Elementos de Programação

Projecto de Computação Evolutiva

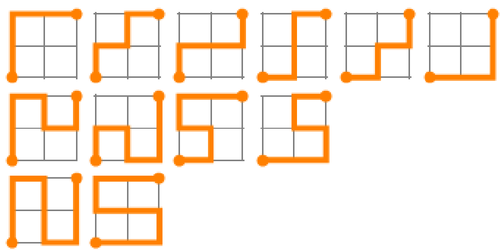
Departamento de Matemática, IST

LMAC, MEBiom
Novembro de 2012

Qual é o melhor caminho?

Considere-se uma grelha de pontos $n \times n$ e uma função f_{custo} que associa a cada ponto (nó) da grelha o seu custo (número real). Pretende-se encontrar o melhor caminho (i.e., o caminho com menor custo acumulado) que, navegando pelas arestas da grelha e sem repetir pontos, una o canto inferior esquerdo (origem - $(0,0)$) ao canto superior direito (destino - (n,n)).

No caso, com generalidade, não é conhecido nenhum algoritmo eficiente para calcular o melhor caminho. No limite, todos os algoritmos conhecidos consistem essencialmente em verificar um a um todos os caminhos possíveis para determinar o de menor custo. No entanto, é fácil de ver que o número de caminhos a considerar é (mais do que) exponencial em n. De facto, o número de caminhos possíveis numa grelha $n \times n$ corresponde à sucessão A007764 da *On-Line Encyclopedia of Integer Sequences* (vide `http://oeis.org/A007764`), cujos primeiros termos são 1, 2, 12, 184, 8512, 1262816, 575780564, 789360053252, 3266598486981642, ... Para ilustração, os 12 caminhos possíveis numa grelha 2×2 são os indicados na figura.



Há muitos exemplos de problemas de grande utilidade prática para os quais não é conhecida uma solução geral eficiente, tal como o problema descrito acima, que tem inúmeras aplicações relevantes, nomeadamente em tarefas de planeamento e optimização em engenharia.

Na verdade, este é um problema dito NP-difícil, uma classe de problemas para os quais se conjectura não existir nenhum algoritmo eficiente. Assim sendo, é usual recorrer-se a soluções aproximadas do problema.

Computação Evolutiva

O objectivo deste projecto é desenvolver em *Mathematica* um programa que calcule uma solução aproximada para o problema do melhor caminho numa grelha $n \times n$ e para uma função de custo

f_{custo} dada, utilizando o paradigma de computação evolutiva.

A ideia consiste na simulação discreta estocástica, durante um período de tempo dado *Tfim*, da evolução de uma população inicial formada por um número dado *Ind* de indivíduos (exploradores) que, inicialmente colocados na origem da grelha, progredirão independentemente avançando ao longo das arestas da grelha até, eventualmente, completarem um caminho até ao destino. A evolução de cada indivíduo far-se-á de acordo com leis aleatórias dependentes da sua adaptação ao problema (que deve ser tanto maior quanto o caminho percorrido tenha menor custo e mais se aproxime do destino). Após a simulação, a solução aproximada calculada corresponderá ao melhor caminho completado por algum indivíduo ou, caso não haja caminhos completos, ao melhor adaptado dos indivíduos da população final.

O coeficiente de adaptação de um indivíduo que até certo instante tenha percorrido um caminho p até ao ponto (x, y) é dado por

$$adapt(p) = \frac{1}{2^{\frac{custo(p)}{compr(p)}}(1 + dist(x, y))}$$

onde $compr(p)$ é o comprimento do caminho p, $custo(p)$ é o custo acumulado do caminho p, e $dist(x, y) = (n - x) + (n - y)$ é a distância mínima de (x, y) até ao nó destino.

Cada indivíduo da população deve ter um identificador único irrepetível, pelo que poderão coexistir vários indivíduos diferentes correspondentes ao mesmo caminho, tal como na população inicial. Ao longo da simulação, cada indivíduo pode sofrer um deslocamento (progresso no caminho para um nó adjacente), reproduzir-se (dando origem a um novo indivíduo com o mesmo caminho do progenitor), ou morrer, de acordo com leis aleatórias que dependem do coeficiente de adaptação do indivíduo. Por razões óbvias, os indivíduos que completem um caminho até ao nó destino param de evoluir. Sempre que a população cresça acima de um limiar *Poplim* dar-se-á um cataclismo que resultará na morte de todos os indivíduos activos da população excepto os *Ind* melhor adaptados.

O simulador deve ser construído de acordo com a técnica de simulação digital estocástica por sequenciamento de eventos pendentes, havendo eventos de 4 tipos:

- *deslocamento*, de certo *indivíduo*, cuja cadência é dada por uma variável aleatória exponencial de valor médio
$$\frac{adapt(p)}{10}$$
onde p é o caminho que lhe está associado, e cuja ocorrência resulta no deslocamento do indivíduo para um dos nós adjacentes (escolhido uniformemente*); caso o indivíduo se desloque para um nó que já ocorre no seu caminho, deve ser eliminado o troço compreendido entre as duas ocorrências desse nó; caso o deslocamento dê azo a um caminho completo até ao nó destino, o indivíduo deve parar de evoluir;

- *reprodução*, de certo *indivíduo*, cuja cadência é dada por uma variável aleatória exponencial de valor médio
$$\frac{1}{1 + adapt(p)}$$
onde p é o caminho que lhe está associado, e cuja ocorrência resulta no nascimento de um novo indivíduo com o mesmo caminho do progenitor; caso o número de indivíduos

*Note que em *Mathematica* a expressão `RandomInteger[{1,N}]` toma uniformemente cada um dos valores de `1` a `N`, isto é, cada valor tem probabilidade $1/N$. Da mesma forma, a expressão `RandomSelect[w]` escolhe uniformemente um dos elementos da lista `w`.

da população exceda *Poplim*, agendar-se-á um *cataclismo* que ocorrerá, uniformemente, dentro das próximas

$$\frac{Ind}{Poplim}$$

unidades de tempo;

- *morte*, de certo *indivíduo*, sendo o tempo de vida de um indivíduo uma variável aleatória exponencial de valor médio

$$0.2 + adapt(p)$$

onde p é o caminho que lhe está associado, e cuja ocorrência resulta na morte do indivíduo em causa;

- *cataclismo*, cuja ocorrência resulta na morte de todos os indivíduos activos da população, excepto os *Ind* melhor adaptados.

Objectivos

O projecto deve ser desenvolvido de acordo com o método de programação modular, por camadas, centrado nos dados.

1. Comece por identificar os tipos de dados relevantes[†], nomeadamente *caminho*, *indivíduo*, *população*, *evento*, e *cadeia de acontecimentos pendentes*, e respectivas operações.
2. Desenvolva de seguida o programa abstracto pretendido sobre a camada que disponibiliza estes objectos.
3. Implemente estas camadas sobre a camada básica do *Mathematica*.
4. Integre o programa obtido em 2 com os pacotes desenvolvidos em 3.
5. Experimente o programa desenvolvido com diversos conjuntos de dados à sua escolha.

O projecto, a ser realizado (preferencialmente) por grupos de 3 alunos, será entregue através do sistema Fénix, após a inscrição do respectivo grupo, até ao final do dia 15 de Dezembro de 2012, **impreterivelmente**. A entrega do trabalho deve consistir de um único arquivo (zip ou rar) contendo o simulador, os pacotes desenvolvidos, e um pequeno relatório que descreva sucintamente a solução obtida, os tipos de dados utilizados e respectivas opções de implementação, e uma discussão dos resultados obtidos.

O trabalho vale 10 valores da nota final da disciplina, que se distribuem da seguinte forma:

- Descrição dos tipos de dados relevantes e suas operações
  (2.5 valores)
- Implementação eficiente dos tipos de dados e sua apresentação sob a forma de pacotes
  (3 valores)
- Simulador
  (3 valores)
- Experimentação
  (1.5 valores)

[†] Note que, que os pontos da grelha podem ser representados directamente por pares de inteiros, tal como os identificadores dos indivíduos podem ser representados directamente por inteiros positivos.