Primer Proyecto de Simulación

Nombre y apellidos: ***Lazaro Jesus Suarez Nuñez***

Grupo: ***C – 412***

Problema: ***Poblado en Evolución***

Principales Ideas para la solución del problema

Se deseaba realizar una simulación del de la evolución de una población de una determinada región por 100 años, contando inicialmente con 50 hombres y 50 mujeres.

La edad de los pobladores iniciales se generó mediante una variable aleatoria discreta con valores de 0 a 100 con igual probabilidad, la cual fue generada por el método de transformada inversa.

Para lograr un cierto grado de realismo en el proceso de embarazo de las féminas, las variables del tiempo fueros interpretadas como año y mes, y fueron transitadas mes a mes hasta que se llega al año 100 (la orientación del problema planteaba un periodo de simulación de 100 años).

La fecha (mes y año) en que un poblador fallece fue fijada en el momento en el que nace. Para generar dicha fecha se utilizó la tabla de probabilidades brindada en la orientación para fijar el intervalo de edad del fallecimiento, y luego fueron generadas dos v.a. discretas para la edad y mes.

El deseo de querer pareja para cada poblador sin pareja fue determinado cada mes según la tabla de probabilidades brindada en la orientación, así como el emparejamiento y el embarazo de las mujeres.

La cantidad de hijos deseados fue calculada en el nacimiento de cada poblador generando v.a. uniformes en (0, 1) para cada número de hijos, y de las que dieron positivo se cogió la menor cantidad de hijos deseados posibles (se podía haber hecho de otra forma pues depende de la interpretación del lector).

El tiempo de luto que los pobladores tomaban después de la muerte de su pareja o una ruptura fue calculado generando una v.a. exponencial mediante el método de la transformada inversa, tomando los valores λ según la tabla brindada en la orientación del problema.

Los datos brindados en la orientación del problema para los embarazos múltiples fueron sutilmente cambiados para que quedara una configuración de probabilidades tal que su suma fuera 1, para así poder trabajar esta característica

como una v.a. discreta y poder generarla por el método de la transformada de la inversa.

Se implementó también una pequeña interfaz en consola a base de comandos para poder analizar parámetros de una simulación mes por mes. Se pueden ver las personas nacidas, fallecidas y el total de hombres y mujeres con vida cada mes, así como muchas de sus características como edad, sexo, entre otras.

Modelo de simulación de eventos discretos desarrollado

Variables de tiempo: y, m (año y mes respectivamente)

Variables de estado del sistema: $M_{y,m}$, $F_{y,m}$, $D_{y,m}$, $B_{y,m}$ (listas de la información de los hombres, mujeres, fallecidos y nacidos en el mes m del año y)

Pseudocódigo:

// iniciación de las variables

y = 1, m = 1

for i -> 50:

idMale = GenerateID() //genera un identificador único

idFemale = GenerateID()

ageMale = GenerateVal(0, 100) // genera un valor aleatorio entre 0 y 100 (v.a. discreta)

ageFemale = GenerateVal(0, 100)

birthdayMale = GenerateVal(1, 12) // para generar el mes cuando cumple años

birthdayFemale = GenerateVal(1, 12)

childrenMale = GenerateDesiredChildren() // genera los hijos deseados

childrenFemale = GenerateDesiredChildren()

deceaseMale = GenerateDeceaseDate(y, m, ageMale, 'varon') // genera fecha de muerte después de la // actual

deceaseFemale = GenerateDeceaseDate((y, m, ageFemale, 'hembra')

male = new Persona(idMale, 'varon', ageMale, birthdayMale, childrenMale, deceaseMale)

female = new Persona(idFemale, 'hembra', ageFemale, birthdayFemale, childrenFemale, deceaseFemale)

$M_{1,1}$*.add(male)*

$F_{1,1}$*.add(female)*

while y < 100:

subjects = $M_{y,m}$ *+* $F_{y,m}$

foreach Person p un subjects:

p.age += 1 // aumenta la edad si la persona p cumple años en el mes m

p.DecreaseLoneTime() // disminuye el tiempo de luto de la persona p si tiene

p.matingDesire = SetMatingDesire() // pone en verdadero o falso el deseo de p de tener pareja

// se separan estos bucles porque por ejemplo establecer pareja depende de la edad y puede que se
// trate de establecer pareja con alguien por el que no se ha iterado y no se ha incrementado su edad

foreach Person p un subjects:

if p.gender == 'hembra' and p.pregnant == true:

p.pregnancyTime -= 1 // disminuye el tiempo de embarazo

if p.pregnancyTime == 0:

for i -> p.numberOfBabies:

idBaby = GenerateID()

genderBaby = GenerateGender() // genera sexo del bebé

birthdayBaby = m

childrenBaby = GenerateDesiredChildren()

if genderBaby = 'varon':

deceaseBaby = GenerateDeceaseDate((y*,* m*, 0, 'varon')*

baby = new Person(idBaby, 'varon', 0, birthdayBaby, childrenBaby, deceaseBaby)

nd = NextDate() // devuelve par año-mes correspondiente a la // fecha siguiente a la actual

$B_{y,m}$*.add(baby)*

If IsDeceaseDate(baby): // si muere en la misma fecha en que // nace

Die(baby) // incluye poner a baby en $D_{y,m}$

else:

M_{nd}*.add(baby)*

else:

deceaseBaby = GenerateDeceaseDate((y*,* m*, 0, 'hembra')*

baby = new Person(idBaby, 'hembra', 0, birthdayBaby, childrenBaby, deceaseBaby)

nd = NextDate() // devuelve par año-mes correspondiente a la // fecha siguiente a la actual

$B_{y,m}$*.add(baby)*

If IsDeceaseDate(baby): // si muere en la misma fecha en que // nace

Die(baby) // incluye poner a baby en $D_{y,m}$

else:

F_{nd}*.add(baby)*

p.childrenCount += p.numberOfBabies

p.babyFather.childrenCount += p.numberOfBabies

if p.mate != none:

if BreakUp(): // metodo que genera verdadero o falso a romper con su pareja

mate = p.mate

mate.mate = none

mate.loneTime = GenerateLoneTime(p.mate.age) // genera un tiempo de luto

p.mate = none

p.loneTime = GenerateLoneTime(p.age)

if p.gender = 'hembra' and p.mate != none and p.pregnant = false:

mate = p.mate

if p.childrenCount < p.desiredChildren and mate.childrenCount< mate.desiredChildren:

```
                    if GetPregnant(p.age): // devuelve verdadero o falso segun la edad
                        p.pregnancyTime = 9
                        p.babyFather = p.mate
                        p.numberOfBabies = GenerateNumberOfBabies() // genera cantidad
                                                                    // de bebes del
                                                                    // embarazo
        // para emparejar a p con alguien
        if p.matingDesire:
            availableMates = FindAvailableMates(p.gender) // devuelve las personas del sexo
                                                          // opuesto con deseos de emparejarse
            foreach Person k in availableMates:
                if MakeMating(Abs(p.age – k.age)):
                    p.matingDesire = false
                    k.matingDesire = false
                    p.mate = k
                    k.mate = p
                    break

        if IsDeceaseDate(p): // si y y m coincide con la fecha de muerte de p
            Die(p)
        else:
            // si no muere p lo pasamos a la lista de las personas de la proxima fecha
            if p.gender == 'varon':
                nd = NextDate()
                M_nd.add(p)
            else:
                nd = NextDate()
                F_nd.add(p)

// avance a la siguiente fecha
y, m = NextDate()
```

En la implementación que se encuentra junto a este archivo se encuentran todos los métodos de generación de variables aleatorias y valores probabilísticos dependientes de los datos sacados de la orden del problema.

También se adjunta un comando en la interface que sirve para examinar una cantidad x de simulaciones mediante el método de Monte Carlo para obtener el valor esperado de la cantidad de personas, hembras y varones en el año 100, así también como el valor esperado del número de personas nacidas y fallecidas hasta este año (esta implementación es bastante humilde y con más de 100 simulaciones demorará más de 30 segundos).

Resultados

Observando los resultados de las simulaciones se llegó a la conclusión de la población tiende a crecer con el tiempo, pues haciendo un análisis con el método de Monte Carlo para 10000 simulaciones se llegó al resultado de que el valor esperado del número de pobladores en el año 100 es de 156 aproximadamente. También se pudieron notar otros resultados como que la tasa de mortalidad de las mujeres es ligeramente mayor que la de los hombres. Otra forma de llegar a este resultado del crecimiento poblacional es comparando el valor esperado de la cantidad de nacidos hasta el año 100 con el del número de fallecidos hasta este año, pues la primera cifra es mayor que la segunda en aproximadamente 56 unidades, lo que afirma el anterior planteamiento del valor esperado de la población de 156 ya que se cuenta inicialmente con 100 pobladores.

Repositorio

El repositorio que contiene la implementación de este simulador en C#, además de este informe se encuentran en https://github.com/LazardStrife/simulation-1.