ECO
Programa de Simulação

António M. Pina
Dep. Informática, Universidade do Minho
Largo do Paço, 4709 Braga, Portugal
Email: `pina@di.uminho.pt`

1 Sistemas adaptativos

O modelo **predador-presa** que a seguir apresentamos é o ponto de partida para o desenvolvimento de um conjunto de experiências que envolvem a paralelização de uma aplicação "realista'.
O problema baseia-se numa classe de modelos de sistemas adaptativos complexos, colectivamente designados por ECO[1][2][1], que abstraem as detalhes físicos dos sistemas reais para se concentrarem num pequeno número de primitivas para a interacção entre agentes e entre agentes e ambiente.
O objectivo é estudar de que forma interacções simples entre agentes podem fazer emergir fenómenos complexos, tais como o fluxo de recursos em sistemas de coordenação e de competição entre agentes.

Predador-presa

A ideia geral é tomar como base uma versão sequencial já realizada de um problema da classe ECO que irá servir de plataforma para a criação de um programa paralelo com funcionalidade semelhante à da versão sequencial.
Tomemos como exemplo um território com uma área rectangular de terreno habitado por raposas e por coelhos. Ao longo dos anos, sucessivas gerações daqueles animais são estimadas de acordo com um modelo simples de predador-presa para determinar a forma como as duas populações evoluem conjuntamente.
A solução do problema parte da divisão do terreno em quadrículas, com 1 quilómetro de lado, e a fixação das populações iniciais. Em cada período de um ano o número de coelhos e de raposas são estimados de acordo com uma fórmula matemática discreta que representa a variação das populações, em cada quadrícula, como função *do nascimento, da morte, ou da migração para outras quadrículas*.

Modelo bi-dimensional

Para um determinado território suponhamos a existência de **NS_Q e EO_Q** quadrículas ao longo dos eixos Norte/Sul e Este/Oeste. Considerarmos, ainda, que os identificadores r_{ij} e c_{ij} representam o número de raposas e de coelhos que vivem em cada quadrícula ij no princípio de cada ano.
O número de coelhos que podem sobreviver até ao fim do ano é calculado segundo a fórmula:

$$c_{ij}^{k+1} = \alpha_c c_{ij}^k + \beta_c r_{ij}^k + \mu_c \Delta_c.$$

Na fórmula α_c, β_c e μ_c são valores constantes que representam, respectivamente, *o nascimento, a morte e a taxa de migração de coelhos* no território. A nova geração de raposas é calculada, de forma semelhante, segundo a fórmula:

$$r_{ij}^{k+1} = \alpha_r c_{ij}^k + \beta_r r_{ij}^k + \mu_r \Delta_r.$$

[1] Uma introdução aos modelos ECO – *ftp santafe.edu – get pub/Users/terry/echo/Echo-01.tar.Z*

Os valores Δ_c, Δ_r são factores de migração que obedecem a expressões próprias[2]. O território é circular, tanto na direcção N/S como na direcção E/O, com período igual ao tamanho da dimensão considerada.

2 Código sequencial

O desenvolvimento do programa paralelo ECO envolve numa primeira fase a compreensão do problema e a execução da versão sequencial fornecida como material pedagógico.
O programa no ficheiro *n2fox.c* corre após compilação, presumivelmente, em qualquer estação de trabalho. Um outro programa, essencialmente equivalente, *ng2fox.c* usa o pacote gráfico `gnuplot` para gerar um gráfico das seis primeiras iterações de cada simulação.
Os ficheiro *macros.h*, *param.h* contêm macro-definições e constantes gerais usadas durante o programa. O ficheiro *Makefile* define o procedimento de compilação e criação dos programas executáveis sequenciais.

O programa sequencial

O código escrito em linguagem `C` usa duas matrizes bi-dimensionais com NS_Q, EO_Q quadrículas, independentes, como estruturas de dados para conter as diferentes populações de animais. Cada elemento $m_{i,j}$ contém o número indivíduos de cada espécie numa quadrícula sendo a população inicial estabelecida através da chamada ao procedimento `Território`.
O procedimento `Geração` calcula o número de indivíduos de cada espécie em cada ano $(k+1)$ a partir da população do ano anterior (k) sendo o valor de cada quadrícula guardado no elemento da matriz de animais correspondente.
Por causa das condições de periocidade em cada dimensão do território, ambas as matrizes possuem uma linha (0) que duplica os elementos da matriz da linha NS_Q. Por motivos idênticos são definidas as linhas NS_Q+1 e as colunas 0 e EO_Q+1. Estas linhas e colunas adicionais são chamadas auréolas[3].
O procedimento `Bordos` é chamado antes do cálculo de cada nova geração para o preenchimento das auréolas.
Em cada iteração a evocação do procedimento `População` permite determinar a população actual de cada espécie.

[2] O cálculo deste valores está descrito no código do programa sequencial.
[3] *Halo* em terminologia anglo-saxónica.

Resumo

- **Declaração de Matrizes**

```
float Coelho [NS_Q+2][EO_Q+2];
float Raposa [NS_Q+2]][EO_Q+2];
```

- **Declaração de constantes**

Constantes	*Coelhos*	*Raposas*
α	-0.2	-1.8
β	0.6	0.6
μ	0.01	0.02

Tabela 1: Constantes na fórmula.

- **População inicial**

$C_{ij} = 128.0{*}(i-1){*}(NS\_Q-i){*}(j-1){*}(EO\_Q-j)/(NS\_Q{*}NS\_Q{*}EO\_Q{*}EO\_Q).$

$R_{ij} = 8.0{*}(i/(NS\_Q)-0.5){*}(i/(NS\_Q)-0.5)+8.0{*}(j/(EO\_Q)-0.5){*}(j/(EO\_Q)-0.5).$

- **Nova geração**

$$C_{ij} = (1.0 + \alpha_c - 4.0 * \mu_c) * C_{ij} + \beta_c * R_{ij} + \mu_c * (C_{ij-1} + C_{ij+1} + C_{i-1j} + C_{i+1j}).$$

$$R_{ij} = AlF * C_{ij} + (1.0 + \beta_r - 4.0 * \mu_r) * R_{ij} + \mu_r * (R_{ij-1} + R_{ij+1} + R_{i-1j} + R_{i+1j}).$$

- **Procedimentos**

`Território` – (*Setland*)
: Estabelecimento das populações iniciais de ambas as espécies.

`População` – (*GetPopulation)*)
: Chamada repetidamente para calcular o número de indivíduos de cada populações após cada iteração.

`Bordos` – (*FillBorder*)
: Preenchimento das auréolas das matrizes de cada população.

`Geração` – (*Evolve*)
: Cálculo das novas populações de coelhos e de raposas no final de cada ano de simulação.

3 Decomposição paralela

Para resolver um problema paralela podem usar-se uma grande variedade de técnicas de decomposição (ver fig. 1). Nesta secção apresentamos apenas três técnicas de decomposição.

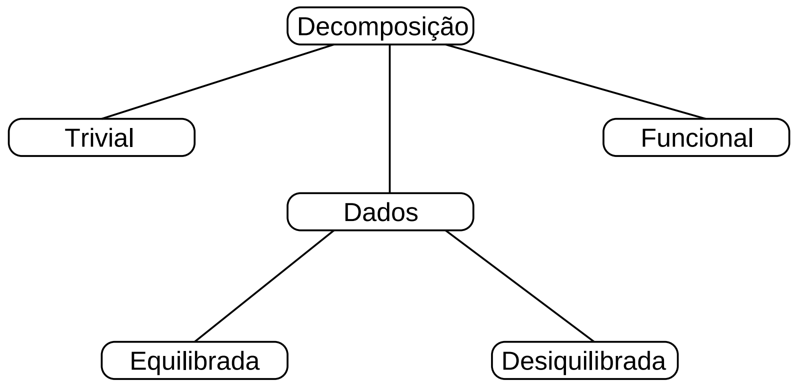

Figura 1: Decomposição paralela.

Domínios regulares

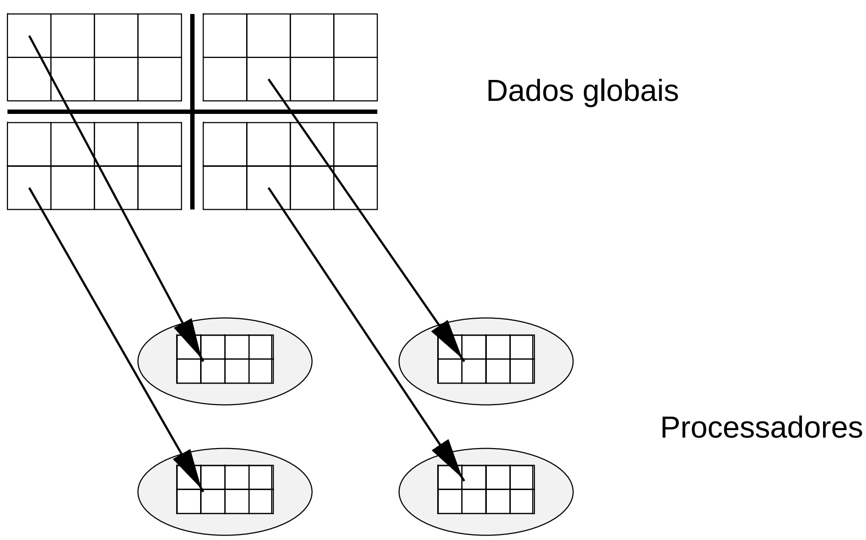

Figura 2: Decomposição regular.

Referências

[1] John Holland. Proposal for a Research in Adaptive Computation. Technical report, Santa Fe Institute, July 1992.

[2] John Holland. *Adaptation in Natural and Artificial Systems.* MIT Press, 1992.