# Reporte práctica seis: Sistemas multiagente

José Anastacio Hernández Saldaña
Posgrado de Ingeniería de Sistemas
1186622
jose.hernandezsld@uanl.edu.mx

18 de septiembre de 2017

**Resumen**

Este es un reporte sobre la práctica seis con respecto al tema de sistemas multiagente que se realizó en la clase de Simulación de sistemas, cómputo paralelo en R.

## 1. Tarea: Implementación paralela de un sistema multiagente

Un sistema multiagente es un sistema computacional donde interactúan varios agentes dentro de un ambiente o entorno, los agentes realizan acciones en el ambiente y a partir del estado del ambiente el agente realiza alguna acción. Este tipo de sistemas son descentralizados, es decir, los agentes no siguen órdenes de una unidad central, sino que cada agente es autónomo.

En esta tarea se realizó una simulación de una epidemia de una cantidad $n$ de agentes, donde cada agente podría estar en alguno de los siguientes estados: sano, infectado e inmune. El agente estaba sano si no se había contagiado al estar cerca de algún otro agente infectado, si estaba a una distancia $d$ del infectado tal que era menor a un umbral $r$, hay una probabilidad $p_i = \frac{r-d}{r}$ de infectarse en otro caso $p_r = 0$, si el agente estaba infectado, tenia una probabilidad $p_r$ de recuperarse y volverse inmune. Cada agente tenia una velocidad horizontal y vertical $dx$ y $dy$ respectivamente, así que en cada paso de la simulación los agentes avanzaban en función de su posición inicial y su velocidad y se volvían a hacer los cálculos para definir su estado, todo esto dentro de un espacio de dimensiones $l \times l$, el cual esta en forma de un torus o dona, ya que al exceder alguno de los bordes, el agente aparece en borde contrario.

### 1.1. Diseño del Experimento

Para el experimento se tomó como base el código de la página del curso, donde se encuentra la simulación ya programada con los siguientes parámetros, umbral de infección $r = 0.1$, probabilidad de recuperación $p_r = 0.2$, con una dimensión de $l = 1.5$, se tienen una cantidad de agentes de $n = 50$, se creaban posiciones $x$, $y$ con velocidades $dx$, $dy$ tomadas de una distribución uniforme tal que $0 \leq x \leq l$ y $\frac{-l}{30} \leq dx \leq \frac{l}{30}$, de manera similar para $y$ y $dy$. El estado inicial de los agentes era asignado con una probabilidad $\alpha_i = 0.05$ de comenzar infectado y una probabilidad $1 - \alpha_i$ de comenzar sano.

Este código no se encuentra paralelizado y cuenta con varias opciones para paralelizarse y optimizarse identificadas, que son las siguientes:

1. Asignación de estado inicial. La asignación se hace de manera iterativa para cada uno de los agentes puede hacerse de manera paralela.
2. Cálculo de la distancia de un agente y los agentes cercanos. El cálculo de distancia en cada agente hacia todos los demás puede hacerse de manera paralela y puede mejorarse si se reduce la cantidad de agentes a comparar.
3. Asignación de posiciones y velocidades iniciales. Esta asignación también se hace de manera iterativa para cada agente, dando la opción de paralelizarse.

Como el cálculo 2 se hace dentro del código de la asignación 1, estas pueden tener problemas si ambas se paralelizan ya que paralelizar procedimientos paralelos los hace competir por los recursos del procesador, así que se decidió paralelizar la asignación de estados, y optimizar el cálculo de la distancia. Otro enfoque también considerado fue utilizar un método de linea de barrido para las asignaciones 1, 3 y que optimiza el cálculo 2 ya que en cada iteración de la linea de barrido considera solo aquellos agentes que se encuentran a una distancia $r$ de la linea, evitando calcular la distancia de agentes más allá del umbral. Aunque el algoritmo de linea de barrido puede paralelizarse en el cálculo

de la distancia 2 no daría mucha mejora ya que la cantidad de agentes es pequeña y la paralelización de las asignaciones depende de la iteración anterior por lo que no podría paralelizarse. Pero tomando en cuenta el enfoque de linea de barrido puede hacerse una versión paralela de este código sí en lugar de llevar un registro de quienes están dentro de la linea de barrido e ir actualizando en cada iteración, puede para cada punto de la línea de barrido calcularse los vecinos cercanos.

Por lo que se tuvieron cuatro algoritmos para comparar, que se clasificaron de la siguiente manera

1. **Original:** Algoritmo original
2. **Original P:** Algoritmo original paralelizado
3. **LdB:** Algoritmo de linea de barrido
4. **LdB P:** Algoritmo basado en linea de barrido paralelizado

Se decidió comparar los cuatro algoritmos para saber si utilizar un buen algoritmo, como lo es el de linea de barrido, contra implementaciones paralelas como el algoritmo original paralelizado de la práctica y la implementación paralela basada en la linea de barrido nos ofrece una ganancia significativa en tiempo y en que medida. Se tomó el algoritmo original como cota superior de lo que podríamos esperar del desempeño de los algoritmos, el código de los cuatro algoritmo esta dentro del repositorio git del curso.

Para comenzar la experimentación, se utilizó una computadora con las siguientes especificaciones, Procesador Intel Core i7-4790 CPU @ 3.6GHz × 8 y memoria RAM de 24 GB utilizando solamente los cuatro núcleos físicos disponibles. Para cada experimento se hicieron 30 réplicas con los parámetros utilizados en la práctica. La comparación se efectuó de manera transformada de acuerdo a la siguiente fórmula $T(x) = \frac{\text{máx}(X)-x}{\text{máx}(X)}$ donde máx$(X)$ es el máximo valor obtenido en nuestros resultados y $x$ el valor actual.

## 1.2. Resultados

Al revisar los resultados de la experimentación, que están graficados en la figura 1, podemos ver como el algoritmo original nos da el mayor tiempo, sin embargo, en los otros algoritmos, tenemos que el algoritmo original paralelizado tuvo un mejor desempeño que el obtenido del algoritmo de línea de barrido o el algoritmo de línea de barrido paralelizado.

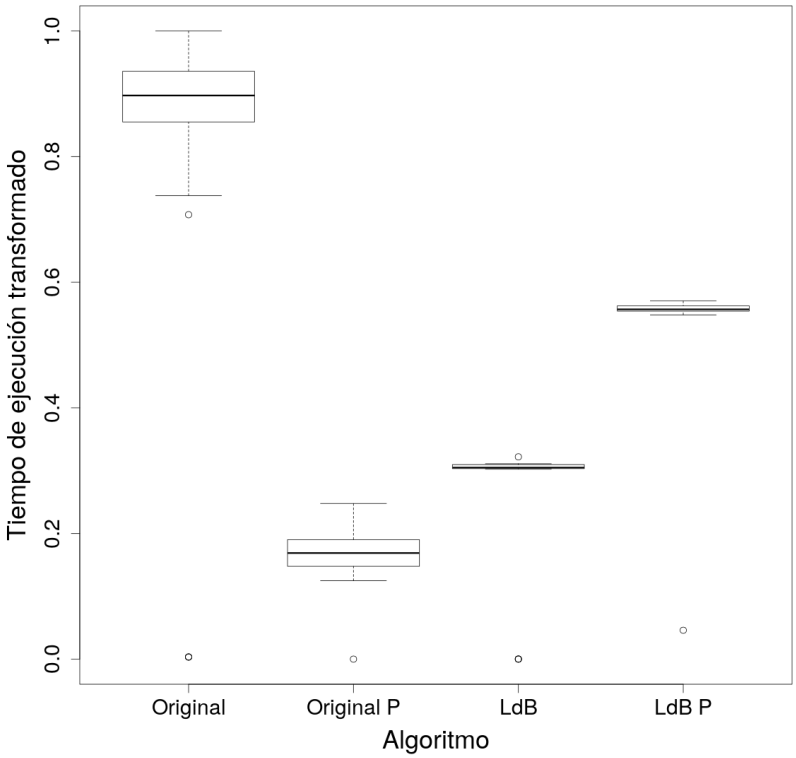


Figura 1: Comparación de tiempo de los cuatro algoritmos

El resultado obtenido el algoritmo de linea de barrido y el algoritmo basado en la linea de barrido

paralelizado fue interesante, ya que al perder la lista de elementos que están en el barrido y tener que calcular los cercanos a cada elemento nos afecto en el tiempo del algoritmo.

# 2. Extra Uno: Efecto de vacunación de los agentes

La actividad extra uno consiste en agregar una probabilidad de vacunado $p_v$ al momento de crear la población inicial, como un agente vacunado es un agente que es inmune a quedar infectado y no tiene efecto sobre los saludables es similar a un agente recuperado, por lo que se utilizó el mismo estado y solo se agregó la posibilidad de agregar este estado desde el inicio de la generación de la población inicial.

## 2.1. Diseño del Experimento

Los parámetros utilizados fueron los mismos que los del experimento anterior, la diferencia fue la probabilidad inicial ya que se probó con valores de probabilidad $p_v$ de 0.1 a 0.9 en intervalos de 0.1, con $p_i = 0.05$, por lo que la probabilidad de estar sano fue $p_s = 1 - (p_v + p_i)$. Se utilizaron treinta réplicas para este experimento.

## 2.2. Resultados

El resultado obtenido de la experimentación mostró que si hay un efecto entre el procentaje de agentes vacunados desde el comienzo al irse incrementándose hasta llegar a tener cerca del 90 % de la población vacunada, esto se puede ver en la figura 2.

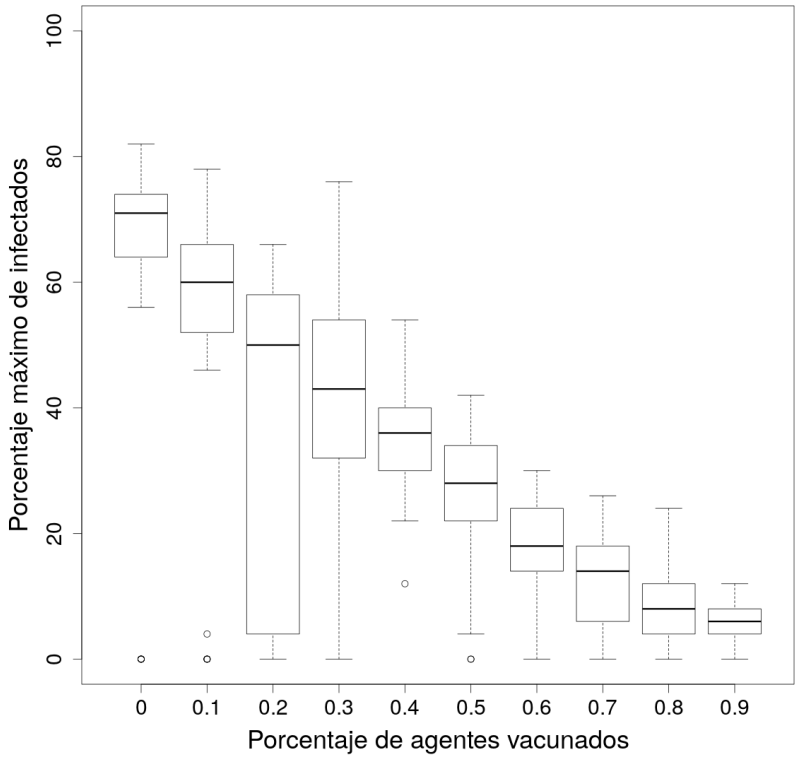


Figura 2: Comparación de máxima cantidad de infectados y el porcentaje de vacunados iniciales

Con esto podemos darnos cuenta que de un 40 % a 50 % de la población de agentes vacunada, la cantidad máxima de infectados es del cerca del 35 % y para valores mayores al 70 % la cantidad máxima de infectados es menor al 20 %.

# 3. Extra Dos: Efecto de la probabilidad inicial de los infectados

La actividad extra dos consistió en examniar el efecto de la probabilidad inicial de infectados $p_i$ al crear la poblacion de agentes de la simulación y ver como esto afectaba a la cantidad máxima de infectados.

## 3.1. Diseño del Experimento

Los parámetros utlizados fueron los mismos que los del experimento anterior,solo que la probabilidad inicial cambio con los valores de probabilidad $p_i$ de 0.05 a 0.9 en intervalos de 0.05, sin probabilidad de vacuna $p_v$ ya que comenzar con agentes vacunados es igual a comenzar con menos agentes sanos iniciales, por lo que tener la mitad de vacunados con 10 % de agentes infectados es igual a una problacion de 25 agentes sanos con 20 % de agentes infectados. La probabilidad de estar sano fué $p_s = 1 - p_i$. De igual manera se utilizaron treinta réplicas para este experimento.

## 3.2. Resultados

El resultado obtenido de la experimentación como se puede ver en la figura 3, que aunque si hay un efecto creciente entre el porcentaje inicial de infectados de la población de agentes y la máxima cantidad de agentes infectados, esta rara vez fue menor al 50 % de los agentes susceptibles a ser infectados, por lo que el contagio puede afectar a gran parte de la población de agentes al mismo tiempo. Se puede también observar que para valores iniciales entre 0.05 % y 30 % no se ve una variación significativa en el valor de las medianas llevándonos a pensar que aunque el tamaño de la población de agentes infectados es pequeño, tiene el mismo efecto que comenzar con un 30 % de población infectada.

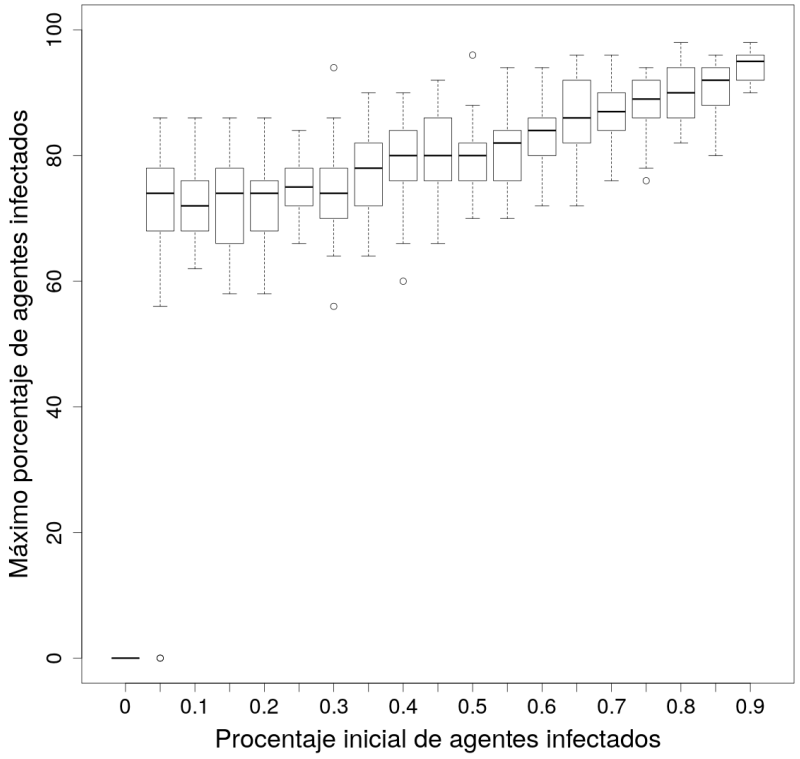


Figura 3: Comparación de máxima cantidad de infectados y el porcentaje de vacunados iniciales

# 4. Conclusiones

En esta práctica pudimos ver por un lado como una implementación paralela ofrece ventajas, aún cuando esta no sea muy compleja, ya que puede tardar menos que una buena implementación iterativa, como lo es el algoritmo de línea de barrido ya que la paralelización nos ofrece una ganancia en tiempo con mayor significancia. Por parte de la vacunación de agentes, esta sí tiene un impacto directo en la máxima cantidad de agentes infectados ya que reduce la cantidad de agentes susceptibles a ser infectados, mostrando la importancia de contar con una mayor cantidad de agentes vacunados. Por último, el efecto de la probabilidad de infección inicial nos muestra que aún con porcentajes bajos, estos alcanzan una tasa máxima muy significativa abarcando más del 60 % de la población de agentes susceptibles, reforzando la conclusión anterior de buscar tener la menor población de agentes susceptibles posibles.