

Elementos de Programação

Relatório da Segunda Parte do Projeto

Departamento de Matemática, IST

2018/2019

Para o simulador decidimos usar os módulos que nós definimos, nomeadamente *event_c*, *cap_c*, *individuo_c* e *grelha_c*. Os módulos utilizados foram os enviados anteriormente exceto a função *eliminarID(ID)* da *cap*. A função foi utilizada foi a seguinte:

```
def eliminarID(self,ID):
    i=len(self.CAP)-1
    while i>=0:
        if self.CAP[i].ID()==ID:
            self.CAP.remove(self.CAP[i])
        i=i-1
```

→ Enviamos em anexo os módulos já corrigidos.

→ Definimos uma variável global *novolD*, que nos permitiu ir criando novos e diferentes identificadores para cada indivíduo.

Funções

Definimos as seguintes funções:

- *texp(x)*
- *novol(e,p,t)*
- *popular(w,Ps,Pi,cen,padj)*
- *Pe(x)*

texp(m)

Dado um tempo *m*, calcula um tempo aleatório usando distribuição exponencial.

novolD(e,p,t)

Dado um estado, uma posição e o tempo atual da simulação, cria um indivíduo com uma nova identificação e o estado “e”, adiciona o mesmo na posição “p” e adiciona à CAP um evento da deslocação, reprodução, morte e (dependendo do estado) avaliação (eventos

“essenciais”) no tempo atual “t” mais o tempo médio de evento aleatório usando distribuição exponencial.

popular(w,Ps, Pi,cen,padj)

Esta função popula a grelha, isto é, acrescenta indivíduos à grelha, dependendo do cenário. Esta distribuição pode ser aleatória (cenário ‘a’) ou agrupar uma percentagem, dada pelo utilizador, de indivíduos em posições adjacentes (cenário ‘b’). O cenário é escolhido através do parâmetro *cen* da função *sim*. e a **percentagem de indivíduos em posições adjacentes pelo parâmetro padj**

Se o cenário escolhido for o cenário ‘a’, os indivíduos são colocados aleatoriamente na grelha.

Caso o cenário escolhido seja o cenário ‘b’, o primeiro indivíduo é colocado aleatoriamente na grelha, e os indivíduos seguintes (até a percentagem ser atingida) são adicionados adjacentes ao anterior. Quando este processo está concluído, os restantes indivíduos são adicionados aleatoriamente na grelha (tal como no cenário ‘a’)

Pe(x)

Dado o ID de um individuo, calcula a probabilidade deste passar de suscetível a exposto. Calcula n1 calculando o tamanho da lista dos individuos infetados da vizinhança 1 do individuo com ID = x e n2 calculando o número de individuos infetados na vizinhança 2 que se encontram em contacto com o individuo com ID = x. Serão estes valores depois a serem aplicados à expressão dada.

While do simulador

Para começar, definimos uma grelha, uma *cap*, populamos a grelha e define-se o tempo(=0).

O simulador tem como condição de paragem o tempo limite da simulação e a *cap* (a simulação acaba quando a *cap* estiver vazia).

Para cada evento, o simulador verifica o tipo de evento que vai ocorrer (deslocamento, reprodução, morte ou avaliação) e dependendo da probabilidade de cada evento, este é realizado ou não. Caso seja realizado, ocorre como explicado acima para cada evento.

Ao longo do *while*, é adicionado a uma lista valores que x e y que irão formar o gráfico. O valor de x corresponde ao tempo decorrido, e começa em 0 e vai até Th. O y é o número de infetados na simulação ao longo do tempo. Este valor aumenta cada vez que um indivíduo exposto passa a infetado e diminui de cada vez que um indivíduo infetado morre ou um indivíduo infetado volta a recuperado.

No final do ciclo é definida uma variável “prox” contendo o próximo evento, assim como o tempo do mesmo (para ser avaliado na condição de paragem) e o ID

Tipos de Eventos

Deslocamento

Se o comprimento da lista das posições livres da vizinhança 1 do indivíduo com ID = x for superior a 0 (quando existe posições adjacentes livres), se o comprimento da lista das posições dos indivíduos infectados da vizinhança 1 do indivíduo com ID = x for superior a 2, este move-se sempre, inserindo na grelha o indivíduo numa dessas posições livres e apagando-o da posição inicial, caso o comprimento da lista das posições dos infectados da vizinhança 1 do indivíduo com ID = x for inferior a 3, este desloca-se (inserindo na grelha o indivíduo numa dessas posições livres e apagando-o da posição inicial) com probabilidade pd .

Quando o evento acaba de ocorrer (havendo ou não deslocação), é criado o próximo evento de deslocação para o indivíduo com ID = x com o tempo de evento igual ao tempo atual mais o tempo médio de deslocação aleatório usando distribuição exponencial e adicionado à CAP.

Reprodução

Se o comprimento da lista das posições livres da vizinhança 1 do indivíduo com ID = x for superior a 0 (quando existe posições adjacentes livres) e se o comprimento da lista das posições ocupadas por indivíduos da vizinhança 1 do indivíduo com ID = x for superior a 0, este reproduz-se, utilizando a função *novoID* para criar, adicionar à grelha, criar os eventos “essenciais” e adicioná-los à CAP numa posição aleatória adjacente ao indivíduo com ID = x com probabilidade pr .

Quando o evento acaba de ocorrer (havendo ou não reprodução), é criado o próximo evento de reprodução para o indivíduo com ID = x com o tempo de evento igual ao tempo atual mais o tempo médio de reprodução aleatório usando distribuição exponencial e adicionado à CAP.

Morte

Se o estado do indivíduo com ID = x for “infectado”, este é apagado da grelha e todos os seus eventos na CAP são também eliminados com probabilidade pm mais 10% de pm ($pm * 1.1$) com o máximo de $pm = 1$. Se no entanto o estado do indivíduo com ID = x não for “infectado”, este apenas é apagado da grelha e todos os seus eventos na CAP são também eliminados com probabilidade pm .

Quando o evento acaba de ocorrer e somente se não houver morte, é criado o próximo evento de morte para o indivíduo com ID = x com o tempo de evento igual ao tempo atual mais o tempo médio de morte aleatório usando distribuição exponencial e adicionado à CAP.

Avaliação

Se o estado do indivíduo com ID = x for “infectado”, este muda o seu estado para “recuperado”

Se o estado do indivíduo com ID = x for “exposto”, este muda o seu estado para “infectado” e é criado o próximo evento de avaliação para o indivíduo com ID = x com o tempo

de evento igual ao tempo atual mais o tempo médio de avaliação “I” aleatório usando distribuição exponencial e adicionado à CAP.

Se o estado do indivíduo com ID = x não for nem “infectado” nem “exposto” (logo, indivíduo com estado “suscetível”, visto que um indivíduo com estado “recuperado” não está sujeito a avaliação) e o comprimento da lista das posições dos indivíduos infectados da vizinhança 1 e 2 do indivíduo com ID = x for superior a 0, este muda o seu estado para “exposto” e é criado o próximo evento de avaliação para o indivíduo com ID = x com o tempo de evento igual ao tempo atual mais o tempo médio de avaliação “E” aleatório usando distribuição exponencial e adicionado à CAP com probabilidade $P_e(x)$. Caso o comprimento da lista das posições dos indivíduos infectados da vizinhança 1 e 2 do indivíduo com ID = x for igual a 0 ou não atinge a probabilidade necessária, o seu estado mantém-se “susceptível” e é criado o próximo evento de avaliação para o indivíduo com ID = x com o tempo de evento igual ao tempo atual mais o tempo médio de avaliação “S” aleatório usando distribuição exponencial e adicionado à CAP.

Criação do ficheiro resultados.txt

É aberto um ficheiro (resultados.txt) onde irá ser adicionada a lista com 4 listas com as coordenadas dos indivíduos susceptíveis, expostos, infectados e recuperados, respectivamente, correspondendo a evolução da grelha ao longo do tempo.

É escrito na primeira linha a dimensão N da grelha e na segunda a lista de obstáculos.

No início de cada ciclo do simulador é definida uma variável “c” booleana com o valor de “False”, e no final de cada evento, se este ocorre (havendo então alteração do aspeto da grelha) esta passa a “True”. Aquando do final do ciclo, se houver de facto alguma alteração, é então escrito no ficheiro uma linha com a lista das 4 listas das posições dos estados. Caso não haja alteração (o evento não se realiza), não escreve, diminuindo assim substancialmente o tamanho do ficheiro.

No final da simulação fecha-se o ficheiro.

Criação do gráfico com a evolução do número de infectados ao longo do tempo

É criada inicialmente aquando da introdução dos indivíduos iniciais na grelha duas listas (“grafx” e “grafy”), sendo que a primeira representa o tempo e a segunda o número de indivíduos “infectados” na grelha.

É também definida uma variável “nInf” que atualiza o número de indivíduos “infectados” na grelha. Esta começa por ser igual a P_i , e como só existe diminuição do número de indivíduos infectados ($nInf = nInf - 1$) aquando da morte ou da avaliação (Inf -> Rec) de um indivíduo infectado e aumento do número de indivíduos infectados ($nInf = nInf + 1$) aquando da avaliação de (Exp -> Inf) de um indivíduo exposto, só se registam (adicionando a “grafx” o tempo atual e a “grafy” o número atualizado de indivíduos infectados na grelha) nestas situações.

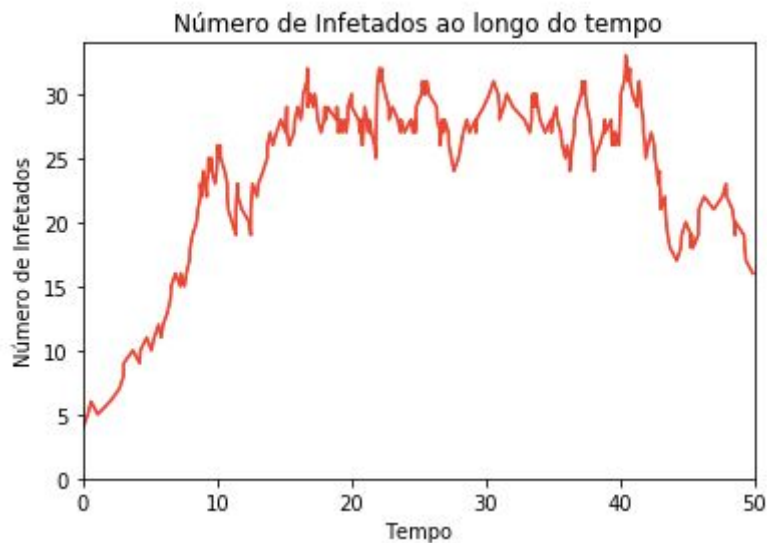
No final da simulação, cria-se o gráfico com os valores de x cada entrada de “grafx” e os valores de y cada entrada de “grafy”, define-se a dimensão dos eixos e adiciona-se legenda ao gráfico.

Resultados

→ **Nota:** conseguimos exportar o ficheiro mp4, contudo a qualidade é bastante reduzida, pelo que anexámos uma pasta para cada cenário que contém a animação da simulação cujos gráficos são apresentados abaixo.

→ Os cenários apresentados abaixo foram obtidos usando o simulador com os parâmetros sugeridos no enunciado.

Cenário 'a' (ver animação1)



Cenário 'b' (ver animação2) (simulação com percentagem 70%)

