

1.- Práctica 6: sistema multiagente

Se realiza una simulación de propagación de una epidemia en un área delimitada por un número de agentes, implementado un sistema de multiagente con enfoque epidemiológico. Los agentes podrán estar en uno de tres estados: susceptibles, infectados o recuperados, tomando en cuenta las variables de número de agentes y probabilidad de infección

2.- Descripción de la tarea

- Identificar las partes de importancia del código que nos sirvan para poder paralelizar sin afectar los resultados que presenta el código original, a su vez estudiar el efecto de probabilidad p_i en el porcentaje máximo de infectados durante las simulaciones.
- El primer reto es iniciar vacunando con probabilidad p_v a los agentes al momento de crearlos de tal forma que están desde el inicio en el estado de inmunidad así no podrán contagiarse ni propagar la infección, estudiar el efecto estadístico de valor p_v en el porcentaje máximo de infectados durante la simulación.

3.- Tarea

Lo primero que se realizó es la forma de paralelizar y lo que se encontró como de mayor ayuda son los datos de los agentes infectados es la parte del código donde se calcula la distancia euclidiana con todos los demás agentes y determinar si se encuentran dentro de una zona de contagio, como el cálculo de la distancia para comparar a cuantos puede contagiar el agente infectado en cuestión es independiente para cada agente, esta parte se puede paralelizar ya que el cálculo de la distancia se puede realizar para todos los agentes al mismo tiempo aplicando la paralelización. Para realizar la paralelización se modificó la parte de contagios del código original y se convirtió en la función contagiados, en esta función lo que se hizo fue el mismo cálculo que se realiza en el código original y al final la función devuelve los índices de los nuevos agentes contagiados. Lo que se agregó fue guardar toda la instrucción de infectados en la función *contagiados* para así en la siguiente secuencia tener la persona infectada y así sucesivamente tener guardados todos los infectados.

Para comprobar que la paralelización fue exitosa realizamos una comparación con 100 agentes y probabilidad de infección del 0.05, con 5 repeticiones como se muestra en la figura 1. Se pudo concluir que el tiempo se reduce 20 segundos al realizar la ejecución con la herramienta de paralelización haciendo más eficiente el tiempo de trabajo.

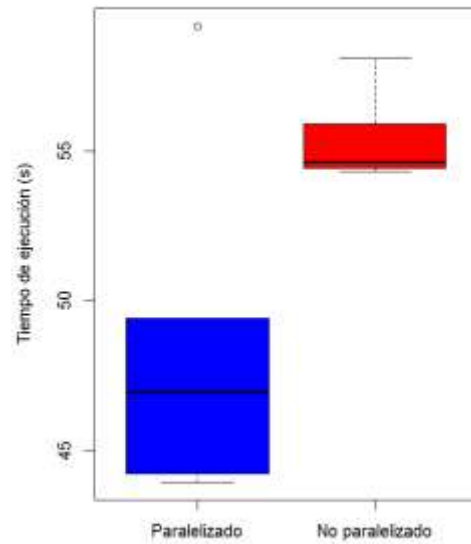


Figura.- 1 Comparación de tiempo de ejecución entre el código paralelizado (azul) y el código original no paralelizado (rojo)

Teniendo el código paralelizado se muestran las siguientes dos imágenes donde se observó el inicio de los 100 agentes respecto a su estado de inicial hasta llegar a un estado de infectado o susceptible.

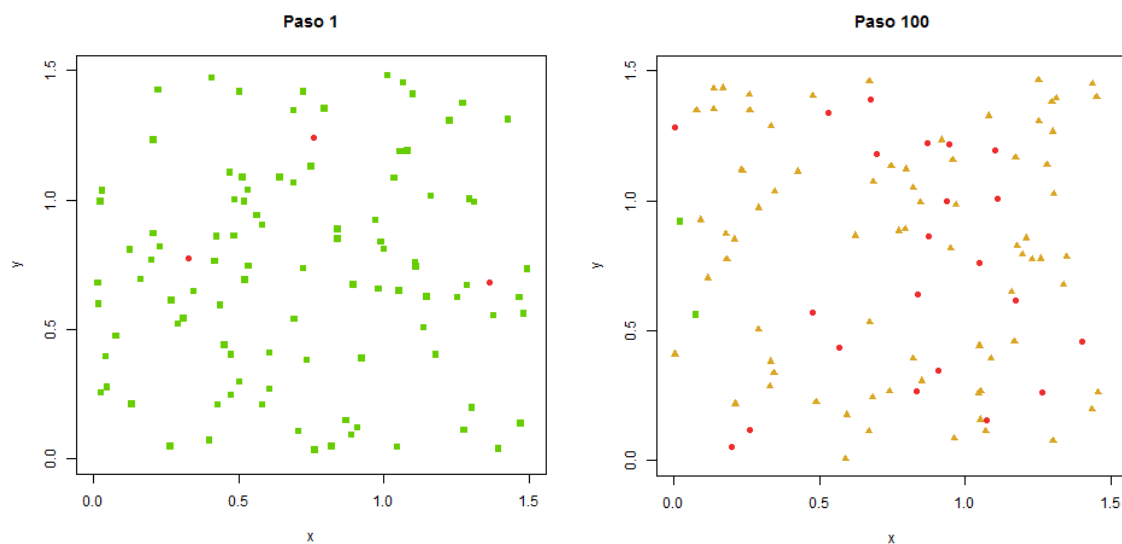


Figura.-2 Imagen inicial y final respectivamente del código original de infectados o susceptible

Teniendo en cuenta los posibles estados que puede presentar nuestros agentes se pasó a realizar la variación de probabilidad pi como se muestra en la figura 3.

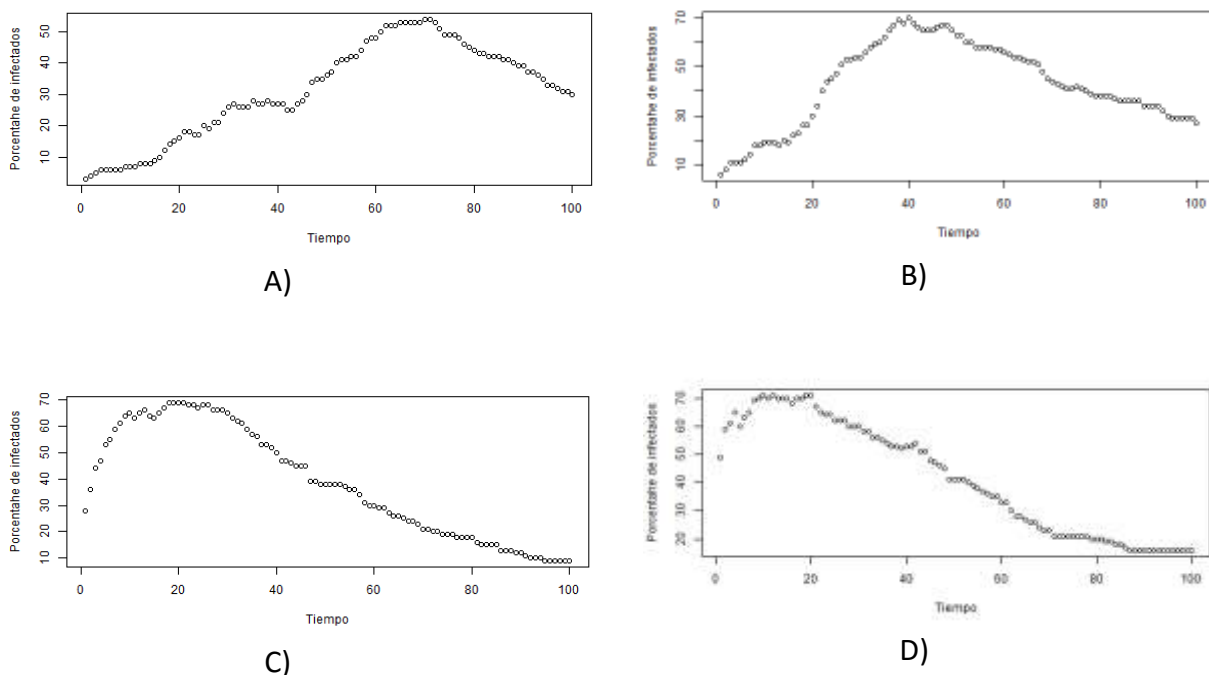
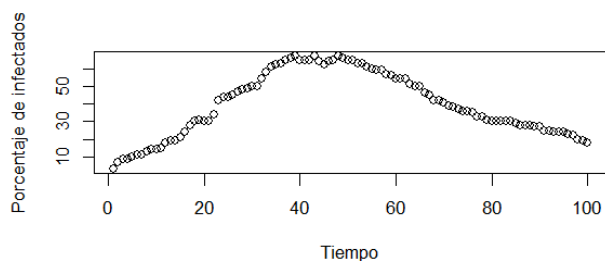


Figura.-3 Grafico de porcentajes de infectados contra tiempo a) $pi = 0.02$, b) $pi = 0.1$, c) $pi = 0.3$, d) $pi = 0.5$.

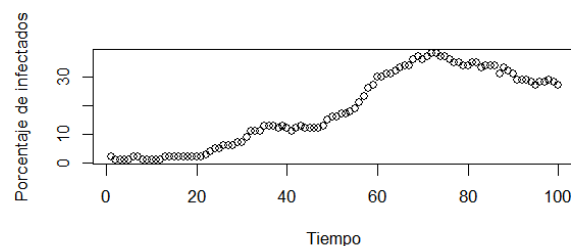
Los resultados que nos muestran en la figura 3, es que al tener un porcentaje mayor de infectados desde un inicio las gráficas tienden a desplazarse hacia la izquierda denotando al estar en contacto con mayor grupo de infectados se llega en un menor tiempo al máximo de infectados como se menciona en los incisos C) y D) de dicha figura, al mismo tiempo al tener mayor número de infectados los agentes pasan a ser de infectados ha recuperado viendo esa drástica caída a la par del tiempo, eso se atribuye a que como desde un inicio se interactúan con un número elevado de infectados todos pasan a cambiar a recuperados por las múltiples relaciones de proximidad de cada uno de los agentes.

4.- Reto 1

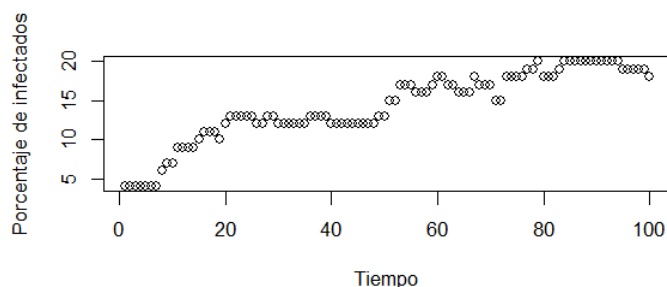
En este reto se hicieron las modificaciones en la parte de la creación agentes para que antes de comenzar a infectar existiera la probabilidad de que el inicio de la simulación se vacunara a los agentes con una probabilidad p_v y esta se fue variando para observar el efecto en el máximo de infectados.



A)



B)



C)

Figura 4. Gráficos de porcentajes de infectados contra tiempo a) $p_v = 0.02$, b) $p_v = 0.3$, c) $p_v = 0.5$.

El aumento en p_v se pueden observar en la figura 3. Se observa que al aumentar la probabilidad de agentes vacunados al inicio los número máximo de infectados presentan una tendencia de desplazamiento hacia la derecha del grafico esto quiere decir que al tener mayor cantidad de vacunados en un inicio el punto máximo de infectados tarda más pasos en alcanzarse.

5.- Conclusiones

Se demostró que el código es factible su paralización dando como resultado una disminución del tiempo de ejecución en casi de la mitad del tiempo en las mayorías de los casos.

Se pudo demostrar que la probabilidad de inicial de infectados es proporcional al número máximo de infección con respecto a un tiempo más corto, es decir, entre más infectados más rápido se llega al máximo de infectados.

A otro punto importante fue al tener de un inicio un mayor número de agentes inmunes se toma más tiempo lograr llegar al máximo punto de infección.

6.- Especificaciones de equipo

Modelo del sistema Inspiron 5420, fabricada por Dell Inc, procesador Intel(R) Core(TM) i5-3210M CPU @2.50Hz 2.50 Hz, memoria instalada (RAM) 8 GB (7.86 GB utilizable), tipo de sistema operativo de 64 bits procesador x64, edición de Windows 10 Pro.