a abordagem desejada. Em geral, o algoritmo está 90% comentado para facilitar o entendimento.

A implementação de cada poda/heurística foi “resumida” em funções. A execução das mesmas é realizada por validações de critérios (escolha poda) necessários para sua execução. Uma estratégia adotada para escolha de execução por poda x, y ou z foi realizada com o acréscimo de uma variável poda representando a abordagem a ser utilizada.

No geral, o algoritmo apresentou demora de execução quando realizado com uso das podas: backtracking com forward checking, em média 5 min. Quando executado com backtracking simples o algoritmo apresentou as soluções, em média, em 1min e 11seg e com o acréscimo do algoritmo do mvr, as soluções foram apresentadas em 17seg. Junto com arquivos desse projeto, foram incluídos arquivos txt com todos os testes por abordagem, incluindo dados quantitativos, como: quantidade de soluções, aproveitamento, quantidade de atribuições e tempo médio.

O critério de parada, 10^6 , só foi acrescentado quando constatado a demora de execução do algoritmo com o uso de alguma das podas, o que não foi necessário.

2. Implementação

1. Backtracking simples

A implementação foi realizada conforme a definição do algoritmo do backtracking, com a busca de variáveis vazias (0) e verificação da validação de um domínio (1 a 9), atribuindo o domínio a variável, caso seja válido.

2. Backtracking com forward checking

Com base na definição do forward checking, a diferença na implementação foi no acréscimo do critério da chamada da função de forward checking para avaliar a validade de valores candidatos -

ok_fc(matriz, valor) – e uma chamada a função src_valores_candidatos para fazer uma busca na linha, coluna e box correspondente para selecionar os domínios candidatos.

3. Backtracking, Forward checking e MVR

Nesse caso, a novidade ficou na implementação de duas funções: *busca_mvr* e *busca_mrv_prox*. A função *busca_mrv* é similar a do *backtracking simples* com a diferença que nessa a validação será realizada com o forward checking e a chamada para a próxima célula da matriz será, agora, para a que possuir menor quantidade de domínios disponíveis.

3. Avaliação de desempenho

Foi realizada uma comparação através da execução do algoritmo para os 95 casos de testes incluídos no arquivo entrada.txt. Com base nos dados, uma demonstração gráfica foi incluída para melhor demonstrar a eficácia da abordagem utilizada pela quantidade total de atribuições para solução do Sudoku. Segue o resultado:

| Abordagem | Backtracking simples | Backtracking + Forward checking | Backtracking + Forward checking + MVR |
|------------------------|----------------------|---------------------------------|---------------------------------------|
| Quantidade atribuições | 393.147.043 | 54.824.062 | 1.493.560 |

