Teoría de grafos 12 de noviembre de 2019

Entrega 3 (Final): Proyecto Teoría de Grafos

Profesor: *Daniel Bojacá* Autor: Oscar Velasco Chiquillo y Jhan Carlos Celi Maldonado

Nota: *Para implementaciones se trabajará en Python3.*

3.1. Introducción

En el presente documento se pretende informar al lector a cerca del **proyecto** planteado para la materia de **Teoría de Grafos**, esperando en lo posibble aplicar los conceptos vistos en dicha asignatura, para el análisis o solución de problemas clásicos, o planteados sobre la cotidianidad, en los cuales los **Grafos** nos faciliten dicha tarea.

3.2. Descripción del Problema

Estudio e implementación de un modelo matemático y gráfico (teniendo en cuenta la Teoría de Grafos) para la representación y el análisis de la interacción entre 2 especies (Depredador-Presa) en un determinado espacio y tiempo basados en el modelo de ecuaciones diferenciales Lotka-Volterra.

3.3. Objetivos

1. **Objetivo General:** Implementar modelo gráfico para la representación y análisis de un ecosistema formado por 2 poblaciones (Depredador-Presa) visto como un grafo.

2. **Objetivos Específicos:**

   a) Solucionar numéricamente el modelo que describa el comportamiento de poblaciones en un espacio y tiempo (Lotka-Volterra).

   b) Obtener la información base a partir del Modelo Lotka-Volterra sobre el comportamiento de las 2 poblaciones (Depredador-Presa).

   c) Implementar la representación gráfica del modelo (Grafo Dinámico) a partir de la información recolectada en el item anterior.

   d) Analizar aspectos y comportamientos importantes basados en el grafo generado en el item anterior.

   e) Aplicar en el análisis del item anterior conceptos de la Teoría de Grafos que nos faciliten generar conclusiones fuertes al respecto.

   f) En caso de que el tiempo sea un factor que nos impida terminar todo lo propuesto, lo ideal es a futuro terminar por completo este proyecto.

3.4. Marco Teórico

3.4.1. Modelo Lotka-Volterra

Las ecuaciones del modelo **Lotka-Volterra**, también conocidas como ecuaciones **depredador-presa**, son un par de ecuaciones diferenciales no lineales de primer orden, frecuentemente utilizadas para describir matemáticamente la interación entre dos especies, una como depredador y la otra como presa. Ambas poblaciónes cambian a través del tiempo, según las siguientes ecuaciones:

$$\frac{dx}{dt} = \alpha x - \beta xy \tag{3.1}$$

$$\frac{dy}{dt} = \delta xy - \gamma y \tag{3.2}$$

donde:

1. x es el número de presas.
2. y es el número de depredadores.
3. $\frac{dx}{dt}$ y $\frac{dy}{dt}$ representan las tasas de crecimiento de las dos poblaciones.
4. t representa el tiempo.
5. $\alpha, \beta, \gamma, \delta$ son parametros enteros positivos que describen la interación entre dos especies.

3.4.2. Resultados del modelo (Lotka-Volterra Discreto)

Para modelar el comportamiento de las ecuaciones de Lotka-Volterra, necesitamos un Método Numérico que aproxime x y y a lo largo del tiempo t. Para ello usamos el método **odeint** de la librería Scipy de Python, el cual bajo unas condiciones iniciales x_0 y y_0 nos retorna el comportamiento de las presas y depredadores en un intervalo de tiempo $[t_0, t_f]$. Por ejemplo observe las siguientes gráficas con $x_0 = 10$ y $y_0 = 2$:

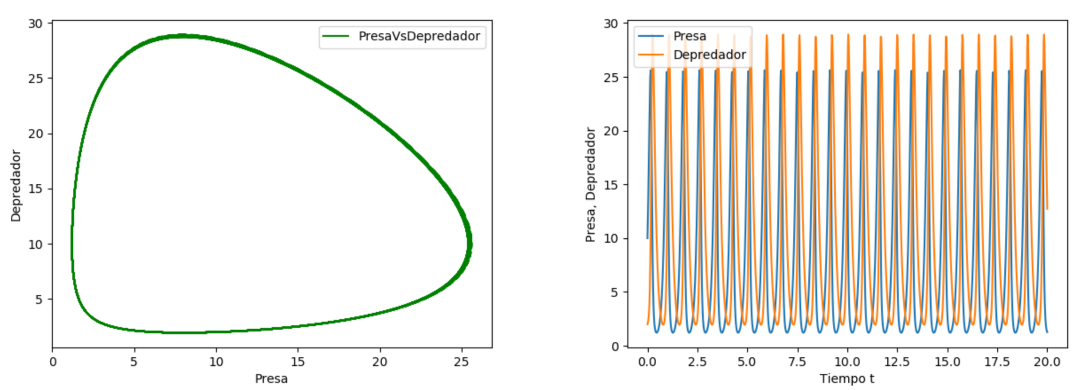

Figura 3.1: Resultados del modelo con las condiciones iniciales anteriores.

Observe que la gráfica de la izquierda representa la relación que describe el modelo entre los depredadores y las presas, mientras que la gráfica de la derecha representa como cambia el número de individuos de cada población en un intervalo de tiempo.

Basados en dichos resultados se implementará la representación gráfica como un grafo sobre una región (ecosistema) cuyo conjunto de vértices representará a los individuos de ambas poblaciones, **rojo para depredador** y **azul para presa**, adicionalmente el conjunto de aristas representará las posibles interacciones (Depredación/Competencia), y cada una tendrá un peso que representará la distancia entre dos vértices, es decir, la distancia entre dos individuos. Basados en los resultados del modelo se espera que el grafo se expanda y contraiga en el intervalo del tiempo evaluado, es decir, como puede verse en las gráficas que ambas poblaciones aumentan y disminuyen periódicamente.

3.4.3. Representación del problema en grafos

Observe a continuación uno de los posibles grafos obtenidos dadas las condiciones iniciales $x_0 = 10$ y $y_0 = 2$, además del grafo que representa las condiciones finales $x_f = 1$ y $y_f = 13$ del inciso anterior:

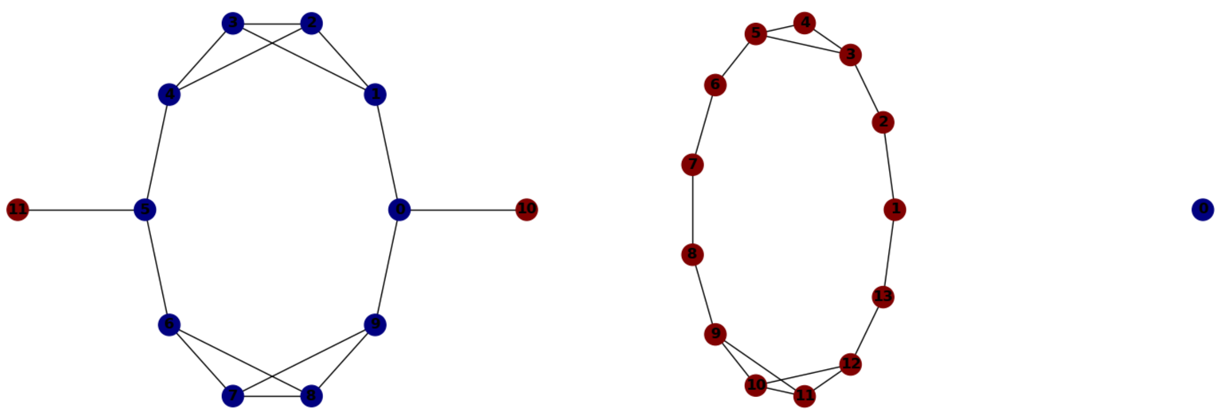

Figura 3.2: Resultados del modelo con las condiciones iniciales (Izq) y finales anteriores (Der) .

Recuerde que la representación del grafo considera que el conjunto de vertices $V(G)$ esta conformado por la unión entre los conjuntos de depredadores Y y las presas X. Donde el color de cada nodo esta definido por:

$$v \in V(G) = \begin{cases} Azul & \text{si } v \in X \\ Rojo & \text{si } v \in Y \end{cases}$$

Además el conjunto de las aristas $E(G)$ representará las posibles interacciones entre individuos. Habrá una arista entre dos nodos *sii* se encuentran a una distacia menor o igual a r. Finalmente el peso de cada arista representará la distancia entre dos nodos en caso de que exista una arista entre los mismos.

3.5. Solución Propuesta

Teniendo ya bien definida la estructura general de un grafo que representa un ecosistema dado. Se procede a plantear la solución del problema.
Se planea elaborar una simulación dirigida por eventos discretos, cuyo conjunto de eventos $\phi(G)$ estará conformado por todas las posibles interacciones que puedan darse en el grafo, es decir, un evento por cada arista del grafo, además de considerar el evento de vértices (individuos) aislados. Como las simulaciones dirigidas por eventos requieren de datos discretos para funcionar, se propone discretizar el modelo Lotka-Volterra. Teniendo acceso así a los valores de x y de y en cualquier tiempo t.

Ahora que tenemos definido $\phi(G)$ la idea es tomar el modelo discreto de Lotka-Volterra e ir graficando cada cambio poblacional que de este resulte, conociendo la cantidad de individuos de cada especie en un tiempo actual t_i y un tiempo posterior t_f. Definamos Δ_x y Δ_y como el cambio en la población de presas y depredadores en el intervalo $[t_i, t_{i+1}]$ respectivamente.

Note que Δ_x y Δ_y almacenan la cantidad de individuos que bien nacen o fallecen de cada población. Teniendo los cambios poblacionales procederemos a modificar el grafo. Para ello elegiremos al azar un evento de $\phi(G)$ que cumpla las condiciones dadas, es decir, aquella donde la cantidad de individuos de x y de y coincida con la del modelo.
Se espera que a partir de multiples simulaciones con distintos valores iniciales, se pueda analizar el comportamiento de los grafos resultantes con base a lo aprendido en clase.

3.6. Librerias utilizadas para la Implementación

1. **numpy:** Estructuras de datos y funciones matemáticas.
2. **scipy:** Metodos de integración numéricos.
3. **matplotlib:** Interfaz gráfica.
4. **networkx:** Representación de los grafos.
5. **random:** Generar números aleatorios.
6. **pylab:** Interfaz gráfica.

3.7. Implementación de la Solución

Link repositorio en GitHub (Códigos Usados): Graphs Proyect

Descripción del Repositorio (Carpetas):

1. **modelo:** Contiene la Solución Numérica del Modelo Lotka Volterra, del cuál extraemos la información para nuestro modelo gráfico.
2. **main:** Contiene el algoritmo planteado para la Solución Propuesta (Parcialmente, ya que la idea es complementar lo mayor posible el proyecto).
3. **tests:** Contiene pruebas parciales de funciones diseñadas y librerías importadas.

3.8. Resultados Obtenidos

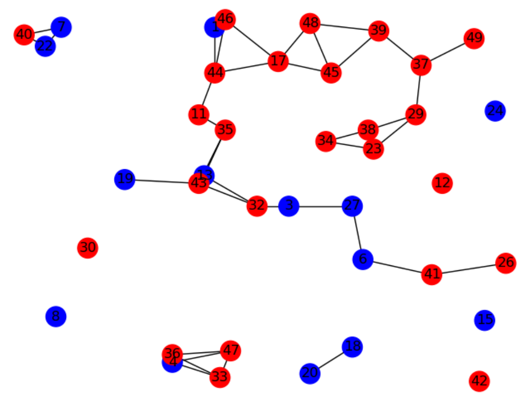

Figura 3.3: Resultados Grafo Dinámico Ponderado

Observe en la anterior figura el grafo ponderado obtenido en un tiempo determinado, es decir que a medida que avanza el tiempo tenemos un grafo distinto.

3.9. Objetivos a Futuro

1. Considerar más eventos en la implementación, teniendo en cuenta que algunos eventos pueden ser más importantes que otros, de modo que el modelo se adapte de una mejor forma al comportamiento de las poblaciones puestas en análisis.

2. Generar movimiento de los nodos del grafo, teniendo en cuenta que las poblaciones se mueven por instinto sobre el espacio en el que se encuentran.

3. Incorporar más poblaciones y factores o parámetros al modelo, es decir generalizarlo de modo que tenga en cuenta la mayor información posible.

4. Programar el modelo como una aplicación de Machine Learning para generar predicciones a cerca del comportamiento de poblaciones sobre un espacio en un lapso de tiempo, así el análisis que se haga puede ser preventivo.

3.10. Conclusiones

- En nuestro modelo nos damos cuenta que las poblaciones se comportan de forma periódica.
- A medida que consideramos más eventos y/o condidiones en nuestro modelo, tiende a ser más leal al comportamiento de las poblaciones en un espacio-tiempo.
- A medida que la magnitud del radio de alcance r crece, el ecosistema tiende a representar un espacio más pequeño, lo que implica que el grafo resultante tiende a ser completo.

Referencias

[1] *Ecuaciones Lotka-Volterra, Alfred J. Lotka & Vito Volterra, Wikipedia https://es.wikipedia.org/wiki/Ecuaciones˙Lotka–Volterra, 02 de Agosto de 2019.*

[2] *Ecuaciones Lotka-Volterra - Modelo Presa Depredador Python, https://pybonacci.org/2015/01/05/ecuaciones-de-lotka-volterra-modelo-presa-depredador/, Alex Saez, 05 de Enero de 2015*

[3] *Libreria Python NetworkX 2.4, https://networkx.github.io/documentation/stable/*