RELACIÓN DE LA PROBABILIDAD DE FRAGMENTACIÓN Y EL NÚMERO DE UNIDADES DE ATAQUE EN UNA SIMULACIÓN DE GUERRA DE GUERRILLAS

Juan Diego Rojas Zambrano[1]

[1] Universidad Nacional de Colombia, Departamento de Física

Resumen

El presente trabajo describe la implementación de un algoritmo que caracteriza la dinámica de fragmentación y coalición de una fuerza insurgente. Se estudia a través de simulaciones la relación entre la probabilidad de fragmentación y el número de unidades de ataque. Los resultados muestran que el número de unidades de ataque decrece con la probabilidad de fragmentación. Cuantitativamente se desarrolla un modelo exponencial para describir este comportamiento.

Palabras clave: Simulación, algoritmo, guerra.

1. Introducción

En [1] y [2] se describen ciertos patrones que subyacen en guerras y hechos de terrorismo que ocurren en el mundo, aún cuando los conflictos están separados geográficamente. Se describe que una fuerza insurgente tiene una fuerza de ataque N total, la cual se reparte en varias unidades de ataque, las cuales poseen una dinámica de coalición y fragmentación. Entonces las distintas unidades de ataque poseerán diferentes cantidades de fuerza de ataque dependiendo de esta dinámica. La característica que se estudia es la relación exponencial entre el número de unidades de ataque n_s con una fuerza s, y la fuerza de ataque s. Esta relación posee varios parámetros, entre los cuales está la probabilidad de fragmentación de una unidad ν. La dinámica de las unidades de ataque viene determinada por el siguiente algoritmo [1]:

jdrojasz@unal.edu.co

1. Seleccionar una unidad de ataque i aleatoriamente, con probabilidad probabilidad proporcional a su fuerza de ataque s_i.

2. Determinar con probabilidad ν, si se fragmenta la unidad i seleccionada.

   2.1 Si hay coalición (no hay fragmentación), se selecciona una unidad de ataque j con probabilidad proporcional a su fuerza de ataque.

   2.2 Realizar la coalición de las unidades i y j e ir al paso 3.

   2.3 Si hay fragmentación, se fragmenta la unidad i en s_i unidades de fuerza de ataque 1.

3. Repetir los pasos 1 y 2.

4. Calcular la distribución n_s vs. s.

Este trabajo tiene como objetivo estudiar la relación entre ν, s y n_s, cuando se tiene una fuerza insurgente con total de fuerza $N = 10000$.

2. Implementación

Para implementar el algoritmo, se describe primero las ideas generales para la descripción de la guerrilla y posteriormente se describe detalladamente los pasos más importantes del algoritmo.

2.1. Idea general de la implementación

Para realizar la implementación del algoritmo, se describe la guerrilla con un vector $\vec{s}$ de N componentes, de tal forma que cada componente (i) represente una unidad de ataque, y el valor de esa componente (s_i) represente la fuerza de ataque de la unidad de ataque i. De esta forma, si la fuerza total de la guerrilla es $N = 10$, y está conformada por 4 unidades, una descripción de esta guerrilla sería:

$$\vec{s} = (3, 2, 1, 4, 0, 0, 0, 0, 0, 0, 0)$$

En este caso, la unidad 1 tiene fuerza 3, la unidad 2 tiene fuerza 2, la unidad 3 tiene fuerza 1, y la unidad 4 tiene fuerza 4. Las demás componentes son 0 porque no existes más unidades. Claramente se debe cumplir que la suma de los valores de todas las componentes es N. Para este ejemplo: $3 + 2 + 4 + 1 = 10$.

Ahora supongamos que la unidad 1 (con fuerza igual a $s_i = 3$) se va a fragmentar. De acuerdo al paso 2.3, se crearán 3 unidades de fuerza 1, por lo que después de la fragmentación el vector queda de la siguiente forma:

$$\vec{s} = (1, 2, 1, 4, 1, 1, 0, 0, 0, 0, 0)$$

Por convención, se irán completando las componentes de $\vec{s}$ hacia la derecha, es decir, $\vec{s}$ siempre tendrá una cola de ceros. Ahora supongamos que se van a coalicionar las unidades 2 y 3. El vector resultante será

$$\vec{s} = (1, 3, 4, 1, 1, 0, 0, 0, 0, 0, 0)$$

Cumpliendo con la convención descrita, la nueva unidad quedará ubicada en la posición de la unidad con el menor índice. Por las características del problema, no se puede tener coalición de una unidad consigo misma. Nótese que las posiciones de las demás unidades se corren para tener la cola de ceros.

Por lo anterior, sí la guerrilla está formada por una unidad se tiene

$$\vec{s}_0 = (10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

Y por lo tanto, con el número máximo de unidades, se tendrá

$$\vec{s}_1 = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$$

De esta forma, y de acuerdo algoritmo descrito, la configuración $\vec{s}_0$ siempre cambia a la configuración $\vec{s}_1$.

2.2. Descripción de la coalición y la fragmentación

En general, el vector $\vec{s}$ es de la forma

$$\vec{s} = (s_1, s_2, s_3, ..., s_u, 0, 0, ...)$$

Donde u es el número de unidades. De esta forma, se cumple

$$\sum_{i=1}^{N} s_i = \sum_{i=1}^{u} s_i = N$$

Por el algoritmo, conviene tener dos funciones que realicen la coalición y la fragmentación de las unidades respectivamente. Si se quiere hacer la fragmentación de la unidad i, desde la configuración

$$\vec{s} = (s_1, s_2, s_3, ..., s_i, ..., s_u, 0, 0, ...)$$

Entonces los pasos a seguir son:

1. Guardar el valor de s_i, por ejemplo $k = s_i$.

2. Asignar $\vec{s_i} = 1$. En este punto se tiene

$$\vec{s} = (s_1, s_2, s_3, ..., \underbrace{1}_{\text{posición } i}, ..., s_u, 0, 0, ...)$$

3. Asignar, a partir de la posición $u+1$ y hasta $k-1$ posiciones después, los valores de 1. Entonces se tiene

$$\vec{s} = (s_1, s_2, s_3, ..., 1, ..., s_u, \underbrace{1, ..., 1}_{k-1 \text{ posiciones}}, 0, 0, ...)$$

Por otra parte, si se quiere hacer la coalición de las unidades i y j, desde la configuración

$$\vec{s} = (s_1, s_2, s_3, ..., s_i, ..., s_j, ..., s_u, 0, 0, ...)$$

Entonces los pasos a seguir son:

1. Si $i > j$, asignar a i con el valor de j y a j con el valor de i.

2. Asignar a la posición i, el valor $s_i + s_j$. El vector queda

$$\vec{s} = (s_1, s_2, s_3, ..., s_i + s_j, ..., s_j, ..., s_{u-1}, s_u, 0, 0, ...)$$

3. A partir de la posición j y hasta la posición u, asignar los valores de las posiciones siguientes. El vector queda

$$\vec{s} = (s_1, s_2, s_3, ..., s_i + s_j, ..., s_{j+1}, ..., s_u, 0, 0, 0, ...)$$

4. Si el número de unidades es el máximo $u = N$, esto es, si se tenia inicialmente la configuración $\vec{s}_1$, se asigna a la última posición un 0.

Nótense que el primer paso se realiza para siempre tener $i < j$ y poder aplicar los demás pasos sin pérdida de generalidad.

2.3. Descripción de la selección de la unidad

El último paso importante es la selección de la unidad. Como la selección de la unidad se debe realizar proporcional a su fuerza, entonces este paso no es sencillo y en el código se realiza una función para esto. Se empieza con un número aleatorio x entre 1 y N. Se supone que x se escoge con distribución uniforme, es decir, todos los valores entre 1 y N tienen igual opción de ser escogidos. Si la unidad i tiene fuerza s_i, entonces convenimos en que la probabilidad de que sea escogida i es s_i/N. Esto cumple con el criterio de que la probabilidad de selección es proporcional a la fuerza. De esta forma, el vector $\vec{s}/N$ puede entenderse como un vector de distribución de probabilidades. Sin embargo, no se va a trabajar con el vector $\vec{s}/N$ sino con el vector $\vec{s}$ directamente por simplicidad en la implementación teniendo en cuenta que la expresión

$$d(i) = \sum_{k=1}^{i} \frac{s_k}{N}$$

representa la probabilidad de seleccionar una unidad con índice menor a i. Luego, la probabilidad de escoger la unidad i es

$$p(i) = d(i) - d(i-1)$$

Donde se debe definir $d(0) = 0$. Si se escoge un número y entre 0 y 1, y ese número esta entre $d(i)$ y $d(i-1)$, es decir, si cayo en el intervalo asignado a la unidad i, entonces se escoge la unidad i. Esto es

$$\text{Si } y \in (d(i), d(i-1)], \text{ entonces se selecciona la unidad } i$$

Multiplicando el criterio por N:

$$\text{Si } x \in \left(\sum_{k=1}^{i} s_k, \sum_{k=1}^{i-1} s_k\right], \text{ entonces se selecciona la unidad } i$$

Entonces, los pasos para seleccionar la unidad son:

1. Escoger un número x con distribución uniforme entre 1 y N. Se crean dos variables $a = 0$ y $b = 0$, y se inicializa un contador: $k = 1$.
2. Se asigna $b = a + s_k$.
3. Si $a < x \leq b$, entonces ir al paso 5. Si no, ir al paso 4.
4. Hacer $a = b$ e ir al paso 2.
5. La unidad seleccionada es la k.

3. Resultados y análisis

Se realizaron simulaciones con $N = 10000$, para 10 valores diferentes de ν y con 20000 iteraciones. Los resultados se muestran en el apéndice. Todas las simulaciones se realizaron en un sistema operativo de 64 bits Intel(R) Core(TM) i3-4160 CPU a 3.60 GHz. Los tiempos de simulación varían entre 15 segundos (con $\nu = 0{,}9$) y 120 segundos (con $\nu = 0{,}01$). La condición inicial de la simulación es

$$\vec{s} = (10000, 0, 0, 0, 0, ...)$$

Los datos muestran que $\log(n_s)$ y $\log(s)$ se relacionan linealmente, por lo tanto, los datos se ajustan a la ecuación

$$\log(n_s) = \alpha \log(s) + b$$

Luego

$$n_s = 10^b s^\alpha$$

El parámetro b varia entre el rango $3{,}8 \pm 0{,}3$. Se considerará por lo tanto que b es aproximadamente constante, y la ecuación toma la forma

$$n_s = 3800 s^\alpha$$

La figura 1 presenta el ajuste (de segundo orden) entre ν y α para los casos estudiados. La ecuación que se obtiene es

$$-\alpha = 1{,}21\nu^2 + 0{,}43\nu + 2{,}18$$

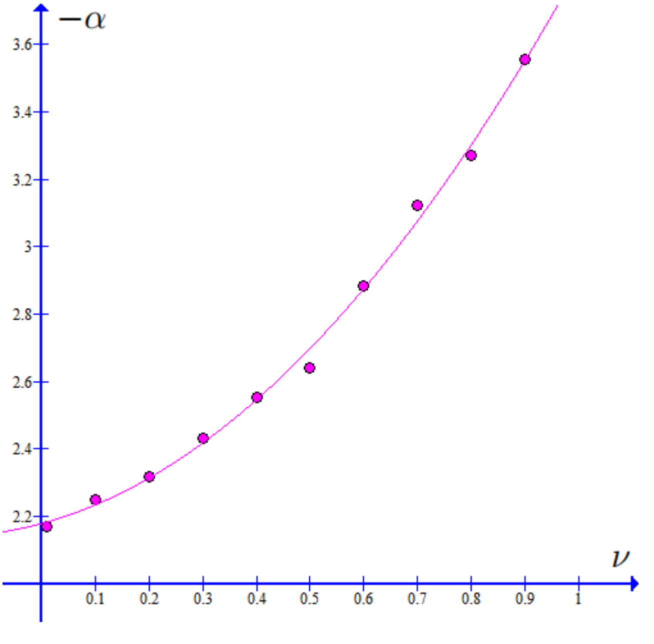

FIGURA 1. *Comparación entre α y ν. La curva tiene ecuación $y = 1{,}21x^2 + 0{,}43x + 2{,}18$ y ajusta con $R^2 = 0{,}996$.*

Entonces, un modelo para la cantidad de unidades (n_s) en función de su fuerza de ataque (s) es:

$$n_s(s, \nu) = 3800 s^{-(1{,}21\nu^2 + 0{,}43\nu + 2{,}18)} \tag{1}$$

Esto muestra que la cantidad de unidades decrece con la probabilidad de fragmentación y aumenta con la fuerza de ataque s.

4. Conclusiones

Los resultados obtenidos muestran que el parámetro α se relaciona cuadráticamente con la probabilidad de fragmentación ν. Por lo tanto el número de unidades de ataque decrece con ν. Esto está de acuerdo con la intuición en el modelo descrito, pues si es más probable que una unidad de fuerza s se fragmente, entonces deben hacer pocas de ellas. Por otra parte, el método descrito no solamente funciona para $N = 10000$. De forma análoga puede emplearse para distintos valores de N.

Referencias

[1] N. F. Johnson *et. al.*, "Universal patterns underlying ongoing wars and terrorism", *arxiv.org*, May. 2016.

[2] N. F. Johnson *et. al.*, "From old wars to new wars and global terrorism", *arxiv.org*, Jun. 2005.

A. Apéndice: Resultados de la simulación

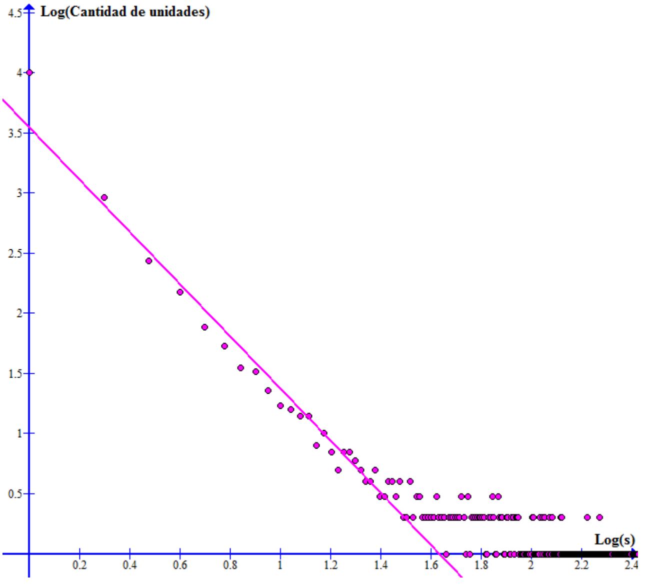

Figura 2. *Resultados de la simulación con* $\nu = 0{,}01$. *La ecuación de la recta es* $y = -2{,}17x + 3{,}55$.

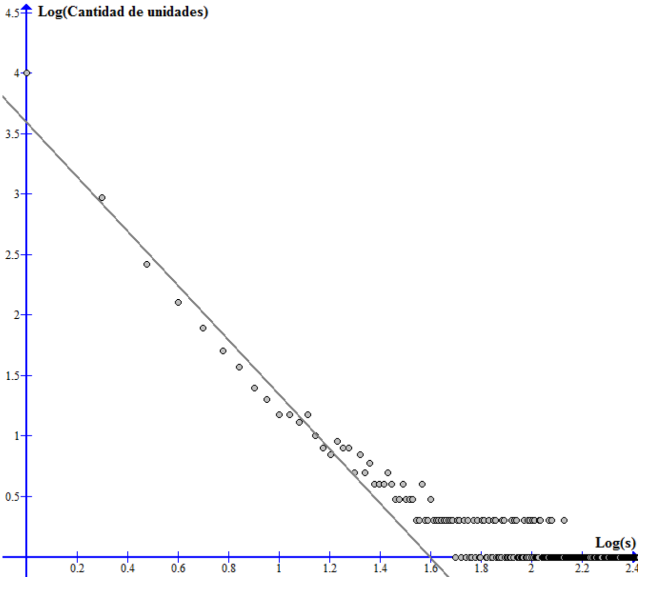

Figura 3. *Resultados de la simulación con* $\nu = 0{,}1$. *La ecuación de la recta es* $y = -2{,}25x + 3{,}59$.

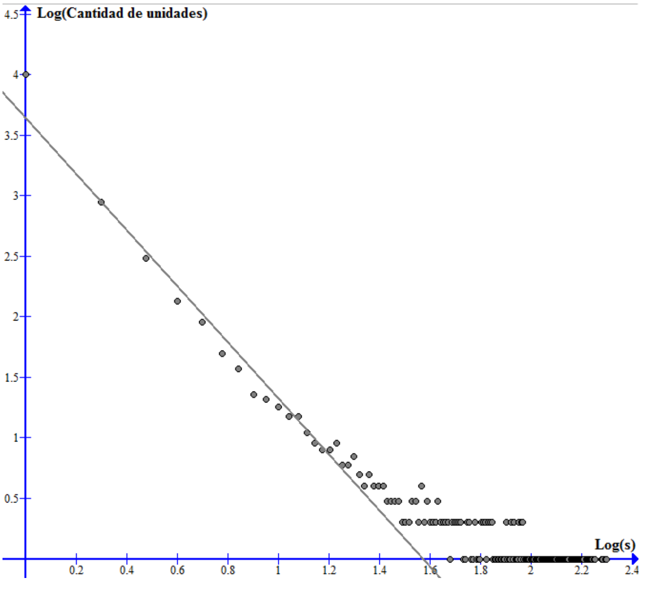

FIGURA 4. *Resultados de la simulación con* $\nu = 0{,}2$. *La ecuación de la recta es* $y = -2{,}31x + 3{,}64$.

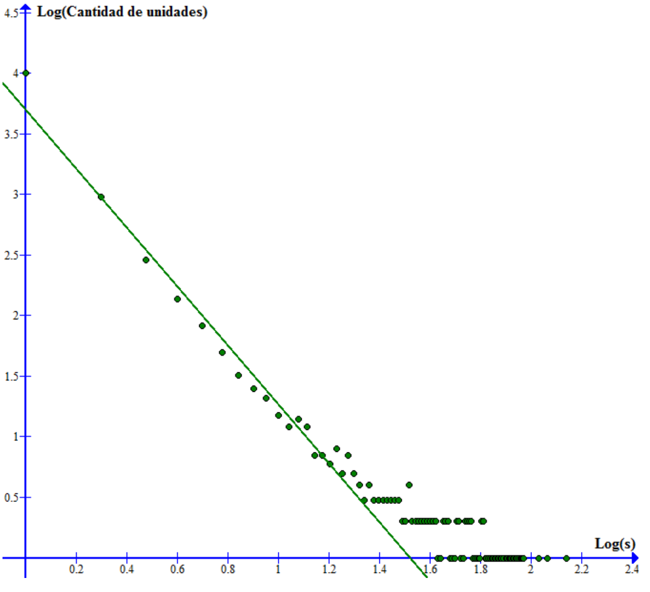

FIGURA 5. *Resultados de la simulación con* $\nu = 0{,}3$. *La ecuación de la recta es* $y = -2{,}43x + 3{,}70$.

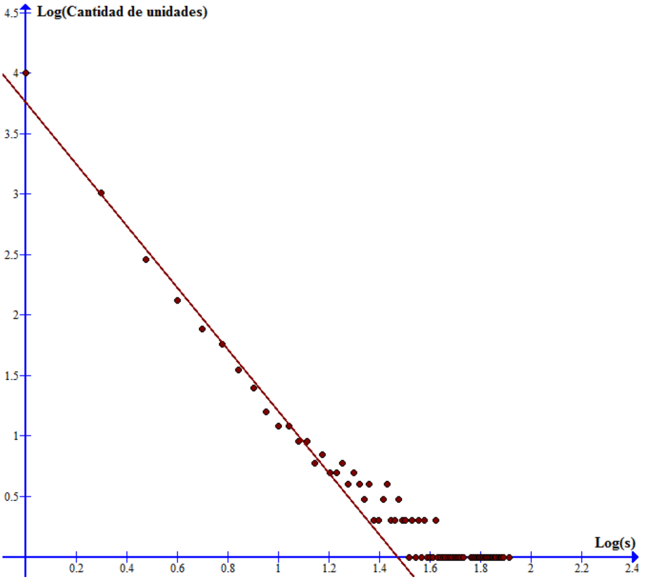

FIGURA 6. *Resultados de la simulación con* $\nu = 0{,}4$. *La ecuación de la recta es* $y = -2{,}55x + 3{,}76$.

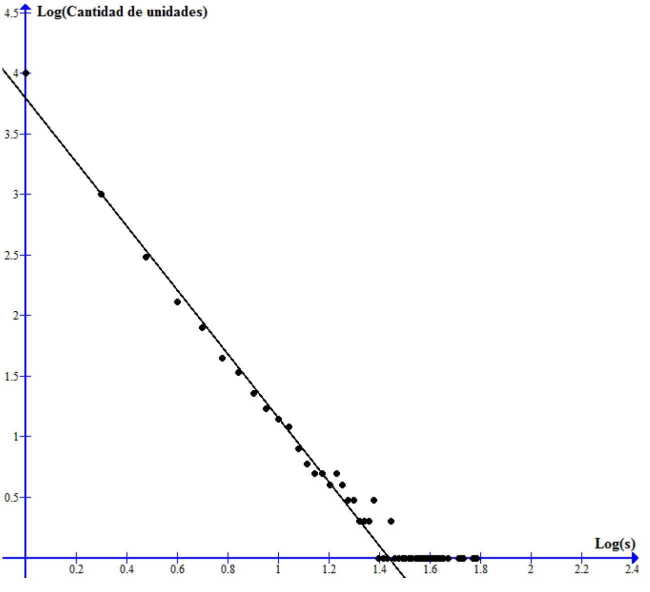

FIGURA 7. *Resultados de la simulación con* $\nu = 0{,}5$. *La ecuación de la recta es* $y = -2{,}64x + 3{,}79$.

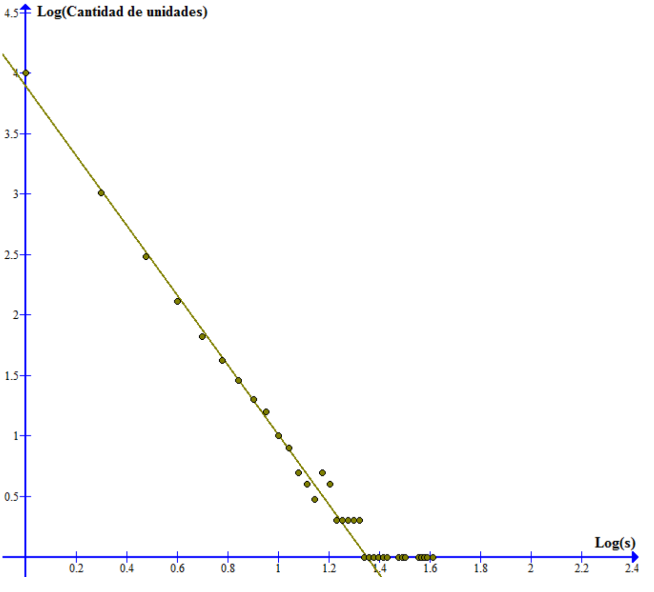

FIGURA 8. *Resultados de la simulación con $\nu = 0{,}6$. La ecuación de la recta es $y = -2{,}88x + 3{,}89$.*

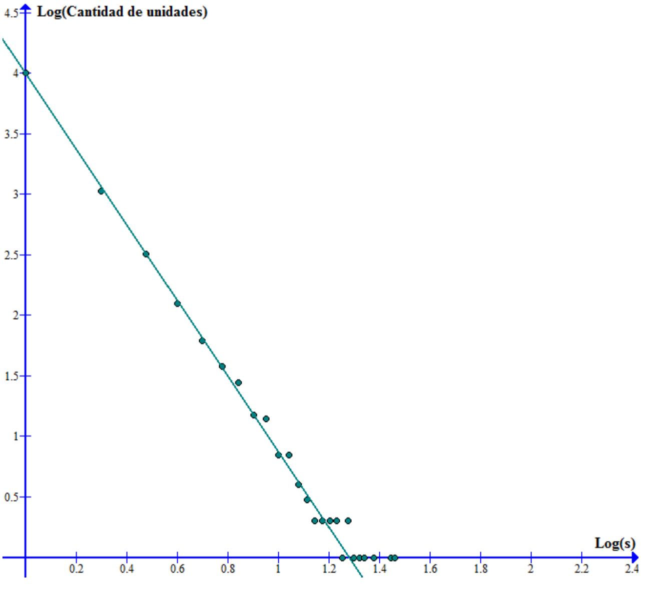

FIGURA 9. *Resultados de la simulación con $\nu = 0{,}7$. La ecuación de la recta es $y = -3{,}12x + 4{,}00$.*

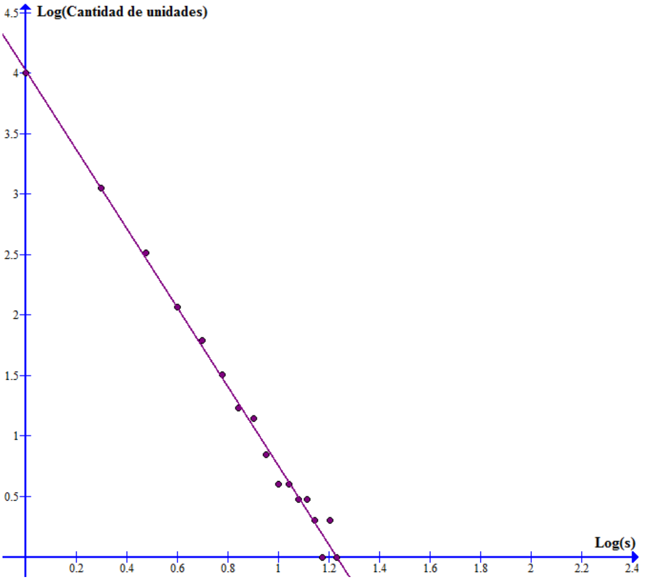

FIGURA 10. *Resultados de la simulación con* $\nu = 0{,}8$. *La ecuación de la recta es* $y = -3{,}27x + 4{,}03$.

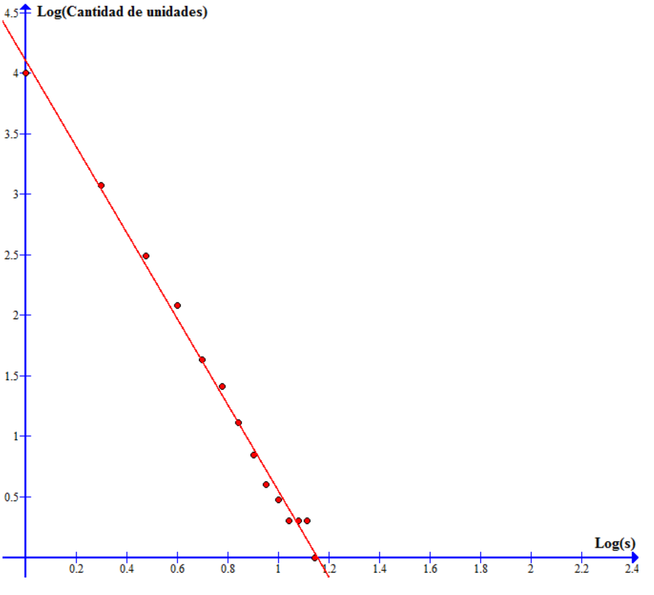

FIGURA 11. *Resultados de la simulación con* $\nu = 0{,}9$. *La ecuación de la recta es* $y = -3{,}55x + 4{,}11$.