Project “Catch” – Projeto de Inteligência Artificial

Nuno Querido

2161320

2161320@my.ipleiria.pt

Paulo Monteiro

2162374

2162374@my.ipleiria.pt

**ABSTRACT**

This report documents the work related to developing the best algorithm for a “Travelling Salesman”-like problem. This algorithm, given a defined problem, must have the capacity to find an optimized itinerary. The majority of the work revolved around implementing functions that evaluated the quality of the solutions found, and also how would the algorithm evolve, while defining the guidelines that would favor individuals that generated better solutions.

**Keywords**

Inteligência Artificial, Algoritmo, Algoritmo Genético, Heurística, Genoma, Fitness, Mutação, Recombinação.

# INTRODUÇÃO

Com a execução deste projeto pretendemos pôr em prática os conhecimentos teóricos adquiridos, aprofundando-os, e também desenvolver a nossa capacidade de implementar programas vocacionados para resolver problemas relacionados com inteligência artificial. As componentes mais significativas deste projeto serão o desenvolvimento de funções de avaliação da qualidade das soluções para os problemas fornecidos, e de operadores genéticos, que neste caso serão algoritmos de recombinação e mutação do genoma dos indivíduos.

# ESTADO DO PROBLEMA

O estado do problema representa toda a grelha a nesse dado momento. Nele podemos verificar a posição atual do agente na grelha bem como as “caixas”, que para serem recolhidos deve o agente deslocar-se à posição de cada uma delas para simbolizar a sua recolha. O objetivo é encontrar um caminho que recolha todas as “caixas” no menor número de movimentos possível.

# FUNÇÃO HEURÍSTICA

A função heurística é a função que estima qual é o custo do caminho entre a posição atual do agente no estado atual e a posição do agente no estado objetivo.

No nosso problema aplicámos o método de pesquisa denominado de *A\* Search*. Este expande os nós sucessores em profundidade, expandido o nó de menor custo primeiro. Os sucessores de um nó expandido são colocados por ordem crescente na fronteira.

# INDIVÍDUOS DO ALGORITMO GENÉTICO

Os indivíduos implementados neste projeto têm informação acerca do problema, nomeadamente a quantidade e localização das caixas, que deve “apanhar”, e da porta, que é o seu objetivo depois de apanhar todas as caixas. Este apenas se pode movimentar uma casa de cada vez, para cima, para baixo, para esquerda ou para a direita.

O genoma destes indivíduos representa a ordem pela qual ele vai “apanhar” as caixas, antes de “sair” pela porta. Utiliza a função *computeFitness* como função de avaliação, descrita no tópico seguinte.

# FUNÇÃO DE AVALIAÇÃO

Esta função é utilizada pelos indivíduos para calcular o custo total do caminho que irá percorrer até atingir o estado objetivo sendo que cada movimento do indivíduo incrementa o valor de custo em uma unidade. Para o nosso problema, são melhores os indivíduos com um fitness mais baixo, ou seja, que façam menos iterações até atingir o estado objetivo.

# CRIAÇÃO DA POPULAÇÃO INICIAL

Os indivíduos da população inicial têm os seus genes gerados de forma aleatória, sendo que os seus descendentes têm o seu genoma gerado através dos operadores genéticos implementados no nosso projeto

# OPEARDORES GENÉTICOS

Os operadores genéticos são aplicados aos indivíduos de uma população numa tentativa de gerar melhores indivíduos para o problema em questão.

## Operadores de Mutação

Altera o material genético de um individuo, proporcionando diversidade dentro da população. São operadores aplicados geralmente após os operadores de recombinação.

### Mutation Insert

São gerados dois pontos de corte aleatórios no genoma do indivíduo, garantido que são diferentes. O gene que consta na posição do segundo corte é então colocado na posição seguinte à do gene constante no primeiro corte, todos os genes que se encontravam entre estes originalmente se movem uma posição no sentido oposto à do gene do primeiro corte.

### Mutation Scramble

São gerados dois pontos de corte no genoma do indivíduo sendo estas posições aleatórias e distintas. Todos os genes compreendidos entre estes dois cortes trocam de posição entre eles de forma aleatória, incluindo os genes que constam nas posições de corte.

### Mutation Inversion

São gerados dois pontos de corte aleatórios no genoma do individuo. Todos os genes que constam entre os pontos de corte são então mudados de posição de forma espelhada, ou seja, o gene que consta na primeira posição do segmento troca de posição com o gene constante da última posição do segmento e assim sucessivamente.

## Operador de Recombinação

Operador que utiliza o material genético de dois indivíduos para gerar dois descendentes.

### Recombination Partially Mapped (PMX)

São geradas duas posições de corte aleatórias e distintas nos dois indivíduos em questão. De seguida são copiados os genes compreendidos entre as duas posições de corte (inclusive) do primeiro pai para o primeiro filho. Para preencher o resto do material genético do primeiro filho olhamos então para o segundo pai, a partir do primeiro corte. Se o gene que consta na posição do segundo pai já se encontra no primeiro filho, não faz nada e passa à posição seguinte, caso contrário, vamos à mesma posição no primeiro pai e vemos em que posição do segundo pai se encontra esse gene. Se essa posição se encontra livre no primeiro filho, colocamos lá o gene do segundo pai, caso contrário continuamos o processo até encontrar uma posição livre. Para o segundo filho repetimos o processo, com as mesmas posições de corte, mas os pais trocam de papel.

### Cycle Recombination

Para gerar o primeiro filho, começando na primeira posição do primeiro pai, guardamos essa posição numa lista de posições visitadas e vemos qual é o gene que consta nessa mesma posição no segundo pai. De seguida voltamos ao primeiro, na posição em que está o elemento do segundo pai visitado na posição anterior, guardamos essa posição na lista e voltamos a ver qual é o gene que consta no segundo pai nessa mesma posição. Repetimos este ciclo até voltarmos à primeira posição visitada no primeiro pai. De seguida copiamos os genes que constam nas posições visitadas para os filhos correspondentes. Para preencher o restante material genético, vamos aos pais contrários e copiamos os genes que constam nas posições que não constam da lista, para os filhos.

### Order 1 Recombination

São geradas duas posições de corte aleatórias e distintas no genoma dos indivíduos. Copiamos os genes do primeiro pai compreendidos entre estas posições, inclusive, para o primeiro filho. De seguida olhamos para o segundo pai, a partir da posição seguinte ao segundo corte. Se o gene que consta nessa posição não se encontra no primeiro filho, colocamos esse gene na primeira posição livre deste, começando também a partir da posição seguinte ao segundo corte. Quando chegamos às posições finais do genoma, passamos para a posição inicial do genoma correspondente e continuamos o processo até o filho ter o material genético completo. Para o segundo filho repetimos o processo, com os pais em papeis inversos.

# RESULTADOS OBTIDOS

# FIGURES/CAPTIONS

Place Tables/Figures/Images in text as close to the reference as possible (see Figure 1). It may extend across both columns to a maximum width of 17.78 cm (7”).

Captions should be Times New Roman 9-point bold. They should be numbered (e.g., “Table 1” or “Figure 2”), please note that the word for Table and Figure are spelled out. Figure’s captions should be centered beneath the image or picture, and Table captions should be centered above the table body.

# SECTIONS

The heading of a section should be in Times New Roman 12-point bold in all-capitals flush left with an additional 6-points of white space above the section head. Sections and subsequent sub- sections should be numbered and flush left. For a section head and a subsection head together (such as Section 3 and subsection 3.1), use no additional space above the subsection head.

## Subsections

The heading of subsections should be in Times New Roman 12-point bold with only the initial letters capitalized. (Note: For subsections and subsubsections, a word like *the* or *a* is not capitalized unless it is the first word of the header.)

### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized and 6-points of white space above the subsubsection head.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

# ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

# REFERENCES

1. Bowman, B., Debray, S. K., and Peterson, L. L. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst., 15,* 5 (Nov. 1993), 795-825.
2. Ding, W., and Marchionini, G. *A Study on Video Browsing Strategies.* Technical Report UMIACS-TR-97-40, University of Maryland, College Park, MD, 1997.
3. Fröhlich, B. and Plate, J. The cubic mouse: a new device for three-dimensional iput. In *Proceedings of the SIGCHI conference on Human factors in computing systems   
   (CHI ’00)* (The Hague, The Netherlands, April 1-6, 2000). ACM Press, New York, NY, 2000, 526-531.
4. Lamport, L. *LaTeX User’s Guide and Document Reference Manual.* Addison-Wesley, Reading, MA, 1986.
5. Sannella, M. J. *Constraint Satisfaction and Debugging for Interactive User Interfaces.* Ph.D. Thesis, University of Washington, Seattle, WA, 1994.

Columns on Last Page Should Be Made As Close As Possible to Equal Length