

Introdução à Inteligência Artificial

TRABALHO PRÁTICO Nº3

Space Shooter



Nome: Joel Filipe Rogão Pires

Nº Estudante: 2014195242

Email: jfpires@student.dei.uc.pt

Nome: Luís Alberto Pires Amaro

Nº Estudante: 2014109216

Email: lapa@student.dei.uc.pt

Nome: Pedro Manuel Cerveira Andrade

Nº Estudante: 2014225147

Email: pmca@student.dei.uc.pt

Introdução

Este trabalho foi desenvolvido no âmbito da unidade curricular de Introdução à Inteligência Artificial do curso de Engenharia Informática da Universidade de Coimbra e regida por Fernando Machado.

Neste trabalho vamos desenvolver e explorar as capacidades de agentes adaptativos. Para isso, vamos tentar desenvolver um controlador para o jogador de um jogo do estilo “Space Shooter”. O controlador tem de fazer com que a nave se desvie dos asteroides. Para o efeito, usámos algoritmos genéticos.

Para desenvolver tal projeto usou-se o motor de jogo Unity e a linguagem de programação C#. O Unity foi-nos útil para poder construir cenários de teste e visualizar os comportamentos manifestados pelo nosso agente adaptativo. A linguagem C# deu vida a esse mesmo agente que manifestava os comportamentos dos algoritmos que nós próprios havíamos programado previamente.

Modelação e Desenvolvimento do AG

Representação

Para representar as soluções candidatas, optou-se por uma representação ternária com os seguintes símbolos -1, 0, 1. As posições da aeronave no mapa são assim codificadas em forma de uma longa cadeia de cromossomas do tamanho de $\text{tamanho_do_mapa} / \text{fator_multiplicativo}$.

Aptidão

O indivíduo é tanto mais apto quanto maior for a sua pontuação no fim de terminar o nível.

Recombinação

A recombinação de dois progenitores faz com que os indivíduos troquem o seu material genético produzindo indivíduos com novas características. No script `ExampleIndividual.cs` implementámos 3 tipos de recombinação:

- **Recombinação com N pontos de corte:** é possível escolher o número de pontos de corte desejados, sendo as posições de corte aleatoriamente escolhidas baseado no tamanho do indivíduo. No nosso caso consideramos que o número de pontos de corte é um número par.
- **Recombinação uniforme:** cada gene tem 50% de hipóteses de ser recombinado.
- **HalfCrossover:** recombinação com apenas um ponto de corte localizado na posição central dos indivíduos.

Mutação

A mutação consiste na alteração do valor de um ou mais genes de um cromossoma. Também é vista como um operador secundário de recombinação, no entanto a mutação desempenha um papel importante no funcionamento dos algoritmos genéticos. Este operador evita, mesmo que utilizado com probabilidades bastantes baixas, que o algoritmo estagne, agitando-o e fazendo com que este explore outras regiões que podem ser mais promissoras. Implementámos 2 tipos de mutação:

- **New Value Mutation:** caso um gene sofra mutação, será modificado aleatoriamente para um dos restantes dois valores (no caso do gene ser -1, após a mutação poderá ser 0 ou 1).
- **Flip Mutation:** quando o gene que sofre mutação é igual a 1 passa a ser -1 e vice versa. No caso do gene ser igual a 0 mantém-se com o valor 0 após a mutação.

Seleção

Os mecanismos de selecção podem ser vistos como a sobrevivência dos mais aptos. Estes mecanismos devem possibilitar que os melhores indivíduos se reproduzam mais vezes, para que, desta forma, a população vá evoluindo até à convergência.

Os métodos de seleção requeridos no enunciado eram a seleção por torneio e a seleção elitista. Abaixo passamos a descrever mais pormenorizadamente cada um deles:

- **Seleção por Torneio**

Este algoritmo baseia-se na escolha de dois indivíduos aleatórios dentro da população e na geração de um número aleatório, r , com valor entre $[0,1]$. Definimos também um valor constante, k , com valor entre $[0,1]$. Se o valor gerado para r for menor que o valor definido para k então é escolhido o melhor indivíduo, caso contrario, é escolhido o pior.

Este algoritmo encontra-se implementado no script *TournamentSelection.cs*.

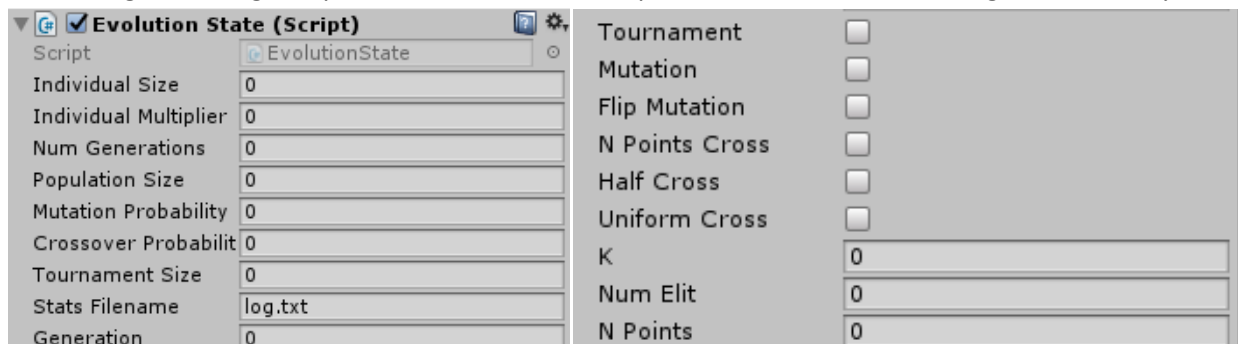
- **Seleção Elitista**

A seleção elitista serve de complemento ao método de seleção acima referido. Este algoritmo evita que os melhores indivíduos da população se percam, retendo um determinado número desses indivíduos para a geração seguinte.

Como se tratava de um **problema de maximização** teve-se o cuidado de ordenar a população por ordem decrescente de aptidão e escolher os mais aptos, que se encontram logo no início.

Parametrização

A seguinte imagem apresenta as variáveis manipuláveis através da interface gráfica do Unity.



Experimentação e Análise

Nesta fase apresentamos todas as análises realizadas. Todos os resultados representam a média dos valores de 30 simulações.

Recombinação

Em primeiro lugar testamos as diferenças entre os tipos de recombinação. Os parâmetros que serviram como base para estes testes são os seguintes:

Nº Gerações	100
Tamanho População	50
Probabilidade Mutação	0.05
Probabilidade Crossover	0.8
Tamanho Torneio	2
Elitismo	2
K	0.8

Algoritmo	Média	Melhor Indivíduo	Pior Indivíduo	Desvio
N pontos de Corte (N = 2)	82.5	201.1	4.6	46.3
N pontos de Corte (N = 4)	78.4	196.2	2.8	45.6
Half Crossover	45.8	182.2	1.2	46.4
Uniforme	89.1	201.3	4.9	45.8

Analisando os resultados da recombinação com N pontos de corte verifica-se que obtemos piores resultados com um N maior. Isto acontece porque o estado atual depende dos estados anteriores, por exemplo, se um indivíduo completar bem a primeira parte de um nível e outro indivíduo completar bem a segunda parte desse mesmo nível, seria de esperar que juntando as duas partes num único indivíduo o tornaria perfeito. No entanto isso não se verifica porque o segundo indivíduo apenas consegue completar bem a segunda parte porque a sua primeira parte lhe permitiu ficar numa posição privilegiada. Ora, com um maior número de pontos de corte, este efeito de dependência torna-se cada vez mais evidente, dando origem a soluções não óptimas.

O algoritmo HalfCrossover é um caso específico do anterior. Neste algoritmo existe apenas um ponto de corte localizado no centro dos indivíduos, sendo o problema explicado para o algoritmo anterior ainda mais evidente. Isto leva a que os resultados sejam maus, comparativamente com os outros.

Verificamos que o melhor algoritmo de recombinação é o uniforme. Este algoritmo também pode ser visto como um caso especial do primeiro algoritmo, sendo que aqui cada gene poderá ser um ponto de corte (com uma probabilidade de 50%). Neste caso o problema de dependência não é tão evidente porque o mais importante não é o tamanho do corte.

Mutação

De seguida testámos as diferenças entre os dois tipos de mutação. Os parâmetros que serviram como base para estes testes são os seguintes:

Nº Gerações	100
Tamanho População	50
Probabilidade Mutação	0.05
Probabilidade Crossover	0.8
Tamanho Torneio	2
Elitismo	2
K	0.8

Algoritmo	Média	Melhor Indivíduo	Pior Indivíduo	Desvio
New Value Mutation	89.1	201.3	4.9	45.8
Flip Mutation	88.2	199.5	3.9	48.5

Analisando os resultados obtidos verificamos que as diferenças entre os dois algoritmos não são tão visíveis como no caso das recombinações. Isto é explicado devido às semelhanças entre os dois algoritmos. No primeiro os valores dos genes mutados são modificados aleatoriamente para um valor diferente, enquanto que no segundo os valores dos genes mutados invertem-se no caso de 1 ou -1 e mantém-se no caso de 0.

Apesar das semelhanças o primeiro algoritmo obteve melhores resultados.

Probabilidade de Recombinação

De seguida testámos diferentes tipos de probabilidade para a recombinação. Os parâmetros que serviram como base para estes testes são os seguintes:

Nº Gerações	100
Tamanho População	50
Probabilidade Mutação	0.05
Tamanho Torneio	2
Elitismo	2
K	0.8

Probabilidade	Média	Melhor Indivíduo	Pior Indivíduo	Desvio
0.7	88.4	200.9	6	46.7
0.8	89.1	201.3	4.9	45.8
0.9	87.6	199.1	3.1	46.2

Com estes resultados verificamos que não existem grandes diferenças ao mudar a probabilidade de ocorrência de recombinação, o que dá a entender que o crossover não é tão relevante como a mutação neste tipo de problema.

Ainda assim, verificamos que os melhores resultados foram obtidos com uma probabilidade de 0.8.

Probabilidade de Mutação

Fazendo o mesmo para a probabilidade de mutação obtemos os seguintes resultados com base nos seguintes parâmetros:

Nº Gerações	100
Tamanho População	50
Probabilidade Crossover	0.8
Tamanho Torneio	2
Elitismo	2
K	0.8

Probabilidade	Média	Melhor Indivíduo	Pior Indivíduo	Desvio
0.01	142.2	200.9	37.2	37.5
0.05	89.1	201.3	4.9	45.8
0.1	70.1	193.6	2.3	55.1

No caso da mutação verificamos grandes diferenças consoante o valor da probabilidade.

Com uma probabilidade de mutação baixa sabemos que irá existir menos diversidade na população, levando a que a média seja elevada e o desvio seja baixo. Temos também melhores resultados no pior indivíduo, sendo que o melhor indivíduo será mais baixo, já que é menos provável que ele explore o espaço ao seu redor e possa seguir um caminho que leve a melhores soluções.

Por outro lado, com uma probabilidade de mutação alta teremos uma população com muita diversidade, levando a que a média seja baixa e o desvio seja alto. A média para o melhor e o pior indivíduo também é baixa porque com esta taxa elevada haverá casos em que os indivíduos passarão ao lado de uma boa solução quando estavam próximos de a alcançar.

Quanto ao desvio pode verificar-se que se encontra dentro do esperado, isto porque é maior quando a probabilidade é maior (quando existe mais diversidade, levando a maiores oscilações) e é menor quando a probabilidade também é menor (quando existe menos diversidade, levando a menores oscilações.)

A solução intermédia de 0.05 acabou por ser a que alcançou melhores resultados ao nível do melhor indivíduo.

Depois de analisados os resultados de mutação e recombinação verificamos que para este problema em específico a mutação é mais relevante do que a recombinação.

Elitismo

De seguida testámos o número de elitismo, isto é, o número de indivíduos que serão guardados para a geração seguinte baseado na sua aptidão. Os parâmetros que serviram como base para estes testes são os seguintes:

Nº Gerações	100
Tamanho População	50
Probabilidade Mutação	0.05
Probabilidade Crossover	0.8
Tamanho Torneio	2
K	0.8

Em primeiro lugar alterámos o número de elitismo para um valor muito grande, procurando perceber as diferenças que ocorrem.

Nº Elitismo	Média	Melhor Indivíduo	Pior Indivíduo	Desvio
2	89.1	201.3	4.9	45.8
40	129.4	184.4	24.3	27.2

Depois de analisar os resultados é fácil perceber o que está a acontecer. No nosso caso, com uma população de 50 indivíduos e com um número de elitismo igual a 40, significa que teremos apenas 10 indivíduos na população que poderão sofrer alterações (mutação e recombinação), ou seja, haverá poucas possibilidades de encontrar uma solução melhor à existente, levando a que a média de aptidão para o melhor indivíduo seja mais baixa. Por outro lado, como existem sempre 40 indivíduos que passam da geração anterior e possuem já boa aptidão, verifica-se que a aptidão média e do pior indivíduo aumentam e o desvio diminui.

O ideal será encontrar um número intermédio que possa manter a média alta e ao mesmo tempo ter boas possibilidades de explorar o espaço e produzir bons indivíduos.

Nº Elitismo	Média	Melhor Indivíduo	Pior Indivíduo	Desvio
8	103.6	200.6	14.3	53.1
16	114.3	201.4	18.4	58.1
20	136.1	204.3	21.2	55.5
22	134.3	198.2	21.7	53.4

Após alguns testes concluímos que o melhor número de elitismo é 20. Podemos verificar um aumento nas aptidões até ao número 20 à medida que o elitismo vai aumentando, tal como seria de esperar.

Quanto ao desvio obtemos algumas oscilações. Isto deve-se ao facto do desvio depender da média das aptidões, isto é, um número baixo de indivíduos com boa aptidão não é suficiente para estabilizar a média e assim baixar o desvio (casos de 8 e 16). Por outro lado, quando começamos a ter bastantes indivíduos que vêm da geração anterior com boa aptidão, a média tende a estabilizar nesses valores dando origem a que o desvio padrão desça (casos 20 e 22).

Constante K (Torneio)

Como referido anteriormente, o algoritmo de seleção por torneio é limitado por uma constante K que determina a probabilidade de ser o escolhido o pior ou o melhor indivíduo como resultado do torneio. Ao alterar esta constante obtemos os seguintes resultados:

Nº Gerações	100
Tamanho População	50
Probabilidade Mutação	0.05
Probabilidade Crossover	0.8
Tamanho Torneio	2
Elitismo	2

Valor de K	Média	Melhor Indivíduo	Pior Indivíduo	Desvio
0.7	66.7	199.8	5.6	45.1
0.8	89.1	201.3	4.9	45.8
0.9	91	200.6	8.7	46.8

Depois de analisarmos os resultados verificamos uma grande diferença na média das aptidões entre 0.7 e 0.8, isto porque com um K igual a 0.7 existe mais probabilidade de ser escolhido o pior elemento, baixando um pouco a média. Entre 0.8 e 0.9 não se verificam grandes alterações nos resultados.

Tamanho da População

De seguida testámos um aumento no tamanho da população de 50 para 100. Os parâmetros que serviram como base para estes testes são os seguintes:

Nº Gerações	100
Probabilidade Mutação	0.05
Probabilidade Crossover	0.8
Tamanho Torneio	2
Elitismo	2
K	0.8

Nº Elitismo	Média	Melhor Indivíduo	Pior Indivíduo	Desvio
50	89.1	201.3	4.9	45.8
100	91.2	206.1	4.7	46.1

Podemos verificar que a aptidão média dos indivíduos é semelhante, bem como o desvio e pior indivíduo. Quanto à aptidão do melhor indivíduo é melhor com uma população maior. Isto é fácil de perceber pelo simples facto que ao existirem mais indivíduos, mais provável será que um deles venha a tornar-se numa ótima solução.

Número de Gerações

Todos os testes anteriores foram realizados sobre 100 gerações. No entanto, foi possível verificar uma estabilização nas aptidões ao fim de 65 - 70 gerações.

Combinação dos melhores parâmetros

Depois de realizados todos os testes apresentamos os melhores resultados obtidos para uma população de 50 indivíduos ao fim de 100 gerações, tendo em conta os parâmetros que obtiveram melhores valores:

Recombinação Uniforme	
New Value Mutation	
Probabilidade Crossover	0.8
Probabilidade Mutação	0.05
Tamanho do Torneio	2
Elitismo	20
K	0.8

Média	Melhor Indivíduo	Pior Indivíduo	Desvio
132.4	208.6	11.4	51.8

Conclusão

Ao analisar todos os testes efetuados, verificámos que existem 2 elementos que têm um grande impacto nos resultados. O primeiro é o número de elitismo. Neste caso, verifica-se que um número muito reduzido pode levar a soluções sub-ótimas, enquanto que um número demasiado elevado, apesar de melhorar significativamente a aptidão média, leva a que o espaço seja pouco explorado e consequentemente seja pouco provável encontrar uma melhor solução.

O segundo é a probabilidade da mutação. Se tivermos uma probabilidade demasiado elevada iremos ter demasiada instabilidade, resultante da enorme diversidade existente, o que pode levar um indivíduo a saltar uma boa solução quando poderia estar perto de a alcançar. Por outro lado, uma probabilidade demasiado baixa pode levar a que os indivíduos fiquem presos a uma solução que pode não ser ótima, isto porque a probabilidade de explorarem o espaço ao seu redor é também ela baixa.

Os restantes elementos não têm o mesmo impacto nos resultados obtidos. No caso da recombinação vimos que não tem tanto impacto como seria de esperar e que não é muito relevante neste problema específico. Isto deve-se ao problema de dependência que explicámos anteriormente, gerando instabilidade nos indivíduos resultantes da recombinação.