

Comprobación de los efectos de los super esparcidores en simulación de epidemia con sistema multiagente

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Victor A. Oviedo

Resumen

En este artículo se pretende observar y comprobar los efectos de los llamados super esparcidores en la simulación de epidemia con sistema multiagente, con características del virus SARS-CoV-2. La totalidad de la simulaciones son realizadas en el software libre Python [2].

Palabras clave: Epidemia, sistema multiagente, super esparcidor.

1. Introducción

Este trabajo surge como continuación a la actividad [5], en el cual es una introducción a los sistemas multiagente tomando como tópico las epidemias. Se tomó la decisión de simular los efectos de los super esparcidores después de leer el artículo [1], en donde el autor habla sobre la decisión de: ¿A quién vacunar?, ya que la epidemia puede ser controlada con mayor eficiencia dependiendo de a quien se le aplique la vacuna. Por lo tanto, hablamos de detener el contagio con el menor número de vacunas.

2. Antecedentes

Para poder entender este tema con claridad es necesario conocer el significado de las palabras clave, ya que son en ellas en las que basamos el tema a generar.

Una pandemia se define como “una epidemia que ocurre en todo el mundo, o en un área muy amplia, que cruza fronteras internacionales y generalmente afectando a un gran número de personas [6].

Un sistema multiagente es una extensión de la tecnología de agentes en la que un grupo de agentes autónomos débilmente conectados actúa en un entorno para lograr un objetivo común. Esto se hace cooperando o compitiendo, compartiendo o no el conocimiento entre ellos [7].

La palabra es un término genérico para un individuo inusualmente contagioso que ha sido infectado con una enfermedad. En el contexto del coronavirus, los científicos no han reducido la cantidad de infecciones que alguien debe causar para calificar como super esparcidor, pero en general supera con creces las dos o tres personas que los investigadores inicialmente estimaron que el paciente infectado promedio podría infectar [8].

3. Trabajos relacionados

En el artículo [3] se tiene la investigación de la dinámica de transmisión del virus MERS-CoV para observar el comportamiento de los super esparcidores, ya que mencionan que aproximadamente el 20 % de los infectados pueden infectar al 80 % de la siguiente generación de infecciones. Sin embargo este artículo no simula ninguna epidemia, en este artículo se analiza la información para

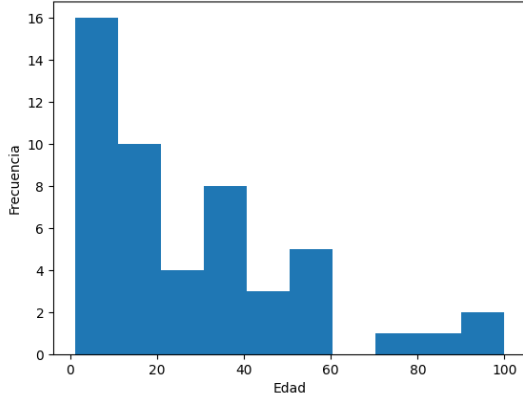


Figura 1: Ejemplo de la asignación de edades con distribución exponencial.

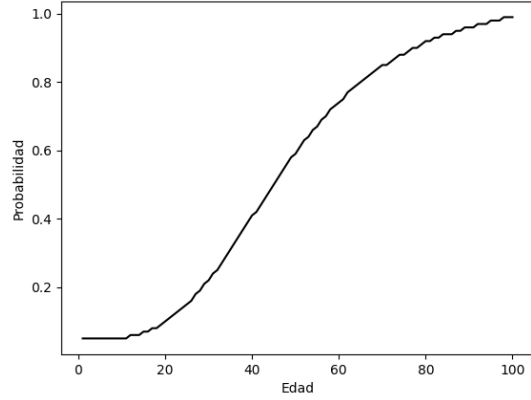


Figura 2: Función de la probabilidad de muerte (pm).

entender la dinámica de la transmisión en un caso real. En cuanto al artículo con mayor similitud en cuanto a idea se refiere, es el artículo [4], ya que en su resumen habla acerca de diferentes modelos de transmisión del virus SARS-CoV-2 como se tratará mas adelante en este artículo. Sin embargo no a sido posible observar el articulo completo.

4. Solución propuesta

Como ya a sido mencionado con anterioridad, este trabajo se basa en la actividad [5] que a su vez se basa en el código de Schaeffer [9]. Para un mejor entendimiento del código base se recomienda leer primero la actividad [5].

Dado que el código base está basado en el Modelo SIR y este no contempla el estado de Muerte este será agregado intentando darle características del virus SARS-CoV-2. Una de estas nuevas características es el tiempo de infección (t_i), el cual es el tiempo en el que los agentes estarán con estado infectado. Una vez pasado el tiempo (t_i) el agente podrá tener el estado Recuperado o Muerto, esto dependerá de la probabilidad de muerte (pm) 4 que a su vez dependerá de la edad del agente evaluando. Esta edad es generada con una distribución exponencial con un rango entre 1 año y 100 años 4.

Para demostrar los efectos de los super esparcidos se cuenta con el espacio ya conocido en la actividad [5] en donde es visible cada uno de los agentes y su movimiento, pero esta vez estará dividido en 4 cuadrantes diferentes contando cada uno de ellos con el mismo número de agentes. La división de estos cuadrantes representa areas aisladas entre sí impidiendo el contagio entre agentes de diferente cuadrante, esto puede verse representado en la Figura 4. Una vez teniendo esta primera instancia en donde tenemos un solo infectado en uno de los cuadrantes podemos correr la simulación para observar el comportamiento que tiene el aislamiento de los agentes en los cuadrantes, esto puede verse representado en la Figura 4. Una vez observada la Figura 4 podemos determinar que el efecto causado por el aislamiento es; impedir la propagación de la infección entre los agentes de diferentes cuadrantes. Ahora se realizará la misma simulación, sin embargo serán agregados los super esparcidos los cuales tendrán la característica de poder moverse entre cuadrantes. El inicio de esta simulación podrá observarse en la Figura 4, mientras el final de la simulación podrá observarse en la Figura 4.

Una vez terminada la simulación con los super esparcidos podemos observar como el aislamiento a sido corrompido y todos los cuadrantes han sido infectados. Para tener una mayor claridad del transcurso de ambas simulaciones se tienen la Figura 4 y 4.

Una vez comprobado los efectos que tienen los super

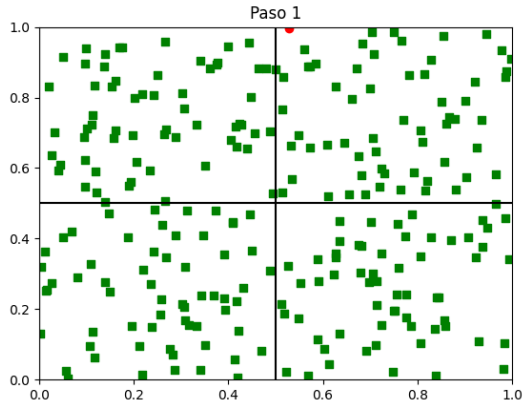


Figura 3: Simulación con 4 cuadrantes sin super esparcidos, paso 1.

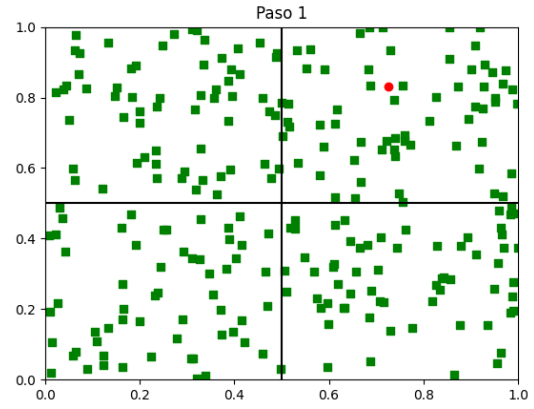


Figura 5: Simulación con 4 cuadrantes con super esparcidos, paso 1.

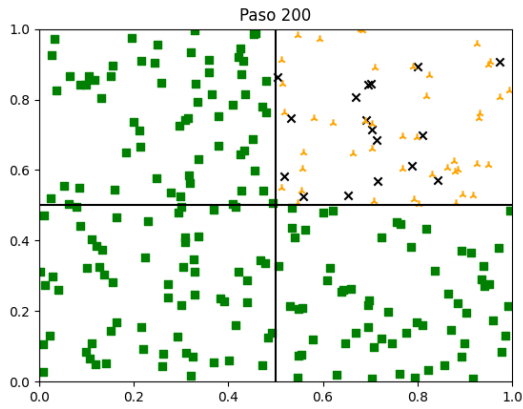


Figura 4: Final de la simulación con 4 cuadrantes sin super esparcidos, paso 200.

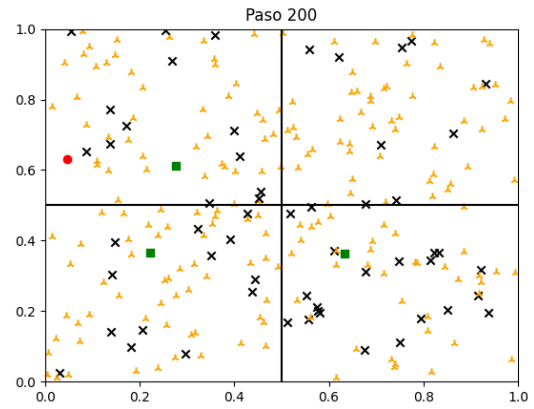


Figura 6: Final de la simulación con 4 cuadrantes con super-esparcidos, paso 200.

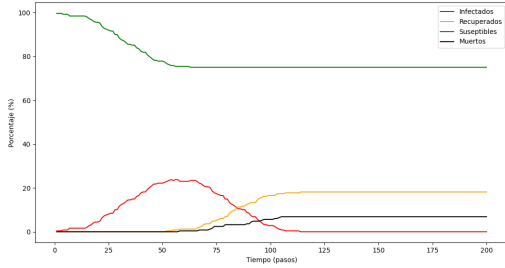


Figura 7: Estadísticas de la simulación con 4 cuadrantes sin super esparcidos.

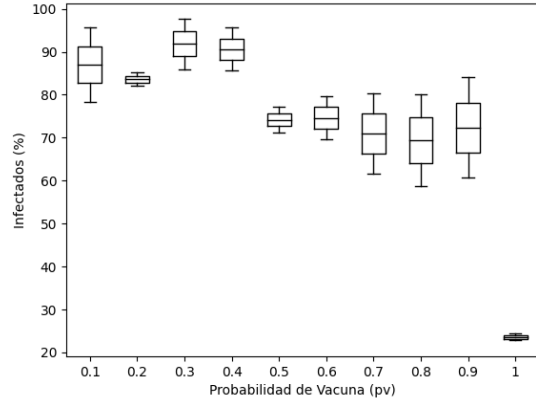


Figura 9: Resultados de la simulación con diferentes valores de (pv) a 3 repeticiones.

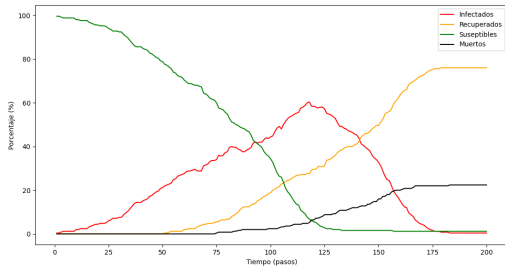


Figura 8: Estadísticas de la simulación con 4 cuadrantes con super esparcidos.

esparcidos en la simulación, el objetivo es encontrar el porcentaje mínimo para detener en cierta medida la epidemia. Por lo tanto se genera una probabilidad de vacuna (pv) por lo que al inicio de generar la instancia de cada agente, estos podrán desde un inicio contar con el estatus Recuperado sin la necesidad de haber sido contagiado. Esta probabilidad solo será aplicado para los super esparcidos. Para comprobar el efecto de la probabilidad de vacuna (pv) sobre la dinámica de la epidemia se realizaran simulaciones con (pv) diferentes, con un rango desde 0 hasta 1. Los resultados de esta simulación se reflejan el la Figura 4.

Con base en los resultados de la Figura 4 podemos afirmar que desde la probabilidad de vacuna (pv) 0.1 hasta 0.4 la pandemia no a sido controlada ya que se tienen porcentajes de infectados mayores al 80 % lo que significa que todos los cuadrantes fueron infectados. Se tiene el Cuadro 4 para las equivalencias de porcentaje de infectados a cuadrantes infectados. Observando la probabilidad de vacuna 0.5 hasta 0.9 observamos que el porcentaje de infectados es menor, por lo tanto se realizan nuevas simulaciones pero esta vez con rango desde 0.8 hasta 0.85 y esta vez a 10 repeticiones cada (pv).

Cuadro 1: Equivalencia de porcentaje de infectados a cuadrantes infectados.

Infectado (%)	Cuadrantes (#)
20	1
40	2
60	3
80	4

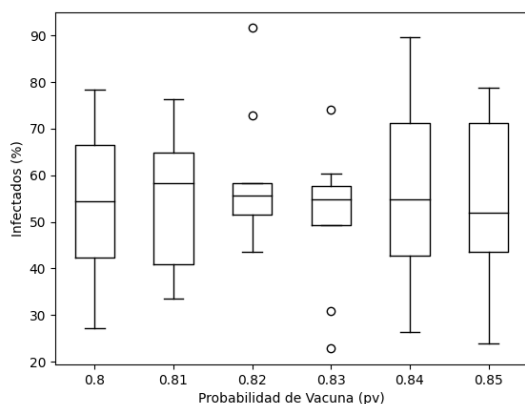


Figura 10: Resultados de la simulación con diferentes valores de (pv) a 10 repeticiones.

5. Conclusiones

Al obtener los resultados de la Figura 4 lo que podemos ver es una muy clara similitud en las simulaciones, sin embargo al comparar los resultados de Figura 4 específicamente los del la probabilidad 0.8, tenemos que su mediana es de aproximadamente el 70 % de infectados mientras que en los datos de Figura 4 su mediana es de aproximadamente el 55 %. Recordando que en una simulación se realizaron 3 repeticiones y en la otra 10 repeticiones, esto nos indica que para una mejor definición de resultados se necesitaría una mayor cantidad de repeticiones, sin embargo al no contar esta simulación con multiprocesamiento el tiempo de simulación es considerable, ya que el tiempo de simulación depende de la cantidad de agentes pero mayor mente de la cantidad de agentes infectados. Por ese motivo para Figura 4 solo se le ha permitido tener 3 repeticiones mientras que Figura 4 se ha permitido tener 10 repeticiones, y aún así el tiempo de la Figura 4 es mayor.

Por otra parte, con los datos obtenidos de la Figura 4, podemos observar que la pandemia solo llega a infectar a aproximadamente el 55 % de la población, por lo tanto podemos decir que vacunando desde aproximadamente el 16 % de la población total es posible controlar la epidemia, sin embargo este 16 % tendría que ser específicamente los super esparcidos, o como se demuestra en la Figura 4 vacunar al 100 % super esparcidos.

Como trabajo futuro sería de suma importancia añadir multiprocesamiento a este código, ya que gran parte de su limitante reside en el tiempo que tarda en realizar completamente una simulación, por lo que al aplicar repeticiones es inviable el tiempo que esta llega a tardar, esto sin tomar en cuenta que la cantidad de agentes en esta simulación es relativamente baja (250 agentes). Por otra parte una mejora considerable al intentar simular el virus SARS-CoV-2 sería agregar un tiempo de inmunidad en el que una vez pasados cierto tiempo los agentes recuperados vuelvan a ser susceptibles y por lo tanto pueden volver a ser contagiados. De ser nuevamente contagiados se podría tener 3 o hasta 4 funciones diferentes de la probabilidad de muerte, en el que dependiendo de la cantidad de veces que haz sido infectado le corresponde una función, tomando en cuenta que la función será cada vez mas severa. De esta

forma podríamos observar verdaderas fluctuaciones en la simulación pudiendo llegar a la extinción.

Referencias

- [1] Christopher Cox. The vulnerable can wait. vaccinate the super-spreaders first. 2020. URL <https://www.wired.com/story/covid-19-vaccine-super-spreaders/>.
- [2] Guido van Rossum. *Python: general-purpose programming language*. Python Software Foundation, 2020. URL <https://www.python.org>.
- [3] Yunhwan Kim, Hohyung Ryu, and Sunmi Lee. Tagent-based modeling for super-spreading events: A case study of mers-cov transmission dynamics in the republic of korea. 2018. URL <https://www.wired.com/story/covid-19-vaccine-super-spreaders/>.
- [4] Egor Malkov. Simulation of coronavirus disease 2019 (covid-19) scenarios with possibility of reinfection. 2019. URL <https://doi.org/10.1016/j.chaos.2020.110296>.
- [5] Victor Oviedo. Práctica 6: sistema multiagente, Oct 2020. <https://github.com/Elitemaster97/Simulacion/blob/master/Tarea6/Tarea6.pdf>.
- [6] Doshi P. The elusive definition of pandemic influenza. 2011. URL <https://who.int/bulletin/volumes/89/7/11-088815/en/>.
- [7] Balaji Parasumanna Gokulan and D. Srinivasan. *An Introduction to Multi-Agent Systems*, volume 310, pages 1–27. 07 2010. ISBN 978-3-642-14434-9.
- [8] Neel V. Patel. What’s a coronavirus superspreader? 2020. URL <https://www.technologyreview.com/2020/06/15/1003576/whats-a-coronavirus-superspreader/>.
- [9] Elisa Schaeffer. Sir.py, Oct 2020. <https://github.com/satuelisa/Simulation/blob/master/MultiAgent/SIR.py>.