Grupo Nº 54



**Inteligência Artificial**

1.º Semestre 2015/2016

**IA-Tetris**

Relatório de Projecto

Índice

[1 Implementação Tipo Tabuleiro e Funções do problema de Procura 3](#_Toc436330760)

[1.1 Tipo Abstracto de Informação Tabuleiro 3](#_Toc436330761)

[1.2 Implementação de funções do problema de procura 3](#_Toc436330762)

[2 Implementação Algoritmos de Procura 4](#_Toc436330763)

[2.1 Procura-pp 4](#_Toc436330764)

[2.2 Procura-A\* 4](#_Toc436330765)

[2.3 Outros algoritmos 4](#_Toc436330766)

[3 Funções Heurísticas 5](#_Toc436330767)

[3.1 Heurística 1 5](#_Toc436330768)

[3.1.1 Motivação 5](#_Toc436330769)

[3.1.2 Forma de Cálculo 5](#_Toc436330770)

[3.2 Heurística n 5](#_Toc436330771)

[4 Estudo Comparativo 6](#_Toc436330772)

[4.1 Estudo Algoritmos de Procura 6](#_Toc436330773)

[4.1.1 Critérios a analisar 6](#_Toc436330774)

[4.1.2 Testes Efectuados 6](#_Toc436330775)

[4.1.3 Resultados Obtidos 6](#_Toc436330776)

[4.1.4 Comparação dos Resultados Obtidos 6](#_Toc436330777)

[4.2 Estudo funções de custo/heurísticas 6](#_Toc436330778)

[4.2.1 Critérios a analisar 6](#_Toc436330779)

[4.2.2 Testes Efectuados 6](#_Toc436330780)

[4.2.3 Resultados Obtidos 6](#_Toc436330781)

[4.2.4 Comparação dos Resultados Obtidos 7](#_Toc436330782)

[4.3 Escolha da procura-best 7](#_Toc436330783)

# Implementação Tipo Tabuleiro e Funções do problema de Procura

## Tipo Abstracto de Informação Tabuleiro

Tratando-se do jogo *“Tetris”*, neste caso com um tabuleiro 18x10, a maneira mais intuitiva de relacionar as linhas e colunas seria por uma matriz contendo 18 linhas e 10 colunas.

Para tal na primeira entrega usamos um array de dimensoes 18 por 10, pré preenchido a nil para representar um tabuleiro vazio.

Apesar de funcionar na perfeição, mudamos para uma implementação com *hash tables* na segunda entrega. A razão para a mudança foi em casos que se acede a um valor a partir de um elemento chave, não precisamos de percorrer a lista toda (O(n)) à procura do elemento mas sim apenas ao *key-value* respectivo (O(1)). Isto irá ser benéfico nas procuras por exemplo.

Neste ponto deve ser abordado, os seguintes tópicos:

•Descrição detalhada das estrutura de dados utilizada para representar o tipo tabuleiro.

•Justificação da escolhas efectuada em relação a possíveis alternativas e às operações implementadas.

## Implementação de funções do problema de procura

Nesta secção deverão descrever as funções para o problema de procura do IA-Tetris, em particular é **obrigatória** a descrição das funções *accoes* e *resultado*. Se houver alguma outra função que considerem relevante descrever podem fazê-lo.

# Implementação Algoritmos de Procura

Nesta secção deverão descrever a vossa implementação dos algoritmos de procura. Em particular devem descrever as seguintes funções/algoritmos:

## Procura-pp

Deverão descrever brevemente o algoritmo de procura em profundidade primeiro.

## Procura-A\*

Aqui deverão descrever o algoritmo de procura A\*, indicando pormenores relevantes como por exemplo a estrutura usada para conjunto de abertos/fronteira e fechados/explorados, se fazem a ordenação dos elementos no conjunto de abertos, ou se alternativamente não fazem ordenação e simplesmente percorrem o conjunto à procura do melhor nó. Também deverão descrever técnicas de optimização que tenham feito sobre o algoritmo (caso o tenham feito).

## Outros algoritmos

Caso tenham implementado outros algoritmos de procura (ex: procura local), podem e devem descrevê-los nesta secção.

# Funções Heurísticas

**NOTA IMPORTANTE:** As várias heurísticas apresentadas devem ter como base uma ideia diferente ou serem variações *relevantes* da mesma ideia, e não simples parametrizações da mesma. Heurísticas com base na mesma ideia que apenas diferem em pequenos factores *são passos na depuração da melhor heurística encontrada para ela e serão considerados como tal.*

## Heurística 1

### Motivação

Abordar de forma detalhada, *pelo menos*, os seguintes aspectos:

* Ideia inicial, informal, que levou à criação da heurística.
* Informação relevante contida no estado usada no cálculo da heurística.
* Processo de depuração que conduziu à forma final da heurística (outras variantes ensaiadas, etc.)

### Forma de Cálculo

Descrição *detalhada* da forma final da heurística (incluindo, eventualmente, uma fórmula que permita calcular o seu valor). É *muito importante* que esta descrição seja *concreta*, de modo a que outra pessoa consiga reproduzir a heurística se necessário. Deve também ser *clara* de modo a que qualquer pessoa que tenha lido a motivação da heurística compreenda a sua forma de cálculo.

Esta descrição deve ser complementada com exemplos da sua aplicação, recorrendo a figuras ou outros meios similares que tornem *perceptível* e *imediata* a compreensão da ideia subjacente à heurística.

**.**

**.**

**.**

## Heurística n

# Estudo Comparativo

NOTA: Embora nas várias secções seguintes seja pedida a descrição dos testes efectuados, não é necessário repetir essa descrição se os testes já tiverem sido descritos anteriormente, bastando referir onde esta descrição se encontra.

Pretende-se, neste projecto, efectuar quer os algoritmos de procura quer as heurísticas implementadas para o jogo IA-Tetris.

## Estudo Algoritmos de Procura

O primeiro estudo foca-se em comparar as duas implementações pedidas para os algoritmos de procura, a procura em profundidade primeiro e a procura A\*. Caso tenham implementado outro algoritmo de procura (ex. procura local)[[1]](#footnote-1), deverão também compará-lo com as outras duas procuras.

### Critérios a analisar

Descrever quais os critérios a ser usados para comparar os algoritmos e justificar a sua escolha. Alguns exemplos de critérios possíveis: qualidade das jogadas, número máximo de jogadas conseguidas, nós visitados, tempo de execução.

### Testes Efectuados

Descrever os problemas utilizados para efectuar o estudo. Justificar a sua *relevância* para os critérios a comparar.

### Resultados Obtidos

Enumerar os resultados obtidos, sob a forma mais adequada (tabela e gráficos, se necessário).

### Comparação dos Resultados Obtidos

Analisar, caso a caso, os resultados, focando a diferença de desempenho dos vários algoritmos entre si. A análise não deve ser somente uma descrição dos resultados obtidos, mas sim uma análise dos factores que possam ter conduzido a esses resultados.

## Estudo funções de custo/heurísticas

O segundo estudo foca-se em comparar as várias funções de custo e heurísticas implementadas, e perceber qual o custo e heurística com melhores resultados para a qualidade do jogador. **Atenção:** os testes com custos e heurísticas devem ser todos feitos com o mesmo algoritmo de procura.

### Critérios a analisar

Descrever quais os critérios a ser usados para comparar as várias heurísticas e justificar a sua escolha.

### Testes Efectuados

Descrever os problemas utilizados para efectuar o estudo. Justificar a sua *relevância* para os critérios a comparar.

### Resultados Obtidos

Enumerar os resultados obtidos, sob a forma mais adequada (tabela e gráficos, se necessário).

### Comparação dos Resultados Obtidos

Analisar, caso a caso, os resultados, focando a diferença de desempenho dos vários custos e das várias heurísticas entre si. A análise não deve ser somente uma descrição dos resultados obtidos, mas sim uma análise dos factores que possam ter conduzido a esses resultados.

## Escolha da procura-best

Com base nos resultados obtidos nas duas comparações anteriores, devem descrever e justificar qual o algoritmo e respectivas funções custo/heurística para implementar a procura-best criando assim o vosso melhor jogador.

1. Atenção, para que fique claro, não é necessária a implementação de outros algoritmos de procura, mas caso o tenham feito, deverão descrevê-los e compará-los. [↑](#footnote-ref-1)