

Simulación de la Propagación de una Epidemia utilizando Modelación basada en Agentes

C. D. Herrera Medina,^{*} A. Y. Bautista Sierra,^{**} A. S. Gómez Fonseca^{***}

Departamento de Física, Facultad de Ciencias, Universidad Nacional de Colombia

Julio 1 de 2020

Resumen

Se propone una simulación para la propagación de una epidemia a partir de un modelo basado en agentes. Este marco se implementa utilizando programación orientada a objetos incluyendo dinámica espacial (interacciones de carácter geográfico entre individuos) y temporal, en términos de los parámetros de infección y cura del modelo SIR. Se obtuvieron resultados coherentes con lo propuesto en otros modelos epidemiológicos, por lo que el modelo construido constituye una poderosa herramienta que indica qué medidas pueden ser más eficientes en el control y erradicación de una epidemia.

Introducción

Actualmente estamos en una emergencia sanitaria debido a la epidemia causada por el virus COVID-19. El estudio de la propagación de una enfermedad resulta una prioridad de salud pública para cualquier gobierno, pero en particular, para países tropicales y endémicos como Colombia. Un modelo preciso permite construir un plan de contingencia, determinar qué políticas y acciones tendrán el mayor impacto en la reducción del brote, entender la dinámica de propagación y posiblemente también, cómo prevenirlo desde un inicio y detener así su propagación.

Los modelos tradicionales se basan en el modelo SIR y sus variaciones (SI, SIS, SEIR, etc.). Estos modelos dividen la población en compartimientos que representan los diferentes estados de la enfermedad. El modelo mas general (SIR) representa la propagación de la siguiente manera: S, representa la proporción susceptible de la población, aquellos que no han sido infectados; I, representa aquellos

que están infectados; y R, representa los individuos que se han recuperado de la enfermedad y ya no hacen parte activa de la propagación. Estos modelos representan la transmisión del virus a través de un conjunto de ecuaciones diferenciales no lineales que asocian la tasa de transición con la movilidad entre compartimientos [1]. Han sido una gran herramienta para guiar y diseñar políticas de salud pública.

Sin embargo, estos modelos fallan en capturar la complejidad e individualidad del comportamiento humano, puesto que modelan la población de forma homogénea y tienen únicamente en cuenta la dinámica temporal de la infección (Friás-Martínez et al. 2011). Modelos de simulación basados en agentes son una alternativa y pueden construir escenarios mucho más complejos [4].

La simulación basada en agentes ha demostrado ser una técnica tremendamente útil para modelar sistemas complejos, y muy especialmente sistemas sociales. Mediante la simulación basada en agentes, el modelador reconoce explícitamente que los sistemas complejos, y en particular los sociales, son producto de comportamientos individuales y de sus interacciones. (Conte et al., 1997; Gilbert

^{*}cdherreram@unal.edu.co

^{**}aybautistas@unal.edu.co

^{***}asgomezf@unal.edu.co

y Troitzsch, 1999; Gilbert y Terna, 2000; Gilbert, 2007) Al modelar una epidemia utilizando este modelo se sigue el progreso de la enfermedad a través de cada individuo, y se rastrea el contacto de cada individuo con otros en las redes sociales y áreas geográficas relevantes, permitiendo así una representación más clara de las interacciones entre los individuos [3][2].

En este proyecto se propone un sistema basado en el modelo de agentes con el fin de simular el comportamiento de una epidemia y el efecto del aislamiento.

Metodología

Existen cuatro componentes principales en un modelo basado en agentes: enfermedad, sociedad, transporte y el entorno. Para modelar la enfermedad se establece que se transmite por contacto directo o por cercanía, representado a través de la variable radio de contagio (r). Esta variable podría dar cuenta de las interacciones entre los individuos o el alcance que tiene el virus. Por otro lado se introduce una probabilidad de contagio (p) que puede estar asociada al tipo de enfermedad, los hábitos del individuo (estilo de vida, higiene, etc.) y otros factores. Para modelar la sociedad se introduce un número de individuos total constante en el tiempo, es decir, no se tiene en cuenta dinámicas de vida (nacimientos y muertes). En cuanto al transporte, a través de la dinámica molecular se establece el movimiento de los individuos. Finalmente, con respecto al entorno, se construyen cuartos con sus respectivas dimensiones: un cuarto cuarentena, donde se aislará una proporción de los individuos infectados, y n cuartos normales.

En una sociedad, los individuos realizan actividades diarias que implican su movimiento entre diferentes lugares, esto puede aumentar las interacciones entre las personas y contribuir a la propagación de la enfermedad. En el modelo basado en agentes se están considerando estos escenarios, pues las personas se mueven libremente en el espacio y en el tiempo, lo que genera que exista contacto entre varios grupos. La metodología empleada para el estudio de la propagación de la enfermedad consistió en la construcción de un algoritmo por medio de programación orientada a objetos en el que se consideraron tres clases para describir a los elementos de la simulación.

Las clases son:

- **Individuo:** es una clase que define los individuos, se tienen en cuenta tres estados; SUSCEPTIBLE (S), INFECTADO (I) y REMOVIDO (R), en concordancia con el modelo clásico de epidemias SIR. Cuenta con un método *CambioStatusPorRecuperacion()*, que permite cambiar el estado de cada persona según el tiempo de recuperación establecido en el que se cura la enfermedad. Además, el método *EsVecino()* permite identificar cuáles son los individuos que se encuentran a una distancia menor al radio de contagio de cada persona. Cada individuo tiene un identificador, y un número de cuarto en el que se encuentra.
- **Cuarto:** es una clase que permite describir el entorno en el que van a interactuar los individuos. Por esta razón, uno de sus principales atributos es un vector de objetos de la clase Individuo (*PersonasEnElCuarto*). El Cuarto es un objeto que se encarga de la dinámica de las personas, para modelar el movimiento se empleó el algoritmo de Leap-Frog, se incluyeron fuerzas de rebote con todas las paredes y fuerzas de disipación. Adicionalmente, uno de sus métodos fundamentales es *CalcularContagio()*, que evalúa si hay vecinos para las personas infectadas, y posteriormente, infecta a los susceptibles según la probabilidad de contagio. Al igual que en la clase anterior, los cuartos tienen un identificador, según si son CUARENTENA ó NORMAL.
- **Organizador:** con el fin de controlar el movimiento de individuos entre los cuartos se creó esta clase, el Organizador tiene un vector de objetos de la clase Cuarto (*vectorCuartos*) y un vector de objetos de la clase Individuo (*vectorIndividuos*). El Organizador es el encargado de darle los atributos iniciales a las personas y a los cuartos, además de agregar las personas a los cuartos para empezar la simulación. Todas estas instrucciones se encuentran en el método *InicializarPorDefecto()*, que crea los individuos según el número de individuos total, el número de infectados que se requiera y la cantidad de cuartos normales, pues por defecto siempre se creará un cuarto de Cuarentena.

Cabe aclarar que todo el proceso de evolución se construyó en la clase Organizador con el método

EmpezarEvolucion(), su función es controlar el movimiento de las personas en todos los cuartos en cada paso de tiempo (*tstep*). Este método tiene como argumentos *pasosEvolucionPosicion*, *pasosContagio*, donde el primero indica los pasos de tiempo para el movimiento, y el segundo muestra el número de veces que se recorrerán todos estos pasos, lo que simulará los días en la evolución de la enfermedad.

Adicionalmente, en los argumentos se tiene una variable tipo bool *incluirAislamiento*, que si es verdadera ejecutará todo el proceso de cambio de cuarto hacia el cuarto Cuarentena para las personas infectadas. Este cambio se realizará según el módulo de la variable que se pase como argumento en *modContagio*. Para simular la propagación de la enfermedad entre los individuos, se llama al método *CalcularContagio()* en cada día de la evolución (*pasosContagio*).

La inicialización del *vectorIndividuos* en Organizador fue realizada de forma paralela con OpenMP, así como la inicialización de los atributos de los individuos, como la masa y la posición. Por otra parte, la evolución de los cuartos para diferentes radios y probabilidades se realizó de paralelamente empleando MPI.

Resultados y Análisis

Cada uno de los resultados expuestos se obtuvo para un valor inicial de población infectada de 6 individuos con un valor de 1000 individuos en total. A través del modelo implementado fue posible reproducir un comportamiento típico del modelo SIR para el caso en que no existe aislamiento. Ver Figura 1(a). Allí se evidencia cómo el número de infectados aumenta rápidamente en un periodo de tiempo muy corto. Para poder visualizar el efecto del aislamiento de una proporción de los individuos infectados tan pronto se detectan, se realizó la simulación bajo los mismos parámetros de radio y probabilidad.

En la Figura 1(b) se evidencia cómo el aislamiento permite un mayor control sobre la epidemia. Existe una reducción del número de infectados a pesar de prolongar su erradicación. Igualmente, surge una pequeña proporción de individuos susceptibles que no resultan infectados.

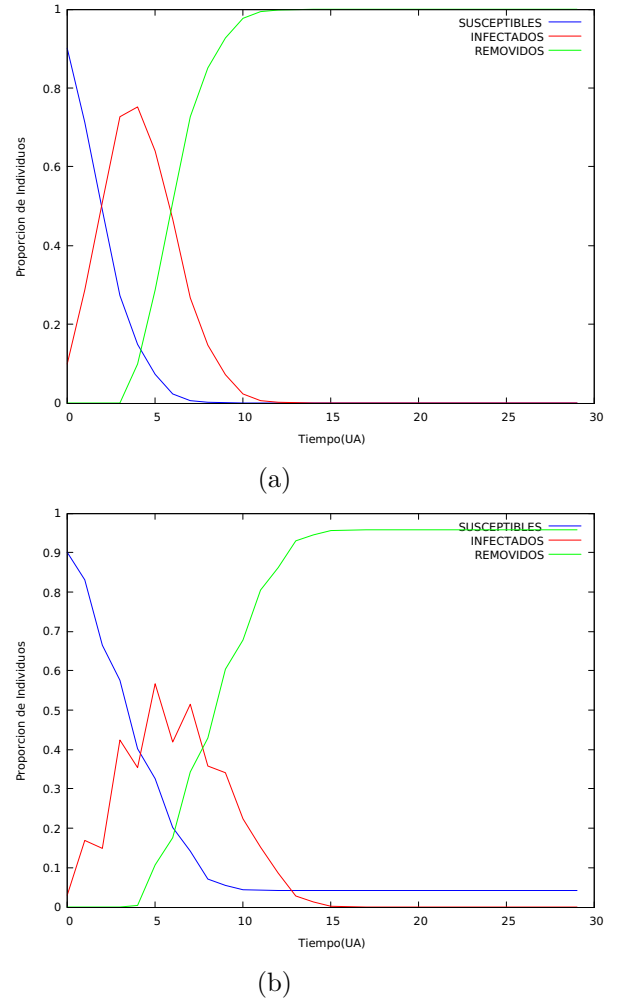


Figura 1: Simulación (a) sin y (b) con aislamiento (proporción de individuos a cuarentena $p_c = 0,7$) para un radio de contagio $r = 1,4$ y probabilidad de contagio $p = 0,14$ fijos.

Para poder visualizar claramente el efecto de cada una de las variables, se aísla cada una de ellas en la simulación. Inicialmente, se realiza una gráfica (Ver Figura 2) que representa la evolución de la proporción de individuos infectados en la población para cinco radios diferentes de contagio.

Allí se refleja cómo la tasa de infección es considerablemente sensible a pequeños cambios en el radio de contagio. Por ejemplo, sólo con doblar su valor, la proporción de individuos infectados alcanza casi la totalidad de la población. Ahora, para analizar el comportamiento de la proporción de infectados con la variación de probabilidad de contagio, se toman cinco valores diferentes de probabilidad, manteniendo las demás variables fijas. Ver Figura 3.

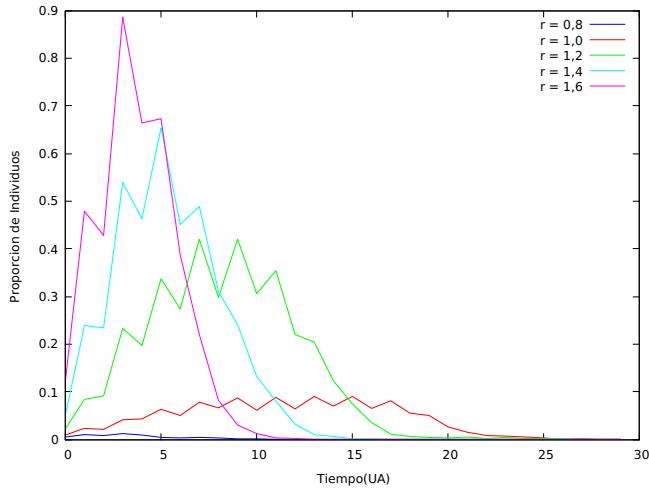


Figura 2: Simulación para una probabilidad de contagio $p = 0,14$ y proporción de individuos a cuarentena $p_c = 0,5$ fijos

El aumento de la probabilidad en cada una de las iteraciones evoca en un mayor número de infectados. Si se relaciona uno de los factores del radio de contagio con la interacción entre los individuos y los buenos hábitos de higiene con la disminución de la probabilidad, es posible afirmar que una reducción de las interacciones puede ser más efectiva controlando la epidemia.

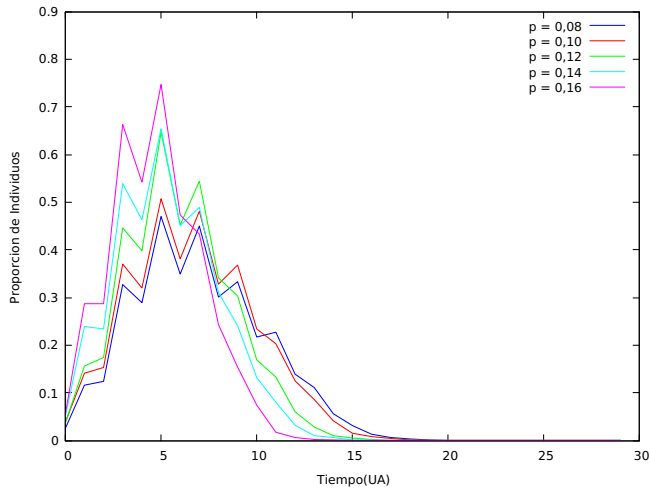


Figura 3: Simulación para un radio de contagio $r = 1,4$ y proporción de individuos a cuarentena $p_c = 0,5$ fijos

Ahora bien, teniendo en cuenta que existe la posibilidad de que no todos los individuos infectados puedan entrar en cuarentena, ya sea debido a que estos son asintomáticos o no, se llevan a cuarentena diferentes proporciones de infectados

y se representa la evolución de la proporción de individuos infectados en la figura 4, esto con el fin de estudiar la dependencia con esta variable.

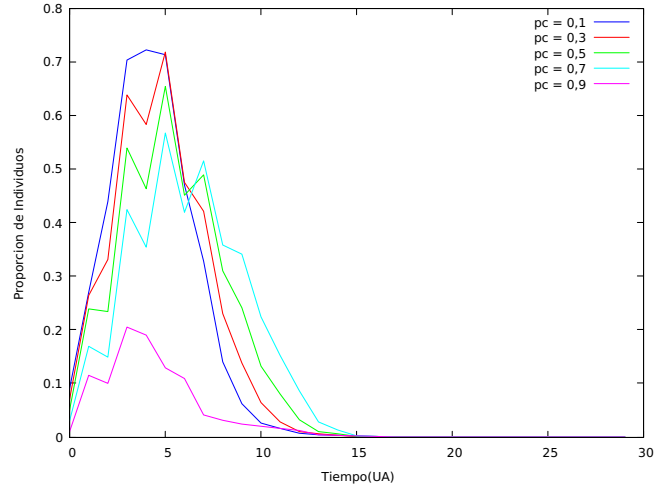


Figura 4: Simulación con un radio de contagio $r = 1,4$ y probabilidad de contagio $p = 0,14$ fijos

Se puede observar que al disminuir la proporción de individuos que van a cuarentena, el pico aumenta de forma considerable, lo que implica un mayor número de individuos que se infectan en un menor lapso de tiempo. Esto llegaría a afectar en gran medida a la población desde que, el sistema de salud se puede ver colapsado y no sería posible atender a todas las personas.

Es posible además afirmar que existe un valor crítico para dicha proporción donde aumenta desproporcionalmente el número de infectados, por lo que el aislamiento de los individuos infectados resulta realmente efectivo en la disminución de la propagación para proporciones altas.

En términos de la eficiencia del programa debido a la paralelización, se tuvo en cuenta implementación con la interfaz OpenMP al interior de algunas funciones para el llenado de algunos vectores, dado que al variar la cantidad de individuos, se hacía necesario una distribución de las tareas para la creación de los objetos y su distribución en los cuartos. Por otro lado, dado que se generaron simulaciones para distintos radios de contagio y probabilidades de contagio, se distribuyó la tarea del main en distintos procesadores utilizando MPI para optimizar la generación de los datos.

Los resultados obtenidos en la paralelización se pueden observar en las gráficas a continuación (Figura 5 y 6). Para poder hacer esta comparación, se evaluaron las paralelizaciones para 1000 individuos en total, iniciando con 20 individuos infectados, una probabilidad de traslado a cuarentena $p_c = 0,5$, 2 probabilidades diferentes para 10 valores de radio y 2 cuartos normales.

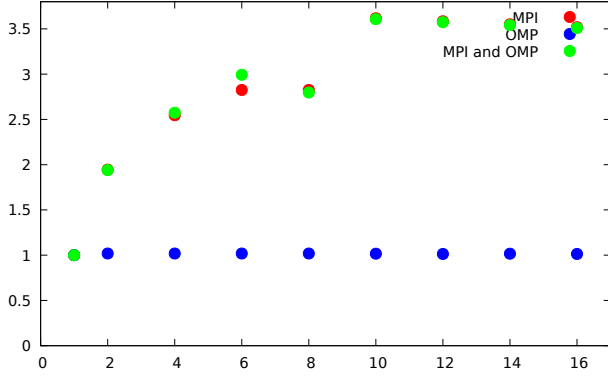


Figura 5: Speedup de la paralelización realizada con OpenMP, MPI y un híbrido entre estos.

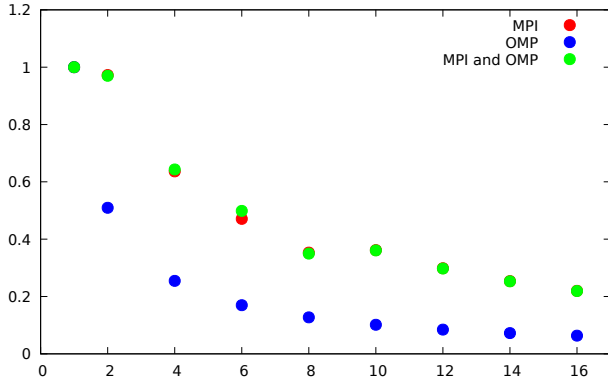


Figura 6: Eficiencia de la paralelización realizada con OpenMP, MPI y un híbrido entre estos.

Se puede observar en ambas gráficas que la paralelización con MPI tiene una mayor influencia en la mejoría de la velocidad del programa debido a que OpenMP se ha implementado únicamente en dos métodos de la clase Organizador, que corresponden a una pequeña proporción de la evolución del sistema. Por otro lado, se evidencia que el tiempo de ejecución aumenta considerablemente hasta 10 procesadores. En primer lugar esto ocurre debido a que la mejoría que puede otorgar la paralelización está limitada por la proporción del código que es serial.

Por otro lado, es la complejidad del algoritmo la que está limitando la mejoría de la velocidad, no el número de procesadores. En este caso, para las variables tomadas, para más de 10 procesadores el algoritmo no se puede paralelizar más. Este valor corresponde con los 10 radios que se están distribuyendo entre los procesadores.

Problemas y Soluciones

A lo largo de la construcción de la simulación surgieron varias complicaciones.

- Al variar el número de individuos total parecía no existir un cambio en el contagio considerable en el tiempo. Esto ocurría debido a que era importante variar no solamente las dimensiones de los cuartos, sino también las posiciones de inicialización de los individuos. Se añade entonces el parámetro de dimensiones del cuarto a la inicialización.
- Al aumentar el número de cuartos normales disminuía la velocidad de propagación. Esto ocurría debido a que los individuos se distribuyen uniformemente entre los cuartos. Se aumenta entonces el número inicial de infectados de acuerdo al número de cuartos.
- Al imprimir las posiciones para las personas en cada paso de tiempo, se observó que cambiaban considerablemente de un paso al siguiente. Esto ocurría porque hay una dependencia entre el número de personas totales, las dimensiones del cuarto y la magnitud de la fuerza de rebote. Se empleó $K = 323.4$ como constante para la fuerza y para evitar este problema, se debe considerar un tamaño para el cuarto adecuado según el número de personas que haya en cada uno.

Conclusiones

- En este estudio se propone una simulación para la propagación de una enfermedad contagiosa a partir de la modelación basada en agentes. Los resultados indican que este modelo permite representar diferentes entornos para la propagación de la enfermedad al incluir el movimiento de los agentes, y aumentar el número de infectados según las interacciones entre los mismos.

El modelo puede ser extendido para simular entornos particulares, como un supermercado, un colegio, una calle, etc, al ajustar de forma adecuada los parámetros del cuarto, sin embargo, en este caso, se hizo un acercamiento muy general en el que se vario la probabilidad y el radio de contagio sin tener en cuenta un entorno en específico.

- Fue posible reproducir los resultados obtenidos previamente por otros modelos epidemiológicos donde se afirma que un pequeño aumento en la probabilidad de contagio o en el radio de contagio puede desencadenar una epidemia con un mayor número de individuos infectados en un intervalo menor de tiempo. Igualmente se obtiene que la proporción de individuos infectados que entran a cuarentena debe ser considerablemente alta para disminuir la propagación.
- En la construcción del algoritmo se tuvo en cuenta solo la comunicación entre cuarto normal y cuarto cuarentena para el envío de una proporción de individuos infectados, sin embargo, debido a la forma en la que está implementado, se puede establecer comunicación entre los cuartos normales, esto con el fin de simular transporte entre diferentes entornos, pues cada cuarto tiene un método *RecibirIndividuos()* y *QuitarIndividuos()* para el envío de individuos entre los cuartos. Se pueden llegar a incluir entonces otras variables importantes, como la tasa de inmigración para estudiar la propagación de la enfermedad. Esto podría variar la implementación de los métodos de paralelización para el sistema, dado que se tendría que compartir el vector de individuos del sistema en total.

- [2] Elizabeth Hunter, Brian Mac Namee y John D Kelleher. “A taxonomy for agent-based models in human infectious disease epidemiology”. En: *Journal of Artificial Societies and Social Simulation* 20.3 (2017).
- [3] Liliana Perez y Suzana Dragicevic. “An agent-based approach for modeling dynamics of contagious disease spread”. En: *International journal of health geographics* 8.1 (2009), pág. 50.
- [4] Jouni T Tuomisto y col. “An agent-based epidemic model REINA for COVID-19 to identify destructive policies”. En: *medRxiv* (2020).

Referencias

- [1] Enrique Frias-Martinez, Graham Williamson y Vanessa Frias-Martinez. “An agent-based model of epidemic spread using human mobility and social network information”. En: *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*. IEEE. 2011, págs. 57-64.