

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

RELATÓRIO
[SCC0218] ALGORITMOS AVANÇADOS E APLICAÇÕES
Projeto 01: Backtracking

Professor Gustavo Batista

Alunos:

Gil Barbosa Reis

NUSPº 8532248

Leonardo Sampaio Ferraz Ribeiro

NUSPº 8532300

I. Implementação

a. compilação e uso

O algoritmo e suas podas foram implementados na classe Sudoku, que organiza a entrada em uma matriz principal (com uma cópia para a solução) e recebe no método solve diferentes flags para ligar ou desligar as podas.

A compilação é feita com o uso de um Makefile, para recompilar use “make all” em um terminal aberto na pasta com o código fonte.

A implementação permite que as flags sejam ligadas através de argumentos na linha de comando, para mais informações execute “./resolverdorDeSudoku -h”.

b. backtracking sem podas

A implementação foi feita baseada no pseudo-código trabalhado durante as aulas e dividida em diversos métodos ajudantes.

c. backtracking com verificação adiante

Para realizar a verificação adiante é criada uma matriz auxiliar de vetores (array de 3 dimensões) que guardam, para cada posição, flags que identificam os valores remanescentes da mesma. Além disso, a última posição do vetor indica o número de valores remanescentes para a posição. Quando alguma posição, dada qualquer atribuição, fica com 0 valores remanescentes, acontece o backtracking.

As flags no vetor são: 1 para valor remanescente e -(profundidade na recursão onde se tornou indisponível) para valores não disponíveis. Desta forma, ao realizar o backtracking, sabendo a profundidade da recursão, o algoritmo restaura os valores remanescentes que foram retirados naquela iteração.

d. backtracking com verificação adiante e mínimos valores remanescentes

Utilizando a matriz auxiliar de valores remanescentes, é criado um vetor que guarda pares (pair<posição, número de valores remanescentes>) e é, a cada novo valor, re-ordenado pelo número de valores remanescentes, fazendo com que aqueles com o mínimo fiquem no início do vetor e permitindo a seleção destas posições para dar continuidade ao algoritmo.

II. Avaliação de Desempenho

Para visualizar com maior clareza a diferença de desempenho entre os diferentes métodos implementados foram plotados dois gráficos comparando os resultados obtidos ao iterar através dos 95 jogos propostos como testes pelo professor.

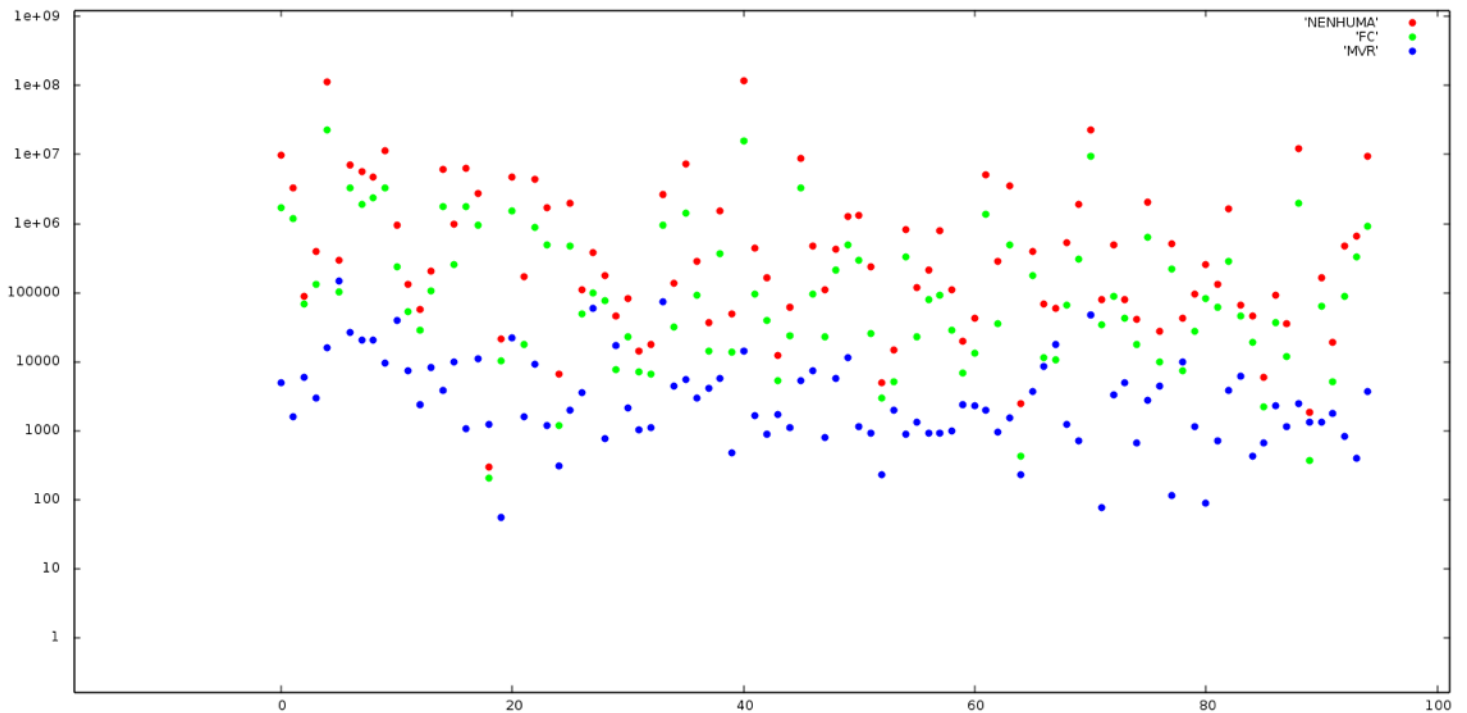


Gráfico 01. Quantidade de Atribuições vs. Número do Sudoku no arquivo “entrada.txt”

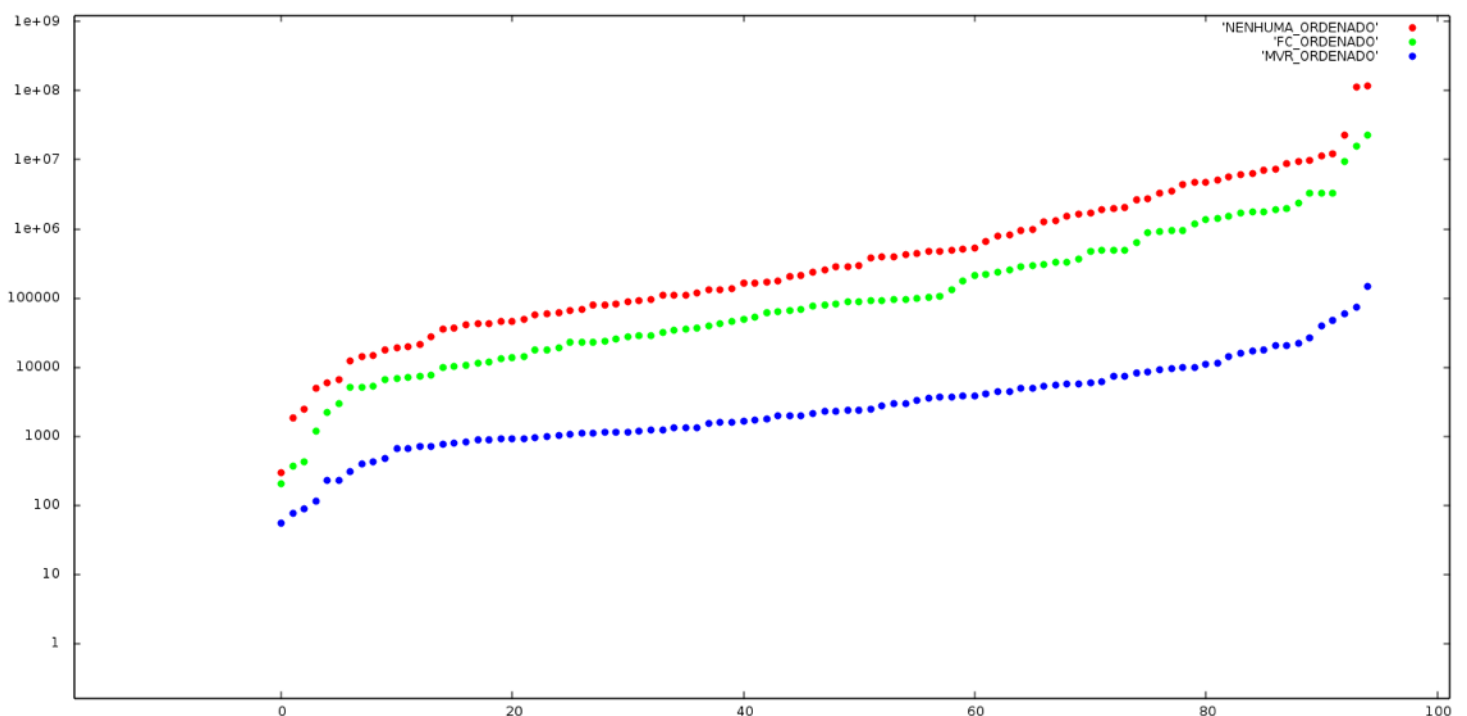


Gráfico 02. Quantidade de Atribuições Ordenada vs. Enésimo Jogo Ordenado por Número de Atribuições

No gráfico 01 é possível notar como a nuvem de pontos do número de atribuições sem poda (“NENHUM”, vermelho) fica acima (e levemente misturada) com a nuvem utilizando verificação adiante (“FW”, verde). A diferença é ainda mais clara quando a nuvem de pontos utilizando MVR é levada em consideração, poucos são os jogos em que o algoritmo usando MVR faz menos atribuições que qualquer um dos outros métodos.

O gráfico 02 compara os jogos ordenados pelo número de atribuições e mostra com clareza ainda maior que, em geral, a verificação adiante faz menos atribuições que o backtracking sem podas e a heurística MVR traz melhorias ainda mais significativas.

Note a escala logarítmica dos gráficos, ela torna a visualização das diferenças mais clara.