



UNIVERSIDAD DE SALAMANCA
MÁSTER UNIVERSITARIO EN MODELIZACIÓN MATEMÁTICA

Diseño de un modelo de combate de naturaleza individual

AUTOR: Marcos López De Castro

TUTOR: Dr. Ángel Martín Del Rey

TUTOR: Dr. Guillermo Sánchez León

Salamanca, Julio 2021

Marcos López De Castro

Diseño de un modelo de combate de naturaleza individual

TRABAJO DE FIN DE MÁSTER

Curso 2020-2021



**VNiVERSiDAD
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

Máster Universitario en Modelización Matemática
Universidad de Salamanca

Memoria presentada como parte de los requisitos para la obtención del título de Máster en Modelización Matemática por la Universidad de Salamanca. El Dr. D. Ángel Martín Del Rey y el Dr. D. Guillermo Sánchez León autorizan la entrega del Trabajo de Fin de Máster titulado *Diseño de un Modelo de Combate de Naturaleza Individual*, realizado bajo su dirección por el alumno D. Marcos López de Castro.

Prof. Tutor Ángel Martín Del Rey.

Prof. Tutor Guillermo Sánchez León.



Resumen

En este trabajo de tesis de Máster se dará a conocer cuales han sido los modelos de combate utilizados históricamente para el modelado de batallas basados en sistemas de ecuaciones diferenciales, haciendo principal incapié en los modelos desarrollados originalmente por Lanchester debido a su carácter pionero en el campo. Tras ello se argumentará por qué los modelos basados en individuos pueden ser una herramienta más apropiada que los sistemas de ecuaciones diferenciales para modelar este tipo de interacciones y finalmente se desarrollará e implementará computacionalmente un modelo de combate basado en un autómata celular. El lenguaje de programación empleado para la implementación computacional ha sido *Wolfram Mathematica*.

Palabras clave

Modelos de combate, Autómatas celulares, Modelos basados en individuos, Modelos de Lanchester.

Abstract

In this Master's thesis, the combat models used historically for modeling battles based on systems of differential equations will be shown, with special emphasis on the models initially developed by Lanchester due to their pioneering nature in the field. After that, it will be argued why individual-based models may be a more appropriate tool than systems of differential equations to simulate this type of interactions. Finally, a model based on cellular automaton will be designed and computationally implemented. The programming language used for the computational implementation has been *Wolfram Mathematica*.

KeyWords

Combat models, Cellular automaton, Individual based models, Lanchester's Laws.

Índice general

| | |
|---|-----------|
| I Estado del Arte | 1 |
| Introducción | 3 |
| 1. Los modelos de Lanchester | 5 |
| 1.1. Modelo de Fuego dirigido | 5 |
| 1.1.1. Descripción y ecuaciones del modelo | 5 |
| 1.1.2. Condiciones de victoria en el modelo de Fuego Dirigido | 6 |
| 1.1.3. Situación tras la victoria | 7 |
| 1.1.4. Resolución numérica del modelo | 8 |
| 1.2. Modelo de Guerrilla o de Fuego de área | 9 |
| 1.2.1. Descripción y ecuaciones del modelo | 9 |
| 1.2.2. Condiciones de victoria | 10 |
| 1.2.3. Situación tras la victoria | 11 |
| 1.2.4. Resolución numérica del modelo | 12 |
| 2. Modelización basada en el individuo | 13 |
| 2.1. Introducción a los modelos individuales | 14 |
| 2.1.1. Diseño de IBMs | 15 |
| 2.1.2. Sistemas complejos | 16 |
| 2.2. Teoría básica de modelización mediante autómatas celulares | 17 |
| 2.3. Teoría básica de redes complejas | 19 |
| II Propuesta de modelo individual | 23 |
| 3. Diseño de un modelo de combate individual | 25 |
| 3.1. Planteamiento | 25 |
| 3.1.1. Fase de Combate. | 27 |
| 3.1.2. Fase de Movimiento | 34 |
| 3.2. Simulaciones del modelo | 39 |
| 3.2.1. Condiciones iniciales | 39 |
| 3.2.2. Escenario con distribución inicial de unidades aleatoria | 40 |
| 3.2.3. Escenario de emboscada | 53 |
| Conclusiones y trabajo futuro | 59 |
| Bibliografía | 63 |

| | |
|---|-----------|
| A. Modelos de combate Lanchesterianos. Ampliación | 65 |
| A.1. Ampliación al modelo de fuego dirigido | 65 |
| A.1.1. Táctica a seguir | 65 |
| A.2. Modelo Mixto o de Emboscada | 67 |
| A.2.1. Descripción y ecuaciones del modelo | 67 |
| A.2.2. Condiciones de victoria en el modelo Mixto | 68 |
| A.2.3. Situación tras la victoria | 68 |
| A.3. Modelo logarítmico | 69 |
| A.3.1. Descripción y ecuaciones del modelo | 69 |
| A.3.2. Condiciones de victoria en el modelo Logarítmico | 69 |
| A.3.3. Situación tras la victoria | 70 |
| A.4. Modelo Universal de Lanchester | 70 |
| A.4.1. Descripción y ecuaciones del modelo | 70 |
| A.4.2. Condiciones de victoria en el modelo Universal de Lanchester | 71 |
| A.4.3. Situación tras la victoria | 72 |
| A.5. Modelo de Morse y Kimball | 72 |
| A.5.1. Descripción y ecuaciones del modelo | 72 |
| B. Código de Mathematica desarrollado | 75 |

Índice de figuras

| | |
|--|----|
| 1.1. Modelo de fuego dirigido. | 8 |
| 1.2. Modelo de guerrilla. | 12 |
| 3.1. Distintos escenarios de conflicto. | 29 |
| 3.2. Ángulos que forman las unidades enemigas de la Figura 3.1 | 30 |
| 3.3. Distintos escenarios de conflicto. | 31 |
| 3.4. Unidades verdes y rojas en un espacio celular con $n = m = 10$ | 35 |
| 3.5. Redes complejas obtenidas para el ejército verde y rojo de la Figura (3.4). | 35 |
| 3.6. Unidades verdes y rojas distribuidas aleatoriamente en un espacio celular con $n = m = 13$. Disposición en el instante de tiempo $t = 1$ | 39 |
| 3.7. 50 unidades rojas emboscan a 50 unidades verdes en un espacio celular con $n = m = 13$. Disposición en el instante de tiempo $t = 1$ | 39 |
| 3.8. Resultados de 10 simulaciones independientes para un mismo número de unidades en el escenario de distribución inicial aleatoria. | 40 |
| 3.9. Resultados de 10 simulaciones independientes para un 10 % más de unidades verdes en el escenario de distribución inicial aleatoria. | 41 |
| 3.10. Resultados de 10 simulaciones independientes para un 20 % más de unidades verdes en el escenario de distribución inicial aleatoria. | 42 |
| 3.11. Resultados de 10 simulaciones independientes para un mismo número de unidades en el escenario de distribución inicial aleatoria. | 42 |
| 3.12. Resultados de 10 simulaciones independientes para un 10 % más de unidades verdes en el escenario de distribución inicial aleatoria. | 43 |
| 3.13. Resultados de 10 simulaciones independientes para un 20 % más de unidades verdes en el escenario de distribución inicial aleatoria. | 43 |
| 3.14. Resultados de 10 simulaciones independientes para un mismo número de unidades en el escenario de distribución inicial aleatoria. | 44 |
| 3.15. Resultados de 10 simulaciones independientes para un 10 % más de unidades verdes en el escenario de distribución inicial aleatoria. | 44 |
| 3.16. Resultados de 10 simulaciones independientes para un 20 % más de unidades verdes en el escenario de distribución inicial aleatoria. | 45 |
| 3.17. Evolución de 250 unidades verdes y 250 unidades rojas en un espacio celular con $n = m = 25$ | 46 |
| 3.18. 250 unidades verdes y 250 unidades rojas en un espacio celular con $n = m = 25$ en $t = 1$ | 47 |
| 3.19. Red compleja obtenida y su distribución de grado para el ejército verde de la Figura 3.18. | 47 |

| | |
|--|----|
| 3.20. Red compleja obtenida y su distribución de grado para el ejército rojo de la Figura 3.18. | 48 |
| 3.21. Unidades verdes y unidades rojas en un espacio celular con $n = m = 25$ en $t = 10$ | 48 |
| 3.22. Redes complejas obtenidas y su distribución de grado para el ejército verde y rojo de la Figura 3.21. | 49 |
| 3.23. Unidades verdes y unidades rojas en un espacio celular con $n = m = 25$ en $t = 20$ | 49 |
| 3.24. Redes complejas obtenidas y su distribución de grado para el ejército verde y rojo de la Figura 3.23. | 50 |
| 3.25. Unidades verdes y unidades rojas en un espacio celular con $n = m = 25$ en $t = 40$ | 50 |
| 3.26. Redes complejas obtenidas y su distribución de grado para el ejército verde y rojo de la Figura 3.25. | 51 |
| 3.27. Unidades verdes y unidades rojas en un espacio celular con $n = m = 25$ en $t = 80$ | 51 |
| 3.28. Redes complejas obtenidas para el ejército verde y rojo de la Figura 3.27. | 52 |
| 3.29. Unidades verdes y unidades rojas en un espacio celular con $n = m = 25$ en $t = 100$ | 52 |
| 3.30. Redes complejas obtenidas y su distribución de grado para el ejército verde y rojo de la Figura 3.29. | 53 |
| 3.31. Resultados de 15 simulaciones independientes para un mismo número de unidades en el escenario de emboscada. | 54 |
| 3.32. Resultados de 15 simulaciones independientes para un 10 % más de unidades verdes en el escenario de emboscada. | 54 |
| 3.33. Resultados de 15 simulaciones independientes para un 10 % más de unidades rojas en el escenario de emboscada. | 55 |
| 3.34. Modelo de fuego dirigido. | 56 |
| 3.35. Modelo de guerilla. | 57 |

Parte I
Estado del Arte

Introducción

En 1916, Frederic Lanchester publicó un libro titulado *Aircraft in Warfare* [20] que consistía principalmente en una recopilación de artículos enviados desde 1914 al *British Journal Engineering*. En uno de los artículos del libro, Lanchester incluyó un sistema de ecuaciones diferenciales ordinarias que pretendían modelizar la dinámica de una batalla entre dos oponentes. Para ello se inspiró en el desgaste y agotamiento de los soldados en los combates aéreos de la Primera Guerra Mundial.

Desde entonces hasta nuestros días se han seguido publicando trabajos relacionados con el tema como, por ejemplo: el de Morse y Kimball [23], el de Brackney [10] o el de Kress [5]. Cabe citar que en 1956 se publicó el artículo tomado del libro de Lanchester de una manera mucho más legible y simple, siendo este el artículo que se suele citar en la bibliografía debido a la dificultad de encontrar el original [21].

Los modelos matemáticos aplicados al combate van desde formas muy simplificadas hasta modelos sumamente complejos, lo cual dependerá de los factores involucrados en el combate o del nivel de realismo que pretendamos reproducir con el modelo. En el caso de los modelos de Lanchester estos se engloban dentro de los modelos compartimentales, es decir, la “población” se divide en diferentes grupos, cada uno de los cuales representa a uno de los dos ejércitos. La población de estos grupos (soldados) dependerá de una única variable independiente: el tiempo. Estos modelos son deterministas, su dinámica viene regida por un sistema de ecuaciones diferenciales y su objetivo es el de predecir el desenlace de un combate. A pesar de que los primeros modelos que se desarrollaron presentan formas muy simples de ecuaciones diferenciales, lo cierto es que pueden llegar a aportar una serie de datos muy importantes, los cuales pueden ser un factor clave en la toma de decisiones o en la preparación discreta de un conflicto. A partir de los modelos originales de Lanchester se desarrollaron multitud de modelos basados en ellos, los denominados modelos Lanchesterianos, siendo los principales y más influyentes los siguientes: Modelo Logarítmico, Modelo Mixto, Modelo Universal de Lanchester y el Modelo de Morse y Kimball.

Los modelos de combate clásicos se basan en las características promedio de las fuerzas en combate para llevar a cabo un análisis de la dinámica de las batallas, lo que puede resultar acertado para modelizar situaciones en las que el número y la homogeneidad de las unidades en liza lo permitan. Sin embargo, si observamos con detalle este tipo de fenómenos vemos que no son capaces de tener en cuenta la distribución de las partes en conflicto ni la coordinación entre las unidades que forman los diferentes bandos, cuestiones que son de gran relevancia a la hora de perseguir el perfeccionamiento de las capacidades

de combate. Aquí entran en juego los modelos basados en individuos, es decir, modelos cuyo objetivo es el de modelizar un fenómeno o sistema a partir de las interacciones de las unidades que los forman. Dentro de este tipo de modelos, se encuentran los modelos basados en autómatas celulares. Los autómatas celulares permiten modelizar y simular sistemas y fenómenos de una manera sencilla, cuyo nivel de complejidad es demasiado elevado como para emplear técnicas basadas en ecuaciones diferenciales o en derivadas parciales.

Los autómatas celulares surgen de la mano de John Von Neuman en 1948 dentro del problema de la vida artificial y adquirieron una gran popularidad gracias al matemático J. H. Conway y su *Juego de la vida*. Fue a partir de entonces cuando se comenzó a estudiar seriamente su potencial para la simulación de problemas de carácter físico, biológico, etc.

El objetivo principal de este trabajo será el de desarrollar e implementar un modelo de combate basado en interacciones individuales. Para ello se empleará una arquitectura basada en un autómata celular. La idea de emplear este tipo de arquitectura para modelizar un combate surge de la aparente poca bibliografía existente sobre este tipo de modelos basados en autómatas celulares, (véase [18]), ya que habitualmente se suelen emplear como sustituto de los sistemas de ecuaciones diferenciales modelos basados en agentes.

Previo a la construcción del modelo, analizaremos en el capítulo 1 los modelos originales de Lanchester, los cuales también se conocen bajo el nombre *Las leyes de Lanchester*. El objetivo del análisis de estos modelos no es otro que el suministrar al lector una base con la que obtener una mayor perspectiva sobre los modelos de combate y poder así analizar las diferencias entre estos modelos clásicos basados en sistemas de ecuaciones diferenciales y los basados en interacciones individuales. Tras ello se argumentará en el capítulo 2 por que una propuesta de modelo basado en interacciones individuales puede ser más realista que un sistema basado en ecuaciones diferenciales para modelar este tipo de fenómenos y finalmente, en el capítulo 3, se desarrollará la propuesta de modelo individual implementando un autómata celular que simule este tipo de situaciones a partir de reglas matemáticas sencillas junto con la ayuda de herramientas más sofisticadas, como lo son las redes complejas, de manera que se pueda caracterizar los comportamientos individuales y a partir de las cuales podamos observar comportamientos no triviales entre las unidades que forman los ejércitos. Es también objetivo de este trabajo que el modelo desarrollado sirva como base para futuros modelos más ambiciosos que permitan explorar posibles variaciones del curso de un combate, dependiendo de la situación en la que este se encuentre y con la ayuda de otro tipo de herramientas o tecnologías que quedan fuera de los objetivos de este trabajo.

Capítulo 1

Los modelos de Lanchester

En este capítulo estudiaremos en detalle los aspectos analíticos de los modelos propuestos originalmente por Lanchester. Los del resto de modelos desarrollados posteriormente se explican brevemente en el Apéndice A. Para una colección completa de este tipo de modelos clásicos véase el trabajo realizado por J. G. Taylor [26, 27] para el ejército de los EEUU en el cual no solo se hace una extensa recopilación de los modelos Lanchesterianos hasta 1980, sino que se discuten ampliamente los aspectos matemáticos de los mismos además de otro tipo de problemas característicos de este tipo de modelos, como el cálculo realista de coeficientes o de la correcta modelización de las últimas fases del combate.

1.1. Modelo de Fuego dirigido

1.1.1. Descripción y ecuaciones del modelo

El llamado *modelo de fuego dirigido* fue desarrollado por Lanchester [21] y pretende describir el combate entre dos fuerzas homogéneas las cuales representan dos ejércitos (en nuestro caso las distinguiremos como ejército rojo y ejército verde) y cuyo tamaño en cualquier instante de tiempo t viene dado por $R(t) > 0$ y $V(t) > 0$. Estos oponentes comienzan su lucha en un instante de tiempo $t = 0$. A medida que se avanza en el tiempo, cada uno destruye un número determinado de unidades con unas tasas de detección r y v respectivamente y siendo éstos los únicos factores que influyen en el modelo. Así obtenemos:

$$\left\{ \begin{array}{l} \frac{dR}{dt} = -vV \\ \frac{dV}{dt} = -rR \end{array} \right. \quad \text{siendo } R(0) = R_0, \quad V(0) = V_0. \quad (1.1)$$

Como se puede comprobar en el sistema de ecuaciones (1.1), Lanchester asume que la variación de cada una de las partes de la igualdad es proporcional a la otra parte, es decir, el poder del ejército R se representa con el producto vV , al igual que el poder de V se representa por el producto rR , lo que significa que el poder de ambas fuerzas disminuirá a medida que una fuerza aniquile a la otra. $\frac{dR}{dt}$ representa el cambio de la variable R en

función del tiempo y es proporcional al poder de V . Por el contrario, $\frac{dV}{dt}$ representa el cambio de la variable V en función del tiempo y es proporcional al poder de R . Es posible calcular la solución implícita del problema de valor inicial dado en (1.1):

$$rR^2 - vV^2 = rR_0^2 - vV_0^2. \quad (1.2)$$

Esta solución nos indica que independientemente del transcurrir del tiempo, la magnitud del lado izquierdo de la igualdad, que representa la diferencia $rR^2 - vV^2$, se tiene que mantener constante e igual a la dada por la del lado derecho de la ecuación (1.2). A continuación se expone la solución explícita del sistema de ecuaciones (1.1) :

$$R(t) = \frac{1}{2} \left(R_0 - \sqrt{\frac{v}{r}} V_0 \right) e^{\sqrt{vr}t} + \frac{1}{2} \left(R_0 + \sqrt{\frac{v}{r}} V_0 \right) e^{-\sqrt{vr}t}, \quad (1.3)$$

$$V(t) = \frac{1}{2} \left(V_0 - \sqrt{\frac{r}{v}} R_0 \right) e^{\sqrt{vr}t} + \frac{1}{2} \left(V_0 + \sqrt{\frac{r}{v}} R_0 \right) e^{-\sqrt{vr}t}, \quad (1.4)$$

donde \sqrt{vr} mide en cierta manera la intensidad de la batalla y nos permite definir un tiempo propio de la batalla:

$$\tau = \frac{1}{\sqrt{vr}}. \quad (1.5)$$

Los coeficientes $\sqrt{\frac{v}{r}}$ y su inverso $\sqrt{\frac{r}{v}}$ miden la efectividad relativa de ambos ejércitos con respecto a su adversario.

1.1.2. Condiciones de victoria en el modelo de Fuego Dirigido

Una vez resuelto el modelo, veamos qué condiciones han de cumplirse para vencer en la batalla. Para ello utilizaremos las soluciones implícita y explícita halladas en el apartado anterior. Teniendo en cuenta la solución implícita:

$$rR^2 - vV^2 = rR_0^2 - vV_0^2, \quad (1.6)$$

podemos suponer que el ejército rojo ganará en el instante de tiempo más pequeño t_f en el que se cumpla: $R(t_f) > 0$ y $V(t_f) = 0$. Sustituyendo en la solución implícita llegamos a la siguiente condición:

$$R(t_f) = \sqrt{R_0^2 - \frac{v}{r} V_0^2} > 0. \quad (1.7)$$

Para calcular el valor de $R(t_f)$ tomaremos sólo el valor positivo de la raíz dado que el valor negativo no tiene sentido real ya que no existe un número negativo de soldados pertenecientes a un ejército y además hemos supuesto que es mayor que cero. En consecuencia, el ejército rojo gana cuando $R_0^2 - \frac{v}{r} V_0^2 > 0$, de donde podemos obtener dos relaciones:

1. En primer lugar, podemos analizar la situación previa al combate:

$$R_0^2 - \frac{v}{r} V_0^2 > 0 \Rightarrow \frac{R_0^2}{V_0^2} > \frac{v}{r} \Rightarrow \frac{R_0}{V_0} > \sqrt{\frac{v}{r}}. \quad (1.8)$$

Si asumimos que las unidades que forman las distintas fuerzas en disputa están bien entrenadas, igualmente equipadas y en buenas o equivalentes condiciones de combate, modificar la parte derecha (efectividad relativa) de la inecuación anterior es

difícil. Esta condición es útil en un escenario previo al combate ya que las variables que intervienen son el número inicial de unidades de cada ejército R_0 y V_0 , sin embargo, la importancia de esta relación radica en que cualquier cambio en la ratio del número de fuerzas iniciales afecta cuadráticamente al balance de la inecuación. Es esta la razón por la cual es tan importante lograr una relación de fuerzas favorable.

2. La situación anterior nos da información en una situación previa al combate, sin embargo, no es útil en el transcurso de la batalla. Si nos valemos de la ecuación (1.2):

$$R_0^2 - \frac{v}{r}V_0^2 > 0 \Rightarrow rR_0^2 - vV_0^2 = rR^2 - vV^2 > 0 \Rightarrow rR^2 > vV^2. \quad (1.9)$$

Esta condición es muy similar a la anterior diferenciándose en que no está valorada en un tiempo inicial. Esto es, que si en algún momento de la batalla se hace una observación de la situación recurriríamos a esta inecuación ya que el valor inicial no sería útil utilizarlo.

Un estudio similar a este se haría en caso de que el ejército verde ganase, obteniendo como condiciones de victoria para el ejército verde:

$$rR^2 < vV^2. \quad (1.10)$$

Además, es sencillo ver que el ejército rojo y el ejército verde empatan si se cumple

$$rR^2 = vV^2. \quad (1.11)$$

1.1.3. Situación tras la victoria

La batalla finaliza cuando uno de los dos ejércitos es aniquilado. Parece lógico tratar de resolver las dos siguientes preguntas:

1. *¿Cuál es el tamaño final del ejército ganador?*
Supongamos que gana el ejército rojo, es decir, para un tiempo suficientemente grande el ejército verde es destruido, verificándose:

$$\lim_{t \rightarrow \infty} (V(t)) = 0, \quad (1.12)$$

luego tomando los límites en la expresión implícita (1.2) obtenemos:

$$\lim_{t \rightarrow \infty} (R(t)) = \sqrt{R_0^2 - \frac{v}{r}V_0^2}, \quad (1.13)$$

que es el tamaño del ejército rojo tras ganar la batalla. En el caso de que el vencedor fuera el ejército verde obtendríamos:

$$\lim_{t \rightarrow \infty} (V(t)) = \sqrt{V_0^2 - \frac{r}{v}R_0^2}. \quad (1.14)$$

2. ¿En qué instante de tiempo se logra la victoria?

Seguimos suponiendo sin pérdida de generalidad que gana el ejército rojo. La ecuación (1.13) nos proporciona el tamaño del ejército rojo al finalizar la batalla en caso de victoria, teniendo por tanto el valor de la función $R(t)$ en el instante final de la batalla t_f de modo que:

$$R(t_f) = \sqrt{R_0^2 - \frac{v}{r}V_0^2}. \quad (1.15)$$

Utilizando la solución explícita del ejército rojo (1.3) obtenemos la siguiente expresión para t_f :

$$t_f = \frac{1}{2} \frac{1}{\sqrt{vr}} \log \left[\frac{R_0 + \sqrt{\frac{v}{r}V_0}}{R_0 - \sqrt{\frac{v}{r}V_0}} \right] = \frac{\tau}{2} \log \left[\frac{R_0 + \sqrt{\frac{v}{r}V_0}}{R_0 - \sqrt{\frac{v}{r}V_0}} \right]. \quad (1.16)$$

En ese instante de tiempo las unidades verdes son aniquiladas y las rojas habrán perdido $1 - R(t_f) = 1 - \sqrt{R_0^2 - \frac{v}{r}V_0^2}$ unidades.

El caso en el que el ejército verde resulta ganador es completamente análogo; Si el ejército verde es el vencedor, se verificará la siguiente relación:

$$V(t_f) = \sqrt{V_0^2 - \frac{r}{v}R_0^2}. \quad (1.17)$$

Utilizando la ecuación explícita del ejército verde (1.4) obtenemos para t_f :

$$t_f = \frac{1}{2} \frac{1}{\sqrt{vr}} \log \left[\frac{V_0 + \sqrt{\frac{r}{v}R_0}}{V_0 - \sqrt{\frac{r}{v}R_0}} \right] = \frac{\tau}{2} \log \left[\frac{V_0 + \sqrt{\frac{r}{v}R_0}}{V_0 - \sqrt{\frac{r}{v}R_0}} \right]. \quad (1.18)$$

1.1.4. Resolución numérica del modelo

Para finalizar la sección se expondrán algunos ejemplos gráficos de la evolución predicha, para diversos escenarios de conflicto por el modelo de fuego dirigido con el objetivo de afianzar la comprensión por parte del lector. La resolución del modelo se hará mediante las técnicas numéricas de las que dispone *Wolfram Mathematica* mediante el comando *NDSolve*.

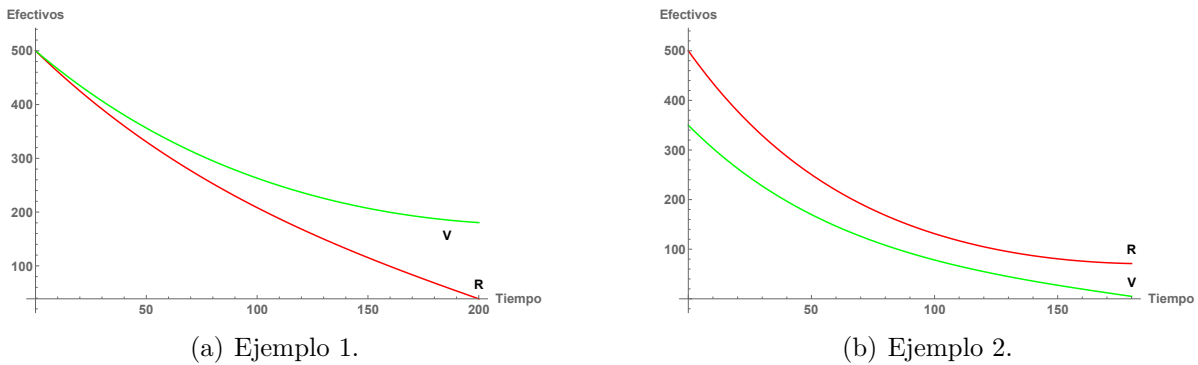


Figura 1.1: Modelo de fuego dirigido.

En la Figura 1.2.4-(a) se muestran los resultados del modelo de fuego dirigido con 500 unidades del ejército verde y 500 unidades del ejército rojo desplegadas inicialmente. Además, la potencia de fuego de las unidades verdes será ligeramente superior a la de las unidades rojas: $v = 0,008$ y $r = 0,007$. Por el contrario, en la Figura 1.2.4-(b) se despliegan 500 unidades del ejército rojo y 350 del ejército verde inicialmente con una potencia de fuego del doble por parte de las unidades verdes: $v = 0,02$ y $r = 0,01$.

1.2. Modelo de Guerrilla o de Fuego de área

1.2.1. Descripción y ecuaciones del modelo

El llamado *modelo de fuego de área* también fue propuesto por el propio Lanchester y al igual que en el modelo anterior, se describe el combate entre dos fuerzas homogéneas, sin embargo, lo que Lanchester pretendía ahora modelizar son las que el denominaba “Ancient Wars”. La idea en la que se basó Lanchester para proponer este modelo es la siguiente: Supongamos que una fuerza de guerrilla permanece oculta y fija en una cierta región fija Q . Las fuerzas enemigas, disparan sobre la región Q ya que saben que las fuerzas de guerrilla enemigas permanecen ocultas en esa región, sin embargo no saben cuándo han conseguido abatir a algún enemigo porque no saben exactamente dónde están.

Lanchester argumentó que es lógico pensar que en este modelo, el ritmo de pérdidas en combate de los ejércitos sea proporcional al número de fuerzas ocultas, ya que, cuantas más sean estas más posibilidades hay de que el enemigo haga blanco en ellas. Además, al igual que ocurría en el modelo anterior, el ritmo de bajas de ambos ejércitos también será proporcional al número de fuerzas enemigas que están disparando sobre Q .

Consecuentemente en este modelo los coeficientes v y r son más complicados de estimar que en el modelo de fuego dirigido. Una propuesta razonable para ellos es la dada por el propio Lanchester:

$$r = -r_r \frac{A_{rr}}{A_v}, \quad v = -r_v \frac{A_{rv}}{A_r}, \quad (1.19)$$

dónde r_r y r_v representan las ratios de disparo (disparos/combatiente/día), A_r y A_v representan el área en la que esta oculta cada guerrilla y A_{rr} y A_{rv} el área donde están los guerrilleros.

Uno de los factores que más influyen, ya que se supone que las capacidades de fuego y coordinación son suficientes para eliminar a todas las fuerzas enemigas, es la capacidad de detectar a las guerrillas. Estas suposiciones pueden resultar un tanto irreales ya que es complicado identificar situaciones de combate en las que la detección sea el factor clave.

Teniendo en cuenta lo comentado anteriormente, y suponiendo que $R(t)$ y $V(t)$ representan el número de efectivos de las guerrillas roja y verde respectivamente en el instante de tiempo t , la dinámica de este modelo viene gobernada por el siguiente sistema de

ecuaciones diferenciales ordinarias:

$$\left\{ \begin{array}{l} \frac{dR}{dt} = -vVR, \\ \frac{dV}{dt} = -rRV, \end{array} \right. \quad \text{siendo } R(0) = R_0, V(0) = V_0. \quad (1.20)$$

Como se puede observar ahora tenemos un modelo no lineal dónde la variación del tamaño de las respectivas fuerzas en conflicto viene dada por una ley de acción de masas cuya constante es la tasa de detección correspondiente. De manera similar al modelo anterior, es posible calcular la solución implícita o ecuación de estado del anterior problema de valor inicial: obtenemos:

$$rR(t) - vV(t) = rR_0 - vV_0. \quad (1.21)$$

Al igual que en el modelo de fuego dirigido, el segundo miembro de la solución implícita (1.21) es constante con respecto al tiempo. Finalmente, las soluciones explícitas del sistema (1.20) son:

$$R(t) = \frac{(vV_0 - rR_0) R_0}{vV_0 e^{(vV_0 - rR_0)t} - rR_0}, \quad (1.22)$$

$$V(t) = \frac{(vV_0 - rR_0) V_0}{vV_0 - rR_0 e^{-(vV_0 - rR_0)t}}. \quad (1.23)$$

1.2.2. Condiciones de victoria

Al igual que en el modelo anterior se considerará que el combate finalizará en el instante de tiempo t_f en el que una de las dos guerrillas quede aniquilada. Supongamos, sin pérdida de generalidad, que es la guerrilla verde la que queda derrotada y la roja la que resulta vencedora, entonces existirá un t_f - dónde t_f es el instante de tiempo más pequeño en el que ocurre esto - tal que $V(t_f) = 0$ y $R(t_f) > 0$. La condición de victoria para el ejército rojo la obtenemos a partir de la expresión (1.21):

$$R(t_f) = \frac{rR_0 - vV_0}{r}, \quad (1.24)$$

y análogamente, en caso de que sea la guerrilla verde la que resulte victoriosa:

$$V(t_f) = -\frac{rR_0 - vV_0}{v}. \quad (1.25)$$

En el caso de las soluciones explícitas obtenidas (1.22) - (1.23) debemos de relajar la condición de victoria entendida como la aniquilación del enemigo, pues ninguna de ellas alcanza de manera efectiva el valor cero, excepto en el límite de tiempos suficientemente grandes. Puesto que el número de efectivos de ambos bandos no puede ser negativo, ya que dicha situación no tendría sentido físico, podemos deducir de la relación (1.24) que cuando la guerrilla roja gana se cumple la siguiente condición:

$$rR_0 > vV_0, \quad (1.26)$$

mientras que a partir de la expresión (1.25) se deduce que en caso de que la guerrilla verde resulte victoriosa se debe cumplir:

$$vV_0 > rR_0. \quad (1.27)$$

A continuación, supongamos que se cumple la condición dada por (1.26). Si tomamos el límite para tiempos lo suficientemente grandes en las soluciones explícitas del problema (1.22) - (1.23), obtenemos:

$$\lim_{t \rightarrow \infty} R(t) = \frac{rR_0 - vV_0}{r}, \quad (1.28)$$

$$\lim_{t \rightarrow \infty} V(t) = 0. \quad (1.29)$$

Como se puede comprobar, la guerrilla roja resulta la vencedora para tiempos lo suficientemente grandes, luego podemos afirmar que si se cumple la condición (1.26), acabará ganando la guerrilla roja. Análogamente, si suponemos que se cumple la condición dada por (1.27) y tomamos el límite para tiempos lo suficientemente grandes en las soluciones explícitas, obtenemos:

$$\lim_{t \rightarrow \infty} R(t) = 0, \quad (1.30)$$

$$\lim_{t \rightarrow \infty} V(t) = \frac{vV_0 - rR_0}{v}, \quad (1.31)$$

luego la expresión (1.27) será condición de victoria de la guerrilla verde sobre la roja. Obsérvese que estas condiciones se basan en el conocimiento del estado inicial del sistema. Un razonamiento similar permitiría extraer condiciones de victoria que dependieran del estado en un instante determinado - no sólo el inicial -; de esa manera, la guerrilla roja ganaría cuando $rR(t) > vV(t)$ para cualquier instante de tiempo t mientras que la guerrilla verde ganaría cuando $vV(t) > rR(t)$.

1.2.3. Situación tras la victoria

Como se ha visto en el apartado anterior, el combate tarda un tiempo infinito en alcanzar las condiciones de victoria para alguno de los participantes en el enfrentamiento ya que, el cumplimiento de la condición de aniquilación es un resultado asintótico. El número de efectivos de la guerrilla superviviente a los enfrentamientos bajo la condición de aniquilación lo determinan las expresiones (1.24) en caso de victoria de la guerrilla roja y (1.25) en caso de victoria de la guerrilla verde. A partir de una de las soluciones explícitas del modelo (1.22) - (1.23) podemos determinar un tiempo τ en el cual las guerrillas verdes y rojas tendrán un determinado tamaño $R(\tau) = R_\tau$ y $V(\tau) = V_\tau$. Luego si evaluamos la expresión (1.22) para un tiempo τ y despejamos, obtenemos el siguiente resultado

$$\tau = \frac{1}{vV_0 - rR_0} \log \left[\frac{vV_0 + rR_\tau - rR_0}{vR_\tau} \left(\frac{R_0}{V_0} \right) \right]. \quad (1.32)$$

Y finalmente, a partir de la ecuación implícita o de estado del modelo (1.21) podemos relacionar R_τ y V_τ :

$$\frac{R_\tau}{R_0} = 1 - \frac{v}{r} \left(\frac{V_0}{R_0} + \frac{V_\tau}{R_0} \right). \quad (1.33)$$

1.2.4. Resolución numérica del modelo

Al igual que con el modelo anterior a continuación se expone la resolución numérica del modelo de guerrilla en un par de escenarios de conflictos para afianzar la comprensión del capítulo por parte del lector.

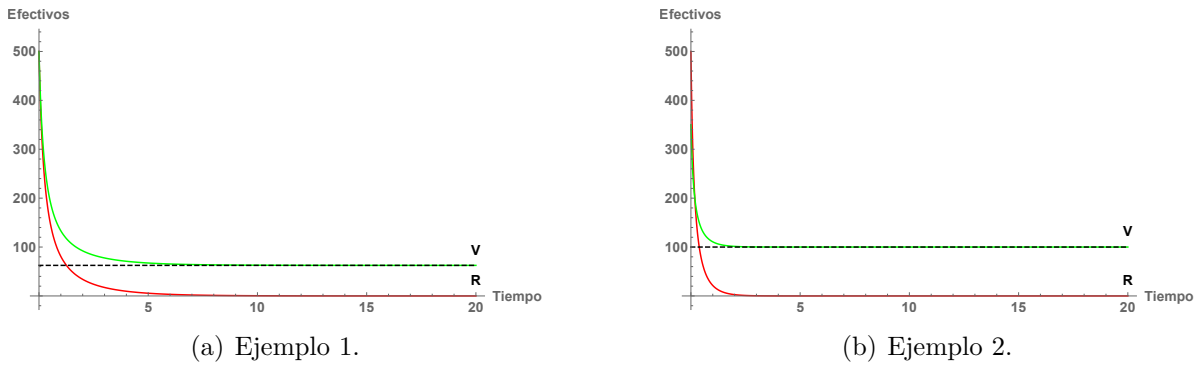


Figura 1.2: Modelo de guerrilla.

En la Figura 1.2-(a) y en la Figura 1.2-(b) resolvemos los mismos casos que en el modelo de guerrilla con exactamente las mismas condiciones iniciales y de potencia de fuego que los mostrados en la Figura , siendo la línea gris horizontal la predicción de las expresiones (1.32) y (1.33) para número final de efectivos del el ejército vencedor.

Capítulo 2

Modelización basada en el individuo

En la sección anterior hemos hablado de los modelos de combate tradicionalmente estudiados, fuertemente influenciados por los modelos originales de Lanchester, y los cuales están descritos en base a sistemas de ecuaciones diferenciales, sin embargo, la realidad refleja para este tipo de problemas una naturaleza mucho más compleja, caótica y basada en interacciones no lineales entre elementos heterogéneos, que se adaptan a condiciones cambiantes con el transcurso del tiempo y que difícilmente los modelos basados en las técnicas tradicionales pueden capturar. En este capítulo de carácter transitorio vamos a ver por qué para la modelización de ciertos fenómenos es más adecuado trabajar con herramientas que reflejen el comportamiento individual de los elementos o individuos en interacción que forman el sistema que se pretende estudiar. El caso de los modelos de combate no es una excepción, de hecho si ignoramos los detalles más íntimos de este tipo de procesos, que son necesarios para una descripción completa del fenómeno, podemos observar que muchos de los comportamientos se corresponden con los definidos por un conjunto simple de reglas dadas entre individuos vecinos, lo cual es susceptible de ser modelado por autómatas celulares. Woodcock et. al. [33], demostraron que se pueden obtener patrones de combate complejos a partir de un pequeño conjunto de reglas para un sistema de autómatas celulares. En el modelo de Woodcock, las reglas se pueden agrupar en cada una de las siguientes variedades: Situación de combate, movimiento a realizar, combate y control jerárquico. Además, este tipo de modelos permiten añadir otro tipo de factores como accidentes geográficos, proporcionar capacidad de reacción a las fuerzas en liza o la sustitución de unidades perdidas o desgastadas en el combate.

En palabras de Woodcock et. al. (véase [17]) el objetivo de este tipo de sistemas no es codificar y simular un conjunto de reglas de manera que el sistema sea lo más parejo a la realidad posible, sino que su objetivo debe ser :

« Encontrar el conjunto de reglas más simple de manera que ambos (ejércitos) puedan generar fenómenos globales de combate similares, no triviales y proporcionar una nueva comprensión del proceso de combate en sí mismo mediante la extracción de la cantidad máxima de complejidad conductual a partir del conjunto de reglas menos complicado ».

A continuación veremos las características básicas de los modelos basados en comportamientos individuales, se dará una introducción a la teoría de autómatas celulares que nos

permita tener una base y el lenguaje necesario para construir un modelo de combate de naturaliza individual en el siguiente capítulo y finalmente una pequeña noción de teoría de redes complejas debido a su utilidad en el análisis de sistemas complejos.

2.1. Introducción a los modelos individuales

Los modelos basados en individuos o IBM - Individual Based Models - son una de las herramientas empleadas a la hora de modelizar y resolver problemas en los que se encuentran individuos en interacción. Para este tipo de problemas los modelos tradicionales, como por ejemplo los basados en ecuaciones diferenciales, o bien no resuelven de manera adecuada el problema o bien simplifican en exceso el mismo obteniendo resultados pobres y que no reflejan la realidad del fenómeno que se pretende estudiar. Un ejemplo paradigmático de lo anterior lo dan Railsback y Grimm en su libro *Agent-Based and Individual-Based Modeling: a practical introduction* [28] para la enfermedad de la rabia: La rabia es una enfermedad que afecta a los mamíferos y puede provocar la muerte en seres humanos. En Europa, su principal vector de transmisión es el zorro rojo, por tanto, se puede controlar esta enfermedad inmunizando a estos animales. Los modelos epidemiológicos clásicos basados en ecuaciones diferenciales predicen que el control de la transmisión de esta población se alcanza a partir del 70% de zorros inmunizados, sin embargo, a partir del trabajo de Jeltsch et al. [4], en el cual se desarrolla un IBM para simular la propagación espacio-temporal de la rabia, y los posteriores desarrollos de Dirk Eisinger y Hans - Hermann adaptando dicho modelo para incluir el efecto de la vacunación en los individuos [2, 3], se observa que no solo no es necesario llegar a un porcentaje de inmunizados tan elevado para lograr la hipotética inmunidad frente a la rabia, lo que implica un ahorro potencial de millones de euros, sino que estrategias de vacunación cuyo éxito predicen los modelos clásicos, fallan. Los resultados de aplicar los modelos basados en interacciones individuales no solo estaban avalados, según los autores, por series de datos históricas, sino que cuando se aplicó la estrategia resultó ser un éxito.

En el ejemplo anterior, la razón por la cual la predicción de población inmunizada se ajusta mejor a partir de los IBMs es debido a que las infecciones se producen de manera local - por contacto- en poblaciones que están distribuidas espacialmente, factor que los modelos clásicos basados en ecuaciones diferenciales no tienen en cuenta. De manera general, una de las características principales que tienen los sistemas formados por individuos es que estos normalmente difieren entre sí e incluso de si mismos a lo largo de diferentes etapas temporales. Además, los individuos pueden tener la capacidad de adaptarse a condiciones cambiantes modificando su comportamiento, modificar el entorno, local o global, a través de sus acciones y suelen tomar sus propias decisiones. Los IBM tratan de capturar esto de manera que las propiedades observadas globalmente, por ejemplo, la inmunidad de los zorros rojos, emerjan a partir de las interacciones individuales y con el entorno.

En muchos campos a los IBMs también se les denomina modelos basados en agentes o ABM - Agent Based Model - sin hacer distinción entre ambos conceptos, sin embargo, podemos considerar diferencias históricas entre ambos términos: cuando hablamos de IBMs

nos solemos centrar en la modelización de las interacciones locales y en la variabilidad del estado del propio individuo, mientras que cuando hablamos de ABMs nos solemos centrar en la toma de decisiones del individuo/agente y en el comportamiento adaptativo del agente ante posibles condiciones cambiantes.

Una vez que la capacidad computacional lo permitió, el campo donde inicialmente se aplicaron este tipo de modelos basados en el individuo fue la ecología con trabajos como los de Botkin et al. [1], en el cual desarrollan un modelo forestal que permitió reproducir con éxito la dinámica poblacional de bosques con diferentes especies, o el modelo de Deangelis et al. [11] en el cual desarrollan un modelo de bancos de peces en los que incluyen los efectos del canibalismo y de la competición entre individuos. Sin embargo, la formalización de los IBMs vendrá de la mano del artículo propuesto por Huston et al. [16] en el cual argumentan por qué los modelos basados en variables de estado, desde los más simples como la ecuación logística a los más complejos, no son capaces de capturar los principios biológicos de la localidad de las interacciones entre los individuos y el hecho de que los individuos, aunque sean de una misma especie, son diferentes entre sí.

2.1.1. Diseño de IBMs

A la hora de desarrollar modelos basados en individuos, es de suma importancia saber capturar de forma adecuada los rasgos de los individuos, de manera que se pueda modelar el comportamiento del sistema u organismo real. Estos rasgos suelen ser adaptativos a la situación existente ya que permiten la toma de decisión dependiendo de la situación específica y genéricamente pueden ser o bien programados, es decir, reglas basadas en sentencias condicionales dentro de bucles o bien aprendidos, donde el uso de técnicas de inteligencia artificial (IA) y aprendizaje automático es de gran utilidad. Teniendo esto en cuenta existen tres aspectos fundamentales a considerar si se pretende desarrollar un IBM:

- Comportamiento de los individuos/agentes: Conjunto de reglas programadas o aprendidas para cada individuo a partir de las cuales surgirán patrones o comportamientos globales.
- Interacción entre individuos: Puede ser de manera directa, es decir, entre los propios individuos/agentes en interacción o de manera indirecta, donde se modifican las características del entorno. La dinámica del conjunto está fuertemente condicionada por este tipo de interacciones.
- El entorno: El entorno local representa el lugar con el cual el sistema u organismo interactúa y se mueve. Se suelen emplear mallas o redes para modelarlo. El entorno tiene una influencia muy importante tanto en el comportamiento de los individuos como en la dinámica del sistema.

Uno de los resultados más importantes que este tipo de modelos proporciona es el comportamiento macroscópico que emerge a partir de las interacciones individuales de los

individuos. La ventaja de los IBMs reside pues en su capacidad de simular comportamientos propios de los sistemas complejos a partir de reglas sencillas.

2.1.2. Sistemas complejos

Hasta ahora, hemos hablado cómo estos modelos basados en el individuo han sido aplicados principalmente a problemas de la ecología o de su posible aplicación a los modelos de combate - que es el objetivo de este trabajo - , sin embargo, la importancia de este tipo de modelos reside en su potencial aplicación a cualquier sistema en el que muchos elementos interaccionan entre sí de manera heterogénea, es decir, cuando nos referimos a individuos/agentes no tienen por qué ser elementos de naturaleza biológica, sino que pueden representar desde partículas en un gas o estrellas en una galaxia hasta vehículos en movimiento o conexiones entre neuronas. A este tipo de sistemas se les denomina genéricamente como *sistemas complejos*. A continuación, se ponen de relieve los rasgos comunes por los que se suelen caracterizar este tipo de sistemas (véase [17]):

- Gran número de elementos en interacción. Normalmente dichos elementos suelen ser heterogéneos entre sí y las interacciones entre ellos de naturaleza no lineal.
- Interacción a distintos niveles. Este tipo de sistemas suele tender a una organización jerárquica, es decir, las partes individuales pueden formar agregados que actúan de manera efectiva a un nivel superior al de los elementos que la forman.
- Múltiples puntos de equilibrio metaestables, es decir, son sistemas que alcanzan distintos puntos de equilibrio estables bajo pequeñas perturbaciones, pero grandes perturbaciones pueden provocar grandes alteraciones en el sistema.
- Los elementos que forman el sistema interaccionan localmente.
- Son sistemas que se autoorganizan. Cuando este tipo de sistemas evolucionan, macroscópicamente se suele observar estructuras organizadas fuera del equilibrio debido a las interacciones de los constituyentes del sistema, que reaccionan y se adaptan al entorno.
- Comportamiento emergente. En este tipo de sistemas pueden surgir patrones globales que no se deducen a partir del comportamiento o interacciones entre las partes que lo forman.
- La evolución de este tipo de sistemas la motiva una continua retroalimentación entre el estado macroscópico y los constituyentes microscópicos.
- El comportamiento de estos sistemas puede ser: ordenado, caótico, intermedio o transitar entre ellos. (Véase [22]).

A continuación, se expondrán algunas de las nociones básicas de los autómatas celulares, cuyo origen surge del estudio algorítmico de la vida y sus aplicaciones han trascendido del mundo puramente biológico ya que son capaces de capturar muchas de las características esenciales de los sistemas complejos observados en los sistemas reales y es esa la razón por la que los emplearemos para construir nuestro modelo individual de combate.

2.2. Teoría básica de modelización mediante autómatas celulares

El concepto de autómata celular, AC, fue propuesto inicialmente por J. Von Neumann y Stanislaw Ulam en 1948. Von Neumann fue junto con A. Turing pionero en trabajar dentro del campo que hoy en día conocemos como vida artificial; más concretamente, Von Neumann estaba interesado en el estudio de máquinas auto replicantes basadas en procedimientos lógico-matemáticos computables, es decir, procedimientos que pudieran ser llevados a cabo por una máquina universal de Turing [29]. También era defensor de la idea de que la vida se puede conseguir fuera de cualquier proceso particular, enfoque actualmente conocido como vida artificial fuerte. A pesar de que inicialmente las aplicaciones prácticas asociadas a este tipo de sistemas eran escasas, a partir de los años 70 con el desarrollo de la computación en paralelo y los trabajos desarrollados por J. H. Conway, S. Wolfram, C. Langton, N. Packard y P. Bak surgió un renovado interés por las posibles aplicaciones prácticas de este tipo de modelos. El campo inicial y más obvio de aplicación para los AC fueron el de las matemáticas y la informática, sin embargo, hoy en día se han empleado para modelizar multitud de fenómenos físicos, químicos, sociológicos, epidemiológicos, etc. Como ejemplo véase el modelo desarrollado por L. H. Encinas et. al. en [19] para la predicción de incendios forestales o el desarrollado por A. M. Del Rey et. al. en [30] para el modelado de epidemias.

Según Michel J. Phipps en [15] los rasgos característicos de los autómatas celulares son: i) el sustrato en el que se desarrolla la acción es generalmente una red de 1,2 o 3 dimensiones y cada celda de la red interacciona con su vecindad, ii) cada celda puede adoptar uno de los m posibles estados y iii) la dinámica temporal y espacial del sistema es discreta y obedece a unas determinadas reglas de transición que tienen en cuenta el estado en el que se encuentran las celdas en una determinada vecindad. En los modelos basados en AC el espacio, tiempo y estados de las celdas son discretos. A continuación, vamos a dar algunas de las definiciones básicas de éste tipo de sistemas en dos dimensiones, ya que estas son las que tendrán utilidad práctica para este trabajo.

Formalmente, podemos definir un AC bidimensional como un sistema dinámico ultradiscreto dado por la 4-upla $\mathcal{A} = (\mathcal{E}, \mathcal{S}, \mathcal{V}, f)$ donde \mathcal{E} es la representación espacial del conjunto de células y se denomina *espacio celular*. En caso de un espacio rectangular bidimensional homogéneo formado por células cuadradas en disposición uniforme el espacio celular vendría definido por el siguiente conjunto:

$$\mathcal{E} = \{\langle 1, 1 \rangle, \dots, \langle n, m \rangle\}, \quad n, m \in \mathbb{N} \quad (2.1)$$

Cabe destacar que existen configuraciones alternativas a la geometría de las células que forman el espacio celular, como por ejemplo la cuadrada, hexagonal, etc.

El término \mathcal{S} representa el conjunto finito (o infinito) de los posibles estados en los que se puede encontrar cada célula en cada instante de tiempo y se denomina *conjunto de estados*. En el caso de AC bidimensionales el estado de la célula $\langle i, j \rangle$ en el instante de tiempo t lo denotaremos como $s_{ij}^t \in \mathcal{S}$.

La *vecindad* \mathcal{V} representa el conjunto de células vecinas a una dada y cuyos estados en el instante de tiempo t suelen influir en el estado de la célula considerada en el instante de tiempo posterior $t+1$. Podemos considerar a la propia célula en la vecindad o no. En caso de no hacerlo decimos que la vecindad es *reducida*. Más rigurosamente, la vecindad es el subconjunto de índices ordenados $\mathcal{V} \subset \mathbb{Z} \times \mathbb{Z}$ tal que, para cada célula $\langle i, j \rangle$ su vecindad es:

$$\mathcal{V}_{ij} = \{\langle i + \alpha_1, j + \beta_1 \rangle, \dots, \langle i + \alpha_m, j + \beta_m \rangle\} \text{ con } (\alpha_k, \beta_k) \in \mathcal{V}. \quad (2.2)$$

Dependiendo del problema a modelar y del espacio celular en que trabajemos existen multitud de vecindades: Von Neumann, Moore, simétricas de radio r , etc.

La denominada *regla de transición* f determina la evolución de los estados en los que se encuentran las células en cada instante de tiempo. Si definimos como $\mathcal{S}^t \mathcal{V}_{ij}$ el conjunto de estados de cada una de las células de la vecindad asociada a la célula $\langle i, j \rangle$ en el instante de tiempo t , entonces en caso de que únicamente las células de la vecindad influyeran en la evolución del estado de una célula podemos definir s_{ij}^{t+1} como

$$s_{ij}^{t+1} = f(\mathcal{S}^t \mathcal{V}_{ij}) \in \mathcal{S}. \quad (2.3)$$

Definimos la *configuración* de un autómata celular en un instante de tiempo t como

$$C^t = (s_{11}^t, s_{12}^t, \dots, s_{nm}^t) \in \mathcal{S} \times \overset{n \cdot m}{\dots} \times \mathcal{S} = \mathcal{S}^{n \cdot m}, \quad (2.4)$$

y finalmente se define la *función de transición global* del un autómata como $\mathcal{F} : \mathcal{S}^n \rightarrow \mathcal{S}^n$, es decir $C^{t+1} = \mathcal{F}(C^t)$.

En caso de espacios celulares finitas es necesario fijar condiciones de contorno que definan la dinámica en los límites del espacio celular. Las condiciones de contorno más usuales son las siguientes:

- Condiciones de contorno nulas. En este caso si la célula no pertenece al espacio celular definido se toma su valor como cero a la hora de aplicar la regla de transición.
- Condiciones de contorno periódicas. Este tipo de condiciones implica que el estado de las células que limitarían con las de los extremos fuera del espacio celular coinciden con las del extremo opuesto, es decir, el espacio celular bidimensional adquiere estructura toroidal, luego este tipo de condición inicial permite extender virtualmente hasta el infinito un autómata celular finito.

En caso de querer modelar algún tipo de fenómeno o sistema específico habrá que adaptar las condiciones necesarias para ello. Una primera clasificación de los autómatas celulares puede hacerse a partir de su regla de transición:

- *Autómatas celulares deterministas*. La regla de transición es determinista, es decir, el estado actualizado de una célula está perfectamente determinado a partir del estado de las células en la vecindad considerada. Un ejemplo paradigmático de este tipo de arquitecturas es el *Juego de la Vida* del matemático J. Conway [14]. Los comportamientos dinámicos asociados a este tipo de autómatas han sido ampliamente estudiados, dónde destacan los trabajos llevados a cabo por S. Wólfram [31, 32] a la hora de caracterizar los llamados autómatas elementales de Wólfram.

- *Autómatas celulares estocásticos.* En este caso, el estado de la célula en el siguiente instante de tiempo estará condicionado por una determinada función de probabilidad. Este tipo de autómatas son empleados para intentar reproducir comportamientos relacionados con las ciencias naturales (física, ecología, biología, etc.) ya que permiten reproducir la aleatoriedad asociada a este tipo de sistemas. Este tipo de AC permiten entre otras cosas incluir alteraciones en la función de transición mediante elementos de refuerzo de manera que el comportamiento se pueda recompensar o castigar, obteniendo así los AC capacidad de aprendizaje o comportamientos adaptativos. Véase Lee et al. [6].

2.3. Teoría básica de redes complejas

En esta sección se introducirán algunos conceptos fundamentales de teoría de redes complejas ya que son una de las herramientas más útiles a la hora de obtener información y estudiar los sistemas complejos. Una red compleja es un grafo $\mathcal{R} = (V, E)$ definido a partir de los siguientes conjuntos

$$V = \{v_1, v_2, \dots, v_n\}, \quad E \subseteq V \times V, \quad (2.5)$$

que representa algún tipo de estructura o relación entre elementos en campos como la física, informática, medicina, economía, transporte, etc. Los elementos $v_i \in V$ con $1 \leq i \leq n$ se les denomina *nodos* y el par $e_{ij} = \{v_i, v_j\} \in E$ define los *enlaces* o *aristas*. Un grafo puede ser dirigido, es decir, el orden de los vértices influye en la definición de las aristas o no dirigido. También puede ser ponderado u ordinario. Un grafo se dice ponderado cuando asociamos a cada arista $e_{ij} = \{v_i, v_j\}$ un peso $w_{ij} \in \mathbb{R}$ mientras que se dice ordinario o no ponderado cuando $w_{ij} = c \in \mathbb{R} \forall e_{ij} \in E$. Finalmente, un camino entre dos nodos v_i, v_j es la secuencia ordenada de nodos y aristas que se encuentran entre ambos nodos y definimos la distancia $d(v_i, v_j)$ entre dos nodos v_i y v_j , como la de la longitud, en el sentido de número de aristas que se recorren, del camino más corto entre ambos nodos.

A la hora de representar grafos o redes, una de las herramientas más empleadas es la matriz de adyacencia A la cual contiene la información de las conexiones entre los nodos de la red. Podemos definir esta matriz para una red simple de N nodos como la matriz cuadrada $N \times N$ cuyos elementos se definen como:

$$A_{ij} = \begin{cases} 1 & \text{si } \exists e_{ij} = \{v_i, v_j\} \in E \\ 0 & \text{cualquier otro caso.} \end{cases} \quad (2.6)$$

Para grafos no dirigidos se cumple $A = A^T$ y si no existen lazos en un mismo nodo, los elementos de su diagonal serán 0.

Un instrumento fundamental en el estudio de redes complejas son los coeficientes estructurales. Este tipo de coeficientes permiten cuantificar características estructurales o topológicas de una red. Más formalmente, un coeficiente estructural de una red $\mathcal{R} = (V, E)$ respecto a X , donde X puede ser V o E , es una función:

$$\begin{aligned} \mathcal{C} : X &\longmapsto \mathbb{R} \\ x &\longmapsto e(x) \end{aligned}$$

tal que, para todo x en X y todo isomorfismo entre grafos $\phi : \mathcal{G} \xrightarrow{\sim} \mathcal{H}$ se cumple $\mathcal{C}(x) = \mathcal{C}(\phi(x))$.

Por otra parte, podemos definir medidas de centralidad en una red. Las medidas de centralidad permiten conocer la importancia estructural de un nodo de dentro de una red, es decir, cuantifica la importancia relativa de un elemento $x \in \mathcal{R}$, siendo x un nodo o arista. El concepto de centralidad de grado es una medida de centralidad fundamental en una red compleja y se define como:

$$\mathcal{C}_D(v_i) = \frac{k_i}{N-1}, \quad (2.7)$$

con k_i el grado de un nodo, el cual representa el número de conexiones de un nodo, y N el número total de nodos. Podemos definir el grado medio de una red como

$$\langle k \rangle = \frac{\sum_{i=1}^n k_i}{N}. \quad (2.8)$$

Puesto que el grado de cada nodo puede ser distinto, podemos definir la distribución de grado de una red como una función de probabilidad $P(k)$ que devuelve la probabilidad de que un nodo elegido al azar tenga un grado k .

Las redes complejas también se pueden clasificar en redes complejas homogéneas y redes complejas heterogéneas. El primer caso ocurre cuando la distribución de grado de los nodos de la red se describe a partir de la distribución de Poisson, alcanzando el máximo en $\langle k \rangle$ y decayendo exponencialmente a ambos lados, tendiendo $k \rightarrow 0, \infty$. Un rasgo característico de este tipo de redes complejas es que los nodos con grado muy alto, $k_i \gg \langle k \rangle$ resultan muy infrecuentes. El ejemplo clásico de este tipo de red lo dan las redes completas, donde todos los nodos están vinculados con todos los nodos. Las redes aleatorias y de mundo pequeño también las podemos englobar en este tipo de redes. Las redes complejas heterogéneas se caracterizan por sus distribuciones de grado, que siguen la forma de una ley de potencias y son independientes de la escala de conectividad, lo que induce una estructura topológica no regular al sistema. Redes muy interesantes con estas características son las llamadas redes libres de escala, propuestas por A.L. Barabási y R. Albert en 1999 [7].

A continuación se detallarán las principales medidas de centralidad que se emplearán en el diseño del modelo de combate. Por definición de medida de centralidad, todas ellas toman valores dentro del rango $[0, 1]$.

- Centralidad de vector propio. La centralidad de vector propio trata de generalizar la centralidad de grado teniendo en cuenta la importancia de los nodos vecinos, es decir, trata de cuantificar como de importante es un nodo dependiendo de lo importantes (en el sentido de conexiones con otros nodos) son los vecinos. Esta medida de centralidad se calcula a partir de la matriz de adyacencia A . Si calculamos el radio espectral de A al que denominaremos λ y su vector propio asociado \mathcal{C}_V , cuyas componentes serán positivas debido al Teorema de Perron-Frobenius, obtenemos que

la centralidad de vector propio asociada a cada nodo v_i viene dada por la siguiente expresión:

$$\mathcal{C}_V(v_i) = \frac{1}{\lambda} \sum_{j=1}^N a_{ji} \cdot \mathcal{C}_v(v_j). \quad (2.9)$$

- Clustering local. Este coeficiente trata de medir la transitividad a nivel local, es decir, cuántos de los vecinos de un nodo v_i son adyacentes entre sí. Se calcula a partir de la siguiente expresión:

$$\mathcal{C}_C(v_i) = \frac{N_a(v_i)}{N_v(v_i)}, \quad (2.10)$$

siendo $N_a(v_i)$ el número de parejas de nodos vecinos a v_i y adyacentes entre si y $N_v(v_i)$ el número de parejas de nodos vecinos a V_i .

- Centralidad de cercanía. Esta medida sirve para cuantificar la distancia a la cual se encuentran el resto de nodos de la red de un determinado nodo v_i . Cuanto mayor sea este coeficiente menor es la longitud de los caminos geodésicos de un nodo al resto de nodos. El coeficiente se define a partir de la siguiente expresión:

$$\mathcal{C}_{C_l}(v_i) = \frac{1}{\sum_{1 \leq j \leq N} d(v_i, v_j)}. \quad (2.11)$$

- Centralidad de intermediación. La idea detras de este coeficiente es la de medir el papel de intermediación que tiene un nodo v_i en una red: Cuanto mayor número de caminos geodésicos pasen por él, mas importancia le dará esta medida. Se define la centralidad de intermediación de un nodo como:

$$\mathcal{C}_B(v_i) = \frac{2}{(N-1)(N-2)} \sum_{1 \leq r < s \leq N} \frac{l_{rs}(v_i)}{l_{rs}} \quad \text{con } r \neq i \text{ y } s \neq i \quad (2.12)$$

siendo l_{rs} el número de caminos geodésicos entre v_l y v_r y $l_{rs}(v_i)$ el número de caminos geodésicos entre v_l y v_r que pasan por v_i .

Parte II

Propuesta de modelo individual

Capítulo 3

Diseño de un modelo de combate individual

En este capítulo se desarrollará la propuesta del modelo de combate de naturaleza individual. Dicho modelo estará basado en un autómata celular bidimensional que constará de dos fases que se desarrollan de manera secuencial: fase de combate y fase de movimiento, lo que permitirá generar una mayor complejidad a partir de las interacciones entre los individuos, obteniéndose patrones de comportamiento que difícilmente se puedan originar mediante el empleo de modelos clásicos basados en sistemas continuos basados en sistemas de ecuaciones diferenciales. El objetivo que subyace al uso de un autómata para modelar este tipo de comportamientos no es el de describir de manera fidedigna la realidad de este tipo de situaciones ya que para ello una arquitectura más adecuada sería el empleo de sistemas multi-agentes como la de EINSTEIN - *Enhanced ISAAC Neural Simulation Toolkit* [17], sino el estudio y el análisis de su naturaleza, es decir, de los patrones que surgen a partir de las interacciones individuales, en nuestro caso basadas en reglas matemáticas, de manera que se pueda aprender como evolucionan las estrategias empleadas en el transcurso de una batalla entre dos ejércitos.

3.1. Planteamiento

A continuación, vamos a definir las principales características del modelo. El área de combate geográfica vendrá representada por el espacio celular \mathcal{E} , el cual supondremos por simplicidad que es rectangular con n células por fila y m por columna. La geometría de cada una de las células será a su vez cuadrangular:

$$\mathcal{E} = \{\langle 1, 1 \rangle, \dots, \langle i, j \rangle, \dots, \langle n, m \rangle\}, \quad i, j \in \mathbb{N} \quad (3.1)$$

En cada instante de tiempo, las células pueden estar en uno de los siguientes tres estados: Ocupadas por una unidad verde, ocupadas por una unidad roja o vacías. Para representar dichos estados se le asignará a cada una de las células en función de su ocupación alguno de los siguientes valores: Si una célula $\langle i, j \rangle$ está ocupada por una unidad del ejército verde le asociaremos el valor 1, es decir, $s_{ij}^t = 1$; si dicha célula está ocupada por una unidad del ejército rojo se le asociará el valor -1 , es decir, $s_{ij}^t = -1$ y finalmente, si la célula $\langle i, j \rangle$ está vacía se le asociará el valor 0, es decir, $s_{ij}^t = 0$. Por tanto, el conjunto de

estados que puede tomar el autómata vendrá dado por $\mathcal{S} = \{1, -1, 0\}$. En el instante de tiempo inicial se desplegarán un número de v unidades verdes y r unidades rojas, siendo $v + r \leq n \cdot m$ y $r, v > 0$.

Cada unidad interactuará con aquellas unidades, ya sean aliadas o enemigas, que se encuentren dentro de la denominada vecindad reducida de Moore, es decir, se considerará que el estado de una célula solo puede ser influido directamente por los estados de las 8 células que la rodean:

$$\begin{aligned} \mathcal{V}_{ij} = & \{ \langle i-1, j-1 \rangle, \langle i-1, j \rangle, \langle i-1, j+1 \rangle, \langle i, j-1 \rangle, \\ & \langle i, j+1 \rangle, \langle i+1, j-1 \rangle, \langle i+1, j \rangle, \langle i+1, j+1 \rangle \}, \\ & \text{con } 1 \leq i \leq n, \quad 1 \leq j \leq m \end{aligned} \quad (3.2)$$

Se impondrán condiciones de contorno nulas, es decir, se considera el estado de las células ficticias que rodearían el espacio celular como vacío: $s_{ij}^t = 0, \quad \forall \langle i, j \rangle \notin \mathcal{E}$. Además, con el objetivo de acotar geográficamente el campo de batalla, supondremos que las unidades en liza no podrán abandonar el espacio celular.

Para caracterizar al autómata celular falta definir las reglas por las cuales las células cambian de estado, por tanto, partiendo de lo comentado en la introducción al capítulo, la función de transición entre estados constará de dos fases: combate y movimiento. Si hemos supuesto que el estado de la célula $\langle i, j \rangle$ en el instante de tiempo t es s_{ij}^t , entonces el estado de dicha célula al acabar la fase de combate la denotaremos como \tilde{s}_{ij}^t y al acabar la fase de movimiento como s_{ij}^{t+1} . En primer lugar vamos a definir la función Π que calcule el estado de la célula tras la fase de combate. Esta función debe depender tanto del estado de la célula s_{ij}^t como de los estados de las células asociadas a su vecindad $s_{i+\alpha, j+\beta}^t$ en el instante de tiempo t .

Formalmente definimos dicha la función como:

$$\begin{aligned} \Pi: \mathcal{S} \times \dots \times \mathcal{S} & \rightarrow \mathcal{S} \\ (s_{ij}^t, s_{i-1, j-1}^t, \dots, s_{i+1, j+1}^t) & \mapsto \Pi(s_{ij}^t, s_{i-1, j-1}^t, \dots, s_{i+1, j+1}^t) = \tilde{s}_{ij}^t \end{aligned}$$

donde dado el caso escribiremos $\Pi_{ij}^t = \Pi(s_{ij}^t, s_{i-1, j-1}^t, \dots, s_{i+1, j+1}^t)$ para simplificar la notación.

Una vez definida la función que actualiza los estados de las células tras la fase de combate, se considerará una función \mathcal{M} que actualice el estado de la célula $\langle i, j \rangle$ durante la fase de movimiento. Esta función dependerá del estado de la vecindad ampliada de Moore de radio 2:

$$\begin{aligned} \mathcal{M}: \mathcal{S} \times \dots \times \mathcal{S} & \rightarrow \mathcal{S} \\ (\tilde{s}_{ij}^t, \tilde{s}_{i-2, j-2}^t, \dots, \tilde{s}_{i+2, j+2}^t) & \mapsto \mathcal{M}(\tilde{s}_{ij}^t, \tilde{s}_{i-2, j-2}^t, \dots, \tilde{s}_{i+2, j+2}^t) \end{aligned}$$

Por tanto, obtenemos que la función de transición local actuará durante la fase de movimiento actualizando, en base a un cierto orden, los estados de las células como:

$$s_{ij}^{t+1} = \mathcal{M}(\tilde{s}_{ij}^t, \tilde{s}_{i-2, j-2}^t, \dots, \tilde{s}_{i+2, j+2}^t), \quad \forall \langle i, j \rangle \in \mathcal{E}. \quad (3.3)$$

A continuación, se detallarán las fases de combate y movimiento.

3.1.1. Fase de Combate.

Durante esta fase se desarrollará el combate entre la unidad que se encuentre en una célula y las unidades enemigas situadas en la vecindad reducida de Moore de dicha célula. Al término de esta fase pueden haber ocurrido tres acciones sobre el estado de las células:

- Las células ocupadas por unidades verdes al principio de la fase de combate pueden permanecer ocupadas por la misma unidad verde o pasar al estado vacío al finalizar la fase de combate:

$$\text{Si } s_{ij}^t = 1 \Rightarrow \tilde{s}_{ij}^t = 1 \text{ ó } 0.$$

- Las células ocupadas por unidades rojas al principio de la fase de combate pueden permanecer ocupadas por la misma unidad roja o pasar al estado vacío al finalizar la fase de combate:

$$\text{Si } s_{ij}^t = -1 \Rightarrow \tilde{s}_{ij}^t = -1 \text{ ó } 0.$$

- Las células en estado vacío principio de la fase de combate deben permanecer vacías al final de la fase de combate:

$$\text{Si } s_{ij}^t = 0 \Rightarrow \tilde{s}_{ij}^t = 0.$$

A continuación, se definirá la parte de la función de transición local que calculará el estado de la célula tras la fase de combate. Antes de llevar a cabo un tratamiento matemático preciso sobre cómo esta función afecta a los estados de las células, vamos a justificar las principales características que deberá tener en cuenta construyéndola paso a paso. Lo primero que hay que considerar es que el proceso que vamos a modelar tiene una componente de aleatoriedad considerable. Para justificar esta afirmación supongamos que una célula $\langle i, j \rangle$ no está ocupada por un soldado de un determinado ejército, sino por un pequeño grupo de soldados que se desplazan y combaten siempre juntos - de ahí que hasta ahora se haya empleado el término *unidad* en vez de *soldado* para referirnos a la ocupación de las células -. Cuando una unidad es atacada o ataca la situación estratégica puede ser de inferioridad, de equivalencia o de superioridad. Sin embargo, puede ocurrir que el combate se resuelva al contrario de lo que cabría esperar de la situación de las unidades en liza, por ejemplo, una unidad que se encuentra en una situación de superioridad con respecto a los enemigos de su vecindad puede que acabe siendo derrotada debido a factores como el adiestramiento, la experiencia, la motivación, etc. Factores que son difícilmente medibles y que pueden incluso ser distintos para una misma unidad en tiempos diferentes. Se resalta el hecho de que en la situación anterior lo más probable es que la unidad que tenga la superioridad estratégica sobreviva y aniquile a la que está en inferioridad, sin embargo, podría ocurrir lo contrario, por ello la función Π debe, de algún modo, reflejar un cierto comportamiento estocástico debido a esta naturaleza probabilística que tienen las situaciones de combate. Con tal fin, se propone diseñar Π a partir de las siguientes

funciones:

$$f^c: \mathcal{S} \times \cdots \times \mathcal{S} \rightarrow \mathbb{R}$$

$$(s_{ij}^t, s_{i-1,j-1}^t, \dots, s_{i+1,j+1}^t) \mapsto f^c(s_{ij}^t, s_{i-1,j-1}^t, \dots, s_{i+1,j+1}^t) = f_{ij}^{c,t}$$

$$\psi: \mathbb{R} \times \mathbb{R} \rightarrow \mathcal{S}$$

$$(f_{ij}^{c,t}, X) \mapsto \psi(f_{ij}^{c,t}, X) = \tilde{s}_{ij}^t$$

donde, por simplicidad, se ha empleado la notación $f_{ij}^{c,t} = f^c(s_{ij}^t, s_{i-1,j-1}^t, \dots, s_{i+1,j+1}^t)$ en la última expresión. La función f^c será una función determinista que dependerá del estado de la célula considerada s_{ij}^t a tiempo t y de cómo interacciona la célula con los elementos de su vecindad y su objetivo es determinar la ventaja estratégica de una unidad, mientras que X será el valor umbral que decidirá cuando una unidad es aniquilada o no. Dicho valor se generará aleatoriamente para cada célula $\langle i, j \rangle$ ocupada en cada instante de tiempo t y seguirá una distribución uniforme continua sobre el intervalo $[a, b]$:

$$X \sim \mathcal{U}(a, b) \quad \text{siendo } a = \min(f_{ij}^{c,t}), \quad b = \max(f_{ij}^{c,t}), \quad \forall \langle i, j \rangle \in \mathcal{E}.$$

Además, se empleará en la función aleatoria ψ , que es la encargada de reflejar la naturaleza estocástica del modelo y que actúa comparando el valor aleatorio de X con el del coeficiente $f_{ij}^{c,t}$ para cada célula de manera que si el valor de X es superior al de $f_{ij}^{c,t}$ se considera que la unidad de la célula $\langle i, j \rangle$ es aniquilada, mientras que en caso contrario sobrevive:

$$\psi(f_{ij}^{c,t}, X) = \begin{cases} s_{ij}^t, & \text{si } X \leq f_{ij}^{c,t} \\ 0, & \text{si } X > f_{ij}^{c,t} \end{cases} \quad (3.4)$$

Por tanto, el estado de una célula tras la fase de combate vendrá determinada por la siguiente expresión:

$$\tilde{s}_{ij}^t = \Pi(s_{ij}^t, s_{i-1,j-1}^t, \dots, s_{i+1,j+1}^t) = \psi(f^c(s_{ij}^t, s_{i-1,j-1}^t, \dots, s_{i+1,j+1}^t), X) \quad (3.5)$$

En los siguientes párrafos vamos a plantear el diseño de la función f^c la cual debe reflejar la relación de combate de la célula en $\langle i, j \rangle$ con el resto de las células de su vecindad. Debido a como hemos construido la expresión (3.4) un mayor valor del coeficiente $f_{ij}^{c,t}$ refleja una situación de combate más favorable para la unidad en la célula considerada y viceversa, luego la función f^c debe traducir una situación de combate favorable o adversa en un mayor o menor valor de la misma.

En un primer acercamiento es lógico suponer que el que una unidad de uno de los dos ejércitos sobreviva dependerá de la cantidad de aliados y enemigos que haya en su vecindad, por tanto, teniendo en cuenta los valores numéricos que definen el estado de una célula, nos podemos inspirar en el modelo de Ising bidimensional [8] para hacer una primera propuesta de la función determinista para modelizar el combate entre células ocupadas por unidades de distintos ejércitos:

$$f_{ij}^c = \sum_{(\alpha,\beta) \in \mathcal{V}_{ij}} s_{ij}^t s_{i+\alpha,j+\beta}^t \quad (3.6)$$

Los posibles valores de la expresión (3.6) pertenecen al conjunto de los enteros, más concretamente estarán dentro del intervalo $[-8, 8]$, lo cual implica que si el valor de la expresión es positivo habrá en la vecindad un mayor número de unidades aliadas que enemigas, mientras que si el resultado es negativo implicará que existen un mayor número de unidades enemigas que aliadas dentro de la vecindad de la célula considerada. Si el resultado de la expresión es nulo, implicará que o bien no hay ninguna unidad en la vecindad de la célula o que las fuerzas aliadas y enemigas están equilibradas en número. En este primer acercamiento, por simplicidad, no se considerarán bajas de efectivos por desertiones, accidentes, fuego amigo, etc. por lo que si no hay ninguna unidad enemiga en la vecindad de una célula ocupada se impondrá automáticamente que dicha unidad sobrevivirá a la fase de combate.

Una vez llegados a este punto debemos hacernos las siguientes preguntas: ¿Influye la disposición de los vecinos, ya sean aliados o enemigos, en el resultado de la fase de combate?, ¿y la composición de sus vecindades? Lógicamente, si queremos que la fase de combate permita capturar detalles más íntimos de este tipo de interacciones, la respuesta a estas preguntas debe ser afirmativa, por tanto, podemos plantear incluir todas las características anteriores en la expresión (3.6) a través de un coeficiente ϕ_{ij}^t que permita cuantificar las cuestiones anteriores:

$$f_{i,j}^{c,t} = \sum_{(\alpha,\beta) \in \mathcal{V}_{i,j}} s_{ij}^t s_{i+\alpha,j+\beta}^t + \phi_{ij}^t. \quad (3.7)$$

Para diseñar ϕ_{ij}^t con rigor vamos a analizar las cuestiones planteadas en el párrafo anterior por separado. En primer lugar, vamos a centrar la atención en cómo influye la distribución de los enemigos en la vecindad. Si nos fijamos en la Figura 3.1 podemos visualizar dos situaciones de combate en la que la célula ocupada por una unidad verde es atacada por tres unidades del ejército rojo situadas en su vecindad. Si para determinar el resultado de la batalla nos basamos en la función de transición dada por la expresión (3.6), ambas situaciones son equivalentes. Sin embargo, parece lógico pensar que el escenario *b*) supone una mayor desventaja para la unidad del ejército verde que el escenario *a*) ya que, dicha unidad debe defenderse tanto en vanguardia como en retaguardia, mientras que en el escenario *a*) las tres unidades enemigas están concentradas en la misma zona del mapa.

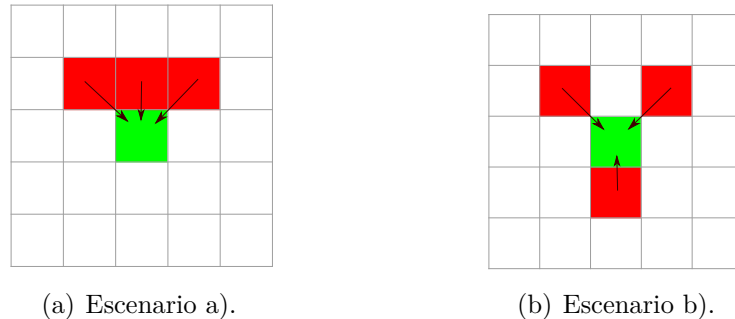


Figura 3.1: Distintos escenarios de conflicto.

Una forma de analizar la dispersión de los enemigos en la vecindad es mediante los

ángulos que estos forman entre sí. Supongamos que dos casillas adyacentes de la vecindad se encuentran a una distancia angular de $\pi/4$ la una de la otra habiendo tomado como centro la célula que se está evaluando - en los ejemplos de la Figura 3.1 dicha célula sería la verde -, entonces en el escenario *a*) las unidades rojas que están en los extremos se encuentran a $\pi/2$ la una de la otra mientras que, las de los extremos con la central se encuentran a $\pi/4$ entre sí. Véase la Figura 3.2 - (a). Por el contrario, en el escenario *b*) las dos unidades enemigas que se encuentran al norte forman un ángulo de $\pi/2$ entre sí, mientras que cada una de ellas con la que se encuentra al sur forman $3\pi/4$. Véase la Figura 3.2 - (b).

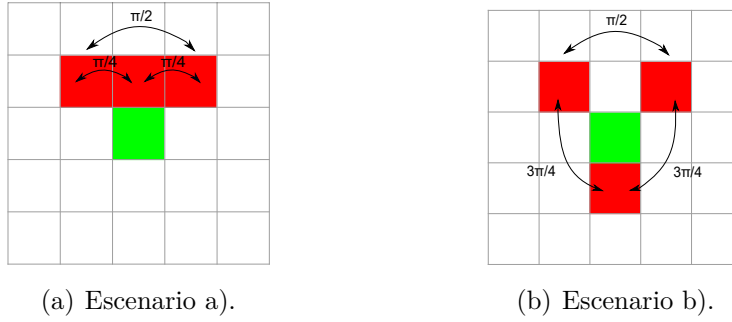


Figura 3.2: Ángulos que forman las unidades enemigas de la Figura 3.1

Es fácil ver que los únicos ángulos que dos unidades pueden llegar a formar en la vecindad entre sí vienen dados por el conjunto $\Omega = \{\frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi\}$ luego podemos definir un coeficiente σ_{ij}^t como :

$$\sigma_{ij}^t = \omega_{\pi} N_{ij}^{\pi,t} + \omega_{\frac{3\pi}{4}} N_{ij}^{\frac{3\pi}{4},t} + \omega_{\frac{\pi}{2}} N_{ij}^{\frac{\pi}{2},t} + \omega_{\frac{\pi}{4}} N_{ij}^{\frac{\pi}{4},t} \quad (3.8)$$

dónde los coeficientes $N_{ij}^{\pi,t}$, $N_{ij}^{\frac{3\pi}{4},t}$, $N_{ij}^{\frac{\pi}{2},t}$, $N_{ij}^{\frac{\pi}{4},t}$ representan el número de parejas de unidades enemigas que forman ángulos entre si de π , $\frac{3\pi}{4}$, $\frac{\pi}{2}$ y $\frac{\pi}{4}$ respectivamente en la vecindad de una célula situada en la posición $\langle i, j \rangle$ en el instante de tiempo t y ω_{π} , $\omega_{\frac{3\pi}{4}}$, $\omega_{\frac{\pi}{2}}$, $\omega_{\frac{\pi}{4}}$ son pesos asociados a cada uno de dichos ángulos y cuyo objetivo es el de ponderar la desventaja que cada una de las cuatro posibles disposiciones de enemigos en la vecindad introducen, luego deben cumplir la siguiente condición: $\omega_{\pi} \leq \omega_{\frac{3\pi}{4}} \leq \omega_{\frac{\pi}{2}} \leq \omega_{\frac{\pi}{4}} < 0$. Los pesos ω los tomaremos como constantes reales dentro del intervalo $(-1, 0]$ por simplicidad, sin embargo, podrían generalizarse, por ejemplo, actualizandose con cada paso temporal - de t a $t + 1$ -, pero siempre respetando la condición impuesta.

Estos pesos están pensados para ser ajustados empíricamente dependiendo de la situación a modelar. En las simulaciones que llevaremos a cabo en la siguiente sección se les asignarán, por simplicidad, los siguientes valores $\omega_{\pi} = -1$, $\omega_{\frac{3\pi}{4}} = -0,75$, $\omega_{\frac{\pi}{2}} = -0,50$, $\omega_{\frac{\pi}{4}} = -0,25$, es decir, la desventaja aumenta de forma constante con el ángulo formado luego el coeficiente σ_{ij}^t tomará valores dentro del intervalo real $[-16, 0]$. Para los escenarios representados en la Figura 3.1 el coeficiente σ_{ij}^t sería, para caso *a*):

$$\sigma_{ij}^t = \omega_{\frac{\pi}{2}} + 2 \omega_{\frac{\pi}{4}} = -1 \quad (3.9)$$

mientras que para el caso b):

$$\sigma_{ij}^t = 2 \omega_{\frac{3\pi}{4}} + \omega_{\frac{\pi}{2}} = -2 \quad (3.10)$$

cumplíndose que σ_{ij}^t del caso b) es menor que el del caso a), lo cual refleja la desventaja del escenario b) sobre el a) para la unidad del ejército verde. Como podemos comprobar, este coeficiente guarda una cierta similitud con el clustering local visto en el capítulo 2. Recordamos que esta medida de centralidad local estima en una red o grafo si los vecinos de un nodo son también adyacentes entre si y podría ser de utilidad emplearla en la fase de movimiento.

Una vez visto como la distribución de los enemigos sobre una célula puede contribuir a decantar el desenlace del combate a favor del enemigo, vamos a estudiar cómo afecta a la capacidad de ataque enemiga su propia vecindad. Supongamos que tenemos las tres situaciones ilustradas en la Figura 3.3 para la célula central ocupada por una unidad del ejército verde.

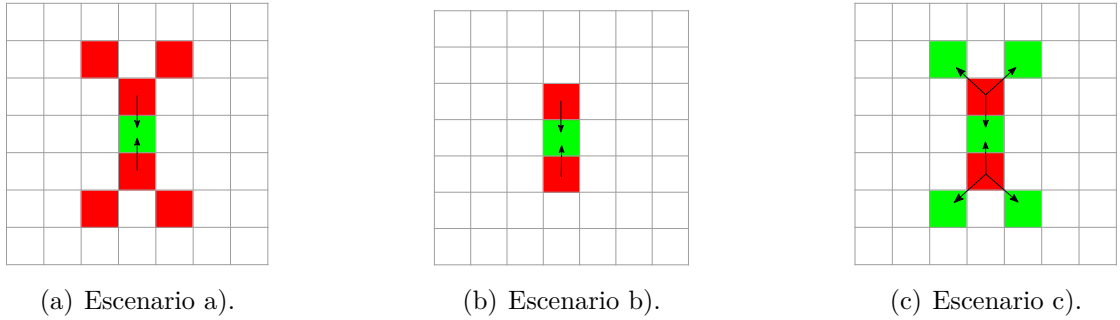


Figura 3.3: Distintos escenarios de conflicto.

Si sólo consideramos el número de efectivos en la vecindad de la célula verde o como están distribuidos los soldados enemigos en ella, las situaciones a), b) y c) son equivalentes, sin embargo, es lógico suponer que el escenario a) es el más desfavorable para la célula central, mientras que el c) es el más favorable ya que, en el primer escenario las unidades enemigas tienen en su vecindad unidades aliadas (rojas) y no tienen que preocuparse de defenderse, mientras que en el escenario c) las unidades enemigas están, a su vez, rodeadas por unidades enemigas (verdes), por lo que no deberían de poder concentrar toda su fuerza de ataque en la célula verde central.

Podemos estudiar cómo influye la vecindad de los vecinos enemigos en su fuerza de ataque mediante un nuevo término Γ_{ij}^t similar al descrito en la expresión (3.6):

$$\Gamma_{ij}^t = \frac{1}{n_{ij}^t} \sum_{(\alpha,\beta) \in \mathcal{V}_{i,j}} \sum_{(\gamma,\delta) \in \mathcal{V}_{i+\alpha,j+\beta}} s_{i+\alpha,j+\beta}^t s_{i+\alpha+\gamma,j+\beta+\delta}^t \quad (3.11)$$

donde n_{ij}^t hace referencia al número de casillas ocupadas por enemigos en la vecindad de $\langle i, j \rangle$ en el instante de tiempo t y pretende normalizar la expresión de acuerdo número de unidades enemigas en la vecindad. El coeficiente Γ_{ij}^t tomará valores reales dentro del

intervalo $[-8, 6]$ y en el caso de los escenarios de la Figura 3.3 dicho coeficiente valdrá para el escenario a):

$$\Gamma_{ij}^t = \frac{1}{2} \cdot ((-1) \cdot (-1 - 1 + 1) + (-1) \cdot (-1 - 1 + 1)) = \frac{1}{2}, \quad (3.12)$$

para el escenario b):

$$\Gamma_{ij}^t = \frac{1}{2} \cdot 0 = 0, \quad (3.13)$$

y para el c):

$$\Gamma_{ij}^t = \frac{1}{2} ((-1) \cdot (1 + 1 + 1) + (-1) \cdot (1 + 1 + 1)) = -3. \quad (3.14)$$

Como podemos observar, que el coeficiente Γ_{ij}^t sea positivo implica que los enemigos que se encuentren en la vecindad de una célula ocupada tienen en sus respectivas vecindades un mayor número de aliados que de enemigos lo cual se traduce en una desventaja para la unidad que se encuentre en $\langle i, j \rangle$ mientras que el que dicho coeficiente sea negativo implica lo contrario: en las vecindades de los enemigos hay más unidades aliadas de la célula $\langle i, j \rangle$ que enemigas, lo cual se traduce en una situación más favorable para dicha célula. Si este coeficiente es cero, o bien no hay ninguna célula ocupada en mi vecindad o las fuerzas que actúan sobre mis vecinos están equilibradas.

En este punto se ha conseguido cuantificar de cara a la resolución de la fase de combate no solo el impacto de la distribución de los enemigos en la vecindad de una célula, sino también como afectan las vecindades de los enemigos en la vecindad a la hora de atacar. Todo ello nos va a permitir definir ϕ_{ij}^t para integrar las cuestiones anteriores en la función de transición local de combate. Conviene recordar que se ha planteado la función f^c de manera que cuanto más pequeño sea el resultado de su evaluación, mayor probabilidad de que al finalizar la fase de combate la unidad que se encuentra en la célula $\langle i, j \rangle$ sea aniquilada y viceversa, por tanto, se propone como expresión para ϕ_{ij}^t la siguiente:

$$\phi_{ij}^t = \begin{cases} s_{ij}^t \sigma_{ij}^t \Gamma_{ij}^t, & \text{si } s_{ij}^t = 0, 1 \\ -s_{ij}^t \sigma_{ij}^t \Gamma_{ij}^t, & \text{si } s_{ij}^t = -1 \end{cases} \quad (3.15)$$

Para justificar la forma de (3.15) vamos a analizar de forma cualitativa como son sus aportaciones a la función que refleja la situación de combate dependiendo de las distintas situaciones que se pueden plantear. Por construcción, siempre se cumplirá que $\sigma_{ij}^t \leq 0$ donde el caso igual a cero solo se da si no hay ninguna casilla ocupada por algún enemigo en la vecindad de una célula ocupada. Además, dependiendo del caso, el término s_{ij}^t ó $-s_{ij}^t$ valdrá siempre 1 si la célula está ocupada y 0 si no lo está. Si el coeficiente Γ_{ij}^t es positivo (+) implica una desventaja para la célula en la posición $\langle i, j \rangle$ como ya se ha argumentado en párrafos anteriores, lo cual a su vez implica que $\phi_{ij}^t < 0$, aumentando así la desventaja en $f_{ij}^{c,t}$. Por el contrario que Γ_{ij}^t sea negativo (-) implica una ventaja para la célula en la posición $\langle i, j \rangle$ siendo $\phi_{ij}^t > 0$ contribuyendo así a las posibilidades de supervivencia de la célula. Cabe resaltar que en caso de que $\Gamma_{ij}^t < 0$ cuanto menor sea

el coeficiente σ_{ij}^t mayor probabilidad de supervivencia tendrá la célula, es decir, cuanto mayor dispersión de unidades enemigas haya en la vecindad mejor. Aunque esto pueda parecer a priori contraintuitivo por la forma en la que se argumentó y construyó dicho coeficiente, lo cierto es que en esta situación que esto sea así si tiene sentido ya que, la unidad en la célula considerada está apoyando el ataque de sus aliados a la célula de su vecindad, y las células enemigas de la vecindad están desconectadas entre sí. Tras este pequeño análisis, cabe destacar que la máxima desventaja que aportará ϕ_{ij}^t al término $f_{ij}^{c,t}$ se dará cuando $\sigma_{ij}^t = -16$ y $\Gamma_{ij}^t = 6$, mientras que la máxima ventaja que aportará se dará cuando $\sigma_{ij}^t = -16$ y $\Gamma_{ij}^t = -8$, por tanto, el intervalo donde el coeficiente ϕ_{ij}^t toma valores será $[-96, 128]$.

Ya tenemos perfectamente caracterizado al coeficiente que modeliza el estado de combate en el que se encuentra la célula $\langle i, j \rangle$ en el instante de tiempo t :

$$f_{ij}^{c,t} = \sum_{(\alpha,\beta) \in \mathcal{V}_{ij}} s_{ij}^t s_{i+\alpha, j+\beta}^t + \phi_{ij}^t. \quad (3.16)$$

Sin embargo, falta plantear cual es la importancia relativa de las partes que forman el coeficiente anterior. Como es lógico, lo más importante a la hora de determinar el resultado de la fase de combate es la cantidad de enemigos y aliados en la vecindad de una célula ocupada, ya que son los que interactúan directamente con ella, por tanto, se propone tratar el segundo término como una corrección a (3.6):

$$f_{ij}^{c,t} = \sum_{(\alpha,\beta) \in \mathcal{V}_{i,j}} s_{ij}^t s_{i+\alpha, j+\beta}^t + \varepsilon \phi_{ij}^t \quad (3.17)$$

donde $\varepsilon \ll 1$ será un parámetro que hará pequeño al segundo término frente al primero. Puesto que el rango en el que toma valores la parte Ising del modelo es $[-8, 8]$ y la parte correctiva toma valores en $[-96, 128]$, se propone que ε se defina de manera que la aportación máxima que pueda llevar a cabo esta última parte al coeficiente $f_{ij}^{c,t}$ no supere un 15% del máximo y mínimo del rango de la parte Ising. Para ello se establecerá para el parámetro pequeño la siguiente cota: $\varepsilon \leq 10^{-2}$, de manera que el rango máximo en el que ahora pueda tomar valores ϕ_{ij}^t será $[-0.96, 1.28]$. Esta propuesta de que la parte correctiva suponga como mucho entorno al 15% de la parte principal o Ising se hace en base a una suposición completamente empírica, ya que es lo suficientemente pequeña respecto al término principal como para tratarse de una corrección pero lo suficientemente significativa como para decantar el resultado del combate a favor de la supervivencia o la aniquilación de la unidad evaluada y podría ser perfectamente modificada siempre que se garantice que $\varepsilon \phi_{ij}^t$ sea pequeño frente a la parte Ising del modelo.

Podemos simplificar la expresión (3.17) sustituyendo valores y sacando factor común:

$$\begin{aligned}
f_{ij}^{c,t} &= \sum_{(\alpha,\beta) \in \mathcal{V}_{i,j}} s_{ij}^t s_{i+\alpha,j+\beta}^t \pm \varepsilon s_{ij}^t \cdot \left(\omega_\pi N_{ij}^{\pi,t} + \omega_{\frac{3\pi}{4}} N_{ij}^{\frac{3\pi}{4},t} + \omega_{\frac{\pi}{2}} N_{ij}^{\frac{\pi}{2},t} + \omega_{\frac{\pi}{4}} N_{ij}^{\frac{\pi}{4},t} \right) \\
&\quad \frac{1}{n_{ij}^t} \sum_{(\alpha,\beta) \in \mathcal{V}_{i,j}} \sum_{(\gamma,\delta) \in \mathcal{V}_{i+\alpha,j+\beta}} s_{i+\alpha,j+\beta}^t s_{i+\alpha+\gamma,j+\beta+\delta}^t \\
&= s_{ij}^t \sum_{(\alpha,\beta) \in \mathcal{V}_{i,j}} s_{i+\alpha,j+\beta}^t \cdot \left(1 \pm \frac{\varepsilon}{n_{ij}^t} \sigma_{ij}^t \sum_{(\gamma,\delta) \in \mathcal{V}_{i+\alpha,j+\beta}} s_{i+\alpha+\gamma,j+\beta+\delta}^t \right) \\
&= \frac{s_{ij}^t}{n_{ij}^t} \sum_{(\alpha,\beta) \in \mathcal{V}_{i,j}} s_{i+\alpha,j+\beta}^t \cdot \left(n_{ij}^t \pm \varepsilon \sigma_{ij}^t \sum_{(\gamma,\delta) \in \mathcal{V}_{i+\alpha,j+\beta}} s_{i+\alpha+\gamma,j+\beta+\delta}^t \right) \quad (3.18)
\end{aligned}$$

Una vez definidos todos los elementos de Π y teniendo en cuenta las expresiones (3.5) - (3.4) podemos detallar explícitamente la función de transición correspondiente a la fase de combate:

$$\begin{aligned}
\text{Si } s_{ij}^t = 0 &\Rightarrow \tilde{s}_{ij}^t = \Pi(0, s_{i-1,j-1}^t, \dots, s_{i+1,j+1}^t) = 0 \\
\text{Si } s_{ij}^t = 1 &\Rightarrow \tilde{s}_{ij}^t = \begin{cases} \Pi(1, s_{i-1,j-1}^t, \dots, s_{i+1,j+1}^t) = 1 & \text{cuando } X \leq f_{ij}^{c,t} \\ \Pi(1, s_{i-1,j-1}^t, \dots, s_{i+1,j+1}^t) = 0 & \text{cuando } X > f_{ij}^{c,t} \end{cases} \\
\text{Si } s_{ij}^t = -1 &\Rightarrow \tilde{s}_{ij}^t = \begin{cases} \Pi(-1, s_{i-1,j-1}^t, \dots, s_{i+1,j+1}^t) = -1 & \text{cuando } X \leq f_{ij}^{c,t} \\ \Pi(-1, s_{i-1,j-1}^t, \dots, s_{i+1,j+1}^t) = 0 & \text{cuando } X > f_{ij}^{c,t} \end{cases}
\end{aligned}$$

Finalmente falta expresar la condición de que si una unidad está únicamente rodeada por sus aliadas o por células vacías dicha unidad no podrá ser aniquilada pues no hay ningún enemigo en su vecindad:

$$\text{Si } s_{i+\alpha,j+\beta}^t \neq -s_{ij}^t \quad \forall (\alpha, \beta) \in \mathcal{V}_{ij} \Rightarrow \tilde{s}_{ij}^t = \Pi(s_{ij}^t, s_{i-1,j-1}^t, \dots, s_{i+1,j+1}^t) = 0.$$

3.1.2. Fase de Movimiento

Completada la fase de combate se inicia la fase de movimiento. Durante esta fase, las unidades supervivientes se desplazarán sobre el espacio celular. Antes de abordar la cuestión del patrón de movimiento que deben de seguir las unidades, hay que decidir qué soldados se moverán primero ya que podría suceder que si movemos al mismo tiempo a todas las unidades dos de ellas se desplacen a la misma casilla, lo cual está prohibido por la condición impuesta de que una célula, de estar ocupada, solo puede estarlo por una única unidad. Para evitar este problema, se propone elaborar un ranking de todas las unidades supervivientes a partir del cual se establecerán las prioridades de movimiento.

Puesto que el modelo que estamos construyendo está basado en individuos, podemos calcular las redes complejas asociadas a cada uno de los dos ejércitos con el objetivo de extraer información útil de cara a esta fase. Denotemos por $\mathcal{R}^{g,t}$ la red asociada al ejército verde y $\mathcal{R}^{r,t}$ al rojo. En primer lugar comenzaremos asociando cada célula del espacio celular, con independencia de si está ocupada o no, con un nodo de una cierta red uniforme $\mathcal{R}^t = (V^t, E^t)$:

$$\forall \langle i, j \rangle \in \mathcal{E}, \exists v_k^t \in V^t \text{ tal que } k = j + (i - 1)n.$$

A continuación, definimos los nodos que forman las redes $\mathcal{R}^{g,t} = (V^{g,t}, E^{g,t})$ y $\mathcal{R}^{r,t} = (V^{r,t}, E^{r,t})$ como subconjuntos del conjunto de nodos V^t de \mathcal{R}^t que cumplen las siguientes condiciones:

$$\begin{aligned} v_k^{g,t} \in V^{g,t} &\iff \tilde{s}_{ij}^t = 1. \\ v_k^{r,t} \in V^{r,t} &\iff \tilde{s}_{ij}^t = -1. \end{aligned}$$

Finalmente imponemos que en el caso de $\mathcal{R}^{g,t}$ y $\mathcal{R}^{r,t}$ los enlaces $e_{kk'}^{g,t}$, $e_{kk'}^{r,t}$ se establecerán entre un par de nodos de una misma red $\{v_k^{g,t}, v_{k'}^{g,t}\}$, $\{v_k^{r,t}, v_{k'}^{r,t}\}$ si sus células asociadas en el espacio celular se encuentran en sus respectivas vecindades, es decir:

$$\begin{aligned} \exists e_{kk'}^{g,t} = \{v_k^{g,t}, v_{k'}^{g,t}\} \in E^{g,t} &\iff \langle i', j' \rangle \in \mathcal{V}_{ij}, \quad \forall \langle i, j \rangle, \langle i', j' \rangle \in \mathcal{E} \text{ y } k \neq k'. \\ \exists e_{kk'}^{r,t} = \{v_k^{r,t}, v_{k'}^{r,t}\} \in E^{r,t} &\iff \langle i', j' \rangle \in \mathcal{V}_{ij}, \quad \forall \langle i, j \rangle, \langle i', j' \rangle \in \mathcal{E} \text{ y } k \neq k'. \end{aligned}$$

Como ejemplo véanse la Figura (3.4) y la Figura (3.5), donde podemos observar las redes de contacto que forman los ejércitos verde y rojo. Como podemos comprobar, solo se forma un enlace entre dos nodos si estos son aliados y están en sus respectivas vecindades.

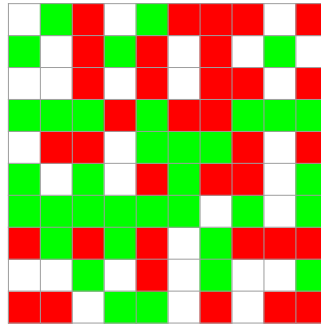
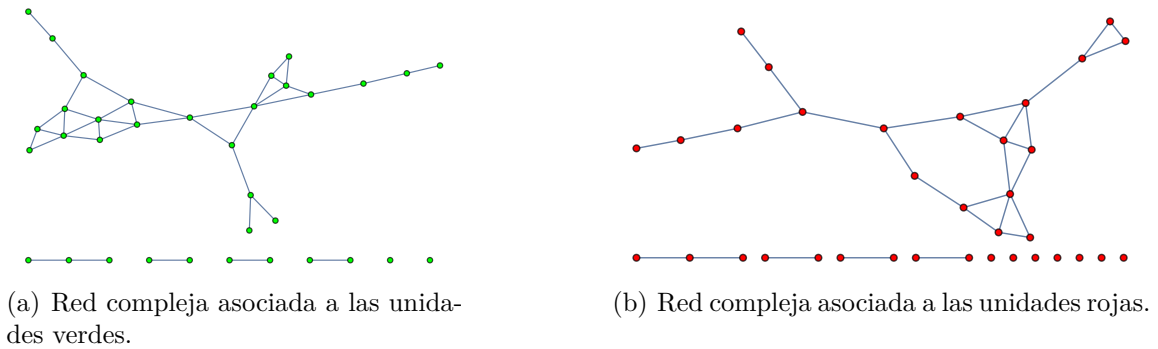


Figura 3.4: Unidades verdes y rojas en un espacio celular con $n = m = 10$.



(a) Red compleja asociada a las unidades verdes.

(b) Red compleja asociada a las unidades rojas.

Figura 3.5: Redes complejas obtenidas para el ejército verde y rojo de la Figura (3.4).

Una vez calculadas las redes de contacto que forman ambos ejércitos, hay que plantear cuales son las características que deben cumplir los nodos que muevan primero. Es lógico suponer que los nodos que mejor conectados estén, tanto con los aliados de su vecindad como con los del resto del clúster, moverán primero ya que dispondrán de un mayor flujo de

información sobre el estado de la contienda. Recordemos que en un modelo que pretenda reproducir fielmente la realidad, los movimientos deberían de producirse simultáneamente, sin embargo, por los motivos descritos en el primer párrafo esto no es posible en el modelo propuesto, por tanto, el ranking que establece la prioridad de movimiento permite reflejar la ventaja que supone tomar una decisión de movimiento con conocimiento previo, o mayor grado de conocimiento, que el resto de los nodos sobre la disposición del resto de unidades. Así pues, podemos basar el ranking propuesto en medidas de centralidad de las redes complejas asociadas a la distribución de unidades de ambos ejércitos.

Se propone emplear las siguientes medidas para construir el ranking: *centralidad de grado* $\mathcal{C}_D(v_k)$, *centralidad de vector propio* $\mathcal{C}_V(v_k)$, *clustering local* $\mathcal{C}_C(v_k)$, *centralidad de cercanía* $\mathcal{C}_{Cl}(v_k)$ y *centralidad de intermediación* $\mathcal{C}_B(v_k)$. La razón por la cual se ha decidido emplear estas medidas es porque en su conjunto permiten reflejar no solo el número de conexiones del nodo con su entorno (centralidad de grado), sino lo bien conectados que están los vecinos entre si o lo importantes que son en la red (clustering local y centralidad de vector propio) o cómo de lejos está la unidad del resto de unidades que forman el clúster (centralidad de cercanía) y el número de caminos mínimos entre nodos que pasan por la unidad seleccionada ya que estos son los caminos por los que, de hacerlo, es lógico que la información fluya entre dos nodos que quieran comunicarse (centralidad de intermediación). Se propone como coeficiente, para caracterizar la posición en el ranking de prioridad de movimiento a cada nodo, el siguiente:

$$\mathcal{I}(v_k) = \frac{1}{5} (\mathcal{C}_D(v_k) + \mathcal{C}_V(v_k) + \mathcal{C}_C(v_k) + \mathcal{C}_{Cl}(v_k) + \mathcal{C}_B(v_k)), \quad (3.19)$$

de manera que, cuanto mayor sea este coeficiente, mayor prioridad para moverse tendrá la unidad que representa el nodo asociado a la célula $\langle i, j \rangle$, estando acotado entre $0 \leq \mathcal{I}(v_k) \leq 1$ por definición de las medidas que lo forman.

Una vez decidido qué unidades se mueven primero, vamos a definir una función Λ que caracterice el movimiento de las unidades por el espacio celular. Esta función no es la función de transición \mathcal{M} que actualiza el estado de la célula tras la fase de movimiento, sino que nos indicará a que casilla de la vecindad debe desplazarse la unidad de la célula $\langle i, j \rangle$:

$$\Lambda: \mathcal{S} \times \overset{25}{\dots} \times \mathcal{S} \rightarrow \mathcal{E}$$

$$(\tilde{s}_{ij}^t, \tilde{s}_{i-2,j-2}^t, \dots, \tilde{s}_{i+2,j+2}^t) \mapsto \Lambda(\tilde{s}_{ij}^t, \tilde{s}_{i-2,j-2}^t, \dots, \tilde{s}_{i+2,j+2}^t) = \langle i + \tilde{\alpha}, j + \tilde{\beta} \rangle.$$

Por tanto, las unidades tienen dos opciones: o abandonan la célula en la que están y se desplazan a una que pertenezca a su vecindad, o no se mueven. Antes de desarrollar formalmente la que será la función de transición local de movimiento, vamos a establecer las condiciones que se deben de satisfacer para esta fase :

- Una unidad podrá desplazarse solo a casillas que pertenezcan a su vecindad o decidir quedarse en la célula que ocupa.
- Si todas las celdas de su vecindad están ocupadas la unidad no podrá desplazarse.

- Ninguna unidad puede abandonar el espacio celular.
- Se ejecutará alguno de los siguientes patrones de movimiento:
 1. Si en la vecindad hay una o ninguna unidad aliada se considerará a esta como una unidad aislada y el movimiento que realizará será aleatorio.
 2. En caso de que haya más de una unidad aliada en la vecindad no se considerará como célula aislada. Se chequearán todas las posibles células a las que la unidad puede desplazarse y en base a criterios lógicos que se establecerán en los próximos párrafos la unidad se moverá a aquella célula que maximice dichos criterios.

La razón por la cual se establece el primer patrón de movimiento (aleatorio) resulta de la incomunicación a la cual una o un par de células aisladas están sometidas, por tanto, esto reflejará en el modelo la falta de información de la que disponen estas células para tomar decisiones en la fase de movimiento que sean coherentes con su situación. Sin embargo, aquellas células que no estén aisladas, como se ha argumentado anteriormente, dispondrán en mayor o menor medida de un flujo de información sobre el estado de la contienda gracias a la conexión con sus aliados, por ello, estas unidades seguirán un patrón de movimiento racional. Debido a como se ha planteado la fase de combate, el patrón de movimiento racional, y en cierta medida obvio, será aquel en el que las unidades comprueben todas las casillas disponibles de su vecindad y calculen en cuál de ellas se maximiza el balance entre aliados y enemigos, puesto que este es el término que más relevancia aporta al coeficiente $f_{ij}^{c,t}$.

A continuación, se expondrá el algoritmo a seguir para completar la fase de movimiento.

1. Una vez calculado el valor $\mathcal{I}(v_k^t)$ para todos nodos de las redes verdes y rojas a tiempo t , se elabora el ranking de prioridad de movimiento con dichos nodos donde, cuanto más alto sea el valor $\mathcal{I}(v_k^t)$ generado por un v_k^t , más cerca estará de las posiciones iniciales siendo el primer nodo del ranking aquel v_k^t que cumpla $\mathcal{I}(v_k^t) = \max \mathcal{I}(v_k^t), \forall v_k^t \in V^{g,t} \cup V^{r,t}$.
2. Una vez elaborado el ranking calculamos las células asociadas a los nodos del ranking. Esto es posible puesto que, para cada k existe un y solo un par de índices $\langle i, j \rangle$ que lo generan.
3. Aplicamos la función Λ a las células en el orden establecido siempre que estas no sean consideradas aisladas. Los índices $(\tilde{\alpha}, \tilde{\beta})$ que indicarán a que célula de la vecindad se moverá la unidad se calculan de manera que sean aquellos que maximicen la siguiente expresión:

$$\max_{(\alpha, \beta) \in \mathcal{V}_{ij}} \left(\tilde{s}_{i+\alpha, j+\beta}^t \sum_{(\gamma, \delta) \in \mathcal{V}_{i+\alpha, j+\beta}} \tilde{s}_{i+\alpha+\gamma, j+\beta+\delta}^t \right) \quad (3.20)$$

donde para este caso en concreto también se valorará la célula $\langle i, j \rangle$ en la que se encuentra la unidad. Si dos o más células maximizaran la expresión anterior, y fueran

accesibles, se escoge de manera equiprobable hacia cual desplazarse. En caso de que la unidad fuera considerada aislada, se asigna $1/9$ de probabilidad de moverse a una de las 8 casillas que la rodean o de quedarse en la célula que ocupa. En caso de que la unidad no esté aislada y la célula a la cual debe desplazarse esté ocupada se escoge la segunda célula que maximice la expresión (3.20) y así sucesivamente. En caso de que ninguna esté disponible la unidad se quedaría en la casilla que ocupa en el instante de tiempo t . En caso de que la célula esté aislada, se escogen células hasta que el movimiento a una de ellas sea válido.

4. Una vez conocida la célula $\langle i + \tilde{\alpha}, j + \tilde{\beta} \rangle$ hacia la cual se desplaza la unidad en $\langle i, j \rangle$ actualizamos el valor de dicha célula como $s_{i+\tilde{\alpha},j+\tilde{\beta}}^{t+1} = \tilde{s}_{ij}^t$ y además para evitar conflictos a la hora de actualizar el resto de las células que faltan establecemos: $\tilde{s}_{i+\tilde{\alpha},j+\tilde{\beta}}^t = \tilde{s}_{ij}^t$ y $\tilde{s}_{ij}^t = 0$. Esto último también puede entenderse dentro del modelo como una cierta capacidad de reacción por parte de las células que no tienen una prioridad de movimiento alta.
5. Repetir para todas las unidades del espacio celular.

Podemos escribir matemáticamente la evolución de la célula $\langle i, j \rangle$ definiendo una función f^m :

$$f^m: \mathcal{E} \rightarrow \mathcal{S}$$

$$\langle i + \tilde{\alpha}, j + \tilde{\beta} \rangle \mapsto f^m(\langle i + \tilde{\alpha}, j + \tilde{\beta} \rangle) = s_{ij}^{t+1}$$

que actualiza el estado de la célula de acuerdo a la siguientes casos:

$$s_{ij}^{t+1} = f^m(\langle i + \tilde{\alpha}, j + \tilde{\beta} \rangle) = \begin{cases} \tilde{s}_{ij}^t, & \text{si } (\tilde{\alpha}, \tilde{\beta}) = (0, 0) \\ 0, & \text{si } (\tilde{\alpha}, \tilde{\beta}) \neq (0, 0) \end{cases} \quad (3.21)$$

dónde $(\tilde{\alpha}, \tilde{\beta})$ son los que resultan del punto 3 del algoritmo descrito. A continuación vamos a escribir explícitamente como actúa la función de transición de la fase de movimiento \mathcal{M} :

$$\text{Si } \tilde{s}_{ij}^t \neq 0 \Rightarrow s_{ij}^{t+1} = \begin{cases} \mathcal{M}(\tilde{s}_{ij}^t, \tilde{s}_{i-2,j-2}^t, \dots, \tilde{s}_{i+2,j+2}^t) = \tilde{s}_{ij}^t & \text{cuando } (\tilde{\alpha}, \tilde{\beta}) = (0, 0) \\ \mathcal{M}(\tilde{s}_{ij}^t, \tilde{s}_{i-2,j-2}^t, \dots, \tilde{s}_{i+2,j+2}^t) = 0 & \text{cuando } (\tilde{\alpha}, \tilde{\beta}) \neq (0, 0). \end{cases}$$

$$\text{Si } \tilde{s}_{ij}^t = 0 \wedge \tilde{s}_{i+\alpha,j+\beta}^t = 0, \quad \forall (\alpha, \beta) \in \mathcal{V}_{ij} \quad \Rightarrow s_{ij}^{t+1} = \mathcal{M}(0, \dots, 0) = 0.$$

$$\text{Si } \tilde{s}_{ij}^t = 0 \wedge \exists \tilde{s}_{i+\alpha,j+\beta}^t \neq 0, \quad \forall (\alpha, \beta) \in \mathcal{V}_{ij} \quad \Rightarrow s_{ij}^{t+1} = \mathcal{M}(0, \tilde{s}_{i-2,j-2}^t, \dots, \tilde{s}_{i+2,j+2}^t) = 0, 1, -1.$$

donde la última expresión pone de manifiesto la posibilidad de que una célula vacía pueda ser ocupada o no si hay al menos una unidad ocupando una célula en la vecindad.

Para ser consistentes con el algoritmo descrito, en caso de que $s_{ij}^{t+1} = 0$, debemos actualizar el estado de la célula de la vecindad, hacia la cual se ha desplazado la unidad, luego se debe añadir la siguiente condición:

$$\text{Si } s_{ij}^{t+1} = 0 \wedge \tilde{s}_{ij}^t \neq 0 \Rightarrow s_{i+\tilde{\alpha},j+\tilde{\beta}}^{t+1} = \tilde{s}_{ij}^t. \quad (3.22)$$

3.2. Simulaciones del modelo

El objetivo de esta sección es testar el modelo construido y estudiar si es capaz de reproducir patrones complejos asociados a fenómenos bélicos o efectos de escala. Se llevarán a cabo simulaciones computacionales para dos escenarios iniciales diferentes con el objetivo de estudiar la dinámica del modelo y las diferencias con respecto a los modelos tradicionales basados en ecuaciones diferenciales.

3.2.1. Condiciones iniciales

Las distintas situaciones iniciales que vamos a analizar serán las siguientes:

- Unidades distribuidas aleatoriamente. El objetivo de esta disposición es el de analizar si el modelo es capaz de generar patrones o un comportamiento complejo, además de analizar el efecto que tiene la escala sobre el mismo. Véase como ejemplo de esta disposición de unidades la Figura 3.6.

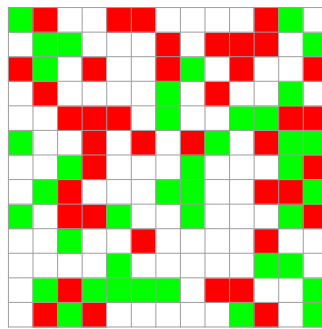


Figura 3.6: Unidades verdes y rojas distribuidas aleatoriamente en un espacio celular con $n = m = 13$. Disposición en el instante de tiempo $t = 1$.

- Modelo de emboscada. Esta disposición inicial tiene como objetivo el de reproducir una situación más realista que la anterior y estudiar como se comporta el modelo en ella. Véase cómo ejemplo de esta disposición de unidades la Figura 3.7.

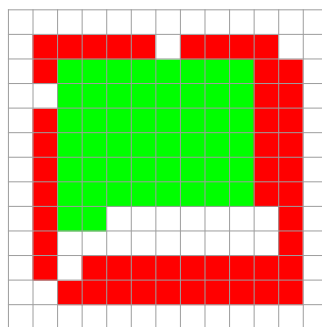


Figura 3.7: 50 unidades rojas emboscan a 50 unidades verdes en un espacio celular con $n = m = 13$. Disposición en el instante de tiempo $t = 1$.

Estos escenarios han sido escogidos ya que representan dos situaciones generales para testar el modelo. Para llevar a cabo las simulaciones numéricas, se recuerda que los pesos asociados al coeficiente σ_{ij}^t se toman en el intervalo $[-1, 0]$ de manera que disminuyan de forma constante según aumenta el ángulo formado por las unidades enemigas en la vecindad : $\omega_\pi = -1$, $\omega_{\frac{3\pi}{4}} = -0,75$, $\omega_{\frac{\pi}{2}} = -0,50$, $\omega_{\frac{\pi}{4}} = -0,25$. Además, el parámetro pequeño lo tomaremos como $\varepsilon = 0,008$. Los límites del intervalo $[a, b]$ en el que toma valores X los estableceremos en $a = -8$ y $b = 8$ ya que estos son el valor mínimo y máximo que toma la parte principal de la función f^c respectivamente.

Debido a que el modelo es no determinista, en el caso de la disposición inicial aleatoria se llevarán a cabo 10 simulaciones independientes para cada uno de los siguientes tamaños del espacio celular: 10×10 , 15×15 y 20×20 con el objetivo de estudiar con una estadística suficiente los resultados proporcionados por el modelo y tener en consideración posibles efectos de escala. Para cada tamaño se estudiarán los siguientes tres casos: Mismo número de unidades para ambos ejércitos, un 10 % más de unidades por parte del ejército verde y un 20 % más de unidades por parte del ejército verde. El objetivo de realizar simulaciones para estos tres escenarios es el de comprobar como se comporta el modelo bajo diferencias en el número inicial de unidades desplegadas.

En el caso de la disposición inicial de emboscada nos centraremos en un tamaño del espacio celular de 15×15 y se estudiarán los siguientes 3 casos: Mismo número de unidades iniciales en ambos bandos, un 10 % más de tropas en el bando emboscado y un 10 % más de tropas en el bando que embosca e intentaremos responder a partir de los datos empíricos obtenidos de 15 simulaciones independientes de los tres casos anteriores las mismas cuestiones que se planteaban para los modelos basados en ecuaciones diferenciales.

3.2.2. Escenario con distribución inicial de unidades aleatoria

Comenzaremos exponiendo los resultados obtenidos en el caso de un tamaño del espacio celular de 10×10 :

- Número de unidades verdes iniciales: 40. Número de unidades rojas iniciales: 40.

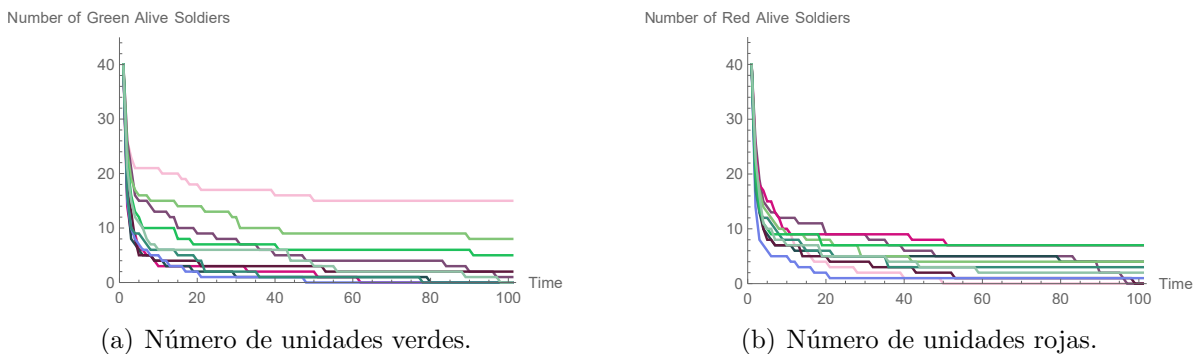


Figura 3.8: Resultados de 10 simulaciones independientes para un mismo número de unidades en el escenario de distribución inicial aleatoria.

Como podemos observar en la Figura 3.8 el comportamiento de ambos ejércitos sigue una tendencia similar: Existe una primera fase en la que se producen un importante número de bajas seguida de una segunda fase en el que el número de bajas parece estabilizarse. Además, se muestra una primera diferencia significativa con respecto a los modelos de Lanchester; Existe una probabilidad considerable de que se llegue a un equilibrio por parte de ambos ejércitos y no se alcance la aniquilación de ninguno de los dos. A continuación vamos a comprobar que ocurre cuando perturbamos ligeramente las condiciones iniciales del número de unidades desplegadas inicialmente.

- Número de unidades verdes iniciales: 44. Número de unidades rojas iniciales: 40.

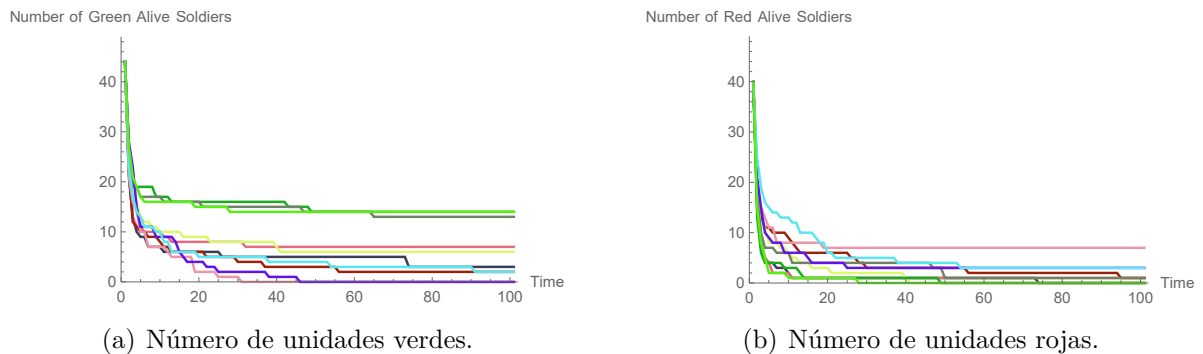


Figura 3.9: Resultados de 10 simulaciones independientes para un 10% más de unidades verdes en el escenario de distribución inicial aleatoria.

Al modificar las condiciones iniciales añadiendo un 10% más de unidades iniciales al ejército verde, la Figura 3.9 muestra como las unidades del ejército rojo siguen tendencias similares en las múltiples simulaciones realizadas: estabilizarse en un número relativamente bajo o nulo de unidades. Sin embargo, en el caso de las unidades verdes observamos claramente dos grupos de resultados en las simulaciones: un escenario en el que las unidades verdes son capaces de prácticamente aniquilar a todas las unidades rojas gracias a su ligera superioridad numérica y un segundo grupo de resultados más parecidos a los de la Figura 3.8 en el que las unidades verdes y rojas se estabilizan con un número de unidades supervivientes similar.

- Número de unidades verdes iniciales: 48. Número de unidades rojas iniciales: 40.

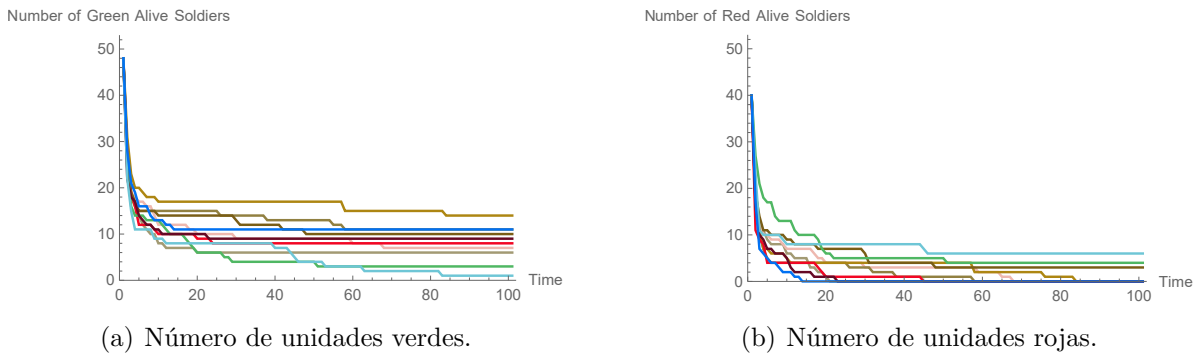


Figura 3.10: Resultados de 10 simulaciones independientes para un 20 % más de unidades verdes en el escenario de distribución inicial aleatoria.

Finalmente, observamos en la Figura 3.10 como la superioridad numérica inicial influye en el aumento de las posibilidades de victoria por parte del ejército verde ya que el número de simulaciones en el que el número final de unidades verdes acaba siendo superior al número de unidades rojas son mayoritarias, por tanto, podríamos afirmar que un aumento considerable en el número de unidades iniciales aumenta notablemente las posibilidades de victoria para un pequeño número de unidades combatientes.

A continuación, se muestran los resultados obtenidos para un tamaño del espacio celular de 15×15 :

- Número de unidades verdes iniciales: 100. Número de soldados rojos iniciales: 100.

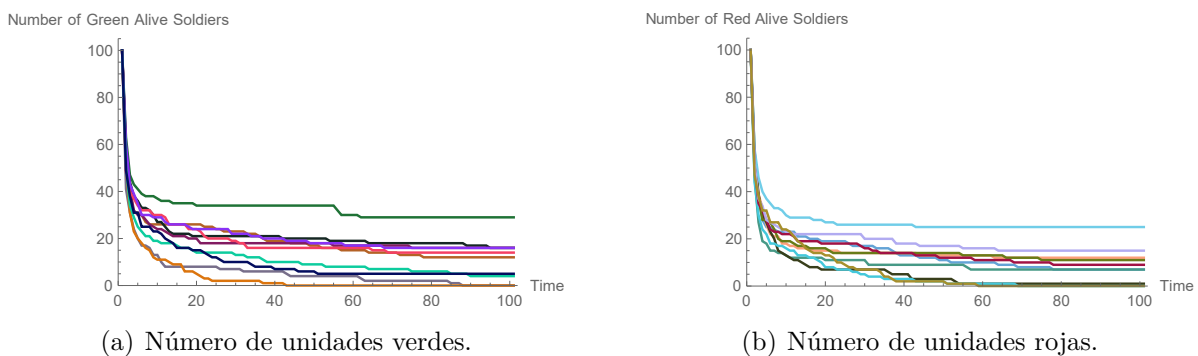


Figura 3.11: Resultados de 10 simulaciones independientes para un mismo número de unidades en el escenario de distribución inicial aleatoria.

La Figura 3.11 muestra al igual que en el caso de la Figura 3.8 dos fases bien diferenciadas y un comportamiento similar por parte de los dos ejércitos, lo cual es lógico debido al mismo número de unidades desplegadas y a la distribución aleatoria de estas que en promedio no debería de privilegiar a ningún ejército.

- Número de unidades verdes iniciales: 110. Número de soldados rojos iniciales: 100.

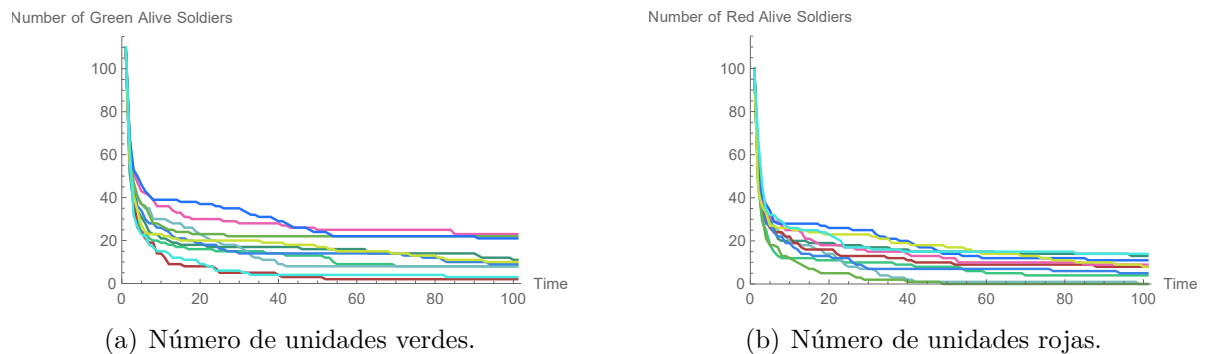


Figura 3.12: Resultados de 10 simulaciones independientes para un 10% más de unidades verdes en el escenario de distribución inicial aleatoria.

En la Figura 3.12 y a diferencia de la Figura 3.9 no se observa de manera clara que una pequeña diferencia a favor en el número de unidades inicialmente desplegadas se traduzca en un aumento considerable de las posibilidades de aniquilación del enemigo. La explicación a este hecho puede estar relacionada con el aumento en el número de soldados en liza por parte de ambos ejércitos, lo cual implica que a pesar de que el aumento porcentual de unidades es el mismo que en los datos expuestos en la Figura 3.9, el número absoluto de unidades en los que se aumenta el ejército verde es demasiado pequeño frente al número total de tropas desplegadas por el ejército rojo. A continuación vamos a comprobar que ocurre si aumentamos en mayor medida la diferencia porcentual a favor del ejército verde.

- Número de unidades verdes iniciales: 120. Número de unidades rojas iniciales: 100.

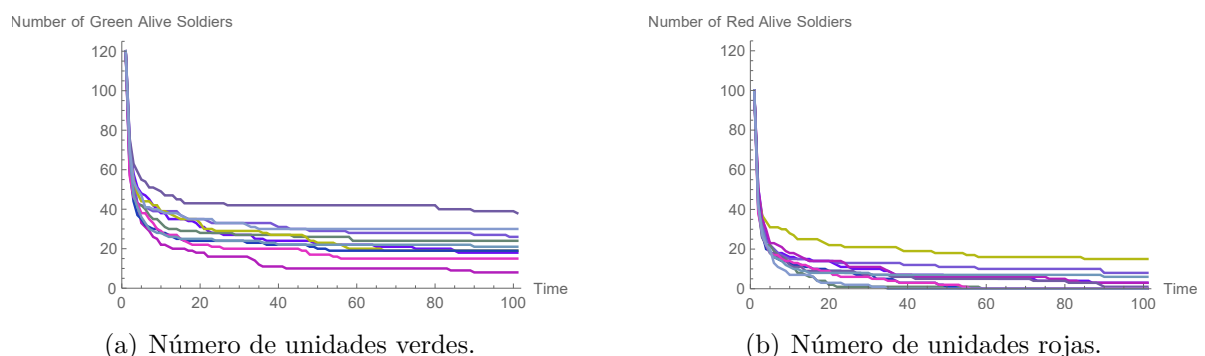


Figura 3.13: Resultados de 10 simulaciones independientes para un 20% más de unidades verdes en el escenario de distribución inicial aleatoria.

En la Figura 3.13 si observamos un claro aumento en el número de unidades supervivientes por parte del ejército verde en el periodo de estabilización, aunque en un

número significativo de casos no consiguen aniquilar a las unidades rojas. Esto último confirma la diferencia entre las soluciones dadas por las simulaciones del modelo construido con las que se obtenían de los modelos basados en sistemas de ecuaciones diferenciales: Es improbable aniquilar al ejército contrario.

Finalmente se exponen los resultados en un espacio celular de 20×20 :

- Número de unidades verdes iniciales: 175. Número de unidades rojas iniciales: 175.

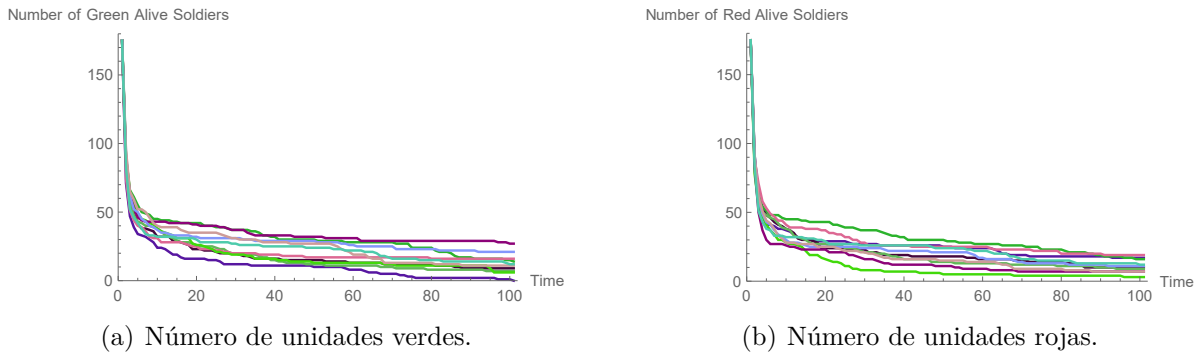


Figura 3.14: Resultados de 10 simulaciones independientes para un mismo número de unidades en el escenario de distribución inicial aleatoria.

La Figura 3.14 muestra un comportamiento similar al de las Figuras 3.8 y 3.11. Sin embargo, se percibe una tendencia mucho más homogénea respecto a las figuras anteriormente citadas en el abanico del número de unidades supervivientes. Esto se debe principalmente a que el número de unidades desplegadas aleatoriamente es mayor, por tanto, las probabilidades de generar una distribución que beneficie a un bando en alguna de las simulaciones es menor, lo que se traducirá en que la diferencia entre los resultados de distintas simulaciones nazca en mayor medida de la aleatoriedad de las interacciones y no de las condiciones iniciales.

- Número de unidades verdes iniciales: 193. Número de unidades rojas iniciales: 175.

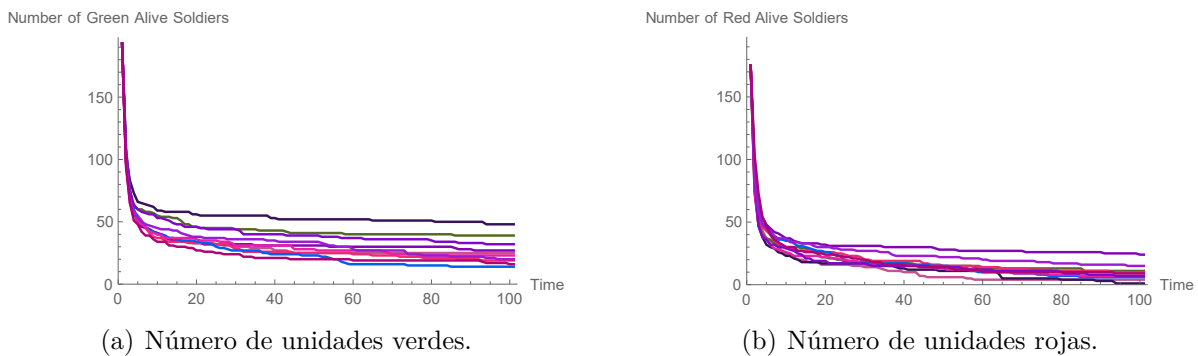


Figura 3.15: Resultados de 10 simulaciones independientes para un 10 % más de unidades verdes en el escenario de distribución inicial aleatoria.

En la Figura 3.15 podemos comprobar como un pequeño aumento porcentual en el número de unidades iniciales si es suficiente para aumentar las posibilidades de supervivencia.

- Número de unidades verdes iniciales: 210. Número de unidades rojas iniciales: 175.

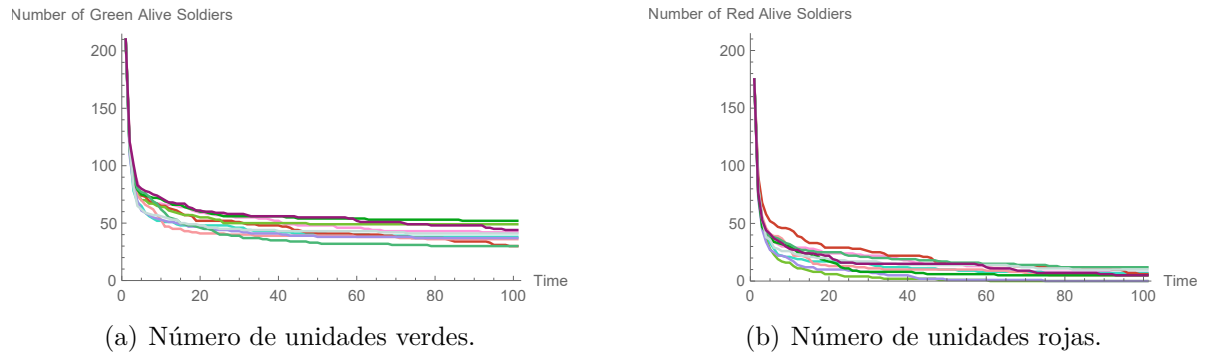


Figura 3.16: Resultados de 10 simulaciones independientes para un 20 % más de unidades verdes en el escenario de distribución inicial aleatoria.

Finalmente, en la Figura 3.16 si observamos como una ventaja significativa en el número inicial de unidades desplegadas prácticamente garantiza la victoria. El espacio celular y el número de unidades desplegadas es considerablemente mayor que en los escenarios anteriores.

Vamos a recapitular lo observado en las simulaciones anteriores. En primer lugar, parece lógico considerar como victorioso al ejército que llegue al final de la simulación con una ventaja clara en el número de unidades frente a su enemigo. En las simulaciones realizadas parecen existir dos periodos bien diferenciados: El primero corresponde a los primeros instantes de la simulación y se caracteriza por la gran virulencia de los enfrentamientos, en el que entorno a la mitad de las tropas de ambos bandos se aniquilan mutuamente. Podemos explicar esta virulencia debido a que en la disposición inicial las unidades se encuentran mezcladas y como consecuencia se producen un gran número de combates. Tras esta etapa podemos observar una estabilización en el número de efectivos de ambos bandos. Esta segunda etapa se caracteriza por largos periodos en los que no se producen bajas junto a instantes puntuales en los que se producen enfrentamientos que causan un número limitado de muertes.

Podemos observar diferencias en el rango del número de unidades supervivientes: En las Figuras 3.8 y 3.11 los resultados proporcionados por el modelo presentan una mayor variabilidad, mientras que en la Figura 3.14 la tendencia que siguen las simulaciones es mucho más parecida como consecuencia del aumento en las dimensiones del espacio celular. A pesar de ello, el modelo propuesto es de naturaleza estocástica, por tanto, siempre existirá una variabilidad inherente al propio modelo lo cual permite explorar distintas situaciones a partir de unas mismas condiciones iniciales.

En este punto cabe preguntarnos al igual que con los modelos basados en sistemas de ecuaciones diferenciales las siguientes cuestiones: ¿Es posible obtener un tiempo de finalización de la batalla?, ¿Cuál es el tamaño final de los ejércitos que se enfrentan?, ¿Existe alguna condición de victoria? Para responder a estas preguntas debemos de tener claro que el modelo es útil a la hora de explorar los distintos escenarios que se pueden desarrollar en un conflicto debido a su naturaleza estocástica. Teniendo como referencia lo observado en las simulaciones de la Figura 3.8 a la Figura 3.16 lo cierto es que es difícil establecer cuando alguno de los dos ejércitos podría ser aniquilado, ya que la segunda fase está caracterizada por una gran estabilidad, aunque parece claro que el aumentar el número de unidades iniciales si contribuye a aumentar la probabilidad de obtener la victoria, sobre todo en escenarios con capacidad de simular un mayor número de unidades. De las simulaciones anteriormente citadas se desprende la posibilidad de establecer un tiempo a partir del cual el autómatas celular pasa de la fase de enfrentamientos inicial a la fase de estabilidad, en la cual uno de los dos ejércitos ya cuenta con la ventaja. Este tiempo parece ser independiente del número de unidades desplegadas y del tamaño del espacio celular y está entorno a $t = 10$ para los coeficientes con los que se han llevado a cabo las simulaciones.

En este punto podemos realizar un pequeño análisis de hacia dónde tienden las soluciones en función de las condiciones iniciales. Si el número de unidades iniciales es similar las soluciones tienden a estabilizarse en un mismo rango por parte de los dos ejércitos. Cuanto más grande sea el espacio celular y el número de soldados desplegados menos variabilidad en el rango de las soluciones habrá. Por el contrario una pequeña diferencia en el número de unidades iniciales no parece ser suficiente para aumentar de manera clara las probabilidades de victoria en caso de que el espacio celular sea pequeño aunque en el caso de un espacio celular suficientemente grande - mayor a 15×15 - sí parece que influye notablemente en ello.

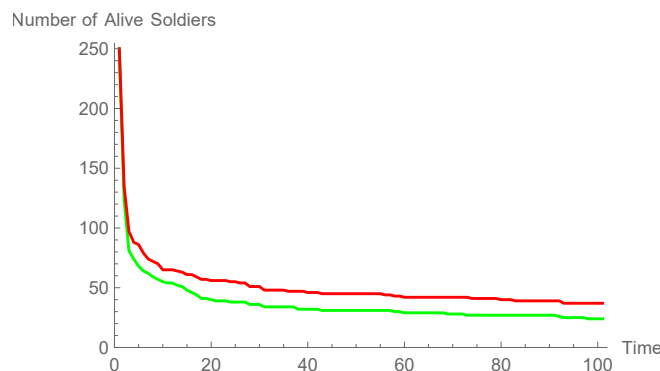


Figura 3.17: Evolución de 250 unidades verdes y 250 unidades rojas en un espacio celular con $n = m = 25$.

Podemos explicar el comportamiento del segundo periodo debido a la formación de clústeres de soldados pertenecientes a un mismo ejército. Estos clústeres se encuentran en un estado metaestable, ya que se ha observado en las simulaciones como son muy sensibles a cualquier perturbación externa, aunque cuando se descomponen sus elementos crean con

relativa facilidad nuevos clústeres. Estas perturbaciones normalmente son provocadas por las unidades consideradas como aisladas, cuyo movimiento es aleatorio. Podemos observar estos comportamientos en la evolución de la topología de las redes complejas que forman los contactos entre nodos. Para ilustrarlo, a continuación se expone como ejemplo la evolución de las redes de contacto que surgen de la simulación dada en la Figura 3.17, la cual se ha llevado a cabo en un espacio celular de 25×25 con 250 unidades de cada bando desplegadas para evitar que la distribución inicial aleatoria pueda beneficiar a alguno de los bandos.

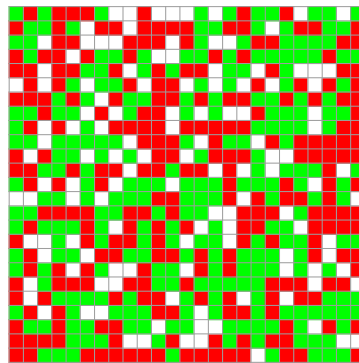


Figura 3.18: 250 unidades verdes y 250 unidades rojas en un espacio celular con $n = m = 25$ en $t = 1$.

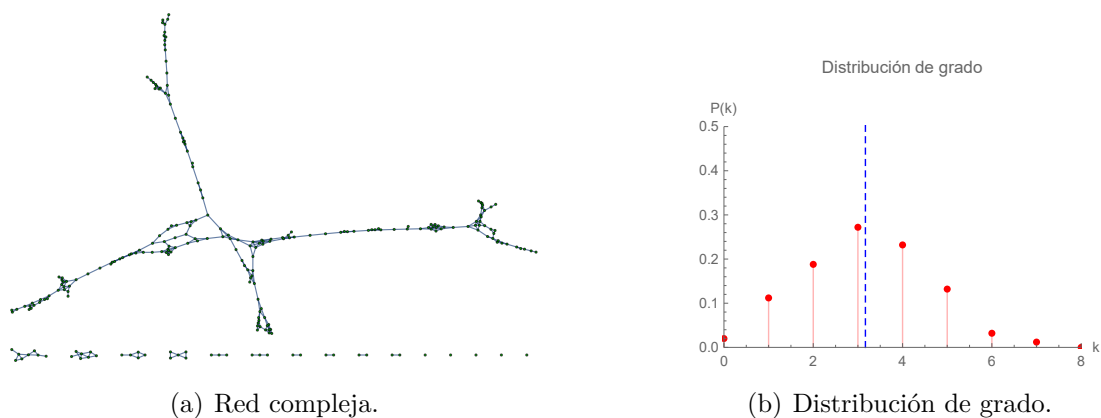
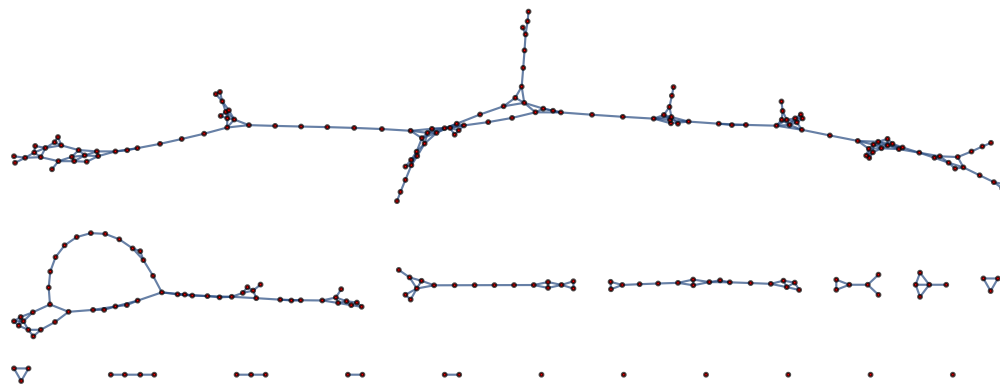
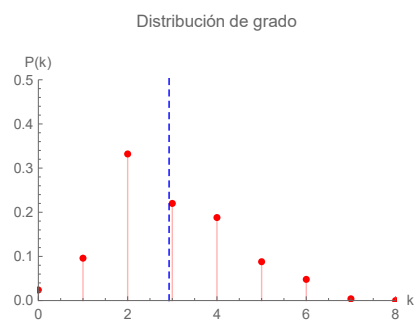


Figura 3.19: Red compleja obtenida y su distribución de grado para el ejército verde de la Figura 3.18.



(a) Red compleja.



(b) Distribución de grado.

Figura 3.20: Red compleja obtenida y su distribución de grado para el ejército rojo de la Figura 3.18.

Como podemos observar en la Figura 3.19-(b) y en la Figura 3.20-(d), las distribuciones de grado de las redes que modelan los contactos entre unidades se aproximan a la de Poisson, pues la probabilidad se aleja de los valores extremos y aumenta en los valores centrales. Esto concuerda con el hecho de haber generado aleatoriamente las posiciones iniciales de las unidades de ambos ejércitos.

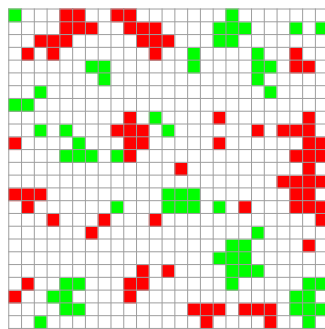


Figura 3.21: Unidades verdes y unidades rojas en un espacio celular con $n = m = 25$ en $t = 10$.

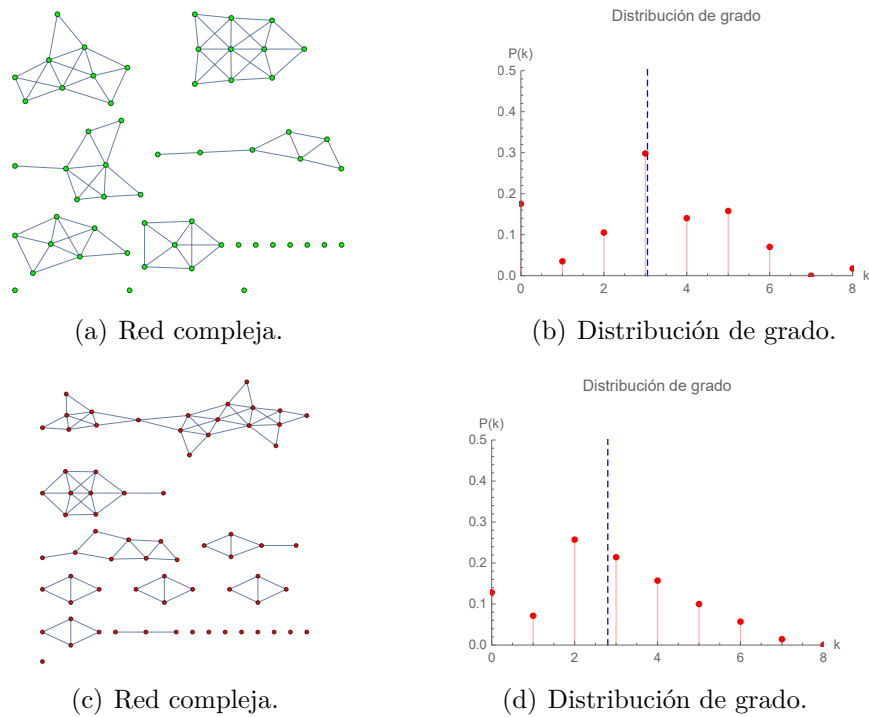


Figura 3.22: Redes complejas obtenidas y su distribución de grado para el ejército verde y rojo de la Figura 3.21.

En la Figura 3.22-(b) y en la Figura 3.22-(d) vemos como la distribución de Poisson se ha desvanecido como consecuencia de los enfrentamientos entre unidades y los patrones de movimiento implementados, cambiando la topología de contactos de los nodos hacia agregados aislados como se apuntaba en párrafos anteriores. Este es el comienzo de la etapa de estabilidad del autómata.

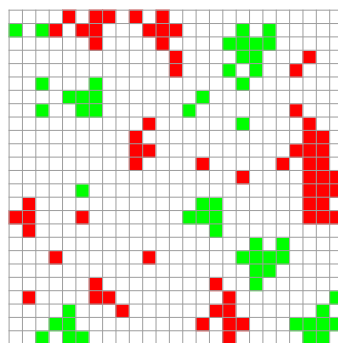


Figura 3.23: Unidades verdes y unidades rojas en un espacio celular con $n = m = 25$ en $t = 20$.

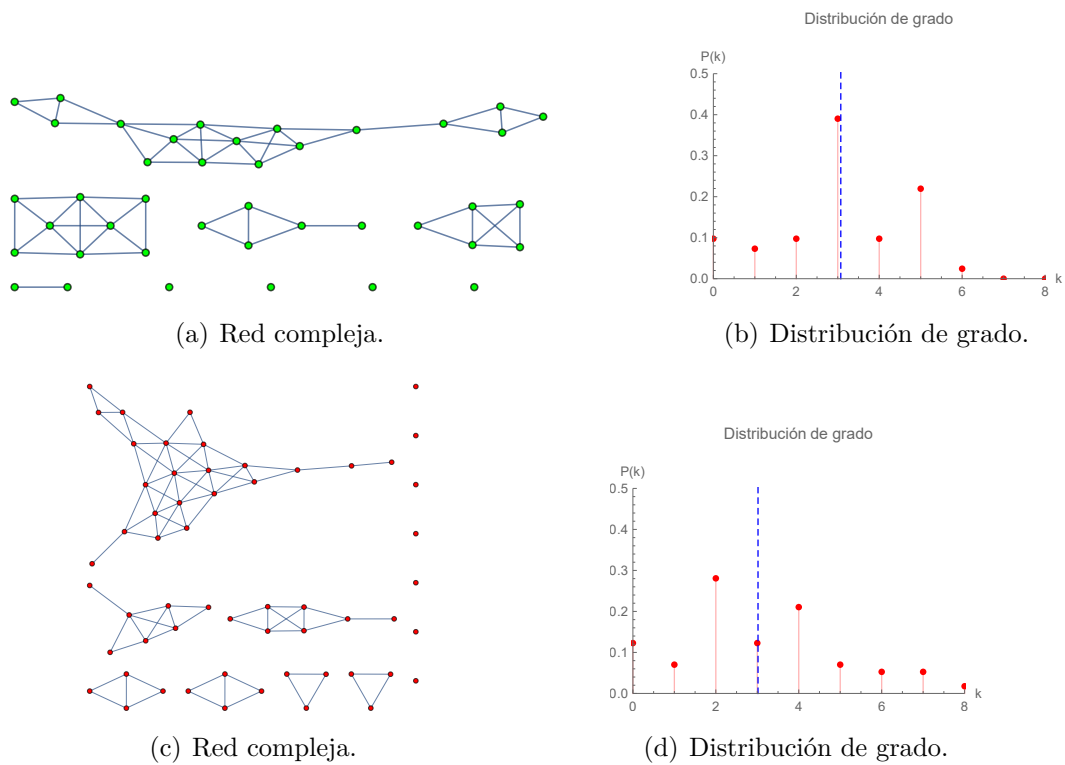


Figura 3.24: Redes complejas obtenidas y su distribución de grado para el ejército verde y rojo de la Figura 3.23.

En la Figura 3.24-(b) y en la Figura 3.26-(d) se observa como muchas de las agrupaciones que surgieron tras la primera etapa de combates intensos aún perduran. Es a partir de ahora cuando las unidades aisladas comenzarán a romper algunas de estas agrupaciones.

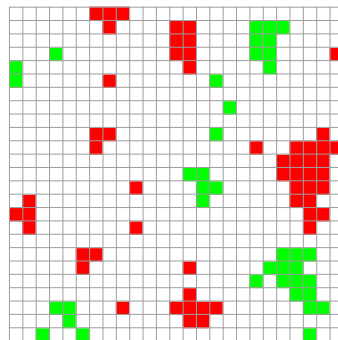


Figura 3.25: Unidades verdes y unidades rojas en un espacio celular con $n = m = 25$ en $t = 40$.

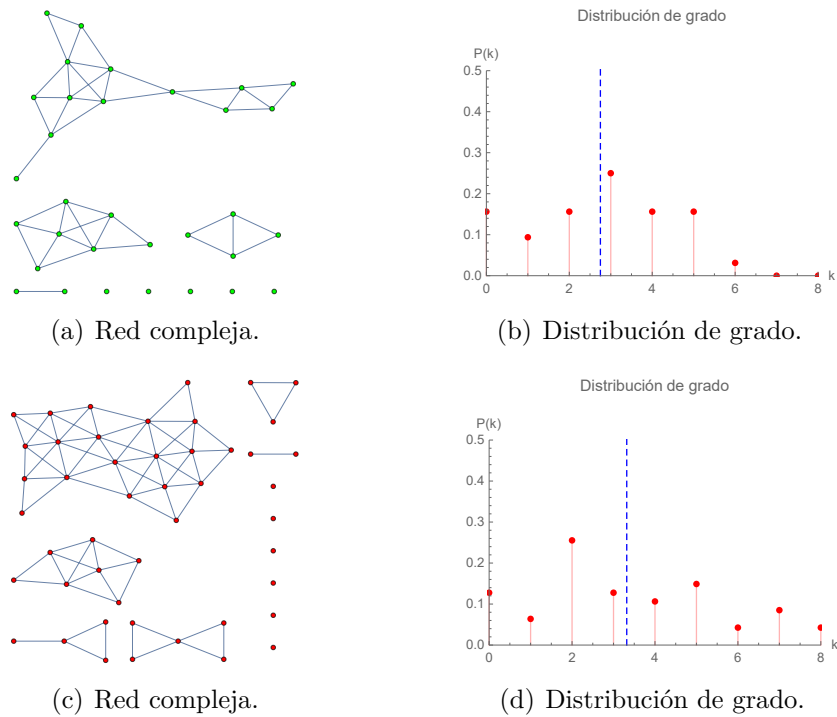


Figura 3.26: Redes complejas obtenidas y su distribución de grado para el ejército verde y rojo de la Figura 3.25.

En la Figura 3.26-(b) y en la Figura 3.26-(d) se observan distribuciones muy parecidas a las dadas en el instante de tiempo $t = 20$.

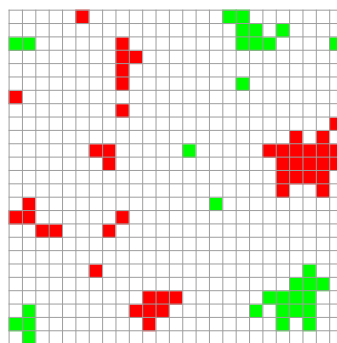


Figura 3.27: Unidades verdes y unidades rojas en un espacio celular con $n = m = 25$ en $t = 80$.

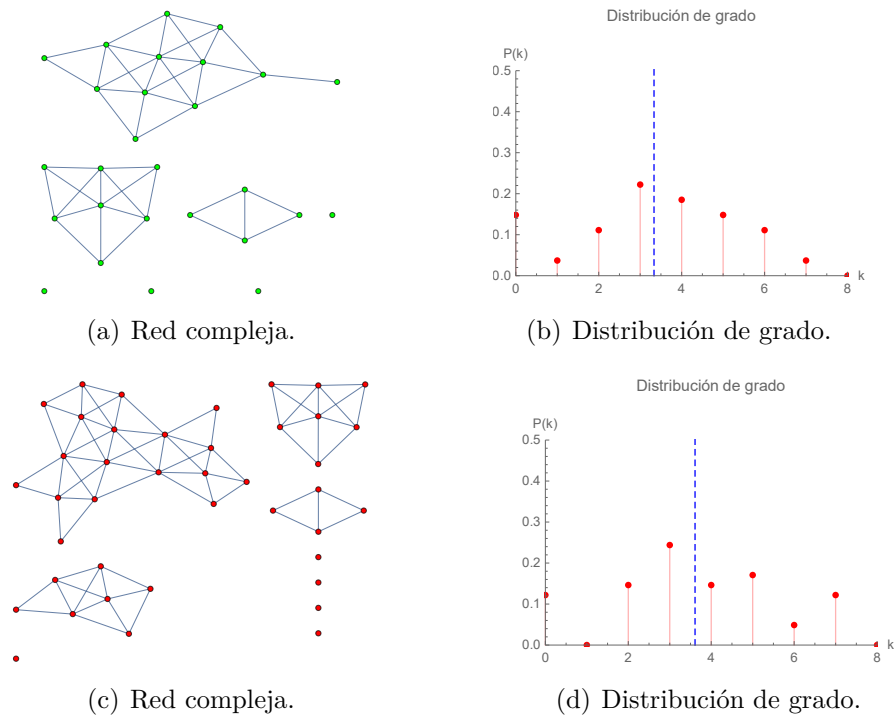


Figura 3.28: Redes complejas obtenidas para el ejército verde y rojo de la Figura 3.27.

Como podemos comprobar, las unidades consideradas aisladas comienzan a desaparecer poco a poco debido a su interacción con los grandes clústeres, lo que afecta a la topología de contactos y se refleja en el comportamiento de la distribución de grado en la Figura 3.28-(b) y en la Figura 3.28-(d). De manera general, la dinámica del sistema a partir de este punto dependerá de los clústeres de unidades que se rompan ya que esto generará nuevas unidades aisladas como consecuencia de la aniquilación de algunos de sus aliados.

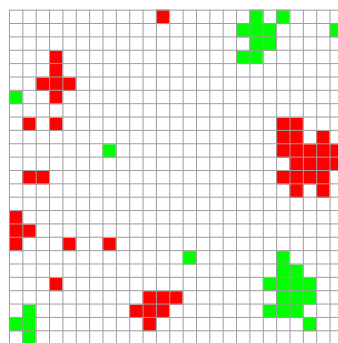


Figura 3.29: Unidades verdes y unidades rojas en un espacio celular con $n = m = 25$ en $t = 100$.

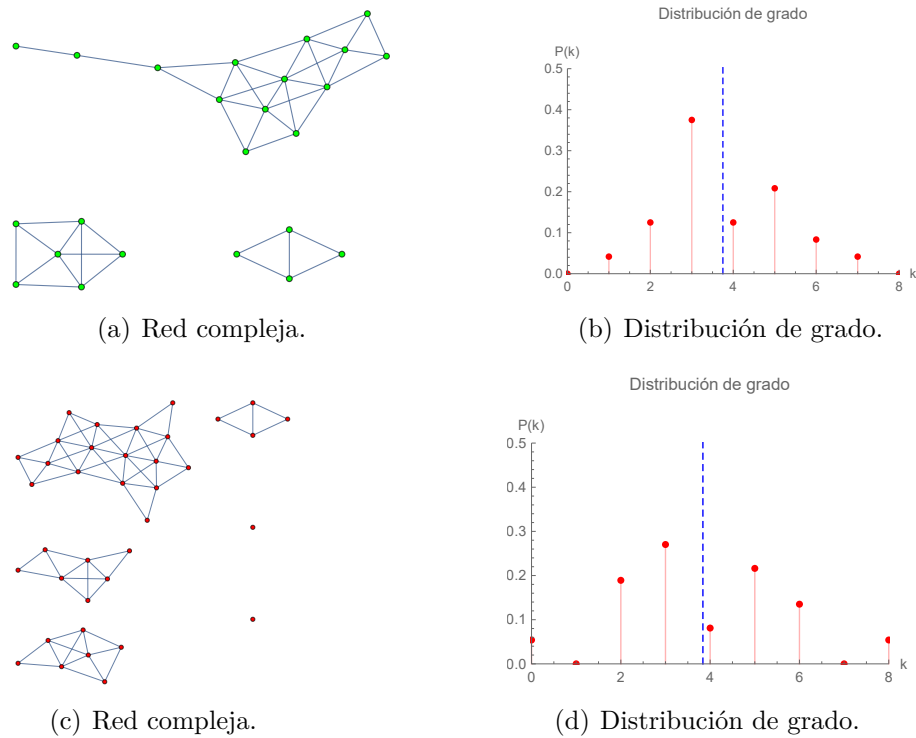


Figura 3.30: Redes complejas obtenidas y su distribución de grado para el ejército verde y rojo de la Figura 3.29.

Recapitulando; la topología de contactos de las redes evoluciona desde unos contactos generados aleatoriamente hasta pequeñas agrupaciones de unidades que se mantienen en el tiempo y son acosadas por las unidades consideradas aisladas, pudiendo ocurrir que el sistema se vuelva completamente estable o las unidades aisladas rompan dicha estabilidad cambiando la topología de contactos de las unidades que forman el sistema. El análisis de la distribución de grado de las redes de contacto obtenidas valida las afirmaciones anteriores.

Finalmente, podemos observar como la superioridad en el número inicial de unidades por parte de uno de los ejércitos aumenta las probabilidades de lograr la victoria en el enfrentamiento. Debido a la aleatoriedad inherente al modelo, no es posible establecer una cota exacta a partir de la cual se garantice la victoria, sin embargo, podemos observar como la diferencia porcentual entre las unidades desplegadas inicialmente permite aumentar de manera significativa las probabilidades de victoria. La distribución inicial de las unidades en este apartado se ha caracterizado por su aleatoriedad. A continuación se expondrá un escenario con una distribución inicial concreta que nos permita analizar como los cambios en ella afectan al resultado del modelo.

3.2.3. Escenario de emboscada

En esta subsección se expondrán los resultados de simular las tres situaciones planteadas dentro del escenario de emboscada. Se recuerda que este escenario consiste en un espacio celular de 15×15 en el que el ejército verde está situado en el centro del mapa

rodeado por el ejército rojo, el cual intentará emboscarlo.

- Número de unidades verdes iniciales: 50. Número de unidades rojas iniciales: 50.

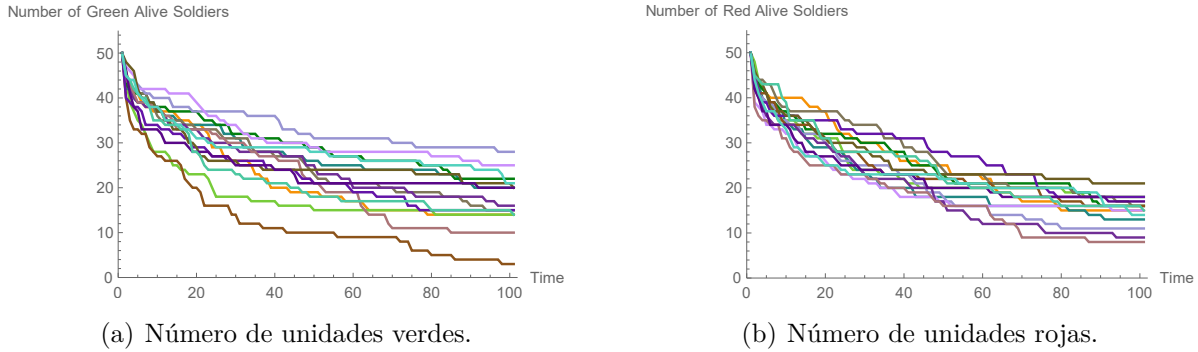


Figura 3.31: Resultados de 15 simulaciones independientes para un mismo número de unidades en el escenario de emboscada.

La primera diferencia evidente de la Figura 3.31 con respecto al escenario en el que la disposición inicial era aleatoria es que ya no se observa una aniquilación de unidades tan pronunciada en la primera fase de la simulación, sino que el sistema tiende a estabilizarse de manera mucho más suave. Otra característica apreciable es la mayor variabilidad en los resultados finales de las unidades verdes frente al de las unidades rojas. Este hecho se explica con la disposición inicial: Si debido al comportamiento estocástico del sistema alguna de las unidades verdes que se encuentran en los extremos de la agrupación se descuelga del resto ésta es rápidamente aniquilada por parte de las unidades rojas que además tienden a autoorganizarse del modo más conveniente, mientras que la capacidad de autoorganización de las unidades verdes es limitada debido al agrupamiento y el estar rodeadas.

- Número de unidades verdes iniciales: 55. Número de unidades rojas iniciales: 50.

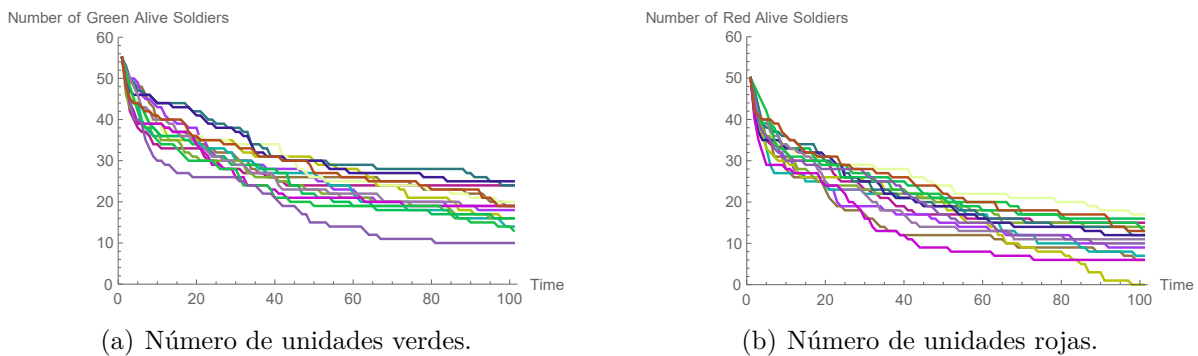


Figura 3.32: Resultados de 15 simulaciones independientes para un 10 % más de unidades verdes en el escenario de emboscada.

En la Figura 3.32 no se muestra que un ligero aumento en el número de unidades rojas tenga efectos significativos para alguno de los dos ejércitos con respecto a la situación planteada en la Figura 3.31 aunque si parece influir en reducir la variabilidad en el rango de las soluciones.

- Número de unidades verdes iniciales: 50. Número de unidades rojas iniciales: 55.

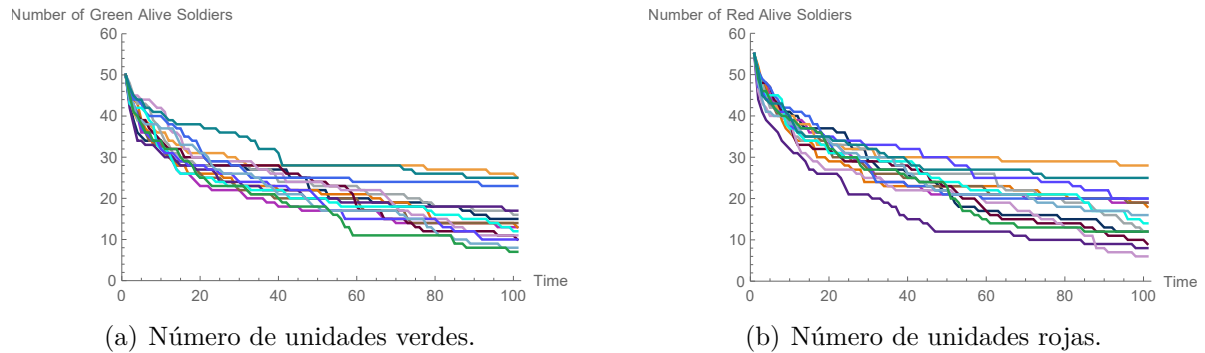


Figura 3.33: Resultados de 15 simulaciones independientes para un 10 % más de unidades rojas en el escenario de emboscada.

Para finalizar, en la Figura 3.33 tampoco se observan diferencias significativas en la tendencia de las soluciones con respecto a la Figura 3.31.

Como podemos observar, en este escenario la fase en la que se produce un mayor número de bajas está mucho más extendida temporalmente. Tras la fase inicial, las simulaciones indican que el sistema evoluciona hacia una relativa estabilidad, caracterizada por el surgimiento de agrupaciones de unidades, y en la cual se producirán enfrentamientos esporádicos entre ambos bandos.

A pesar de la dificultad de realizar un análisis analítico para sistemas basados en autómatas celulares podemos intentar responder empíricamente, es decir, en base a los resultados obtenidos hacia dónde tienden las soluciones en función de las condiciones iniciales. Lo primero que se deduce de la Figura 3.31, la Figura 3.32 y la Figura 3.33 es que la distribución inicial de emboscada es estable frente a pequeñas diferencias en el número de unidades iniciales tanto por parte del ejército que embosca como por parte del ejército que es emboscado. Además, en caso de un mismo número de unidades las soluciones del ejército emboscado parecen ser mucho más impredecibles que las del ejército que embosca.

También podemos intentar responder a las mismas preguntas que se planteaban para los modelos basados en ecuaciones diferenciales. En cuanto al tamaño final de los ejércitos, las múltiples simulaciones sugieren que se encontrará por debajo de la mitad de las unidades desplegadas inicialmente, aunque parece improbable que se llegue a una aniquilación completa de alguno de los dos ejércitos. Las dificultades para calcular un tiempo en el que se alcanza la victoria por parte de alguno de los dos ejércitos son mayores en este escenario debido a que la separación entre las fases se produce de manera suave y

no bruscamente, aun así las simulaciones muestran que a partir del instante de tiempo $t = 50$ la mayoría de las soluciones han alcanzado la fase de estabilidad. No es posible establecer una condición de victoria clara. La distribución de emboscada parece favorecer a las unidades que emboscan debido a que la variabilidad de los resultados finales del número de unidades supervivientes es menor lo que permite hacer previsiones más acertadas, sin embargo, el modelo propuesto no parece privilegiar claramente en el escenario de emboscada a ninguno de los dos bandos en liza.

Para finalizar el análisis de esta sección vamos a calcular la contrapartida determinista de las tres situaciones anteriores empleando los modelos de Lanchester. Estableceremos que la fuerza de fuego de ambos ejércitos es exactamente la misma ya que así se considera en el modelo propuesto.

- Modelo de fuego dirigido:

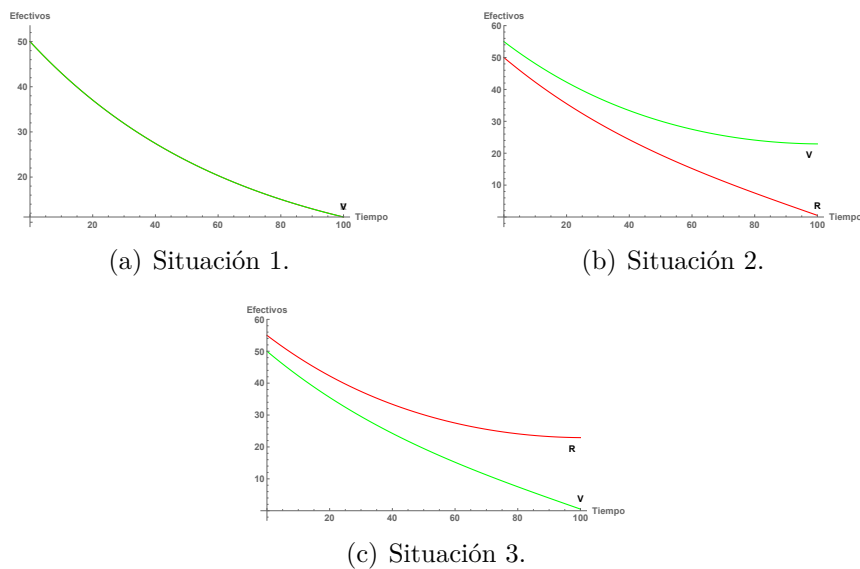


Figura 3.34: Modelo de fuego dirigido.

Como podemos observar en la Figura 3.34 la dinámica del ejército victorioso guarda similitudes con la dinámica observada en el escenario de emboscada, sin embargo, la dinámica del ejército derrotado no se corresponde con la predicha por nuestro modelo en el escenario de emboscada. Esto se puede explicar debido a que los modelos de Lanchester no consideran la posibilidad de que dos ejércitos puedan llegar a un equilibrio entre ellos.

- Modelo de guerrilla:

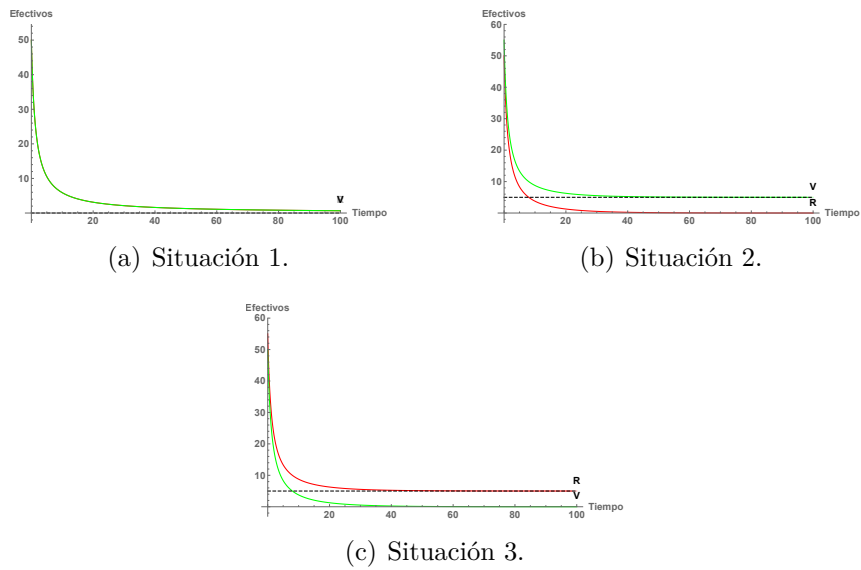


Figura 3.35: Modelo de guerrilla.

La dinámica observada en la Figura 3.35 recuerda a las simulaciones realizadas en el escenario de la distribución inicial aleatoria ya que se produce un cambio de tendencia brusco de un periodo de un elevado número de bajas hacia un periodo de estabilidad. Sin embargo, existe una diferencia notable con respecto a los resultados del modelo propuesto: Se llega a la estabilidad debido a la aniquilación del enemigo mientras que en el modelo propuesto se alcanza de manera frecuente la coexistencia de grupos de soldados en la fase de estabilidad.

Conclusiones y trabajo futuro

El diseño y modelado de acontecimientos bélicos ha sido un campo de intenso trabajo y estudio desde comienzos del siglo XX por la ventaja práctica que estos suponen a la hora de hacer preparativos frente a un escenario de conflicto y la discreción con la que pueden ser llevados a cabo. Este campo, que comenzó siendo estudiado mediante sistemas de ecuaciones diferenciales, ha avanzado en los últimos años hacia sistemas multiagentes gracias a las cada vez mayores prestaciones de los ordenadores los cuales permiten hacer predicciones de escenarios concretos.

El objetivo principal de este trabajo ha sido el de diseñar e implementar un modelo de combate basado en interacciones individuales mediante un autómata celular y desde un punto de vista estocástico, motivado por la poca literatura que se puede encontrar sobre este tipo de arquitecturas aplicadas a este campo. Las simulaciones realizadas en el capítulo 3 muestran como a partir de reglas sencillas el modelo propuesto es capaz de reproducir una dinámica de combate propia, que presenta notables diferencias respecto a los modelos clásicos. Estas diferencias surgen de la no linealidad de las interacciones entre los múltiples individuos, y de la propia filosofía con la que se construye el modelo: mientras que el objetivo de los modelos clásicos es obtener cotas y rangos a partir de los cuales predecir con un cierto error si una batalla puede ser ganada o no, es decir, describir una posible batalla, el modelo propuesto en este trabajo no persigue ese objetivo, sino que persigue la comprensión de los factores que a nivel individual caracterizan este tipo de interacciones e intentar reproducirlos.

Podemos concluir que el modelo propuesto cumple su objetivo de reproducir la dinámica característica de los modelos de combate a partir del comportamiento individual de las unidades en liza, observándose la formación de conductas complejas como lo son las múltiples agrupaciones de unidades que evolucionan hasta posiciones metaestables o las dos aparentes fases que surgen del modelo: Una primera caracterizada por un gran número de bajas, que puede ser más o menos pronunciada dependiendo de la disposición inicial de unidades, y una segunda fase caracterizada por una cierta estabilidad en la que se producen un bajo número de bajas y espaciadas en el tiempo. Ninguna de estas características conductuales pueden inferirse a partir de las reglas que inicialmente rigen el modelo, lo cual, junto con un elevado número de elementos en interacción, es una de las características propias de los sistemas complejos.

A pesar del párrafo anterior, el modelo sufre de diversas limitaciones. Una de las más notables es la homogeneidad de las unidades, es decir, la única diferencia entre dos unidades es el de la pertenencia a alguno de los dos ejércitos en lucha. Otro rasgo limitante

del modelo es la falta de datos sobre el entorno en el que se desarrolla la acción, por ello a continuación se expondrán algunos de los posibles itinerarios por los cuales el trabajo presentado podría continuar siendo desarrollado:

- En primer lugar, se debería de dotar a cada una de las unidades de potencia de fuego y capacidad defensiva propias con el objetivo de heterogeneizar a las unidades. En el modelo propuesto todas las unidades tienen la misma capacidad de ataque y su principal fuerza se basa en el número de aliados que se encuentran en la vecindad de una célula.
- El modelo podría mejorarse añadiendo características del terreno que permitieran añadir una ventaja o desventaja al enfrentamiento. Otra vía de desarrollo en esta línea es el de añadir una dimensión extra al modelo, esto no solo permitiría describir variaciones del terreno de una manera intrínseca, sino que permitiría la implementación de otro tipo de unidades como lo son las aéreas añadiendo realismo y complejidad al modelo.
- Sería interesante considerar una estructura jerarquizada de varios autómatas celulares, en la que cada uno de ellos se encargara de procesar la información de las distintas partes del conflicto para luego suministrar los resultados a un agente que tomara decisiones en base a dichos resultados.
- Otra posible línea es la de implementar la toma de decisiones por parte de todas o algunas unidades. Estas unidades podrían cubrir el papel que tienen los oficiales en una batalla, lo que resultaría en un modelo híbrido de autómatas celulares y sistema basado en agentes.
- Otra línea de investigación que podría ser interesante es la de estudiar el sistema cuando a uno o a los dos ejércitos se le permiten emplear sistemas de refuerzo del comportamiento a sus unidades. El modelo puede ser un buen pretexto para estudiar nuevas técnicas o variaciones de las ya existentes dentro del campo del aprendizaje automático o inteligencia artificial, actualmente en auge.
- El modelo propuesto podría ser adaptado para modelizar otras situaciones de combate, como puede ser el enfrentamiento de colonias de animales o entre sistemas biológicos elementales dentro de un organismo.
- Finalmente, sería interesante estudiar mediante teoría de percolaciones y medios de la física estadística las distintas etapas del modelo y comprobar si este cambio pudiera tratarse como un cambio de fase.

Bibliografía

- [1] D. Botkin et al. «Some Ecological Consequences of a Computer Model of Forest Growth». En: *Journal of Ecology* 60(3) (1972), págs. 849-872. DOI: [10.2307/2258570](https://doi.org/10.2307/2258570).
- [2] D. Eisinger et al. «Emergency vaccination of rabies under limited resources – combating or containing?» En: *BMC Infect Dis* 5 5 (2005), pág. 10. DOI: <https://doi.org/10.1186/1471-2334-5-10>.
- [3] D. Eisinger et al. «Spatial pattern formation facilitates eradication of infectious diseases.» En: *The Journal of Applied Ecology* 45(2) (2008), págs. 415-423. DOI: <https://doi.org/10.1111/j.1365-2664.2007.01439.x>.
- [4] F. Jeltsch et al. «Pattern formation triggered by rare events: lessons from the spread of rabies». En: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 264 (1997), págs. 495-503. DOI: [10.1098/rspb.1997.0071](https://doi.org/10.1098/rspb.1997.0071).
- [5] M. Kress et al. «Lanchester model for three-way combat». En: *European Journal of Operational Research* 264 (2018), págs. 46-54. DOI: <https://doi.org/10.1016/j.ejor.2017.07.026>.
- [6] Y. C. Lee et al. «Adaptive Stochastic Cellular Automata: Theory». En: *Cellular Automata: Theory and Experiment*. Cambridge, MA, USA: MIT Press, 1991, págs. 159-180. ISBN: 0262570866.
- [7] A-L. Barabási y A. Albert. «Emergence of scaling in random networks». En: *Science* 286 (1999), págs. 509-512. DOI: [10.1126/science.286.5439.509](https://doi.org/10.1126/science.286.5439.509).
- [8] Somendra M. Bhattacharjee y Avinash Khare. «Fifty years of the exact solution of the two-dimensional Ising model by Onsager». En: *Current Science* 69 (1995), págs. 816-821. ISSN: 00113891. URL: <http://www.jstor.org/stable/24097007>.
- [9] J. Bracken. «Lanchester models of the ardennes campaign». En: *Naval Research Logistics* 42(4) (1995), págs. 559-577. DOI: [https://doi.org/10.1002/1520-6750\(199506\)42:4<559::AID-NAV3220420405>3.0.CO;2-R](https://doi.org/10.1002/1520-6750(199506)42:4<559::AID-NAV3220420405>3.0.CO;2-R).
- [10] H. Brackney. «The Dynamics of Military Combat». En: *Operations Research* 7(1) (1959), págs. 30-44. ISSN: 0030364X, 15265463. URL: <http://www.jstor.org/stable/167591>.
- [11] D.L. DeAngelis, D.K. Cox y C.C. Coutant. «Cannibalism and size dispersal in young-of-the-year largemouth bass: Experiment and model». En: *Ecological Modelling* 8 (1980), págs. 133-148. DOI: [https://doi.org/10.1016/0304-3800\(80\)90033-2](https://doi.org/10.1016/0304-3800(80)90033-2).

- [12] S. J. Deitchman. «A Lanchester Model of Guerrilla Warfare». En: *Operations Research* 10.6 (1962), págs. 818-827. DOI: [10.1287/opre.10.6.818](https://doi.org/10.1287/opre.10.6.818). eprint: <https://doi.org/10.1287/opre.10.6.818>. URL: <https://doi.org/10.1287/opre.10.6.818>.
- [13] R. Fricker. «Attrition models of the Ardennes Campaign». En: *Naval Research Logistics* 45 (feb. de 1998), págs. 1-22. DOI: [10.1002/\(SICI\)1520-6750\(199802\)45:13.0.CO;2-D](https://doi.org/10.1002/(SICI)1520-6750(199802)45:13.0.CO;2-D).
- [14] M. Gardner. «Mathematical Games - The Fantastic Combinations of John Conway's New Solitaire Game's Life». En: *Scientific American* 223 (1970), págs. 120-123. DOI: [10.1038/scientificamerican1070-120](https://doi.org/10.1038/scientificamerican1070-120).
- [15] M. Huston, D. L. DeAngelis y W. Post. «From Local to Global: The Lesson of Cellular Automata». En: *Individual-Based Models and Approaches in Ecology: Populations, Communities and Ecosystems (1st ed.)*, Chapman and Hall/CRC (1992), págs. 165-187.
- [16] M. Huston, D. L. DeAngelis y W. Post. «New Computer Models Unify Ecological Theory». En: *BioScience* 38 (1988), págs. 682-691. URL: <http://www.jstor.org/stable/1310870>.
- [17] A. Ilachinski. *Artificial War. Multiagent-Based Simulation of Combat*. (1st ed.) World Scientific, 2004. DOI: [10.1142/5531](https://doi.org/10.1142/5531).
- [18] A. V. Kuznetsov. «A simplified combat model based on a cellular automaton». En: *Journal of Computer Systems Sciences International* 56(3) (mayo de 2017), págs. 397-409. URL: <https://www.proquest.com/scholarly-journals/simplified-combat-model-based-on-cellular/docview/1914965967/se-2?accountid=17252>.
- [19] S. H. White L. H. Encinas y G. R. Sánchez A. M. Del Rey. «Modelling forest fire spread using hexagonal cellular automata». En: *Applied Mathematical Modelling* 31 (2007), págs. 1213-1227. DOI: <https://doi.org/10.1016/j.apm.2006.04.001>.
- [20] F. W. Lanchester. *Aircraft in Warfare; The Dawn of the Fourth Arm*. Constable y Co., 1916.
- [21] F. W. Lanchester. *Mathematics in Warfare, in The World of Mathematics*. Ed. Newman, J.R., Simon y Schuster, 1956, págs. 2138-2157.
- [22] C. G. Langton. «Computation at the edge of chaos: Phase transitions and emergent computation». En: *Physica D: Nonlinear Phenomena* 42 (1990), págs. 12-37. DOI: [https://doi.org/10.1016/0167-2789\(90\)90064-V](https://doi.org/10.1016/0167-2789(90)90064-V).
- [23] P. M. Morse y G. E. Kimball. *Methods of operations research*. (2nd ed.) Chapman London, 1951.
- [24] R. H. Peterson. «Letter to the Editor—On the “Logarithmic Law” of Attrition and its Application to Tank Combat». En: *Operations Research* 15.3 (1967), págs. 557-558. DOI: [10.1287/opre.15.3.557](https://doi.org/10.1287/opre.15.3.557). eprint: <https://doi.org/10.1287/opre.15.3.557>. URL: <https://doi.org/10.1287/opre.15.3.557>.

- [25] R. H. Peterson. «Methods of Tank Combat Analysis.» En: *Report of Fifth Tank Conference Held at Aberdeen, H. Goldman and G. Zeller(Eds.), Report No. 918, Ballistic Research Laboratories, Aberdeen Proving Ground, Maryland, July 1953 (AD 46 000) ()*, págs. 134-150.
- [26] James G. Taylor. *Lanchester-type models of warfare. Vol. I.* Naval Postgraduate School, 1980.
- [27] James G. Taylor. *Lanchester-type models of warfare. Vol. II.* Naval Postgraduate School, 1980.
- [28] S. F. Railsback y V. Grimm. *Agent-Based and Individual-Based Modeling: a practical introduction.* (2nd ed.) Princeton University Press, 2019.
- [29] J. Von Neumann. «The general and logical theory of automata». En: *Cerebral Mechanisms in Behaviour.* Ed. por L. A. Jeffress. Wiley, 1951.
- [30] S. Hoya White, A. Martín del Rey y G. Rodríguez Sánchez. «Modeling epidemics using cellular automata». En: *Applied Mathematics and Computation* 186(1) (2007), págs. 193-202. DOI: <https://doi.org/10.1016/j.amc.2006.06.126>.
- [31] S. Wolfram. «A New Kind of Science». En: *Champaign, IL: Wolfram Media* (2002).
- [32] S. Wolfram. «Statistical Mechanics of Cellular Automata». En: *Rev. Mod. Phys.* 55 (1983).
- [33] A. E. R. Woodcock, L. Cobb y J.T. Dockery. «Cellular Automata: A New Method for Battlefield Simulation». En: *Signal* (1988), págs. 39-45.

Apéndice A

Modelos de combate Lanchesterianos. Ampliación

Para aquellos lectores interesados, en este apéndice vamos a estudiar algunos de los modelos desarrollados tras los propuestos originalmente por Lanchester y que mayor impacto han tenido dentro de la comunidad.

A.1. Ampliación al modelo de fuego dirigido

En esta sección se expondrá una pequeña ampliación de uno de los modelos de Lanchester como ejemplo de la información que estos modelos deterministas pueden llegar a aportar a pesar de su simplicidad.

A.1.1. Táctica a seguir

Vamos a estudiar qué táctica debería seguir un ejército en una contienda dependiendo de su tamaño y su eficacia en relación con otro ejército. Supongamos que el ejército rojo tiene el doble de unidades que el ejército verde: $R_0 = 2V_0$ pero su eficacia es un tercio de la del ejército verde: $v = 3r$, entonces sustituyendo en la condición de victoria para el ejército rojo (1.7) vemos que:

$$R_0^2 - \frac{v}{r}V_0^2 = 4V_0^2 - \frac{3r}{r}V_0^2 = V_0^2 > 0 \quad \Rightarrow \quad \text{El ejército rojo gana.} \quad (\text{A.1})$$

En consecuencia, teniendo en cuenta las condiciones de victoria en situaciones particulares, aunque el ejército rojo tenía menos eficacia, al tener un mayor número de efectivos ganará la batalla. Recordemos que en este modelo (de Fuego Dirigido) las unidades de cada ejército se relacionan con las del enemigo indistintamente.

Podríamos decir que la táctica a seguir cuando se tiene menor eficacia y mayor número de unidades es una lucha todos contra todos de la manera más rápida posible. Obviamente esta táctica, como acabamos de ver en el ejemplo anterior, no es la más adecuada para el ejército verde. ¿Podría ganar el ejército verde en alguna situación? Vamos a ver qué ocurre cuando el ejército verde sigue la máxima latina *divide et impera*, es decir, de dividir al ejército contrario y luchar secuencialmente contra dichas dos fuerzas. Para ilustrar esta

situación, volvamos al ejemplo anterior en el que el ejército rojo tiene el doble de unidades que el ejército verde pero su eficacia es un tercio de la del verde. El ejército verde, para intentar ganar, luchará secuencialmente contra el ejército rojo de manera que en la primera batalla se enfrente a la mitad de las tropas de dicho ejército y en la segunda batalla contra el resto. La situación es la siguiente:

- *Primera batalla.*

Las fuerzas en disputa serán:

$$\left. \begin{array}{l} R_0 = V_0 \\ V_0 = V_0 \end{array} \right\} \quad (\text{A.2})$$

$$\Rightarrow R_0^2 - \frac{v}{r}V_0^2 = V_0^2 - \frac{3r}{r}V_0^2 = -2V_0^2 < 0 \quad \Rightarrow \quad \text{El ejército verde gana.}$$

Cuando acaba esta batalla, los soldados rojos que han participado en la primera batalla han sido aniquilados, pues es la condición de victoria del ejército verde, y nos queda la otra mitad para luchar en la segunda batalla. El número de unidades verdes que permanecen son los supervivientes de la primera batalla, que sabemos calcular gracias a la expresión (1.17).

- *Segunda batalla.*

Las fuerzas en disputa serán:

$$\left. \begin{array}{l} R_0 = V_0 \\ V_0 = \sqrt{V_0^2 - \frac{r}{3r}V_0^2} = \sqrt{\frac{2}{3}} \cdot V_0 \end{array} \right\} \quad (\text{A.3})$$

$$\Rightarrow R_0^2 - \frac{v}{r}V_0^2 = V_0^2 - \frac{3r}{r} \frac{2}{3}V_0^2 = V_0^2 - 2V_0^2 = -V_0^2 < 0 \quad \Rightarrow \quad \text{El ejército verde gana.}$$

Luego podemos concluir que la estrategia del ejército verde tiene que ser dividir al ejército rojo en dos mitades de forma que según llegue el ejército rojo, el ejército verde sea ágil a la hora de empezar a combatir y no esperar a que el ejército rojo acabe de formarse.

Una vez visto este ejemplo es lógico plantearnos la pregunta de cuál es el umbral de hombres a los que un ejército más efectivo, pero menos numeroso tiene que dividir al ejército contrario para ganar la batalla. Supongamos por tanto que el ejército rojo tiene más efectivos que el ejército verde, $R_0 > V_0$ ($R_0 = \lambda V_0$ con $\lambda > 1$) pero es menos eficaz: $r = \mu \cdot v$ con $\mu < 1$. Se puede demostrar que la condición de victoria para el ejército verde será: $\lambda < \frac{1}{\sqrt{\mu}}$. Como en nuestro caso el ejército rojo es más numeroso que el verde, la condición de victoria para el verde se dará cuando: $1 < \lambda < \frac{1}{\sqrt{\mu}}$, es decir, hay que dividir al ejército más numeroso (en este caso el rojo) en R'_0 y $R''_0 = 1 - R'_0$ de manera que se cumpla:

$$R'_0 < \frac{V_0}{\sqrt{\mu}}, \quad (\text{A.4})$$

$$R''_0 < \frac{\sqrt{V_0^2 - \mu R_0'^2}}{\sqrt{\mu}}, \quad (\text{A.5})$$

donde el enfrentamiento se produce en primer lugar con las fuerzas R'_0 . En el resto del apéndice vamos a ver distintas variantes que nacen de las ecuaciones de Lanchester.

A.2. Modelo Mixto o de Emboscada

A.2.1. Descripción y ecuaciones del modelo

El llamado *modelo mixto* fue propuesto por Brackney en 1956 [10] y estudiado posteriormente por Deitchman en 1962 [12] basándose en los modelos originales de Lanchester. Este modelo consiste en una mezcla entre un modelo de Fuego Dirigido y un modelo de Guerrilla, ya que representa una situación de combate en la que un ejército es atacado por sorpresa, es decir, emboscado. De esta manera, el ejército que embosca tendrá cierta ventaja sobre el ejército emboscado, ya que conoce la ubicación del contrincante y podrá anticipar sus objetivos y ver el daño causado. En el lado contrario, el ejército emboscado no podrá determinar el daño causado ni la localización exacta del ejército que embosca, es decir, estará usando el modelo de guerrilla. Teniendo en cuenta lo comentado anteriormente, y suponiendo que $R(t) > 0$ y $V(t) > 0$ representan el número de efectivos del ejército rojo y de la guerrilla verde en el instante de tiempo t , la dinámica de este modelo viene gobernada por el siguiente sistema de ecuaciones diferenciales ordinarias:

$$\begin{cases} \frac{dR}{dt} = -vV \\ \frac{dV}{dt} = -rVR \end{cases} \quad \text{Con } R(0) = R_0, V(0) = V_0. \quad (\text{A.6})$$

Otra posible interpretación podría ser que el ejército rojo avanza sobre un terreno abierto buscando y luchando contra el ejército verde, de modo que el ejército verde está en una posición oculta para el ejército rojo. La tasa de detección es el único factor limitante para el ejército rojo, siendo la potencia de fuego la única limitación para la guerrilla verde. Esta forma de las ecuaciones de Lanchester puede considerarse como un modelo para las primeras etapas de un enfrentamiento de unidades pequeñas. Durante esta etapa inicial, ambas unidades fueron tomadas por sorpresa y vulnerables de cara al fuego enemigo. Es una situación en la que la fuerza más grande presenta más objetivos al fuego enemigo. Es sencillo calcular la ecuación de estado del anterior problema de valor inicial:

$$\frac{rR(t)^2}{2} - vV(t) = \frac{rR_0^2}{2} - vV_0. \quad (\text{A.7})$$

Como se puede comprobar, el segundo miembro de la solución implícita (A.7) se mantiene constante con respecto al tiempo. Al igual que en los modelos anteriores, también es posible calcular las soluciones explícitas del sistema (A.6):

$$R(t) = \left(\sqrt{\frac{2vV_0}{r} - R_0^2} \right) \left[\tan \left[\arctan \left[R_0 \sqrt{\frac{r}{2vV_0 - rR_0}} \right] - \frac{t}{2} \sqrt{2vrV_0 - r^2R_0^2} \right] \right], \quad (\text{A.8})$$

$$V(t) = \left(V_0 - \frac{rR_0^2}{2v} \right) \left(1 + \left[\tan \left[\arctan \left[R_0 \sqrt{\frac{r}{2vV_0 - rR_0}} \right] - \frac{t}{2} \sqrt{2vrV_0 - r^2R_0^2} \right] \right]^2 \right) \quad (\text{A.9})$$

A.2.2. Condiciones de victoria en el modelo Mixto

Al igual que en los modelos del capítulo 1, supondremos que el combate finalizará en el instante de tiempo t_f en el que una de las dos fuerzas de combate queda aniquilada. Supongamos que es la guerrilla verde la que queda derrotada y el ejército rojo el que resulta vencedor, es decir, existe un t_f tal que $R(t_f) > 0$ y $V(t_f) = 0$ (teniendo en cuenta que t_f es el instante de tiempo más pequeño en el que ocurre esto). Si evaluamos la expresión (A.7) en dicho instante de tiempo obtenemos el siguiente resultado:

$$R(t_f) = \sqrt{R_0^2 - \frac{2v}{r}V_0}. \quad (\text{A.10})$$

Como $R(t_f) > 0$, entonces:

$$\sqrt{R_0^2 - \frac{2v}{r}V_0} > 0 \Leftrightarrow R_0^2 - \frac{2v}{r}V_0 > 0 \quad (\text{A.11})$$

Hemos llegado a una condición de victoria para el ejército rojo que depende de R_0 y V_0 . Podemos generalizar la condición de victoria del ejército rojo para cualquier instante de tiempo en el que transcurra la batalla:

$$R(t)^2 - \frac{2v}{r}V(t) > 0 \quad \forall t < t_f \quad (\text{A.12})$$

Análogamente se demuestra que la condición de victoria para la guerrilla verde es:

$$\frac{rR(t)^2}{2v} - V(t) > 0 \quad \forall t < t_f \quad (\text{A.13})$$

A.2.3. Situación tras la victoria

Finalmente calculemos el instante de tiempo t_f en el que finaliza la batalla y el número final de fuerzas vencedoras existentes en ese momento. Al igual que antes podemos suponer, sin pérdida de generalidad, que el ejército vencedor es el rojo, entonces siguiendo la expresión (A.7) al finalizar el enfrentamiento, el número de efectivos supervivientes será:

$$R(t_f) = \sqrt{R_0^2 - \frac{2v}{r}V_0}. \quad (\text{A.14})$$

Teniendo en cuenta la ecuación (A.8) el instante de tiempo t_f en el que se alcanza la victoria será:

$$t_f = \frac{\frac{\pi}{2} - 2 \arctan \left[\frac{R_0 \sqrt{\frac{r}{2vV_0 - rR_0}}}{\sqrt{2vrV_0 - r^2R_0^2}} \right]}{\sqrt{2vrV_0 - r^2R_0^2}} \quad (\text{A.15})$$

De manera análoga obtendríamos que si el ganador fuera la guerrilla verde, entonces el número de supervivientes tras vencer en el combate sería:

$$V(t_f) = \frac{rR_0^2}{2v} - V_0 \quad (\text{A.16})$$

A.3. Modelo logarítmico

A.3.1. Descripción y ecuaciones del modelo

El llamado *modelo logarítmico* fue propuesto por Peterson en 1953 [24, 25] para modelizar una batalla de tanques de la IIGM. Al igual que el modelo de fuego dirigido, describe el combate entre dos fuerzas homogéneas y los únicos factores que influyen son v y r que representan las tasas de detección. Suponiendo entonces que $R(t) > 0$ y $V(t) > 0$ representan el número de efectivos de los ejércitos rojo y verde respectivamente en el instante de tiempo t , la dinámica de este modelo viene gobernada por el siguiente sistema de ecuaciones diferenciales:

$$\left\{ \begin{array}{l} \frac{dR}{dt} = -vR \\ \frac{dV}{dt} = -rV \end{array} \right. \quad \text{Con } R(0) = R_0, \quad V(0) = V_0. \quad (\text{A.17})$$

La ecuación de estado o solución implícita del anterior problema de valor inicial será:

$$\left(\frac{R(t)}{R_0} \right)^r = \left(\frac{V(t)}{V_0} \right)^v, \quad (\text{A.18})$$

mientras que la solución explícita del sistema (A.17) será:

$$R(t) = R_0 e^{-vt}, \quad (\text{A.19})$$

$$V(t) = V_0 e^{-rt}. \quad (\text{A.20})$$

A.3.2. Condiciones de victoria en el modelo Logarítmico

Supondremos que la victoria se produce cuando uno de los dos ejércitos aniquila completamente al contrario. Si por ejemplo tomamos que el ejército verde es aniquilado por el rojo, entonces existirá un tiempo t_f tal que $V(t_f) = 0$ y $R(t_f) > 0$, siendo t_f el instante de tiempo más pequeño en el que ocurre esto. Observemos que el comportamiento de las soluciones obtenida es el siguiente:

$$R(t) = R_0 e^{-vt} \Rightarrow \frac{dR}{dt} = -vR_0 e^{-vt} < 0 \Rightarrow R(t) \text{ es decreciente.} \quad (\text{A.21})$$

$$V(t) = V_0 e^{-rt} \Rightarrow \frac{dV}{dt} = -rV_0 e^{-rt} < 0 \Rightarrow V(t) \text{ es decreciente.} \quad (\text{A.22})$$

Por tanto, el resultado de victoria por aniquilación resulta ser un resultado asintótico:

$$\lim_{t \rightarrow \infty} (R(t)) = \lim_{t \rightarrow \infty} R_0 e^{-vt} = R_0 \cdot 0 = 0 \quad (\text{A.23})$$

$$\lim_{t \rightarrow \infty} (V(t)) = \lim_{t \rightarrow \infty} V_0 e^{-rt} = V_0 \cdot 0 = 0 \quad (\text{A.24})$$

Consecuentemente dicho límite no se alcanzará en tiempo finito, luego la definición de victoria que debemos imponer debe ser algo distinta: El ejército rojo gana cuando existe un t_f tal que:

$$V(t_f) = u, \quad R(t_f) > u \text{ y } u > 0. \quad (\text{A.25})$$

donde u será el número de unidades que consideremos adecuada para otorgar la victoria. Empleando la expresión (A.18) obtenemos que la condición para que resulte ganador el ejército rojo es:

$$\frac{R_0^r}{V_0^v} > u^{r-v} \quad (\text{A.26})$$

Análogamente, el ejército verde vencerá cuando se cumpla

$$\frac{V_0^v}{R_0^r} > u^{v-r} \quad (\text{A.27})$$

A.3.3. Situación tras la victoria

Finalmente calculemos el instante de tiempo t_f en el que finaliza la batalla y el número de fuerzas vencedoras existentes en ese momento. t_f es el instante de tiempo en el que $V(t_f) = u$, entonces utilizando la solución explícita obtenemos:

$$t_f = -\frac{1}{r} \log \left(\frac{u}{V_0} \right) \quad (\text{A.28})$$

Como hemos supuesto ya que el ejército rojo gana, veamos ahora cuál es su tamaño final $R(t_f)$. Utilizando el valor de t_f y la expresión explícita de la solución obtenemos tras un sencillo cálculo:

$$R(t_f) = \frac{uR_0}{V_0} e^{\frac{v}{r}} \quad (\text{A.29})$$

De una manera análoga se calcula el tamaño del ejército verde y el instante de la victoria en el caso de victoria verde:

$$u = V(t_f) = V_0 e^{-vt_f}, \quad (\text{A.30})$$

$$t_f = -\frac{1}{v} \log \frac{u}{V_0}. \quad (\text{A.31})$$

A.4. Modelo Universal de Lanchester

A.4.1. Descripción y ecuaciones del modelo

El llamado *modelo universal de Lanchester* fue empleado por Bracken en 1995 [9] con el objetivo de modelizar el resultado de la campaña de las Ardenas durante la IIGM. La idea detrás de este tipo de modelos es lidiar con la falta de éxito de los modelos de fuego dirigido y no dirigido a la hora de modelizar ciertas situaciones de combate, ya que se estos se basan en suposiciones previas y así poder ajustar los parámetros del modelo a partir de los datos de los que disponemos del enfrentamiento. También cabe citar el trabajo posterior hecho por Ronald D. Fricker [13]. Puesto que este tipo de modelos se

basan en el ajuste de datos, vamos a ver brevemente las principales características en un modelo generalizado. Suponiendo que $R(t) >$ y $V(t) >$ representan el número de efectivos de los ejércitos rojo y verde respectivamente en el instante de tiempo t , la dinámica de este modelo viene dada por el siguiente sistema de ecuaciones diferenciales ordinarias:

$$\left\{ \begin{array}{l} \frac{dR}{dt} = -vR^qV^p \\ \frac{dV}{dt} = -rR^pV^q \end{array} \right. \quad \text{Con } R(0) = R_0, \quad V(0) = V_0, \quad (\text{A.32})$$

donde p y q son parámetros a determinar empíricamente a través de observaciones históricas o experimentación. Como en modelos anteriores, es posible calcular la ecuación de estado o la solución implícita del anterior problema de valor inicial:

$$rR^{p-q+1} - vV^{p-q+1} = rR_0^{p-q+1} - vV_0^{p-q+1}. \quad (\text{A.33})$$

El segundo miembro de la solución (A.33) se mantiene constante con el transcurrir del tiempo. En este modelo no se puede obtener una solución explícita que nos del valor de $R(t)$ y $V(t)$ de una forma general.

A.4.2. Condiciones de victoria en el modelo Universal de Lanchester

Al igual que en los modelos anteriores, supondremos que el combate finalizará en el instante de tiempo t_f en el que uno de los dos ejércitos quede aniquilado. Supongamos, sin pérdida de generalidad, que es el ejército rojo el que resulta vencedor y el ejército verde el que queda derrotado; entonces teniendo en cuenta lo comentado en la sección anterior existe un t_f tal que $V(t_f) = 0$ y $R(t_f) > 0$, dónde t_f es el instante de tiempo más pequeño en el que ocurre esto. Imponiendo estas condiciones en la expresión (A.33) obtenemos la condición de victoria para el ejército rojo:

$$\left(\frac{R_0}{V_0} \right)^{p-q+1} > \frac{v}{r}. \quad (\text{A.34})$$

De manera análoga, en caso de victoria del ejército verde la condición será:

$$\left(\frac{R_0}{V_0} \right)^{p-q+1} < \frac{v}{r}. \quad (\text{A.35})$$

Como hemos visto en los modelos anteriores, estas expresiones las podemos reformular en el sentido de que no se tenga en cuenta qué sucede en el instante inicial. Así, las nuevas condiciones de victoria del ejército rojo serían:

$$rR(t)^{p-q+1} - vV(t)^{p-q+1} > 0 \quad (\text{A.36})$$

ó

$$\left(\frac{R(t)}{V(t)} \right)^{p-q+1} > \frac{v}{r}. \quad (\text{A.37})$$

A.4.3. Situación tras la victoria

Finalmente calculemos el instante de tiempo t_f en el que finaliza la batalla y el número de fuerzas vencedoras existentes en ese momento. Al igual que antes podemos suponer, sin pérdida de generalidad, que el ejército vencedor es el rojo, entonces:

$$\lim_{t \rightarrow \infty} (R(t)) = R_0^{p-q+1} - \frac{v}{r} V_0^{p-q+1}, \quad (\text{A.38})$$

con lo que $R_0^{p-q+1} - \frac{v}{r} V_0^{p-q+1}$ es el tamaño del ejército rojo tras resultar ganador. De manera análoga demostraríamos que, si el vencedor es el ejército verde, entonces

$$\lim_{t \rightarrow \infty} (V(t)) = V_0^{p-q+1} - \frac{r}{v} R_0^{p-q+1}. \quad (\text{A.39})$$

A.5. Modelo de Morse y Kimball

A.5.1. Descripción y ecuaciones del modelo

Para finalizar el apéndice se comentará brevemente el modelo de Morse y Kimball, aunque no se desarrollarán los apartados anteriores debido a su dificultad y a que el objetivo de este apéndice es únicamente dar a conocer estos modelos para el lector interesado. En 1951 Philip M. Morse y George E. Kimball publicaron *Methods of operations research* dónde recogían algunos de los resultados obtenidos por Morse para la marina de los EEUU durante la IIGM [23]. Uno de los modelos planteados y que a continuación estudiaremos permite extender las situaciones de combate bajo dos hipótesis:

1. Pérdidas operacionales de dos fuerzas homogéneas en combate. Las pérdidas operacionales son bajas provocadas por causas ajenas a la acción enemiga como por ejemplo las debidas a accidentes, deserciones, enfermedades, etc.
2. Dos fuerzas homogéneas primarias en combate a las que se le suma la acción de fuerzas de apoyo a las que el enemigo debe detectar. Un ejemplo podría ser infantería (primaria) con fuerzas de apoyo que emplean el modelo de fuego de área sobre el enemigo.

Las ecuaciones que rigen el modelo son:

$$\left\{ \begin{array}{l} \frac{dR}{dt} = -aV - \beta R \\ \frac{dV}{dt} = -bR - \alpha V \end{array} \right. \quad \text{Con } R(0) = R_0, \quad V(0) = V_0, \quad (\text{A.40})$$

donde a y b serán el ritmo con el que se producen las bajas para un modelo de fuego dirigido mientras que β y α serían los esperados en un modelo de guerrilla o de fuego de área. Como en modelos anteriores, es posible calcular la ecuación de estado del anterior

problema de valor inicial, sin embargo generalmente resultará mucho más complejo:

$$\begin{aligned}
 & V(t) \left(\theta + \left(\frac{\beta - \alpha}{2} \right) \right) - bR(t) = \\
 & \left[V_0 \left(\theta + \left(\frac{\beta - \alpha}{2} \right) \right) - bR_0 \right] \left[\frac{V_0 \left(\theta - \left(\frac{\beta - \alpha}{2} \right) \right) + bR_0}{V(t) \left(\theta - \left(\frac{\beta - \alpha}{2} \right) \right) + bR(t)} \right]^v \quad (\text{A.41}) \\
 & \text{con } \theta = \sqrt{ab + [(\beta - \alpha)/2]^2}, \quad v = (\theta - (\alpha + \beta)/2) / (\theta + (\alpha + \beta)/2).
 \end{aligned}$$

Apéndice B

Código de Mathematica desarrollado

En este apéndice se muestra el código empleado en las simulaciones realizadas del modelo propuesto.

Programa Principal.

Universidad de Salamanca - Máster Universitario en Modelización Matemática

Trabajo de Fin De Máster: Diseño de un Modelo de Combate de Naturaleza Individual.

Autor: Marcos López De Castro.

```
(*ANTES DE EVALUAR ESTE CUADERNO HAY QUE
EVALUAR EL CUADERNO Modulos.nb O NO FUNCIONARÁ*)
|notación O

n = 10;           (* Numero de celdas por lado *)
timesteps = 100; (* Iteracciones (Combate y movimiento) *)
DisposicionInicial = 2; (*Distribución inicial de las unidades*)
Verdes = 48;     (* Número de unidades verdes desplegadas*)
Rojos = 40;     (* Número de unidades verdes desplegadas*)
NumberPath = 10; (* Número de simulaciones a realizar *)
eps = 0.008;
(*Establecemos el peso de la corrección a la parte Ising (epsilon)*)

Gpaths = {};
Rpaths = {};

(* Comienzo de la simulación *)
Do [
|repite
  ClearAll[Matrix, MatrixAuxiliar, SoldadosVerdes,
|borra todo
  SoldadosRojos, Posiciones, PosicionesV, PosicionesR];
  Matrix = {};
  SoldadosVerdes = {};
  SoldadosRojos = {};
  Posiciones = {};
  PosicionesV = {};
  PosicionesR = {};

  (*Calculamos la disposición inicial
  de las unidades en función de la escogida*)
  Which [
|cuál
```

```

[...
DisposicionInicial = 1, (* Número y unidades desplegadas aleatoria *)
Do[AppendTo[Matrix, RandomInteger[{-1, 1}, n]], n],
[... [añade al final] [entero aleatorio]

DisposicionInicial = 2, (* Unidades desplegadas aleatoriamente *)
Matrix = ConstantArray[0, {n, n}];
[... [arreglo constante]
Do[AppendTo[Posiciones, {i, j}], {i, 1, n}, {j, 1, n}];
[... [añade al final]
PosicionesV = RandomSample[Posiciones, Verdes];
[... [muestra aleatoria]
Posiciones = Complement[Posiciones, PosicionesV];
[... [complemento]
PosicionesR = RandomSample[Posiciones, Rojos];
[... [muestra aleatoria]
Do[Matrix[[PosicionesV[[i, 1]], PosicionesV[[i, 2]]]] = 1, {i, 1, Verdes}];
[... [repite]
Do[Matrix[[PosicionesR[[i, 1]], PosicionesR[[i, 2]]]] = -1, {i, 1, Rojos}];
[... [repite]

DisposicionInicial = 3, (* Escenario de emboscada *)
Matrix = ConstantArray[0, {n, n}];
[... [arreglo constante]
Do[AppendTo[Posiciones, {i, j}], {i, 2, n - 1}, {j, 2, n - 1}];
[... [añade al final]
k = 1;
While[Length[PosicionesV] < Verdes,
[... [mient... [longitud]
ClearAll[PosicionesV]; PosicionesV = {};
[... [borra todo]
Do[AppendTo[PosicionesV, {i, j}], {i, Round[n/2 - k * n/8],
[... [añade al final] [entero más próximo]
Round[n/2 + k * n/8]}, {j, Round[n/2 - k * n/8], Round[n/2 + k * n/8]}}];
[... [entero más próximo] [entero más próximo] [entero más próximo]
k = k + 1];
Posiciones = Complement[Posiciones, PosicionesV];
[... [complemento]
PosicionesR = RandomSample[Posiciones, Rojos];
[... [muestra aleatoria]
Do[Matrix[[PosicionesV[[i, 1]], PosicionesV[[i, 2]]]] = 1, {i, 1, Verdes}];
[... [repite]
Do[Matrix[[PosicionesR[[i, 1]], PosicionesR[[i, 2]]]] = -1, {i, 1, Rojos}];
[... [repite]
];

(*Guardamos el número de unidades iniciales*)
VerdesIniciales = Count[Matrix, 1, 2];
[... [conteo]
AppendTo[SoldadosVerdes, VerdesIniciales];
[... [añade al final]
RojosIniciales = Count[Matrix, -1, 2];
[... [conteo]

```

```

AppendTo[SoldadosRojos, RojosIniciales];
[añade al final

plot = {};
grafosV = {};
grafosR = {};

(*Calculo de redes complejas y
medidas de centralidad de la distribución inicial*)
PerVerdes = ClusterGraphs[n, Matrix, 1];
PerRojos = ClusterGraphs[n, Matrix, -1];

(*Plot de la situación inicial*)
[representación gráfica
AppendTo[plot,
[añade al final
ArrayPlot[Matrix, ColorRules -> {0 -> White, 1 -> Green, -1 -> Red}, Mesh -> True]];
[representación de arre... [reglas de color [blanco [verde [rojo [malla [verdadero
AppendTo[grafosV, PerVerdes];
[añade al final
AppendTo[grafosR, PerRojos];
[añade al final

Do[
[repite
ClearAll[MatrixAuxiliar];
[borra todo
MatrixAuxiliar = Matrix;
PerVerdes = ClusterGraphs[n, MatrixAuxiliar, 1];
PerRojos = ClusterGraphs[n, MatrixAuxiliar, -1];

Do[ Do[
[re... [repite
Aleatoriofila = i;
Aleatoriocolumna = j;
Elemento = Matrix[[Aleatoriocolumna, Aleatoriofila]];

(*FASE DE COMBATE*)

(*Chequeamos su vecindad (en este caso Moore) e imponemos CC nulas*)
If[Aleatoriocolumna - 1 == 0, Norte = 0,
[si
Norte = Matrix[[Aleatoriocolumna - 1, Aleatoriofila]];
If[Aleatoriocolumna + 1 > n, Sur = 0,
[si
Sur = Matrix[[Aleatoriocolumna + 1, Aleatoriofila]];
If[Aleatoriofila - 1 == 0, Oeste = 0,
[si
Oeste = Matrix[[Aleatoriocolumna, Aleatoriofila - 1]];
If[Aleatoriofila + 1 > n, Este = 0,
[si
Este = Matrix[[Aleatoriocolumna, Aleatoriofila + 1]];

```

```

If[Aleatoriocolumna - 1 == 0 || Aleatoriofila - 1 == 0, NorOeste = 0,
  [si
  NorOeste = Matrix[[Aleatoriocolumna - 1, Aleatoriofila - 1]];
If[Aleatoriocolumna - 1 == 0 || Aleatoriofila + 1 > n, NorEste = 0,
  [si
  NorEste = Matrix[[Aleatoriocolumna - 1, Aleatoriofila + 1]];
If[Aleatoriocolumna + 1 > n || Aleatoriofila - 1 == 0, SurOeste = 0,
  [si
  SurOeste = Matrix[[Aleatoriocolumna + 1, Aleatoriofila - 1]];
If[Aleatoriocolumna + 1 > n || Aleatoriofila + 1 > n, SurEste = 0,
  [si
  SurEste = Matrix[[Aleatoriocolumna + 1, Aleatoriofila + 1]];
(*Calculo de la parte Ising del modelo *)
IsingPart = Elemento *
  (Norte + Sur + Este + Oeste + NorOeste + SurOeste + NorEste + SurEste);

(*Correcciones a la parte Ising: Sigma y Gamma*)
[gamma de Euler

If[Aleatoriocolumna - 1 == 0 ||
  [si
  MatrixAuxiliar[[Aleatoriocolumna - 1, Aleatoriofila]] == 0,
  IsingNorte = Infinity, IsingNorte =
[infinito
  FuncionChequeo[Matrix, n, Aleatoriofila, Aleatoriocolumna - 1]];

If[Aleatoriocolumna + 1 > n || MatrixAuxiliar[[
  [si
  Aleatoriocolumna + 1, Aleatoriofila]] == 0, IsingSur = Infinity,
[infinito
  IsingSur = FuncionChequeo[Matrix, n, Aleatoriofila, Aleatoriocolumna + 1]];

If[Aleatoriofila - 1 == 0 ||
  [si
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila - 1]] == 0,
  IsingOeste = Infinity, IsingOeste = FuncionChequeo[
[infinito
  Matrix, n, Aleatoriofila - 1, Aleatoriocolumna]];

If[Aleatoriofila + 1 > n || MatrixAuxiliar[[
  [si
  Aleatoriocolumna, Aleatoriofila + 1]] == 0, IsingEste = Infinity,
[infinito
  IsingEste = FuncionChequeo[Matrix, n, Aleatoriofila + 1, Aleatoriocolumna]];

If[Aleatoriocolumna - 1 == 0 || Aleatoriofila - 1 == 0 ||
  [si
  MatrixAuxiliar[[Aleatoriocolumna - 1, Aleatoriofila - 1]] == 0,
  IsingNorOeste = Infinity, IsingNorOeste =
[infinito
  FuncionChequeo[Matrix, n, Aleatoriofila - 1, Aleatoriocolumna - 1]];

```

```

If[Aleatoriocolumna - 1 == 0 || Aleatoriofila + 1 > n ||
[si
  MatrixAuxiliar[[Aleatoriocolumna - 1, Aleatoriofila + 1]] == 0,
  IsingNorEste = Infinity, IsingNorEste =
[infinito
  FuncionChequeo[Matrix, n, Aleatoriofila + 1, Aleatoriocolumna - 1]];

If[Aleatoriocolumna + 1 > n || Aleatoriofila - 1 == 0 ||
[si
  MatrixAuxiliar[[Aleatoriocolumna + 1, Aleatoriofila - 1]] == 0,
  IsingSurOeste = Infinity, IsingSurOeste =
[infinito
  FuncionChequeo[Matrix, n, Aleatoriofila - 1, Aleatoriocolumna + 1]];

If[Aleatoriocolumna + 1 > n || Aleatoriofila + 1 > n ||
[si
  MatrixAuxiliar[[Aleatoriocolumna + 1, Aleatoriofila + 1]] == 0,
  IsingSurEste = Infinity, IsingSurEste =
[infinito
  FuncionChequeo[Matrix, n, Aleatoriofila + 1, Aleatoriocolumna + 1]];

(*Tenemos el término Ising de cada elemento de la vecindad*)

ClearAll[ListaIsing, NumEleVecindad, Gama, Sigma, Listaposicion];
[borra todo

(*Pasamos a considerar solo enemigos de elemento,
luego, como ya hemos almacenado la info de la vecindad,
tratamos a los aliados como casillas vacias*)
If[Elemento == NorOeste, IsingNorOeste = Infinity];
[si [infinito
If[Elemento == Norte, IsingNorte = Infinity];
[si [infinito
If[Elemento == NorEste, IsingNorEste = Infinity];
[si [infinito
If[Elemento == SurOeste, IsingSurOeste = Infinity];
[si [infinito
If[Elemento == Sur, IsingSur = Infinity];
[si [infinito
If[Elemento == SurEste, IsingSurEste = Infinity];
[si [infinito
If[Elemento == Oeste, IsingOeste = Infinity];
[si [infinito
If[Elemento == Este, IsingEste = Infinity];
[si [infinito

ListaIsing = {IsingNorOeste, IsingNorte, IsingNorEste,
  IsingEste, IsingSurEste, IsingSur, IsingSurOeste, IsingOeste};
(*El orden de ListaIsing es importante, aqui se almacena el
término Ising de los elementos de la vecindad enemiga*)

(*Calculo de \sigma *)
Listaposicion = {};

```

```

Do[If[ListaIsing[[k]] ≠ Infinity, AppendTo[Listaposicion, 1],
  r... [si [infinito [añade al final
    AppendTo[Listaposicion, 0]], {k, 1, Length[ListaIsing] }];
  [añade al final [longitud
NSigma = MideAngulos[Listaposicion];
Sigma = 0;
Omega = {-0.25, -0.5, -0.75, -1}; (*Establecemos los pesos Omega*)
Do[Sigma = Sigma + Omega[[k]] * NSigma[[k]], {k, 1, 4}];
[repite
(*Calculo de \Gamma *)
  [gamma de Euler
NumEleVecindad = 8 - Count[ListaIsing, Infinity];
  [conteo [infinito
(*Contamos el numero de enemigos*)
Do[If[ListaIsing[[k]] == Infinity, ListaIsing[[k]] = 0],
  r... [si [infinito
  {k, 1, Length[ListaIsing] }];
  [longitud
If[NumEleVecindad != 0, Gama = Total[ListaIsing] / NumEleVecindad, Gama = 0];
[si [total

Which[
  [cuál
  Elemento == 1,
  CorrectionCoeficient = Sigma * Gama,
  Elemento == -1,
  CorrectionCoeficient = Sigma * Gama,
  Elemento == 0,
  CorrectionCoeficient = 0
];

(*FUNCIÓN DE TRANSICIÓN DE COMBATE *)
H = IsingPart + eps * CorrectionCoeficient;

(*Vive/Muere*)
random = RandomReal[{-8, 8}];
  [real aleatorio
(*Numero aleatorio para ver si aceptamos el movimiento*)
If[N[H] < random, MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = 0,
[si [valor numérico
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = Elemento];
(*Si un elemento está en blanco debe seguir estándolo
al finalizar la fase de combate*)
If[Elemento == 0, MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = 0];
[si
(*Si no está rodeado por nadie o solo por aliados no se muere*)
If[Elemento * Norte ≥ 0 && Elemento * Sur ≥ 0 &&
[si
  Elemento * Este ≥ 0 && Elemento * Oeste ≥ 0 && Elemento * NorOeste ≥ 0 &&
  Elemento * NorEste ≥ 0 && Elemento * SurEste ≥ 0 && Elemento * SurOeste ≥ 0,

```

```

MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = Elemento];

, {i, 1, n}], {j, 1, n}];

AppendTo[plot, ArrayPlot[MatrixAuxiliar,
  añade al final representación de arreglo
  ColorRules -> {0 -> White, 1 -> Green, -1 -> Red}, Mesh -> True]];
  reglas de color blanco verde rojo malla verdadero
AppendTo[grafosV, PerVerdes];
  añade al final
AppendTo[grafosR, PerRojos];
  añade al final
Matrix = MatrixAuxiliar;
AppendTo[SoldadosVerdes, Count[Matrix, 1, 2]];
  añade al final conteo
AppendTo[SoldadosRojos, Count[Matrix, -1, 2]];
  añade al final conteo

(* FASE DE MOVIMIENTO *)

ClearAll[MedidasVerdes, MedidasRojas,
  borra todo
  GreenArmyNodeDegree, GreenArmyEigenCentrality, RedArmyNodeDegree,
  RedArmyEigenCentrality, Indices, EigencentalityMatrix];
(* Calculamos las medidas de centralidad de las redes complejas asociadas *)
MedidasVerdes = ComplexNetworkMeasuresMovPhase[n, Matrix, 1];
MedidasRojas = ComplexNetworkMeasuresMovPhase[n, Matrix, -1];

GreenArmyNodeDegree = MedidasVerdes[[1]];
GreenArmyEigenCentrality = MedidasVerdes[[2]];
GreenArmyClusterCoefficient = MedidasVerdes[[3]];
GreenArmyCercania = MedidasVerdes[[4]];
GreenArmyIntermediacion = MedidasVerdes[[5]];

RedArmyNodeDegree = MedidasRojas[[1]];
RedArmyEigenCentrality = MedidasRojas[[2]];
RedArmyClusterCoefficient = MedidasRojas[[3]];
RedArmyCercania = MedidasRojas[[4]];
RedArmyIntermediacion = MedidasRojas[[5]];

(*Calculamos el Ranking*)
IndexMatrix =
  (GreenArmyNodeDegree + GreenArmyEigenCentrality + GreenArmyClusterCoefficient +
  GreenArmyCercania + GreenArmyIntermediacion + RedArmyNodeDegree +
  RedArmyEigenCentrality + RedArmyClusterCoefficient +
  RedArmyCercania + RedArmyIntermediacion) / 5;
Indices = IndexRank[n, IndexMatrix];

(* Empezamos con el movimiento*)
MatrixAuxiliar = Matrix;
Do[
  write

```



```

[repite
Aleatoriofila = Indices[[i]][[2]];
Aleatoriocolumna = Indices[[i]][[1]];
Elemento = Matrix[Aleatoriocolumna, Aleatoriofila]];

If[Elemento ≠ 0,
[si
(*Comprobamos si la unidad está aislada o no*)
contaje = Count[MideAislado[MatrixAuxiliar,
[conteo
n, Aleatoriofila, Aleatoriocolumna], Elemento]];
If[contaje == 0 || contaje == 1, Aislado = "Si", Aislado = "No"];
[si
If[Aislado == "No",
[si

(* Calculamos el H de las casillas de alrededor *)

If[Aleatoriocolumna - 1 == 0, IsingNorte = Infinity,
[si [infinito
If[ MatrixAuxiliar[[Aleatoriocolumna - 1, Aleatoriofila]] ≠ 0,
[si
IsingNorte = Infinity, IsingNorte = FuncionChequeoMov[MatrixAuxiliar,
[infinito
n, Aleatoriofila, Aleatoriocolumna - 1, Elemento]]];
If[Aleatoriocolumna + 1 > n, IsingSur = Infinity,
[si [infinito
If[ MatrixAuxiliar[[Aleatoriocolumna + 1, Aleatoriofila]] ≠ 0,
[si
IsingSur = Infinity, IsingSur = FuncionChequeoMov[MatrixAuxiliar,
[infinito
n, Aleatoriofila, Aleatoriocolumna + 1, Elemento]]];
If[Aleatoriofila - 1 == 0, IsingOeste = Infinity,
[si [infinito
If[ MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila - 1]] ≠ 0,
[si
IsingOeste = Infinity, IsingOeste = FuncionChequeoMov[MatrixAuxiliar,
[infinito
n, Aleatoriofila - 1, Aleatoriocolumna, Elemento]]];
If[Aleatoriofila + 1 > n, IsingEste = Infinity,
[si [infinito
If[ MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila + 1]] ≠ 0,
[si
IsingEste = Infinity, IsingEste = FuncionChequeoMov[MatrixAuxiliar,
[infinito
n, Aleatoriofila + 1, Aleatoriocolumna, Elemento]]];
If[Aleatoriocolumna - 1 == 0 || Aleatoriofila - 1 == 0 ||
[si
MatrixAuxiliar[[Aleatoriocolumna - 1, Aleatoriofila - 1]] ≠ 0,
IsingNorOeste = Infinity, IsingNorOeste = FuncionChequeoMov[
[infinito
MatrixAuxiliar, n, Aleatoriofila - 1, Aleatoriocolumna - 1, Elemento]]];

```

```

If[Aleatoriocolumna - 1 == 0 || Aleatoriofila + 1 > n ||
_si
  MatrixAuxiliar[[Aleatoriocolumna - 1, Aleatoriofila + 1]] ≠ 0,
  IsingNorEste = Infinity, IsingNorEste = FuncionChequeoMov[
    infinito
    MatrixAuxiliar, n, Aleatoriofila + 1, Aleatoriocolumna - 1, Elemento]];

If[Aleatoriocolumna + 1 > n || Aleatoriofila - 1 == 0 ||
_si
  MatrixAuxiliar[[Aleatoriocolumna + 1, Aleatoriofila - 1]] ≠ 0,
  IsingSurOeste = Infinity, IsingSurOeste = FuncionChequeoMov[
    infinito
    MatrixAuxiliar, n, Aleatoriofila - 1, Aleatoriocolumna + 1, Elemento]];

If[Aleatoriocolumna + 1 > n || Aleatoriofila + 1 > n ||
_si
  MatrixAuxiliar[[Aleatoriocolumna + 1, Aleatoriofila + 1]] ≠ 0,
  IsingSurEste = Infinity, IsingSurEste = FuncionChequeoMov[
    infinito
    MatrixAuxiliar, n, Aleatoriofila + 1, Aleatoriocolumna + 1, Elemento]];

ClearAll[ListaIsing];
_borra todo
(*Lista con las posiciones accesibles para
 mover y el cálculo de la parte Ising en esa posición *)
ListaIsing = {IsingNorte, IsingSur, IsingEste, IsingOeste,
  IsingNorOeste, IsingNorEste, IsingSurOeste, IsingSurEste};

Do[If[ListaIsing[[k]] == Infinity, ListaIsing[[k]] = -10],
_r..._si_ infinito
  {k, 1, Length[ListaIsing]}];
_longitud

If[Count[ListaIsing, -10] == 8, Replazo = 0, Replazo = Max[ListaIsing]],
_si_ conteo_ máximo
(*CUIDADO CON LA COMA AQUI!!!!!!!!!!!!!!*)

(*Movimiento unidades aisladas*)
MovAleatorio = RandomInteger[{1, 8}];
_entero aleatorio

ClearAll[check];
_borra todo
check = ChequeoContorno[MatrixAuxiliar,
  n, Aleatoriofila, Aleatoriocolumna, MovAleatorio];

If[check == 0, MovAleatorio = 0];
_si
IsingNorte = 10;
IsingSur = 10;
IsingEste = 10;
IsingOeste = 10; IsingNorOeste = 10; IsingNorEste = 10; IsingSurOeste = 10;
IsingSurEste = 10;
Which[
_cuál

```

```

[cuál
  MovAleatorio == 1, IsingNorOeste = MovAleatorio;
  Replazo = MovAleatorio,
  MovAleatorio == 2, IsingNorte = MovAleatorio;
  Replazo = MovAleatorio,
  MovAleatorio == 3, IsingNorEste = MovAleatorio;
  Replazo = MovAleatorio,
  MovAleatorio == 4, IsingOeste = MovAleatorio;
  Replazo = MovAleatorio,
  MovAleatorio == 5, IsingEste = MovAleatorio;
  Replazo = MovAleatorio,
  MovAleatorio == 6, IsingSurOeste = MovAleatorio;
  Replazo = MovAleatorio,
  MovAleatorio == 7, IsingSur = MovAleatorio;
  Replazo = MovAleatorio,
  MovAleatorio == 8, IsingSurEste = MovAleatorio;
  Replazo = MovAleatorio,
  MovAleatorio == 0, Replazo = 0];

];

(*Replazo de casillas tras establecerse el movimiento a realizar*)
Which[
cuál
  Replazo == IsingNorte,
  MatrixAuxiliar[[Aleatoriocolumna - 1, Aleatoriofila]] = Elemento;
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = 0,

  Replazo == IsingSur,
  MatrixAuxiliar[[Aleatoriocolumna + 1, Aleatoriofila]] = Elemento;
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = 0,

  Replazo == IsingEste,
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila + 1]] = Elemento;
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = 0,

  Replazo == IsingOeste,
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila - 1]] = Elemento;
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = 0,

  Replazo == IsingNorOeste,
  MatrixAuxiliar[[Aleatoriocolumna - 1, Aleatoriofila - 1]] = Elemento;
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = 0,

  Replazo == IsingNorEste,
  MatrixAuxiliar[[Aleatoriocolumna - 1, Aleatoriofila + 1]] = Elemento;
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = 0,

  Replazo == IsingSurOeste,
  MatrixAuxiliar[[Aleatoriocolumna + 1, Aleatoriofila - 1]] = Elemento;
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = 0,

  Replazo == IsingSurEste,
  MatrixAuxiliar[[Aleatoriocolumna + 1, Aleatoriofila + 1]] = Elemento;

```

```

MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] = 0,

Replazo == {},
MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] =
  Matrix[[Aleatoriocolumna, Aleatoriofila]],

Replazo == 0,
MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] =
  Matrix[[Aleatoriocolumna, Aleatoriofila]]
];

, MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]] =
  MatrixAuxiliar[[Aleatoriocolumna, Aleatoriofila]]];

, {i, 1, n^2}];

Matrix = MatrixAuxiliar;
AppendTo[plot, ArrayPlot[Matrix,
  añade al final representación de arreglo
  ColorRules -> {0 -> White, 1 -> Green, -1 -> Red}, Mesh -> True]];
  reglas de color blanco verde rojo malla verdadero

If[Mod[t, 10] == 0,
  si operación módulo
  Print["iteration number: ", t, "/", timesteps]];
  escribe
, {t, 1, timesteps}]; (*Fin de la simulación*)

Print["Initial Green Army Combat Network:", grafosV[[1, 1]]];
  escribe verde

Print["Initial Red Army Combat Network:", grafosR[[1, 1]]];
  escribe rojo

Color = RandomColor[90];
  color aleatorio

G = ListPlot[SoldadosVerdes, PlotStyle -> Color,
  representación de lista estilo de representación
  AxesLabel -> {"Time", "Number of Green Alive Soldiers"},
  etiqueta de ejes número verde
  PlotRange -> {{0, timesteps + 2}, {0, Max[SoldadosRojos, SoldadosVerdes] + 5}},
  rango de representación máximo
  PlotStyle -> ScientificForm, Joined -> True];
  estilo de repr... forma científica unido verdadero

R = ListPlot[SoldadosRojos, PlotStyle -> Color,
  representación de lista estilo de representación
  AxesLabel -> {"Time", "Number of Red Alive Soldiers"},
  etiqueta de ejes número rojo
  PlotRange -> {{0, timesteps + 2}, {0, Max[SoldadosRojos, SoldadosVerdes] + 5}},
  rango de representación máximo
  PlotStyle -> ScientificForm, Joined -> True];
  estilo de repr... forma científica unido verdadero

```

```

AppendTo[Gpaths, G];
AppendTo[Rpaths, R];
, {kk, 1, NumberPath}]
(*Fin de todas las simulaciones en caso de que Numerpath > 1*)

Show[Gpaths]
Show[Rpaths]
Print["Final Green Army Combat Network:"]
grafosV[[timesteps, 1]]
Print["Final Red Army Combat Network:"]
grafosR[[timesteps, 1]]

(*Creamos el video la simulación*)
ListAnimate[Table[plot[[t]], {t, 1, Length[plot]}],
DefaultDuration → 0.2 * timesteps]

```

Modulos auxiliares.

Autor: Marcos López De Castro

Los módulos que se exponen a continuación deben de ser ejecutados previamente al programa principal.

```

(*Limpiamos todas las variables antes de ejecutar*)
Clear["Global`*"]
(*Este módulo calcula la matriz de adyacencia a partir
de la situación de las células, las medidas de centralidad y
asocia a cada célula su medida de centralidad correspondiente*)
ClusterGraphs[n_, matrix_, soldierNumber_] :=
Module[{Adjacent, color, elemento, AdjacentList, grafo,
subgrafo, MatrixClosenessCentrality, MatrixLocalClustering,
aux1, aux2, ClosCen, LocalClust, salida},
salida = {};
Adjacent = {};
color = {};
MatrixClosenessCentrality = {};

```

```

MatrixLocalClustering = {};
Do[Do[
  elemento = j + (i - 1) * n; (*Mas importante,
  es lo que codifica el i,j como un k en la lista*)
  ClearAll[AdjacentList];
  AdjacentList = {};
  Do[Do[
    Which[{l, m} == {i, j},
      AppendTo[AdjacentList, 0],
      Abs[m - j] > 1,
      AppendTo[AdjacentList, 0],
      Abs[l - i] > 1,
      AppendTo[AdjacentList, 0],
      matrix[[l, m]] == soldierNumber,
      AppendTo[AdjacentList, 1],
      matrix[[l, m]] != soldierNumber,
      AppendTo[AdjacentList, 0]
    ];
    , {m, 1, n}], {l, 1, n}];
  If[matrix[[i, j]] != soldierNumber, ClearAll[AdjacentList];
  AdjacentList = Table[0, n^2], AppendTo[color, elemento]];
  AppendTo[Adjacent, AdjacentList];
  , {j, 1, n}], {i, 1, n}];
If[soldierNumber == 1,
  grafo = AdjacencyGraph[Adjacent, VertexSize -> Automatic, ImageSize -> Medium,
  VertexStyle -> Table[color[[i]] -> Green, {i, 1, Length[color]}]];
  subgrafo = Subgraph[grafo, color];
  grafo = AdjacencyGraph[Adjacent, VertexSize -> Automatic, ImageSize -> Medium,
  VertexStyle -> Table[color[[i]] -> Red, {i, 1, Length[color]}]];
  subgrafo = Subgraph[grafo, color];

```

```

[Subgrafo]
(*Cualquier calculo con grafo,
subgrafo solo sirve para la representación gráfica*)
ClosCen = ClosenessCentrality[grafo];
[centralidad de cercanía]
LocalClust = LocalClusteringCoefficient[grafo];
[coeficiente de acumulación local]
(*Reconstruimos la matriz con las medidas calculadas*)
Do[
[repite]
  ClearAll[aux1, aux2];
  [borra todo]
  aux1 = {};
  aux2 = {};
  Do[
  [repite]
    elemento = j + (i - 1) * n;
    AppendTo[aux1, ClosCen[[elemento]]];
    [añade al final]
    AppendTo[aux2, LocalClust[[elemento]]];
    [añade al final]

    , {j, 1, n}];
  AppendTo[MatrixClosenessCentrality, aux1];
  [añade al final]
  AppendTo[MatrixLocalClustering, aux2],
  [añade al final]

  {i, 1, n}];
AppendTo[salida,
[añade al final]
  {subgrafo, MatrixClosenessCentrality, MatrixLocalClustering}];
salida[[1]]

];

```

```

(* Este modulo chequea la vecindad de una célula ocupada *)
FuncionChequeo[Matrix_, n_, Aleatoriofila_, Aleatoriocolumna_] :=
Module[{Norte, Sur, Este, Oeste, SurEste,
[módulo]
  NorEste, NorOeste, SurOeste, Elemento, Salida},
  Elemento = Matrix[[Aleatoriocolumna, Aleatoriofila]];

  (*Chequeo Vecindad de Moore y CC NULAS*)
  If[Aleatoriocolumna - 1 == 0, Norte = 0,
  [si]
  Norte = Matrix[[Aleatoriocolumna - 1, Aleatoriofila]]];
  If[Aleatoriocolumna + 1 > n, Sur = 0,
  [si]
  Sur = Matrix[[Aleatoriocolumna + 1, Aleatoriofila]]];
  If[Aleatoriofila - 1 == 0, Oeste = 0,
  [si]

```

```

[si
  Oeste = Matrix[ [Aleatoriocolumna, Aleatoriofila - 1] ]];
If[Aleatoriofila + 1 > n, Este = 0,
[si
  Este = Matrix[ [Aleatoriocolumna, Aleatoriofila + 1] ]];
If[Aleatoriocolumna - 1 == 0 || Aleatoriofila - 1 == 0, NorOeste = 0,
[si
  NorOeste = Matrix[ [Aleatoriocolumna - 1, Aleatoriofila - 1] ]];
If[Aleatoriocolumna - 1 == 0 || Aleatoriofila + 1 > n, NorEste = 0,
[si
  NorEste = Matrix[ [Aleatoriocolumna - 1, Aleatoriofila + 1] ]];
If[Aleatoriocolumna + 1 > n || Aleatoriofila - 1 == 0, SurOeste = 0,
[si
  SurOeste = Matrix[ [Aleatoriocolumna + 1, Aleatoriofila - 1] ]];
If[Aleatoriocolumna + 1 > n || Aleatoriofila + 1 > n, SurEste = 0,
[si
  SurEste = Matrix[ [Aleatoriocolumna + 1, Aleatoriofila + 1] ]];

Salida =
  Elemento * (Norte + Sur + Este + Oeste + NorOeste + SurOeste + NorEste + SurEste);
Salida
];

```

(*La siguiente función es como la anterior pero solo se ejecuta en células que no estén ocupadas*)

```

FuncionChequeoMov[Matrix_, n_, Aleatoriofila_,
  Aleatoriocolumna_, Elemento_] := Module[
  [módulo
    {Norte, Sur, Este, Oeste, SurEste, NorEste, NorOeste, SurOeste, Salida},

    (*Chequeo Vecindad de Moore y CC NULAS*)
    If[Aleatoriocolumna - 1 == 0, Norte = 0,
[si
  Norte = Matrix[ [Aleatoriocolumna - 1, Aleatoriofila] ]];
If[Aleatoriocolumna + 1 > n, Sur = 0,
[si
  Sur = Matrix[ [Aleatoriocolumna + 1, Aleatoriofila] ]];
If[Aleatoriofila - 1 == 0, Oeste = 0,
[si
  Oeste = Matrix[ [Aleatoriocolumna, Aleatoriofila - 1] ]];
If[Aleatoriofila + 1 > n, Este = 0,
[si
  Este = Matrix[ [Aleatoriocolumna, Aleatoriofila + 1] ]];
If[Aleatoriocolumna - 1 == 0 || Aleatoriofila - 1 == 0, NorOeste = 0,
[si
  NorOeste = Matrix[ [Aleatoriocolumna - 1, Aleatoriofila - 1] ]];
If[Aleatoriocolumna - 1 == 0 || Aleatoriofila + 1 > n, NorEste = 0,
[si
  NorEste = Matrix[ [Aleatoriocolumna - 1, Aleatoriofila + 1] ]];
If[Aleatoriocolumna + 1 > n || Aleatoriofila - 1 == 0, SurOeste = 0,
[si
  SurOeste = Matrix[ [Aleatoriocolumna + 1, Aleatoriofila - 1] ]];

```



```

If[Aleatoriocolumna + 1 > n || Aleatoriofila + 1 > n, SurEste = 0,
  si
  SurEste = Matrix[[Aleatoriocolumna + 1, Aleatoriofila + 1]];

Salida = Elemento * (Norte + Sur + Este +
  Oeste + NorOeste + SurOeste + NorEste + SurEste - Elemento);
Salida
];

(*Este módulo es similar al primero pero se emplea en la fase de Movimiento*)
ComplexNetworkMeasuresMovPhase[n_, matrix_, soldierNumber_] :=
Module[{Adjacent, color, elemento, AdjacentList, grafo, aux1, aux2, aux3,
  módulo
  aux4, aux5, MatrixClusteringLocal, MatrixCentralidadIntermediacion,
  MatrixCentralidadCercania, MatrixCentralidadVectorPropio,
  CentralidadVectorPropio, ClusteringLocal, CentralidadIntermediacion,
  CentralidadCercania, MatrixGradoVertice, GradoVertice, salida},
  salida = {};
  Adjacent = {};
  color = {};
  MatrixGradoVertice = {};
  MatrixCentralidadVectorPropio = {};
  MatrixClusteringLocal = {};
  MatrixCentralidadCercania = {};
  MatrixCentralidadIntermediacion = {};

  Do[Do [
    r... repite
    elemento = j + (i - 1) * n; (*Mas importante,
    es lo que codifica el i,j como un k en la lista*)
    ClearAll[AdjacentList];
    borra todo
    AdjacentList = {};
    Do[Do [
      r... repite
      Which[{1, m] == {i, j},
        cuál
        AppendTo[AdjacentList, 0],
        añade al final
        Abs[m - j] > 1,
        valor absoluto
        AppendTo[AdjacentList, 0],
        añade al final
        Abs[1 - i] > 1,
        valor absoluto
        AppendTo[AdjacentList, 0],
        añade al final
        matrix[[1, m]] == soldierNumber,
        AppendTo[AdjacentList, 1],
        añade al final
        matrix[[1, m]] != soldierNumber,
        AppendTo[AdjacentList, 0]
        añade al final

```

```

];

, {m, 1, n}], {1, 1, n}];
If[matrix[[i, j]] ≠ soldierNumber, ClearAll[AdjacentList];
AdjacentList = Table[0, n^2], AppendTo[color, elemento]];
AppendTo[Adjacent, AdjacentList];
, {j, 1, n}], {i, 1, n}];
If[soldierNumber == 1,

grafo = AdjacencyGraph[Adjacent, VertexSize → Automatic, ImageSize → Medium,
VertexStyle → Table[color[[i]] → Green, {i, 1, Length[color]}]],
grafo = AdjacencyGraph[Adjacent, VertexSize → Automatic, ImageSize → Medium,
VertexStyle → Table[color[[i]] → Red, {i, 1, Length[color]}]];

(*Cualquier calculo con grafo,
subgrafo solo sirve para la representación gráfica*)

GradoVertice = VertexDegree[grafo];
CentralidadVectorPropio = EigenvectorCentrality[grafo];
GradoVertice = VertexDegree[grafo];
CentralidadVectorPropio = EigenvectorCentrality[grafo];
ClusteringLocal = LocalClusteringCoefficient[grafo];
CentralidadCercania = ClosenessCentrality[grafo];
CentralidadIntermediacion = BetweennessCentrality[grafo];

(*Reconstruimos la matriz con las medidas calculadas*)
Do[
ClearAll[aux1, aux2];
aux1 = {};
aux2 = {};
aux3 = {};
aux4 = {};
aux5 = {};
Do [

```

```

elemento = j + (i - 1) * n;
AppendTo[aux1, GradoVertice[[elemento]]];
añade al final
AppendTo[aux2, CentralidadVectorPropio[[elemento]]];
añade al final
AppendTo[aux3, ClusteringLocal[[elemento]]];
añade al final
AppendTo[aux4, CentralidadCercania[[elemento]]];
añade al final
AppendTo[aux5, CentralidadIntermediacion[[elemento]]];
añade al final

, {j, 1, n}];
AppendTo[MatrixGradoVertice, aux1];
añade al final
AppendTo[MatrixCentralidadVectorPropio, aux2];
añade al final
AppendTo[MatrixClusteringLocal, aux3];
añade al final
AppendTo[MatrixCentralidadCercania, aux4];
añade al final
AppendTo[MatrixCentralidadIntermediacion, aux5],
añade al final
{i, 1, n}];
AppendTo[salida,
añade al final
{MatrixGradoVertice, MatrixCentralidadVectorPropio, MatrixClusteringLocal,
MatrixCentralidadCercania, MatrixCentralidadIntermediacion}];
salida[[1]]

];

(*Este modulo chequea la vecindad de una célula en la vecindad *)
ChequeoContorno[Matrix_, n_, Aleatoriofila_, Aleatoriocolumna_, Movimiento_] :=
Module[{Norte, Sur, Este, Oeste, SurEste,
módulo
NorEste, NorOeste, SurOeste, Elemento, Salida},
Elemento = Matrix[[Aleatoriocolumna, Aleatoriofila]];
Salida = 1;
(*Chequeo No salirse del grid establecido *)
Which[
cuál
Movimiento == 2, If[Aleatoriocolumna - 1 == 0, Salida = 0,
si
If[Matrix[[Aleatoriocolumna - 1, Aleatoriofila]] != 0, Salida = 0]],
si
Movimiento == 7, If[Aleatoriocolumna + 1 > n, Salida = 0,
si
If[Matrix[[Aleatoriocolumna + 1, Aleatoriofila]] != 0, Salida = 0]],
si
Movimiento == 4, If[Aleatoriofila - 1 == 0, Salida = 0,
si

```

```

    If[Matrix[[Aleatoriocolumna, Aleatoriofila - 1]] ≠ 0, Salida = 0]],
  _si
Movimiento == 5, If[Aleatoriofila + 1 > n, Salida = 0,
  _si
    If[Matrix[[Aleatoriocolumna, Aleatoriofila + 1]] ≠ 0, Salida = 0]],
  _si
Movimiento == 1, If[Aleatoriocolumna - 1 == 0 || Aleatoriofila - 1 == 0, Salida =
  _si
    0, If[Matrix[[Aleatoriocolumna - 1, Aleatoriofila - 1]] ≠ 0, Salida = 0]],
  _si
Movimiento == 3, If[Aleatoriocolumna - 1 == 0 || Aleatoriofila + 1 > n, Salida = 0,
  _si
    If[Matrix[[Aleatoriocolumna - 1, Aleatoriofila + 1]] ≠ 0, Salida = 0]],
  _si
Movimiento == 6, If[Aleatoriocolumna + 1 > n || Aleatoriofila - 1 == 0, Salida = 0,
  _si
    If[Matrix[[Aleatoriocolumna + 1, Aleatoriofila - 1]] ≠ 0, Salida = 0]],
  _si
Movimiento == 8, If[Aleatoriocolumna + 1 > n || Aleatoriofila + 1 > n, Salida = 0,
  _si
    If[Matrix[[Aleatoriocolumna + 1, Aleatoriofila + 1]] ≠ 0, Salida = 0]]
  _si
];
Salida
];

(*Este módulo calcula la posición en el ranking*)
IndexRank[n_, matrix_] := Module[{posicion, temp, position, ii, jj, salida},
  _módulo
  temp = matrix;
  posicion = {};
  Do[
  _repite
    ClearAll[position];
    _borra todo
    position = Position[temp, Max[temp]][[1]];
    _posición _máximo
    ii = position[[1]];
    jj = position[[2]];
    AppendTo[posicion, position]; (*Se guarda como (fila,columna)*)
    _añade al final
    temp[[ii, jj]] = -1, {kk, 1, n^2}];
  salida = posicion;
  salida];

(*Este modulo calcula el ángulo que forman dos unidades en la vecindad*)
MideAngulos[Entrada_] := Module[{VectorAngulos, Dif, DiferenciasAngulos, Salida},
  _módulo
  VectorAngulos = {};
  DiferenciasAngulos = {};
  Salida = {0, 0, 0, 0};
  Do[If[Entrada[[i]] == 1, AppendTo[VectorAngulos, (i - 1) * Pi / 4]],
  _si _añade al final _numero pi

```

```

    {i, 1, Length[Entrada] }];
  Do[Do[
    Dif = Abs[VectorAngulos[[i]] - VectorAngulos[[j]]];
    AppendTo[DiferenciasAngulos, Simplify[Dif]];
    , {j, i + 1, Length[VectorAngulos]}, {i, 1, Length[VectorAngulos]};
  Salida[[1]] = Count[DiferenciasAngulos, Pi/4];
  Salida[[1]] = Salida[[1]] + Count[DiferenciasAngulos, 7 * Pi/4];
  Salida[[2]] = Count[DiferenciasAngulos, Pi/2];
  Salida[[2]] = Salida[[2]] + Count[DiferenciasAngulos, 6 * Pi/4];
  Salida[[3]] = Count[DiferenciasAngulos, 3 * Pi/4];
  Salida[[3]] = Salida[[3]] + Count[DiferenciasAngulos, 5 * Pi/4];
  Salida[[4]] = Count[DiferenciasAngulos, Pi];
  Salida];

(*Este módulo se emplea en el cálculo de si una célula está o no aislada*)
MideAislado[Matrix_, n_, Aleatoriofila_, Aleatoriocolumna_] :=
Module[{Norte, Sur, Este, Oeste, NorOeste, SurOeste, NorEste, SurEste, Salida},
  If[Aleatoriocolumna - 1 == 0, Norte = 0,
  Norte = Matrix[[Aleatoriocolumna - 1, Aleatoriofila]];
  If[Aleatoriocolumna + 1 > n, Sur = 0,
  Sur = Matrix[[Aleatoriocolumna + 1, Aleatoriofila]];
  If[Aleatoriofila - 1 == 0, Oeste = 0,
  Oeste = Matrix[[Aleatoriocolumna, Aleatoriofila - 1]];
  If[Aleatoriofila + 1 > n, Este = 0,
  Este = Matrix[[Aleatoriocolumna, Aleatoriofila + 1]];
  If[Aleatoriocolumna - 1 == 0 || Aleatoriofila - 1 == 0, NorOeste = 0,
  NorOeste = Matrix[[Aleatoriocolumna - 1, Aleatoriofila - 1]];
  If[Aleatoriocolumna - 1 == 0 || Aleatoriofila + 1 > n, NorEste = 0,
  NorEste = Matrix[[Aleatoriocolumna - 1, Aleatoriofila + 1]];
  If[Aleatoriocolumna + 1 > n || Aleatoriofila - 1 == 0, SurOeste = 0,
  SurOeste = Matrix[[Aleatoriocolumna + 1, Aleatoriofila - 1]];

```

```
If[Aleatoriocolumna + 1 > n || Aleatoriofila + 1 > n, SurEste = 0,  
  [si  
    SurEste = Matrix[[Aleatoriocolumna + 1, Aleatoriofila + 1]]];  
  Salida = {Norte, Sur, Este, Oeste, NorOeste, SurOeste, NorEste, SurEste}  
]
```