Faculdade de Ciências e Tecnologia da Universidade de Coimbra  
Departamento de Engenharia Informática

Introdução à Inteligência Artificial

Trabalho Prático nº 2

***Curva Braquistócrona***

Trabalho realizado pelos alunos:

2006125142 João Claro

2008114843 Ricardo Lopes

2008115099 Rui Chicória

**Índice**

Autores 3

Introdução 4

Requisitos 5

Meta 1 – Modelação e desenvolvimento do algoritmo genético 6

Meta 2 – Experimentação e análise 7

Conclusão 18

**Autores**

**João Claro**

Número de aluno: 2006125142

Email: jclaro@student.dei.uc.pt

Tempo de estudo: 2 horas

Tempo de implementação: 20 horas

Contributo para o trabalho: Testes da meta 1, programação geral, relatório

**Ricardo Lopes**

Número de aluno: 2008114843

Email: rplopes@student.dei.uc.pt

Tempo de estudo: 5 horas

Tempo de implementação: 35 horas

Contributo para o trabalho: Cenários da meta 2 e 3, programação geral, relatório.

**Rui Chicória**

Número de aluno: 2008115099

Email: chicoria1@student.dei.uc.pt

Tempo de estudo: 4 horas

Tempo de implementação: 35 horas

Contributo para o trabalho: Cenários da meta 2 e 3, programação geral, relatório.

**Introdução**

Com a elaboração deste trabalho pretende-se desenvolver e utilizar um agente adaptativo com o objectivo de encontrar uma boa aproximação da curva braquistócrona que une dois pontos.

O problema da curva braquistócrona é um problema de optimização, visto que se pretende encontrar a curva que permite que um dado objecto, deslizando sem atrito sobre a cruva, sujeito apenas à força da acção da gravidade, a atravesse num tempo mínimo. Assim, a utilização de uma abordagem adaptativa apresenta-se como uma boa solução para procurar uma boa aproximação para esse resultado óptimo.

Para tal, pretende-se seguir uma divisão do trabalho a efectuar em duas etapas distintas. Na primeira etapa, tem-se por objectivo o desenvolvimento do programa em Python capaz de efectuar a abordagem adaptativa ao problema da curva braquistócrona. Na última etapa, pretende-se efectuar uma grande sucessão de testes experimentais que permitam tirar conclusões relativas à eficácia, eficiência e robustez das diferentes configurações possíveis do algoritmo.

Para a realização deste trabalho, recorre-se à linguagem Python para a programação do algoritmo, e ao módulo matplotlib para obter a representação gráfica dos resultados obtidos na experimentação. É ainda utilizado o ficheiro disponibilizado pelos docentes, BrachFitness.py, que permite avaliar a aptidão e validade dos indivíduos.

**Requisitos**

Para a execução correcta desta aplicação, deve-se ter em conta os seguintes requisitos:

Módulos de Python a instalar:

- matplotlib (para instalar em Linux, executar sudo apt-get install python-matplotlib)

Ficheiros/directorias necessários:

- Brachistochrone.py  
- BrachFitness.py

- Testes.py  
- conf.txt  
- testes (directoria)

**Meta 1 – Modelação e desenvolvimento do algoritmo genético**

Esta meta tem como objectivo desenvolver o algoritmo genético de aproximação da curva braquistócrona na linguagem Python.

Optou-se, para tal, por implementar duas representações diferentes para os indivíduos, uma em que os valores das abcissas dos pontos das curvas são fixos, todos a igual distância dos seus vizinhos, e outra em que os valores das abcissas dos pontos das curvas são dinâmicos, podendo variar ao longo da execução do algoritmo.

Implementou-se ainda os métodos de recombinação de n pontos e de mutação para a geração de novos descendentes em cada geração, os métodos de selecção por torneio e por roleta para a selecção dos progenitores em cada geração, e um parâmetro de elitismo para manter os melhores indivíduos da população de uma geração para a seguinte.

De modo a poder analisar os resultados das experiências, incluiu-se ainda no programa a capacidade de gerar a representação visual de indivíduos da população e a mostrar a evolução ao longo das gerações do valor do melhor indivíduo, do pior indivíduo, da média da aptidão da população e do seu desvio padrão. Todos esses valores são ainda guardados num ficheiro de texto, juntamente com a data do teste, os parâmetros utilizados e ainda o número total de recombinações e mutações efectuadas.

O algoritmo adaptativo, presente no ficheiro Brachistochrone.py, recorre ainda aos métodos do ficheiro BrachFitness.py, um ficheiro disponibilizado pelos docentes da disciplina, para o cálculo da aptidão dos indivíduos e para verificar se estes são ou não válidos.

Para ser possível definir os parâmetros do algoritmo adaptativo sem ser necessário alterar o código ou ter de os especificar individualmente para cada nova experiência, recorreu-se a um ficheiro de configuração, onde se pode definir os parâmetros desejados uma única vez, podendo-se alterar qualquer campo à escolha com bastante facilidade. Caso o programa não encontre o ficheiro de configuração ou esse se encontre corrompido ou com dados errados, um novo ficheiro de configuração é criado, com os parâmetros pré-definidos.

No ficheiro de configuração é possível definir as coordenadas dos dois pontos da curva braquistócrona, o número de gerações do algoritmo, o número de indivíduos de uma população, o número de genes de cada indivíduo, o número de indivíduos seleccionados no método do torneio (se for zero, então é usado o método de roleta para a selecção dos progenitores), o número de pontos de recombinação, a probabilidade de recombinação, a probabilidade de mutação, o tamanho em percentagem da elite e o tipo de representação desejado (pontos com abcissas fixas ou dinâmicas).

**Meta 2 – Experimentação e análise**

O objectivo desta etapa é efectuar uma sucessão de testes, fazendo variar os diversos parâmetros do algoritmo adaptativo, de forma a poder tirar conclusões sobre as configurações que permitem uma maior eficácia, eficiência e robustez do mesmo. Para tal, considerou-se as seguintes variações de parâmetros:

**Número de gerações:** 10, 20, 50, 100 default, 250

**Número de indivíduos de uma população:** 50, 100, 200 default, 500

**Número de genes de um indivíduo:** 15, 30 default, 60

**Método de selecção:** Roleta, Torneio (tamanho 2, 5, 10)

**Número de pontos de recombinação:** 1, 3, 5

**Probabilidade de recombinação:** 5%, 10%, 50% default

**Probabilidade de mutação:** 5%, 10% default, 15%

**Taxa de elitismo:** 0%, 10%

**Representação usada:** Abcissas fixas, Abcissas variáveis

**Pontos que definem a curva:** A(0,100) B(30,50), A(0,10) B(3,5), A(10,100) B(20,10), A(0,20) B(100,10)

 

**2**

**1**

 

**4**

**3**

Para cada uma destas variações, como se trata de testes de cariz aleatório, procedeu-se à sua experimentação 30 vezes, de modo a obter-se a média de todos os resultados, resultando assim num output mais credível. De seguida apresentam-se os resultados obtidos. Por motivos de legibilidade, apenas se apresentam os valores médios resultantes das 30 experimentações, evitando-se assim sobrecarregar o relatório com os 30 resultados diferentes para cada variação de parâmetro.

**Meta 2 – Conclusões**

(Intro, dizer onde estão as tabelas, que as conclusões são apenas para os pârametros usados, etc..)

Conjuntos de Pontos

Através da análise das diversas tabelas, foi possível analisar as diferenças de valores das aptidões dos diversos conjuntos de pontos. Essa análise permitiu chegar a algumas conclusões que são de seguida enunciadas:

O segundo conjunto de pontos equivale ao primeiro conjunto de pontos numa escala 10 vezes inferior. Como se pode ver nos resultados de qualquer tabela, a sua aptidão é sempre muito melhor que a do primeiro conjunto de pontos. Isso acontece porque, como os pontos estão a uma escala menor, encontram-se mais próximos e que a curva que os une e que traça o caminho do objecto é mais curta. Assim, é esperado que o objecto demore menos tempo a percorrer essa curva.

O terceiro conjunto de pontos define uma recta com um grande declive, e o quarto conjunto de pontos define uma recta com um declive muito reduzido. Como se pode verificar, as aptidões do terceiro conjunto de pontos são sempre melhores que as aptidões do quarto conjunto de pontos. Isso acontece porque o efeito da gravidade num percurso mais íngreme (terceiro conjunto de pontos) faz-se sentir muito mais do que num percurso mais plano, o que leva a que o objecto percorra a curva num menor espaço de tempo.

Tabela A.1.1

Para 15 genes nota-se que em 20 gerações a aptidão é sempre pior, mas que para as restantes parece não ter uma relação directa. Tal significa que, entre as 20 e as 50 gerações, para 15 genes, a população convergiu para um máximo. Para 30 genes já se verifica que sempre que o número de gerações aumenta a aptidão melhora. Esse comportamento encontra-se dentro do esperado, pois a quanto mais gerações a população está sujeita, mais esta evolui e melhora a sua aptidão.

Assim podemos concluir que para uma menor precisão (menos genes) o algoritmo vai convergir muito mais rapidamente, visto que há muito menos informação para ser modificada ao longo das gerações.

Quanto maior for o número de indivíduos de uma população, melhor é a sua aptidão. Isto deve-se ao facto de haver mais código genético diferente a ser recombinado. Esta relação é mais notória para indivíduos com um maior número de genes. Tal comportamento é explicado por, tal como se verificou anteriormente, as populações de indivíduos com um número de genes menor converger muito mais rapidamente para um máximo, ou seja, o efeito do número de indivíduos dessa população é minorado.

Tabela A.2.1

Quanto maior a probabilidade de recombinação melhor a aptidão e menor o desvio padrão. Este resultado é esperado porque são gerados mais índividuos novos e o processo evolutivo é mais rápido.

Para valores maiores do torneio há uma tendência para um desvio padrão menor. Quanto maior o tamanho do torneio mais probabilidade há de se seleccionar os individuos com melhor aptidão o que faz com que as próximas gerações fiquem com muitos indíviduos semelhantes ou até iguais.

Nota-se que o desvio padrão da roleta é muito superior ao do torneio e muitas vezes as aptidões são piores. Assim se pode concluir que converge mais lentamente. Isso é explicado pelo facto das aptidões dos indíviduos serem muito semelhantes, logo a probabilidade de escolher um bom indíviduo é praticamente a mesma de escolher um pior índividuo o que vai levar que a selecção seja mais aleatória e menos baseada no mérito.

Para uma probabilidade de mutação maior a convergência não se dá tão facilmente. As aptidões e desvio padrão têm tendência para melhorar. Isto é explicado pelo facto da mutação trazer mais aleatoriedade à criação de novos individuos, o que traz também uma maior capacidade de fugir a um máximo local. Como na recombinação não se criam genes novos é mais fácil convergir para um máximo local.

A mutação cria maior diversidade de indíviduos (desvio padrão) e mais capazes (aptidão)