



UNIVERSIDAD
DE MÁLAGA

TESIS DOCTORAL

GENERACIÓN AUTOMÁTICA DE CONTENIDO PARA JUEGOS
DE ESTRATEGIA EN TIEMPO REAL

RAÚL LARA CABRERA

Directores

ANTONIO JOSÉ FERNÁNDEZ LEIVA

CARLOS COTTA PORRAS

Doctorado en Tecnologías Informáticas
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

2015



Publicaciones y
Divulgación Científica

AUTOR: Raúl Lara Cabrera

 <http://orcid.org/0000-0002-7959-1936>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): riuma.uma.es

Raúl Lara Cabrera: *Generación automática de contenido para juegos de estrategia en tiempo real*, © 2015

*Nuestra tecnología, nuestras máquinas,
son parte de nuestra humanidad.
Las hemos creado para mejorarnos a nosotros mismos,
y esto es lo que hace único al ser humano.*

— Ray Kurzweil

AGRADECIMIENTOS

¡Esto se acaba! Otra etapa más de mi vida que se cierra, y, como siempre, rodeado de personas que hacen que todo esto merezca la pena. Voy a intentar acordarme de todas ellas así que prepárate para una larga lista de nombres.

En primer lugar, como no podía ser de otra forma, quiero dar las gracias a mis padres y a mi hermana, gracias a los cuales me encuentro hoy aquí escribiendo estos agradecimientos.

A Patricia, mi mujer y alma gemela, por acompañarme en todas mis aventuras, ¡y las que nos quedan!

A mi familia que, aunque pequeña, es capaz de brindarme el mismo apoyo que otras con cientos de miembros.

A todos mis amigos, nos vemos cada semana o una vez al año: Juanma, Isa, Ismael, Josemi, Aragón, Miguel, Denis, Dani, Desi, Manu, Sheila, Manolo, Virginia, Juanlu, Vane, Javi, Merche, Antonio, Eva, Dani-chan, Juanjo. Un millón de gracias.

A Carlos y Antonio, mis directores de tesis, por confiar en mí y servirme de guía para cumplir mis objetivos, y a Mariela, mi compañera de proyecto.

Gracias a la gente del departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, y en especial, a aquellos que me han soportado diariamente en el laboratorio y durante el café: Jamal, Daniel, Javier, Yesnier, Enrique, Gabriel, Francis, Lisa y Lola.

A los colegas de conferencias y proyectos: Pablo, Antares, Antonio y JJ (Granada), y Paco (Extremadura).

Y por último, pero no menos importante, gracias a ti, lector, por mostrar interés en saber de mi trabajo.

Este trabajo ha sido parcialmente financiado por la Junta de Andalucía dentro del proyecto P10-TIC-6083 (DNEMESIS¹), el Ministerio español de Economía y Competitividad dentro de los proyectos ANYSELF² (TIN2011-28627-C04-01) y UMA-EPHEMECH³ (TIN2014-56494-C4-1-P), y la Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

¹ <http://dnemesis.lcc.uma.es/wordpress/>

² <http://anyself.wordpress.com/>

³ <http://blog.epheme.ch>

ÍNDICE GENERAL

| | | |
|-------|---|----|
| i | INTRODUCCIÓN Y FUNDAMENTOS | 1 |
| 1 | INTRODUCCIÓN | 3 |
| 1.1 | Contexto y objetivo de la tesis | 3 |
| 1.2 | Contribuciones de la tesis | 6 |
| 1.3 | Publicaciones | 6 |
| 2 | ANTECEDENTES | 9 |
| 2.1 | Juegos e inteligencia artificial | 9 |
| 2.1.1 | Generación de contenido para juegos de ETR . . | 10 |
| 2.1.2 | Juegos de estrategia en tiempo real e intelligen- cia artificial | 14 |
| 2.1.3 | Generación de mapas y niveles para otros géneros | 18 |
| 2.1.4 | Planet Wars: Descripción del juego | 19 |
| 2.2 | Introducción a la computación evolutiva | 21 |
| 2.2.1 | Reseña histórica | 22 |
| 2.2.2 | Componentes de un algoritmo evolutivo | 23 |
| 2.2.3 | Algoritmos genéticos | 25 |
| 2.2.4 | Estrategias de evolución | 26 |
| 2.2.5 | Programación evolutiva | 26 |
| 2.2.6 | Programación genética | 27 |
| 2.2.7 | Optimización multiobjetivo | 27 |
| 2.2.8 | Otras técnicas | 28 |
| 3 | RED DE CO-AUTORÍA | 31 |
| 3.1 | Obtención de los datos bibliográficos | 32 |
| 3.2 | Análisis de la red de colaboración | 33 |
| 3.2.1 | Propiedades macroscópicas básicas | 34 |
| 3.2.2 | Estructura global de la red | 37 |
| 3.2.3 | Mecánicas de crecimiento de la red: vinculación preferente | 38 |
| 3.2.4 | Patrones de colaboración y clustering | 42 |
| 3.2.5 | Estructura mesoscópica de la red | 46 |
| 3.2.6 | Análisis de centralidad | 50 |
| 3.3 | Resultados y contribuciones | 52 |
| ii | LA GENERACIÓN AUTOMÁTICA DE MAPAS | 55 |
| 4 | UN GENERADOR AUTOMÁTICO DE MAPAS | 57 |
| 4.1 | Descripción del algoritmo | 57 |
| 4.2 | Operadores | 60 |
| 5 | MEJORA DE LA EXPERIENCIA DE JUEGO | 63 |
| 5.1 | Equilibrio | 63 |
| 5.2 | Dinamismo | 69 |
| 5.3 | Equilibrio vs. Dinamismo | 77 |

| | | |
|-----|---|-----|
| 5.4 | Equilibrio y Dinamismo: un enfoque multiobjetivo . . . | 79 |
| 5.5 | Resultados y contribuciones | 84 |
| 6 | EVOLUCIÓN DE LA ESTÉTICA DE LOS MAPAS | 87 |
| 6.1 | Medidas geométricas | 87 |
| 6.2 | Medidas topológicas | 88 |
| 6.3 | Resultados y contribuciones | 93 |
| 7 | CONCLUSIONES Y TRABAJO FUTURO | 99 |
| 7.1 | Trabajo futuro | 101 |
| iii | APÉNDICES | 103 |
| A | APÉNDICES | 105 |
| A.1 | Conferencias, revistas y palabras clave para el análisis de la red | 105 |
| | REFERENCIAS | 109 |

ÍNDICE DE FIGURAS

| | | |
|------------|---|----|
| Figura 2.1 | Captura de pantalla del juego <i>Ants</i> . Los puntos de colores representan a las hormigas de cada equipo mientras que los círculos negros son los hormigueros. | 17 |
| Figura 2.2 | Una partida de <i>Planet Wars</i> en marcha. Los círculos representan los planetas mientras que el color de los mismos y el número que incluyen representan el jugador propietario y el número de naves de la flota defensora, respectivamente. Los números entre planetas son flotas de naves que se encuentran en movimiento. . . | 20 |
| Figura 2.3 | Esquema general del funcionamiento de un algoritmo evolutivo | 22 |
| Figura 3.1 | Evolución del número total de autores en la red. La gráfica interna muestra el número de artículos publicados cada año. | 33 |
| Figura 3.2 | Evolución temporal del número de autores nuevos y volátiles (datos anuales). | 34 |
| Figura 3.3 | Número medio de autores por artículo y artículos por autor (figura interior). Las barras verticales marcan el error estándar de la media. | 35 |
| Figura 3.4 | Evolución del número de componentes conectadas (escala log) en la red acumulada. La línea sólida muestra el mejor ajuste ($y = \alpha(1 + \beta)^x$, $\beta = 0.47$, $R^2 = 0.98$ con $1997 \leq x \leq 2005$). La figura interna muestra el mismo ajuste para la red efectiva ($\beta = 0.35$, $R^2 = 0.99$ con $2001 \leq x \leq 2005$). | 36 |
| Figura 3.5 | Ratio entre el tamaño de la componente más grande de la red acumulada y $N^{2/3}$ (línea sólida). Ratio entre ese mismo tamaño y el de la componente gigante de un grafo aleatorio con el mismo número de nodos y aristas (línea discontinua). La figura interior muestra los mismos datos para la red efectiva. | 37 |
| Figura 3.6 | Componente gigante de la red de IC y videojuegos en (a) 2006 y (b) 2011. Los nodos representan a los autores y las aristas, al menos una colaboración entre ellos. | 39 |

| | | |
|-------------|---|----|
| Figura 3.7 | Grados de los nodos para la red acumulada G_{2011} (escala log-log). La línea con guiones es un ajuste a la ley de potencias $P(k) \sim k^{-\gamma}$ con $\gamma = 3.27$ y $R^2 = 0.96$ ($2 \leq k \leq 25$). La línea sólida es un ajuste log-normal $P(k) \sim \exp(\alpha \ln k - \beta (\ln k)^2)$ con $\alpha = -10.5908$ y $\beta = -2.9923$ ($R^2 = 0.98$). | 40 |
| Figura 3.8 | Probabilidad relativa acumulada (para 2007-2011) de que los nuevos autores colaboren con los que ya están en la red dependiendo del número de colaboraciones previas que tienen los segundos. Los datos siguen una ley de potencia ($\alpha + 1 = 1.64$ y $R^2 = 0.97$). Figura interna: probabilidad relativa acumulada de nuevas colaboraciones (autores ya existentes colaboran por primera vez entre ellos). Los datos siguen una ley de potencia ($\alpha + 1 = 2.01$ and $R^2 = 0.91$). En ambos casos el número de colaboradores previos se calcula usando la red acumulada. | 41 |
| Figura 3.9 | Porcentaje de cada tipo de enlace para los datos acumulados. Las líneas sólida, punteada y discontinua muestran un ajuste lineal al número de enlaces entre nodos existentes ($\Delta_x = 0.0036, R^2 = 0.68$), entre un nodo existente y otro nuevo ($\Delta_x = 0.0217, R^2 = 0.83$) y entre nodos nuevos ($\Delta_x = -0.0279, R^2 = 0.85$), respectivamente. Figura interna: los mismos datos para la red efectiva. Los parámetros de los ajustes son $\Delta_x = 0.0087, R^2 = 0.75$, $\Delta_x = 0.0269, R^2 = 0.0268$ y $\Delta_x = -0.0356, R^2 = 0.92$ respectivamente. | 41 |
| Figura 3.10 | Probabilidad relativa de nueva colaboración dependiendo del número de colaboradores en común. La línea sólida es el mejor ajuste lineal ($\Delta_x = 18.44, R^2 = 0.98$). La figura interior muestra los mismos datos para la red efectiva ($\Delta_x = 11.59, R^2 = 0.66$). En ambos casos se excluye el punto $x = 4$, cuya condición de valor atípico se atribuye al pequeño tamaño de la red) | 43 |
| Figura 3.11 | Evolución del coeficiente de clustering. La línea sólida es el mejor ajuste lineal ($\Delta_x = 0.013, R^2 = 0.93$). La figura interior muestra la evolución de este coeficiente para la red efectiva y su ajuste lineal ($\Delta_x = 0.0093, R^2 = 0.94$). | 44 |

| | | |
|-------------|--|----|
| Figura 3.12 | Distribución del número de colaboraciones previas. El eje Y está en escala logarítmica. La línea sólida muestra el mejor ajuste exponencial ($P(y) \sim a^x, a = 0.4343, R^2 = 0.94$). | 45 |
| Figura 3.13 | Probabilidad relativa de nueva colaboración en base al número de colaboraciones previas entre dos autores en 2011. Eje Y en escala logarítmica. La línea sólida muestra el mejor ajuste exponencial ($a = 1.6318, R^2 = 0.97$). La figura interior muestra los mismos datos en grupos de 5 años y su ajuste exponencial ($a = 1.3763, R^2 = 0.80$). Los datos se corresponden en ambos casos a la red acumulada y para el cálculo del ajuste se han omitido el primer y último punto de los datos. | 45 |
| Figura 3.14 | Evolución del tamaño medio de comunidad en la componente gigante de la red acumulada. La figura interior muestra el número de comunidades en la componente gigante. | 46 |
| Figura 3.15 | Evolución de la centralización por grado en la componente gigante de la red acumulada. Se incluye un ajuste lineal ($\Delta_x = -0.065, R^2 = 0.93$). La figura interior muestra estos datos para la red efectiva. | 47 |
| Figura 3.16 | (a) Grafo de interacción entre comunidades de la componente gigante hasta 2006 (figura interior) y 2011, usando datos acumulados. Cada nodo representa una comunidad. El tamaño del nodo es proporcional al grado de intermediación del nodo. (b) Comunidad más grande y conectada de la componente gigante hasta 2011 usando datos acumulados. El tamaño del nodo es proporcional a su grado de intermediación mientras que el de la arista se corresponde con el número de colaboraciones entre los autores. | 48 |

| | | |
|-------------|--|----|
| Figura 3.17 | Grafo que representa las palabras más frecuentes en el título de los artículos escritos por al menos un miembro de la comunidad más grande y central. Cada nodo representa una palabra y una arista entre dos nodos indica que esas palabras aparecen en el mismo título. El tamaño de los nodos representa la frecuencia de la palabra, mientras que el de las aristas representa la frecuencia de la relación entre ambas palabras. El color de cada nodo indica a que comunidad pertenece dentro del grafo de palabras | 49 |
| Figura 3.18 | Grafo que representa las palabras más frecuentes en el título de los artículos escritos por al menos un miembro de los autores más centrales de la red efectiva de 2011. Cada nodo representa una palabra y una arista entre dos nodos indica que esas palabras aparecen en el mismo título. El tamaño de los nodos representa la frecuencia de la palabra, mientras que el de las aristas representa la frecuencia de la relación entre ambas palabras. El color de cada nodo indica a que comunidad pertenece dentro del grafo de palabras | 51 |
| Figura 4.1 | Esquema sobre la representación de un mapa de Planet Wars como individuo del algoritmo evolutivo | 58 |
| Figura 4.2 | Ejemplo del operador de cruce “corte y empalme”. La línea roja indica el punto de corte elegido para cada padre. Nótese que los hijos tienen un tamaño diferente al de sus padres. | 60 |
| Figura 5.1 | Evolución del valor de aptitud medio para la versión auto-adaptativa (línea sólida) y con número fijo de planetas. | 65 |
| Figura 5.2 | Evolución del número de planetas del mapa más equilibrado. En ambos casos, el área sombreada indica el error estándar de la media. | 65 |
| Figura 5.3 | Un mapa equilibrado creado de forma automática por el generador. Los números sobre los planetas indican el número inicial de naves mientras que el color define si el planeta es neutral al inicio (gris) o pertenece a alguno de los dos jugadores (rojo y azul). | 66 |
| Figura 5.4 | Reglas difusas para el equilibrio (bal). HI, MED y LO representan un nivel alto, medio y bajo, respectivamente. | 68 |

| | | |
|-------------|---|----|
| Figura 5.5 | Funciones de pertenencia a los conjuntos difusos. Se corresponden con las variables (a) Π, Γ, Ξ, T, K (b) $\Delta^\pi, \Delta^\gamma, \Delta^\xi$ (c) Z y (d) bal y dyn | 69 |
| Figura 5.6 | Reglas difusas para el dinamismo. | 72 |
| Figura 5.7 | Evolución del valor de aptitud medio para ambas funciones objetivo. El área sombreada muestra el error estándar de la media. Las figuras izquierda y central muestran la evolución del dinamismo medido según ambos enfoques (recursos y confrontación) cuando se está optimizando uno de ellos, indicado por el título de la figura. En la derecha se muestra la evolución del equilibrio cuando se optimiza el dinamismo según ambos enfoques. Cada línea muestra la media de 10 ejecuciones del algoritmo. | 73 |
| Figura 5.8 | Evolución de las diferentes variables recogidas en las reglas de los conjuntos difusos. B y K son los ratios de batallas y conquistas, respectivamente, las variables PD representan los picos de diferencia entre los recursos de cada jugador, S se corresponde con las naves destruidas, T es la longitud de la partida y Z se corresponde con el ratio de reconquista. | 74 |
| Figura 5.9 | Varias características de los mejores mapas generados en cada ejecución del algoritmo y para cada función objetivo: número de planetas en el mapa, distancia media entre esos planetas, correlación entre el tamaño de los planetas y el número inicial de naves de cada uno y la desviación típica del tamaño de los mismos. | 75 |
| Figura 5.10 | Dos ejemplos de mapas dinámicos. | 76 |
| Figura 5.11 | Evolución de los indicadores de rendimiento para el frente no dominado en cada una de las ejecuciones del algoritmo. La figura interior muestra la misma medida para el frente acumulado de ambos algoritmos. (a) Hipervolumen (b) R_2 | 78 |
| Figura 5.12 | Diagrama de cajas de los indicadores de rendimiento para los dos enfoques de algoritmo multiobjetivo. | 79 |
| Figura 5.13 | Comparativa de los frentes no dominados globales obtenidos por ambos algoritmos. | 80 |

| | | |
|-------------|---|----|
| Figura 5.14 | (a) Número de planetas en cada solución del frente no dominado. (b) Distribución de las distancias interplanetarias. El círculo negro se corresponde con el percentil 25 %, el gris con la mediana y el blanco con el percentil 75 % . . . | 81 |
| Figura 5.15 | Distribución de los tamaños de los planetas en cada solución del frente no dominado. El círculo negro se corresponde con el percentil 25 %, el gris con la mediana y el blanco con el percentil 75 %. | 82 |
| Figura 5.16 | Coefficiente de correlación de Pearson entre el tamaño del planeta y el número inicial de naves que se encuentran en él al principio de la partida. Un círculo blanco indica correlación positiva mientras que uno negro indica una correlación negativa. | 83 |
| Figura 6.1 | Un grafo de esferas de influencia. Hay una arista entre dos puntos solo si sus esferas de influencia intersecan. Estas esferas están centradas en los puntos y sus radios son la distancia mínima desde el punto central hasta el resto de puntos. | 89 |
| Figura 6.2 | Conjunto acumulado de soluciones no dominadas (círculo) y mapas estéticos (triángulo) y no estéticos (cuadrado) pertenecientes a los conjuntos de entrenamiento tanto para el enfoque geométrico (a) como el topológico (b) . . . | 91 |
| Figura 6.3 | VARIABLES DE CARACTERIZACIÓN PARA LOS MAPAS NO DOMINADOS Y ENTRENAMIENTO, TANTO PARA EL ENFOQUE GEOMÉTRICO (a) COMO PARA EL TOPOLÓGICO (b) | 92 |
| Figura 6.4 | Distribución de los mapas sobre el mapa auto-organizado tanto para el enfoque geométrico (a) como el topológico (b). El rojo para los no estéticos, el verde para los estéticos y el azul para los no dominados. (c) y (d) muestran las soluciones del enfoque topológico proyectadas sobre el mapa auto-organizado geométrico y viceversa. | 93 |
| Figura 6.5 | Soluciones topológicas no dominadas sobre el espacio de aptitud de las soluciones geométricas (a) y a la inversa (b). | 95 |

| | | |
|------------|---|----|
| Figura 6.6 | Algunos mapas de ejemplo: pertenecientes al conjunto de entrenamiento con buena (a) y mala (b) estética, generados mediante el enfoque geométrico (c) (d) y el topológico (e) (f). El color de los planetas representa quién los controla: rojo y verde para los jugadores 1 y 2 respectivamente, marrón para indicar que el planeta es neutral. El número junto a los planetas indica el total de naves al inicio de la partida en cada planeta. | 97 |
|------------|---|----|

ÍNDICE DE TABLAS

| | | |
|-----------|---|----|
| Tabla 2.1 | Caracterización de algoritmos de generación automática de contenido | 10 |
| Tabla 5.1 | Parámetros de los algoritmos para generar mapas equilibrados | 64 |
| Tabla 6.1 | Parámetros del clasificador de bosques aleatorios | 90 |

ACRÓNIMOS

- AE** Algoritmos Evolutivos
- DBLP** Digital Bibliography and Library Project
- ETR** Estrategia en Tiempo Real
- GAC** Generación Automática de Contenido
- GEI** Grafo de Esferas de Influencia
- IA** Inteligencia Artificial
- IC** Inteligencia Computacional

Parte I

INTRODUCCIÓN Y FUNDAMENTOS

Los dos primeros capítulos sirven como introducción y motivación de la tesis, así como un estudio del estado del problema y una introducción a la computación evolutiva. Además de esto, se presentan las principales contribuciones de la tesis.

INTRODUCCIÓN

1.1 CONTEXTO Y OBJETIVO DE LA TESIS

Los videojuegos se han convertido en un pilar fundamental de la industria del entretenimiento, generando un destacado volumen de negocio a nivel internacional y causando un gran impacto a una buena parte de la población. Solo en Estados Unidos ya existen 155 millones de jugadores que gastaron 15400 millones de dólares en 2014 [37]. Este hecho no ha pasado desapercibido para la comunidad científica, que se ha visto surtida de nuevos retos y desafíos que abordar.

Ya desde sus inicios, los videojuegos han sido objeto de estudio por parte del mundo académico, abarcando distintas ramas del conocimiento, pero incidiendo especialmente en la investigación de nuevas técnicas de Inteligencia Artificial (IA) [97]. Su principal atractivo recae en la amplia variedad de problemas susceptibles de ser resueltos y optimizados, como puede ser la planificación de tareas en tiempo real [133], la búsqueda de caminos en entornos con información imperfecta [14] o el desarrollo de estrategias de IA que simulen el comportamiento racional de los humanos [4].

La investigación en videojuegos no se limita al ámbito de la inteligencia artificial, sino que abarca otras disciplinas tan heterogéneas como la psicología, involucrada en la mejora y modelado de la satisfacción del jugador [119], el marketing y la gamificación [65], la salud y los juegos serios con objetivos más allá del mero entretenimiento [105], o los gráficos por computador [1].

Tradicionalmente ha existido una considerable separación entre la industria del videojuego y el mundo académico. Hasta hace unos años, la calidad de un videojuego estaba directamente relacionada con su aspecto gráfico, por lo que los desarrolladores han estado centrando su esfuerzo en mejorar dicha calidad gráfica. Sin embargo, se ha llegado a un punto en el cual es muy difícil que un juego destaque por sus gráficos, que se encuentran cercanos al fotorrealismo, por lo que los consumidores han empezado a demandar características adicionales, cambiando su forma de valorar los videojuegos. Es por ello que la industria ha empezado a incorporar a sus proyectos avances científicos fruto de la investigación en videojuegos, estableciéndose una sinergia entre ambas partes.

Entre todos los avances científicos que se están consiguiendo con la investigación en videojuegos y que la industria está empezando a incorporar con mayor frecuencia a sus proyectos de desarrollo se encuentran las técnicas de Generación Automática de Contenido (GAC),

*Generación
automática de
contenido*

también conocidas como generación de contenido por procedimientos, técnicas en las cuales se centra esta tesis. La GAC son un conjunto de técnicas que permiten generar, mediante algoritmos, contenido específico para videojuegos. El contenido, que puede ser indispensable para la construcción del videojuego o, por el contrario, opcional, varía desde los mapas y niveles en los que se desarrolla la partida hasta la música, pasando por las reglas, las misiones que el jugador tiene que cumplir, los modelos tridimensionales de los personajes y objetos o sus texturas, por poner solo unos ejemplos. A su vez, el proceso de generación puede suceder durante la fase de desarrollo del juego, lo que podríamos denominar como generación estática o a priori, o de forma dinámica al mismo tiempo que se está jugando. Según sea el proceso de generación, se puede establecer una caracterización de los métodos de GAC, que pueden seguir un esquema constructivo, donde el contenido se genera mediante unas reglas de transformación, uno de generación y prueba simple, donde se somete a validación el contenido obtenido mediante el proceso constructivo, o un esquema basado en búsquedas [153], que basa su funcionamiento en buscar el contenido más adecuado en todo el espacio de soluciones posibles usando para ello, por ejemplo, metaheurísticas. Los métodos de generación de contenido incluidos en la presente tesis se encuadran en esta última categoría.

*Ventajas/desventajas
del uso de GAC*

Una de las ventajas inmediatas del uso de estas técnicas es el ahorro en el coste que supone diseñar el contenido de forma manual, y es que gran parte del presupuesto de un videojuego se dedica precisamente al diseño. Más allá de sustituir al especialista humano, el empleo de técnicas de GAC sirve a su vez como fuente de inspiración para el diseñador, ya que debido a su funcionamiento, es posible obtener contenido que nunca se le habría ocurrido o que sirva como base para nuevos contenidos. También se puede contemplar un ahorro en el espacio que ocupa el juego, tanto en memoria como en almacenamiento físico, pues es posible generar dinámicamente contenido y objetos del juego que de otra forma tendrían que ser cargados desde el disco a la memoria principal. Pero la ventaja más destacable es la posibilidad para adaptar el juego al jugador de manera que la experiencia de juego sea única y personalizada, contribuyendo al objetivo de incrementar la satisfacción del jugador, que a fin de cuentas es el objetivo final de un videojuego. Con respecto a las desventajas, cabe destacar la inversión inicial de tiempo requerida por el uso de GAC para estudiar los requisitos del contenido a generar, modelar y crear la herramienta de generación y someter a validación el resultado obtenido.

*Ejemplos del uso de
GAC en videojuegos
comerciales*

Estas ventajas son vistas por la industria como una mejora del proceso de desarrollo de videojuegos así como una extensión de la vida comercial de los mismos, por lo que se pueden encontrar numerosos ejemplos del uso de este tipo de técnicas en videojuegos comerciales.

Por ejemplo, la saga de juegos de acción en primera persona *Borderlands* [55] usa un sistema de GAC para crear los diferentes objetos y armas que pueden usar los jugadores, dando lugar a un amplio espectro de ítems con diferentes características y ventajas. *Terraria* [130] y *Minecraft* [108] son juegos de construcción libre y supervivencia cuyos mapas son generados automáticamente mediante algoritmos. En el primer caso, el mapa se genera la primera vez que el jugador empieza una partida mientras que en el segundo, el mapa se va expandiendo de forma dinámica a la vez que el jugador lo va explorando. Uno de los juegos que hace un uso extensivo de la generación automática de contenido es *Spore* [103], en el cual la GAC forma parte de la propia mecánica de juego, ya que el jugador diseña criaturas que irán evolucionando de forma autónoma en unos planetas que, al igual que la música, se han generado algorítmicamente.

El objetivo de esta tesis es:

Objetivo

1. Definir una metodología para la generación automática de contenido para videojuegos de Estrategia en Tiempo Real (ETR).
2. Dicho contenido tiene que cumplir una serie de características orientadas a incrementar la satisfacción del jugador y que van más allá del mero cumplimiento de los requisitos funcionales para que el mapa se considere jugable.
3. Esta metodología propone el uso de Algoritmos Evolutivos (AE) para la generación del contenido.

Para la consecución de este objetivo se ha realizado en primer lugar un estudio de las contribuciones aportadas al campo de la GAC para videojuegos (Capítulo 2) así como un análisis de la comunidad (Capítulo 3) dedicada a la inteligencia computacional y videojuegos. Este análisis ha servido para identificar las contribuciones que se pueden hacer dentro de la comunidad y centrar los objetivos de esta tesis en una dirección concreta. Es por ello que se propone un generador automático de contenido basado en la computación evolutiva (Capítulo 4) con el cual se ha diseñado un método de creación de mapas con el objetivo de incrementar la satisfacción del jugador, obligando al contenido generado a cumplir los objetivos de equilibrio y dinamismo (Capítulo 5). La condición de equilibrio hace que los mapas no ofrezcan ninguna ventaja a un jugador sobre el otro, mientras que con el objetivo del dinamismo se consigue que las partidas jugadas en el mapa sean más interesantes. Los mapas se han generado para un juego de ETR llamado *Planet Wars*, ambientado en el espacio y con unas reglas muy sencillas. Tras observar la falta de estética en los mapas obtenidos por el anterior generador, se ha realizado un estudio sobre la estética de los mismos y se ha diseñado otro generador de mapas que busca que los resultados tengan una buena apariencia y, por tanto, no provoquen rechazo o desconcierto en el jugador (Capítulo 6).

Por último, el lector puede encontrar las conclusiones obtenidas a lo largo de todo el proceso en el Capítulo 7.

1.2 CONTRIBUCIONES DE LA TESIS

A continuación se pueden encontrar las contribuciones de la tesis junto con los capítulos en los cuales se describen:

- Se ha realizado un análisis de la red de co-autoría de la comunidad de investigación de IC y videojuegos, el cual ha arrojado datos interesantes relacionados con las dinámicas que rigen dicha red (Capítulo 3).
- En el Capítulo 4 se presenta un generador automático de mapas basado en algoritmos evolutivos para el juego de *ETR Planet Wars*. Este generador va más allá de crear contenido que cumpla las reglas del juego y es capaz de crear mapas con diferentes propiedades mediante el ajuste de su función de aptitud y objetivo de optimización.
- Se han planteado y estudiado dos propiedades deseables para los mapas generados automáticamente con el fin de incrementar la satisfacción del jugador: equilibrio y dinamismo. Además de analizar la relación entre ambas, se ha comprobado que es posible crear mapas con estas propiedades tanto de forma individual como conjunta, siguiendo varios enfoques (ver Capítulo 5).
- En el Capítulo 6 se ha abordado el estudio de otra propiedad de los mapas, esta vez relacionada con la estética de los mismos. Usando dos enfoques diferentes para medir la estética de los mapas, se ha comprobado que el generador también es capaz de crear mapas estéticos para el juego en cuestión.

1.3 PUBLICACIONES

A continuación se detallan las publicaciones en las que se basa esta tesis.

- Revistas:
 - R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, "On balance and dynamism in procedural content generation with self-adaptive evolutionary algorithms," *Natural Computing*, vol. 13, num. 2, pp. 157–168, 2014
Índice de impacto (ISI): 0.757
Índice de impacto (SCImago): 0.655
Cuartil: Q3

- R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, “An analysis of the structure and evolution of the scientific collaboration network of computer intelligence in games,” *Physica A: Statistical Mechanics and its Applications*, vol. 395, num. 0, pp. 523–536, 2014
Índice de impacto (ISI): 1.731
Índice de impacto (SCImago): 0.653
Cuartil: Q2
 - R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, “Geometrical vs topological measures for the evolution of aesthetic maps in a RTS game,” *Entertainment Computing*, vol. 5, num. 4, pp. 251–258, 2014
Índice de impacto (SCImago): 0.477
Cuartil (SCImago): Q2
 - R. Lara-Cabrera, M. Nogueira, C. Cotta, y A. J. Fernández-Leiva, “Procedural content generation for real-time strategy games,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 3, num. 2, pp. 40–48, 2015
- Conferencias:
- R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, “Procedural map generation for a RTS game,” en *13th International Conference on Intelligent Games and Simulation - GAMEON 2012*, 2012, pp. 53–58
 - R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, “A review of computational intelligence in RTS games,” en *Foundations of Computational Intelligence (FOCI), 2013 IEEE Symposium on*, 2013, pp. 114–121
 - R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, “A procedural balanced map generator with self-adaptive complexity for the real-time strategy game Planet Wars,” en *Applications of evolutionary computation*, ser. Lecture Notes in Computer Science, A. Esparcia-Alcázar, Ed. Springer Berlin Heidelberg, 2013, vol. 7835, pp. 274–283
Índice de impacto (SCImago): 0.348
Cuartil (SCImago): Q2
 - R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, “Evolving aesthetic maps for a real time strategy game,” en *I Simposio Español de Entretenimiento Digital*. Universidad Complutense de Madrid, 2013, pp. 61–71
 - R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, “Using self-adaptive evolutionary algorithms to evolve dynamism-oriented maps for a real time strategy game,” en *Large-Scale Scientific Computing*, ser. Lecture Notes in Computer Science, I. Lirkov, S. Margenov, y J. Waśniewski, Eds. Springer

Berlin Heidelberg, 2014, vol. 8353, pp. 256–263

Índice de impacto (SCImago): 0.339

Cuartil (SCImago): Q2

- R. Lara-Cabrera, C. Cotta, y A. Fernández-Leiva, “A self-adaptive evolutionary approach to the evolution of aesthetic maps for a rts game,” en *Evolutionary Computation (CEC), 2014 IEEE Congress on*, 2014, pp. 298–304
CORE Rank: B
- R. Lara-Cabrera, M. Nogueira, C. Cotta, y A. J. Fernández-Leiva, “Optimización en videojuegos: retos para la comunidad científica,” en *Actas del X Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, Mérida, España.*, F. Chávez de la O, R. Luque Baena, F. Luna, y F. Fernández de Vega, Eds., 2015, pp. 463–470
- R. Lara-Cabrera, M. Nogueira, C. Cotta, y A. J. Fernández-Leiva, “Game artificial intelligence: Challenges for the scientific community,” en *Actas del II Congreso de la Sociedad Española para las Ciencias del Videojuego, Barcelona, España*, 2015, pp. 1–12

ANTECEDENTES

Este capítulo tiene como objetivo poner en contexto al lector sobre el trabajo que se desarrolla en los siguientes capítulos. En primer lugar se realiza una revisión del estado del arte de la GAC para videojuegos, poniendo especial énfasis en aquellos trabajos enfocados hacia los juegos de ETR, pues éste es el género objetivo de la presente tesis. Posteriormente el lector puede encontrar un capítulo de introducción a la computación evolutiva, cuyas técnicas forman el núcleo de los métodos desarrollados en posteriores capítulos.

2.1 JUEGOS E INTELIGENCIA ARTIFICIAL

La investigación en IA e IC empezó en los 50 y ha estado creciendo desde entonces, con el objetivo de dotar a los ordenadores con la capacidad de realizar tareas que se suponían exclusivas de la inteligencia humana. Desde sus orígenes, este campo de investigación ha necesitado nuevos problemas a los que enfrentarse. Y aquí es donde los videojuegos entran en escena, proporcionando nuevos e interesantes desafíos para la comunidad, incluyendo, entre otros, el desarrollo de jugadores artificiales, agentes que se adaptan al resto de jugadores, sistemas que analizan y aprenden sobre el comportamiento durante el juego de otros jugadores y sistemas de construcción de contenido para juegos, siendo éste último el problema tratado en la presente tesis. Ya que los videojuegos se diseñan para ser desafiantes e interesantes para los humanos, muchos de los problemas de IA relacionados con los videojuegos son relevantes a su vez para entender los procesos cognitivos y la creatividad humana.

Esta sección está dedicada a realizar un repaso por los trabajos que se han publicado en el campo de la inteligencia artificial y videojuegos, en concreto con aquellos trabajos relacionados con la GAC. En primer lugar se reseñan los trabajos enfocados en los juegos de ETR para pasar después a realizar un breve recorrido por los distintos ejemplos en los que este tipo de juegos han servido como entorno de pruebas para técnicas de IA. Por último, se nombran diferentes trabajos que si bien no están enfocados en los juegos de ETR se dedican a la generación automática de mapas y niveles, motivo por el cual han sido incluidos.

2.1.1 Generación de contenido para juegos de ETR

La Generación Automática de Contenido (GAC) se refiere a la creación, mediante algoritmos, de contenido para videojuegos con una contribución humana nula o limitada. El término “contenido” abarca un amplio rango de elementos de los videojuegos (niveles, misiones, personajes, ...), aunque quedan excluidos de la definición tanto el motor del juego como las estrategias de IA. Siguiendo la taxonomía de Togelius *et al.* [153], los algoritmos de GAC se pueden categorizar atendiendo a distintas características, como son el momento en el cual se genera el contenido, su importancia, el nivel de parametrización del algoritmo, el nivel de aleatoriedad y el esquema de generación utilizado (ver Tabla 2.1).

| Característica | Opciones | |
|---------------------------|--------------|---------------------|
| Momento de la generación | Online | Offline |
| Importancia del contenido | Esencial | Opcional |
| Parametrización | Aleatorio | Vector paramétrico |
| Nivel de aleatoriedad | Estocástico | Determinista |
| Esquema de generación | Constructivo | Generación y prueba |

Tabla 2.1: Caracterización de algoritmos de generación automática de contenido

Una primera distinción se basa en el momento en el cual se genera el contenido, que puede ser durante la ejecución del videojuego y por tanto estaríamos ante un algoritmo de generación *online*, o por el contrario la generación se lleva a cabo durante la fase de desarrollo del videojuego, siendo entonces un algoritmo de generación *offline*. Por otra parte, los algoritmos pueden ser diferenciados por el tipo de contenido que producen, ya que éste puede ser esencial para el progreso del jugador a lo largo del juego y, por tanto, debe ser siempre correcto, u opcional. Cabe destacar que un contenido puede ser esencial para un tipo de juego y, a la vez, ser opcional para otros géneros. Otra posible distinción la establece el nivel de parametrización que admite el algoritmo, el cual puede construir el contenido a partir de valores obtenidos por lo que el único parámetro necesario es la semilla del generador pseudo-aleatorio. En el otro extremo estaría el algoritmo que como entrada toma un conjunto de valores que especifican las propiedades del contenido que se va a generar. Con respecto al nivel de aleatoriedad, si sucesivas ejecuciones del algoritmo producen resultados similares, estaremos ante un algoritmo determinista. Por el contrario, el algoritmo será estocástico si sucesivas ejecuciones del algoritmo producen diferente contenido.

Con respecto a la última caracterización, el esquema de generación, se dice que un algoritmo es constructivo si genera el contenido una

única vez, ejecutando secuencias de operaciones que garantizan que el contenido generado es válido. Por el contrario, un algoritmo que sigue un esquema de generación y prueba incorpora ambos mecanismos: cuando un contenido candidato es generado se somete a un proceso de validación según cierto criterio, rechazándose en caso de no resultar válido y volviendo a generarse uno nuevo, proceso que se repite hasta que se obtiene contenido válido. Existe una especialización dentro de los algoritmos de generación y prueba: la generación de contenido basada en búsquedas. En esta categoría, el proceso de validación del contenido va más allá de la aceptación o rechazo, se le asigna un valor de validez al contenido que sirve como retroalimentación al proceso de generación en la siguiente iteración. El algoritmo que se presenta en el Capítulo 4 pertenece a esta categoría.

Además de la anterior categorización, se puede abordar la GAC utilizando distintas familias de técnicas de IA. Yannakakis y Togelius [166] han estudiado las técnicas empleadas en el campo de la IA aplicada a los videojuegos y han encontrado que, en el caso de la GAC, hay una técnica dominante: la computación evolutiva, aunque también se han encontrado algunos casos en los que se usó la planificación de tareas. La explicación de los autores a este fenómeno es que la computación evolutiva encaja perfectamente en el esquema de generación basada en búsquedas, además de que la GAC se realiza, principalmente, de forma offline, por lo que el elevado coste computacional de esta técnica no supone un problema.

Los juegos de ETR son un género de videojuegos en los cuales se requiere la gestión de diferentes tipos de unidades y recursos en tiempo real. Por norma general, los participantes posicionan y manejan unidades y estructuras bajo su control para asegurar áreas del mapa y/o destruir las de los oponentes. En un juego de ETR típico cabe la posibilidad de construir nuevas unidades y edificios en el transcurso de la partida, aunque esto se limita en base a los recursos acumulados, que se recogen precisamente con edificios y unidades específicas para ello. Estas características convierten a los juegos de ETR en una herramienta idónea para la investigación en inteligencia computacional/artificial [19], ya que sus jugadores necesitan solucionar varios problemas como la gestión de recursos, el razonamiento espacial, la planificación estratégica, la anticipación a los oponentes o incluso la GAC para crear distintos mapas y unidades. Esto se refleja en el elevado número de artículos sobre inteligencia artificial/computacional y juegos de ETR [86].

Como se ha comentado anteriormente, los juegos de ETR son susceptibles de ser objetivo de la GAC, como sucede con los trabajos de Frade *et al.* [46, 47, 48, 49] en los cuales se estudia la generación automática de mapas de altura que representan terrenos y mapas para videojuegos. Mediante la programación genética se consiguen obtener diversos generadores de terrenos los cuales, a su vez, son capaces

de generar distintos tipos de terrenos con características en común cuando se le proporcionan valores de entrada diferentes. Los autores incorporan también la interactividad humana en el proceso de evolución de los generadores, para que los terrenos obtenidos cumplan con algún criterio estético por parte del usuario. Los autores también han estudiado la evaluación automática de los generadores en función de unos parámetros de accesibilidad en lugar de la evaluación interactiva por parte del experto.

En [152], los autores estudian un generador automático de mapas para el juego *Starcraft* empleando para ello una optimización multiobjetivo. Los mapas se codifican como vectores de números reales que especifican la distribución y características de los principales elementos de un mapa, como la situación de las bases y recursos, así como las propiedades de las zonas no transitables del mapa. Para la evaluación de los mapas se utilizan ocho medidas de calidad diferentes, todas ellas relacionadas con distancias calculadas mediante el algoritmo A^* [66] y que buscan diversas características deseables como la jugabilidad y la imparcialidad con respecto a los jugadores.

Siguiendo con la generación de mapas para juegos de ETR, Mahlmann *et al.* [102] presentan un sistema de creación de mapas para el juego *Dune 2* basado en un algoritmo evolutivo. El generador consta de dos partes diferenciadas, una primera que se encarga de transformar los individuos del algoritmo evolutivo en mapas adecuados para el juego (es decir, una transformación de genotipo a fenotipo), y una segunda que es el sistema evolutivo en sí. La evaluación de los individuos se realiza después de la transformación, la cual convierte un vector de parámetros, que son optimizados por el algoritmo evolutivo, en una matriz de valores que determinará la distribución de los distintos materiales y objetos que conforman un mapa jugable.

Por su parte, Uriarte y Ontañón [157] han diseñado un generador de mapas para *Starcraft: Brood War* que han denominado *PSMAGE*. Este sistema divide el mapa en regiones mediante un diagrama de Voronoi y, posteriormente, asigna diferentes propiedades a cada una de las regiones, como la altura de la misma. También selecciona qué región va a ser el punto de partida de cada jugador y distribuye los recursos entre el resto de regiones, todo ello intentando mantener la simetría del mapa de forma que se mantenga equilibrado.

Siguiendo un enfoque similar, Liapis *et al.* [95] presentan un generador automático de bocetos de mapas para juegos de ETR. Estos bocetos son matrices de baja resolución (es decir, con pocos elementos) que distribuyen en el espacio los tipos de terreno y recursos. Posteriormente, los bocetos se expanden, usando autómatas celulares, y se convierten en mapas jugables que respetan la distribución original del boceto.

Mahlmann *et al.* [100] han diseñado un lenguaje de descripción de juegos de estrategia (*Strategy Game Description Language, SGDL*) con

varios criterios de diseño en mente: el lenguaje tenía que ser completo para poder modelar prácticamente todos los aspectos de los juegos de estrategia, que fuese posible su evolución de forma que fuese sencillo aplicar algoritmos de búsqueda y optimización, y entendible por las personas. El lenguaje utiliza una representación basada en árboles, al estilo de la programación genética. Este lenguaje se ha empleado en la generación automática de conjuntos de tipos de unidades que se complementan entre sí, destacando cada una de ellas en alguna característica en particular [101]. También se ha usado para la creación automatizada de conjuntos de reglas o escenarios de juegos de estrategia. En [100] se presentan tres métodos para evaluar la calidad de los posibles conjuntos de reglas generados de forma automática a partir del lenguaje de descripción.

Además de los anteriores artículos, que estaban enfocados directamente en los juegos de ETR, hay otros trabajos de generación automática de contenido que, sin centrarse específicamente en un género concreto de videojuegos, pueden aplicarse a este género en cuestión. La generación de terrenos virtuales por procedimientos es un problema ampliamente estudiado en la literatura [141] y cuyos resultados pueden ser útiles si se aplican a los videojuegos de ETR, ya que estos terrenos pueden servir como base para los mapas del juego. Un ejemplo de generación procedimental de terrenos es el trabajo de los autores de van Eck y Lamers [159], los cuales presentan un estudio preliminar sobre la posibilidad de generar terrenos para videojuegos a partir de organismos vivos. En concreto, convierten imágenes de cultivos de hongos y bacterias mediante el escaneo de las placas de Petri en mapas de altitud que representan los terrenos virtuales. Otro ejemplo es el trabajo de Ashlock *et al.* [5] que utiliza una representación indirecta del terreno (concretamente polígonos codificados usando *L-systems*) que da indicaciones sobre cómo construirlo en lugar de ser una representación directa del mismo y que permite el uso de heurísticas a la hora de realizar la transformación al modelo final del terreno. Un enfoque distinto es el de Doran y Parberry [32] que proponen la utilización de agentes inteligentes especializados en construir diferentes tipos de terreno (playas, colinas, llanuras, ...) de forma que en conjunto son capaces de crear terrenos virtuales que obedecen una serie de parámetros que se especifican al inicio del proceso.

Otro tipo de contenido genérico que se puede adaptar para los juegos de ETR es el de la música [23]. Por ejemplo, Plans y Morelli [127] han estudiado la generación de música y efectos de sonido de forma dinámica y automática para que se adapte a la experiencia del jugador en todo momento. Para ello se analizan las acciones del jugador, que influyen en tres funciones que definen el nivel de frustración, desafío y diversión del mismo. Estos niveles, a su vez, influyen en tiempo real en el algoritmo de generación de música, adaptándose

ésta última a la situación particular que se esté produciendo en el juego.

2.1.2 Juegos de estrategia en tiempo real e inteligencia artificial

La estrategia en tiempo real es un género de videojuegos de gran importancia dentro de la industria. Desde los primeros éxitos comerciales como *Dune II: The building of a Dynasty* [164] y *Warcraft: Orcs & Humans* [11] hasta éxitos más recientes como *Starcraft II* [12] o *Supreme Commander 2* [54], pasando por verdaderos pesos pesados como las sagas *Age of Empires* [36] y *Command & Conquer*, [165] este género ha sabido atraer a una gran cantidad de público. Más allá del éxito comercial del género, los juegos de ETR también se han hecho un hueco dentro de la investigación en Inteligencia Computacional (IC), debido principalmente a los numerosos desafíos y problemas que proporcionan este tipo de juegos.

Planificación de estrategias

El problema más tratado dentro del campo de los juegos de ETR es, sin duda, la planificación de estrategias, que consiste en determinar secuencias de acciones de forma que, al ejecutarse, se cumpla un determinado objetivo. En el caso de este género, el hecho de que existan adversarios sumado a que las acciones se llevan a cabo en tiempo real y en ausencia de información perfecta complica la planificación de estrategias.

Aha *et al.* [2, 109] presentaron un algoritmo de selección de estrategias denominado *Case-based Tactician (CaT)* que basa su funcionamiento en el conocimiento del problema y el razonamiento basado en casos, es decir, realizar la selección de estrategias en base a las decisiones tomadas en partidas anteriores. Este algoritmo se mejoró posteriormente [110] para poder razonar y aprender a partir de espacios continuos de acciones, en lugar de aproximaciones discretas como en el caso del algoritmo original. Otra mejora a la selección de estrategias mediante el razonamiento basado en casos es la propuesta por Mishra *et al.* [107], quienes introdujeron en el algoritmo una representación de alto nivel sobre el estado global de la partida, mejorando su eficiencia. Por su parte, Sharma *et al.* [140] describieron una arquitectura que permite el aprendizaje automático de criterios tácticos que pueden reutilizarse en otras instancias del problema. Este mecanismo de aprendizaje permite que el conocimiento adquirido previamente se use para mejorar el rendimiento de la planificación de estrategias en las siguientes partidas. Ontañón *et al.* [122, 123] propusieron la extracción de comportamientos a partir de demostraciones de expertos y su posterior reutilización como un generador de comportamientos basado en casos, además de presentar un marco de trabajo para la planificación basada en casos para juegos de ETR. Otro posible enfoque para la planificación es el empleado por Alcázar *et al.* [3], quienes usaron el lenguaje *PDDL* para definir un dominio de

planificación que puede ser usado en los juegos de ETR, de la misma forma que Kovarsky y Buro [80], quienes también usaron PDDL en este caso para la optimización del orden de construcción de los diferentes edificios del juego. Por su parte, Weber *et al.* [161] estudiaron un enfoque dirigido por objetivos para desarrollar jugadores artificiales que modifican sus objetivos en base a los eventos de la partida que se van sucediendo.

La computación evolutiva también se ha usado de forma frecuente a la hora de la planificación de estrategias, como en el trabajo de Jang *et al.* [73] donde se usa un algoritmo evolutivo para mejorar el rendimiento de un jugador artificial, el trabajo de Lichoki *et al.* [96] que usa programación de expresiones de genes, el de Miles *et al.* [106] con un algoritmo genético, o el de Ponsen *et al.* [128] y el trabajo de Traish y Tulip [156], que usan aprendizaje evolutivo.

Otra técnica muy utilizada para la planificación es el *scripting* dinámico [145], que es una técnica de aprendizaje por refuerzo y que fue extendida por Dahlbom y Niklasson [29]. Algunos ejemplos de su uso para juegos de ETR incluyen el trabajo de Ponsen *et al.* [128] donde se combinó con un algoritmo evolutivo, y el de Ludwig y Farley [98], donde se extendió la técnica para comprobar su efectividad en un juego táctico abstracto.

La gestión dentro de los juegos de ETR se pueden dividir en dos niveles: macro-gestión (es decir, tomar decisiones estratégicas como el edificio que se construirá a continuación o qué zona del mapa debería ser explorada) y micro-gestión (formación de las unidades así como designación de sus objetivos). Al igual que la planificación, la micro-gestión es un problema que se ha tratado ampliamente en la literatura, utilizando diferentes técnicas para ello, tales como el aprendizaje por refuerzo [60, 163].

Hagelbäck y Johansson [63, 62, 64, 61] estudiaron el uso de campos potenciales y esquemas multi-agente para el manejo de las unidades teniendo en cuenta la niebla de guerra de este tipo de juegos, que no es otra cosa que la incapacidad de observar regiones del mapa en las cuales no se controlen unidades. La misma técnica ha sido empleada tanto por Avery y Louis [6] como por Keaveney y O'Riordan [76, 77], quienes han combinado los mapas potenciales con la computación evolutiva. En el mismo sentido, Synnaeve y Bessiere [147] propusieron controlar las unidades con un modelo Bayesiano.

Por su parte, Ng *et al.* [117] describen un método para planificar la división de las unidades en sub-grupos o formaciones adecuadas para cumplir los objetivos de la partida. Siguiendo con la gestión de las formaciones de las unidades, Preuss *et al.* [129] utilizaron para ello una combinación de mapas auto-organizados y algoritmos evolutivos para adaptarlos al juego, mientras que van der Heijden *et al.* [158] se basaron en el aprendizaje computacional.

El reconocimiento de estrategias se refiere a las acciones llevadas a

Micro-gestión

Reconocimiento y predicción de estrategias

cabo por un jugador con la intención de predecir las acciones futuras, intenciones u objetivos del oponente. Esto tiene una vital importancia a la hora de desarrollar un jugador artificial proactivo que sea capaz de adelantarse a los acontecimientos. En [21] se presenta una introducción al reconocimiento de estrategias basado en casos para juegos de ETR, un enfoque que se utilizó en [72], donde se analizaron y categorizaron diferentes estilos de juego mediante los datos recogidos sobre la secuencia en la construcción de los edificios. Estos datos, que en realidad eran repeticiones de partidas de *Starcraft*, se emplearon para entrenar un sistema de decisión basado en casos capaz de predecir la estrategia del oponente. Este enfoque de análisis de repeticiones de partida también se utilizó para extraer modelos probabilísticos del comportamiento de los oponentes [31, 58].

Por su parte, Schadd *et al.* [134] estudiaron el uso de modelos jerárquicos estructurados para modelar el comportamiento del oponente, además de comprobar la efectividad de este enfoque mediante la evaluación de dos clasificadores, mientras que en el caso de Synnaeve y Bessiere [146, 148], la predicción se llevó a cabo utilizando un modelo Bayesiano. Otro enfoque para la predicción es el propuesto por Weber y Mateas [162], quienes plantearon el uso de técnicas de minería de datos para extraer patrones de comportamiento de una colección numerosa de partidas guardadas.

Juegos de ETR como entornos de prueba

Se puede observar que los juegos de ETR pueden ser de utilidad a la hora de desarrollar e investigar nuevas técnicas y algoritmos de IA. Esto ha propiciado que se desarrollen herramientas y librerías para facilitar la tarea de usar estos juegos como entorno de pruebas para evaluar el rendimiento de los algoritmos.

Starcraft es uno de los juegos que aparece frecuentemente en la literatura gracias a la librería de código abierto *BWAPI - The Brood War API*¹ que permite interactuar con el motor del juego controlando las unidades individualmente y accediendo a todos los aspectos relevantes del estado del juego. La mayoría de los trabajos que han usado *Starcraft* se corresponden con el desarrollo de jugadores artificiales como, por ejemplo, el artículo de Justesen *et al.* [75] que presenta un estudio sobre secuencias óptimas de acciones que deben tomar las unidades enfrentadas en combate. También hay trabajos [121, 162] que utilizan técnicas de aprendizaje computacional y minería de datos para extraer y analizar patrones de comportamiento de jugadores humanos.

Aunque la librería permita conectar jugadores artificiales con *Starcraft*, éste no se diseñó con ese objetivo en mente, justo al contrario que *ORTS (Open Real-Time Strategy)* [18]. *ORTS* es un motor de juegos de ETR diseñado explícitamente para servir como herramienta de apoyo a la investigación de problemas de IA en tiempo real. Está desarrollado siguiendo una arquitectura cliente/servidor que incluye

¹ <https://github.com/bwapi/bwapi>

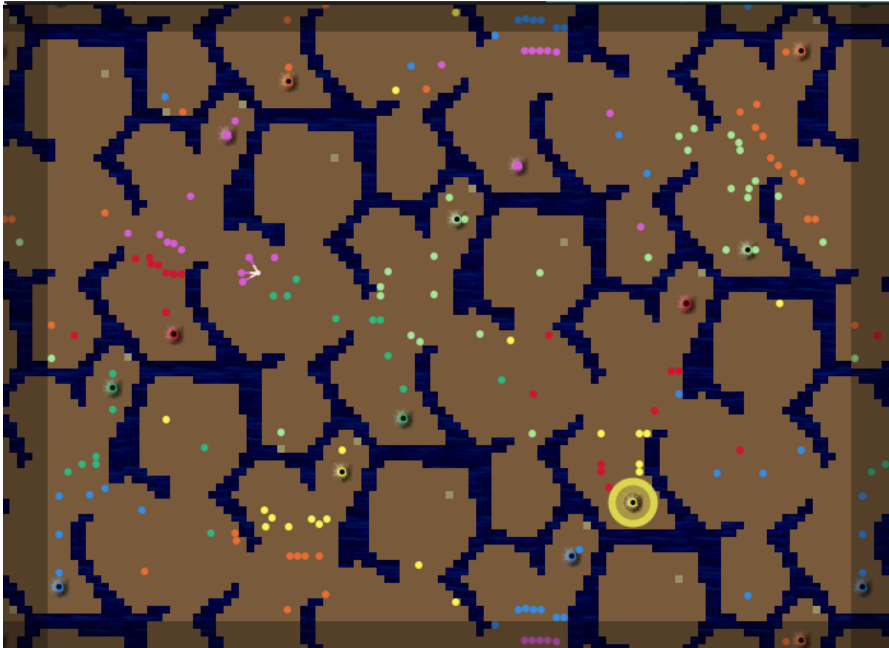


Figura 2.1: Captura de pantalla del juego *Ants*. Los puntos de colores representan a las hormigas de cada equipo mientras que los círculos negros son los hormigueros.

visualización 3D al nivel de juegos comerciales, mientras que el motor de juego se puede personalizar a través de *scripts* que describen los tipos de unidades, las estructuras disponibles y las interacciones entre ambos. Un ejemplo de uso de esta herramienta es el trabajo de Hagelbäeck y Johansson [63], que estudia el uso de campos potenciales multi-agentes para controlar el movimiento de las unidades del juego generando atracción y repulsión en distintas zonas del mapa dependiendo del edificio o unidad que allí se encuentre.

En la línea del anterior juego, hay otros dos que se crearon con el mismo objetivo en el marco de la competición de *IA Google AI Challenge*², que ha sido una competición internacional de IA patrocinada por Google. En primer lugar se encuentra el juego *Planet Wars*, que es el empleado en la presente tesis y cuya descripción completa se puede encontrar en la sección 2.1.4. Mora *et al.* [111] utilizan el *Planet Wars* para estudiar el efecto que tienen las funciones de aptitud con un elevado número de perturbaciones en el desarrollo de estrategias de IA para construir jugadores artificiales, además de diseñar un jugador artificial de *Planet Wars* mediante AE que obtuvo un mejor rendimiento que otro jugador virtual programado según el conocimiento de un experto. Otro trabajo que utilizar a este juego como base es el de Fernández-Ares *et al.* [41], que presenta un método rápido y online para el análisis de los mapas de este juego, de forma que el conocimiento obtenido sirve para definir estrategias especializadas para un

² <http://aichallenge.org/>

jugador artificial [39, 40]. Nogueira Collazo *et al.* [118, 120] también han estudiado el diseño de jugadores artificiales para el *Planet Wars*, en este caso, mediante el uso de algoritmos coevolutivos competitivos, enfoque similar al empleado en [42].

En segundo lugar, y también en el marco de la *Google AI Challenge*, se desarrolló otro juego de ETR denominado *Ants*, en el cual los jugadores tienen que gestionar una colonia de hormigas con el objetivo de recoger alimentos y conquistar los hormigueros del resto de jugadores (ver Figura 2.1. Ambos juegos permiten conectar los jugadores artificiales al motor del juego de forma inmediata, ya que al haber sido desarrollados en el marco de una competición de IA, se han desarrollado precisamente de forma que esto sea posible.

2.1.3 Generación de mapas y niveles para otros géneros

Más allá de los juegos de ETR hay una gran cantidad de artículos dedicados a la generación automática de mapas y niveles para otros géneros de videojuegos. Por ejemplo, Togelius *et al.* [151] han diseñado un sistema capaz de generar circuitos de carreras a partir de un vector de parámetros que se interpreta, a su vez, como los parámetros de una secuencia de curvas Bezier que forman el circuito en sí. La evaluación de los circuitos se realiza por simulación, utilizando un conductor artificial cuyo rendimiento a la hora de conducir en el circuito forma parte del valor de aptitud del mismo. Con el mismo juego como objetivo, Cardamone *et al.* [20] presentan una herramienta accesible desde la web para la generación de circuitos que basa su funcionamiento en la evolución interactiva, es decir, incorporar al humano en el proceso de evaluación de la calidad de los circuitos obtenidos por la herramienta, para ayudar al algoritmo a obtener mejores soluciones.

Por su parte, Jennings-Teats *et al.* [74] presentan un estudio preliminar sobre la creación automatizada de niveles para juegos de plataforma que adaptan su dificultad al nivel del jugador durante la partida. Siguiendo con los juegos de plataformas, Pedersen *et al.* [125] modificaron una versión de código abierto del juego *Super Mario Bros* de forma que fuese posible la generación de niveles personalizados. En este caso, los niveles se codifican en un vector de parámetros indicando el número y tamaño de los distintos huecos en el nivel. El resto de elementos se coloca siguiendo un proceso estocástico. En el caso de Compton *et al.* [24, 142] los niveles del juego de plataformas se generan combinando diferentes patrones de forma rítmica y teniendo en cuenta la física del avatar controlado por el jugador a la hora de efectuar saltos y desplazamientos. Snodgrass y Ontañón [143] presentan un estudio preliminar sobre el uso de cadenas de Markov en el proceso de creación de niveles para un juego de plataformas. El método que presentan es capaz de aprender patrones estadísticos a partir de

niveles diseñados por expertos para, posteriormente, generar nuevos mapas de forma automática en base a los patrones aprendidos.

En el ámbito de los puzzles, Shaker *et al.* [137, 138] proponen un sistema evolutivo para generar niveles del juego *Cut the rope!*. La evaluación de los niveles generados combina un conjunto de restricciones de diseño así como una comprobación de jugabilidad basada en la simulación con un agente inteligente que juega los niveles.

Usando un enfoque algo más amplio, Sorenson y Pasquier [144] estudiaron una representación genérica para los niveles de videojuegos de forma que abarcasen varios géneros y, por tanto, generalizar el proceso de creación automática de los mismos. La representación de los niveles se basa en lo que los autores han llamado “elementos de diseño” cuya composición da como resultado niveles completos.

2.1.4 *Planet Wars: Descripción del juego*

Como ya se ha descrito en la Sección 2.1.2, muchos juegos de ETR son usados de forma frecuente como entorno de pruebas para la investigación en técnica de IA. Para este trabajo se ha elegido el juego Planet Wars como entorno de pruebas para el generador automático de mapas que se propone en el Capítulo 4 así como para el resto de experimentos realizados. El motivo de esta elección no es otro que la sencillez en la representación de los mapas, la disponibilidad del código fuente del mismo así como la capacidad de encapsular la esencia de los juegos de ETR con un conjunto sencillo de reglas.

Planet Wars es un juego abstracto desarrollado en el marco de la competición de estrategias de IA *Google AI Challenge 2010*³ e inspirado en el arcade *Galcon*⁴. El juego tiene lugar en el espacio y los jugadores toman el control de una flota de naves con el objetivo de conquistar todos los planetas del mapa o bien eliminar todas las flotas del enemigo.

Cada partida se desarrolla en un mapa bidimensional compuesto por planetas de diferentes tamaños (ver Figura 2.2). El tamaño de un planeta indica su capacidad de producción de unidades, por lo que los planetas más grandes son a su vez más atractivos, ya que generan nuevas naves en una cantidad directamente proporcional a dicho tamaño, siempre que el planeta esté controlado por algún jugador (los planetas neutrales no producen unidades). Por su parte, las naves de los jugadores se organizan en flotas o grupos de naves que, a su vez, pueden encontrarse en movimiento desde un planeta a otro o bien defendiendo un determinado planeta. Aunque los jugadores emiten las órdenes de movimiento de flotas turno a turno, se puede considerar que se realiza en tiempo real, puesto que dichas órdenes se ejecutan en paralelo.

Ambientación y objetivo

Mecánicas de juego

³ <http://planetwars.aichallenge.org>

⁴ <http://www.galcon.com>

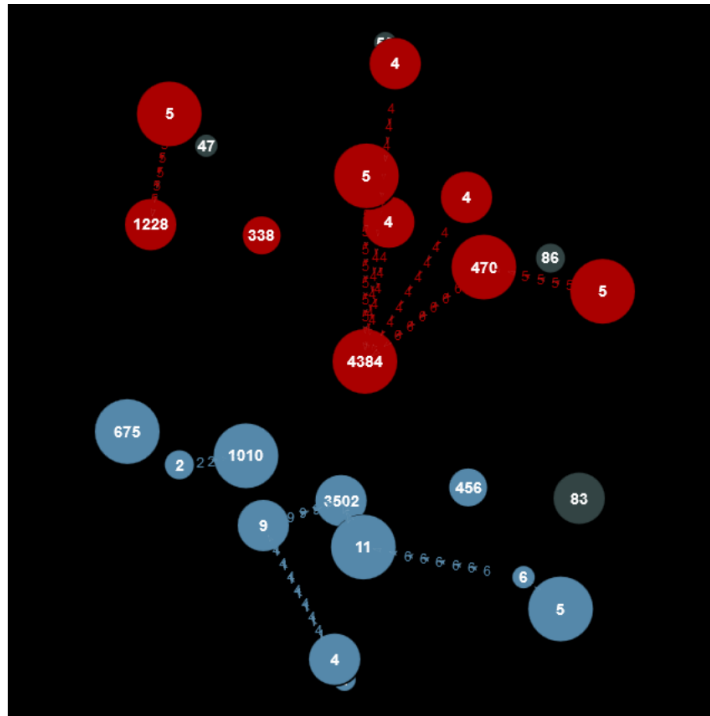


Figura 2.2: Una partida de *Planet Wars* en marcha. Los círculos representan los planetas mientras que el color de los mismos y el número que incluyen representan el jugador propietario y el número de naves de la flota defensora, respectivamente. Los números entre planetas son flotas de naves que se encuentran en movimiento.

*Conquista, batalla,
refuerzos y
movimientos*

En el transcurso de un turno, el jugador puede ordenar el envío hacia un planeta objetivo de todas las naves que se encuentren en un planeta que controle. Esta flota tardará en llegar a su destino un tiempo relacionado con la distancia que existe entre el planeta de origen y de destino. Es imposible reasignar el planeta objetivo de una flota que se encuentre en movimiento. Si el planeta objetivo también es controlado por el jugador, las naves que forman la flota se suman a las que ya hubiese en el planeta, pasando a formar parte de la defensa del mismo. Por su parte, si el objetivo no está controlado por el jugador (es neutral o de un jugador enemigo), se produce el combate entre la flota atacante y la defensora. El combate se resuelve de forma simple, se van destruyendo naves de ambas flotas en una proporción de una atacante por una defensora, hasta que alguna de las dos flotas se quede sin naves, resultando ser la perdedora. El control del planeta pasa a ser de la flota ganadora y las naves restantes se incorporan a la defensa del mismo. Como ya se ha comentado, al final del turno cada planeta, dependiendo de su tamaño, incrementa el número de naves de la flota defensora.

Final del juego

El juego llega a su fin cuando se cumple alguna de las siguientes condiciones: un jugador consigue conquistar todos los planetas del

mapa o eliminar todas las naves del resto de jugadores, o bien se han jugado un número determinado de turnos, por lo que el jugador con mayor número de naves es el que se convierte en el ganador.

2.2 INTRODUCCIÓN A LA COMPUTACIÓN EVOLUTIVA

La computación evolutiva [7, 35] es la rama de la informática dedicada al estudio de una clase de algoritmos que se inspiran en la teoría de la evolución Darwiniana, concretamente en la selección natural, así como en la genética molecular, razón por la que se les conoce como algoritmos bio-inspirados.

Dentro de los algoritmos evolutivos existen diferentes variantes que comparten la misma idea principal: las soluciones a un determinado problema forman una población de individuos, cuya medida de calidad global se ve incrementada por los mecanismos de selección natural que provocan que solo sobrevivan los más fuertes. Una vez se tiene definida una función de aptitud que mide cómo de buena es una determinada solución, se pueden generar éstas de forma aleatoria y someterlas a evaluación con la función de aptitud, que indicará las posibilidades de supervivencia de la solución. En base a este valor de aptitud, se eligen aquellos individuos (soluciones) de entre la población que tienen un valor más alto y se les somete a un proceso de recombinación y mutación para generar nuevos individuos que formarán la nueva población de soluciones. Este proceso se puede repetir de forma iterativa hasta que alguno de los individuos alcance un valor de aptitud objetivo o se consuman un número máximo de iteraciones o recursos computacionales definidos al inicio del proceso.

La conjunción de la diversidad introducida por los operadores de mutación y recombinación y la orientación hacia la calidad del proceso de selección forman la base de los sistemas evolutivos. Cabe destacar que en ambos casos los procesos son estocásticos. Por su parte, la función de aptitud se puede considerar como una función heurística que ofrece una aproximación de la calidad de la solución.

Como ya se ha mencionado, dentro de la computación evolutiva existen diferentes vertientes como la programación genética, las estrategias de evolución y algoritmos genéticos, que, aun difiriendo en los detalles técnicos de implementación tales como la forma de representación de los individuos, siguen el esquema general reflejado en la Figura 2.3. El proceso se inicia creando aleatoriamente la población inicial de individuos, entonces comienza un proceso iterativo que terminará cuando se alcance la condición de parada deseada. Durante el proceso iterativo, en cada paso, se seleccionan unos progenitores que son sometidos a recombinación y mutación para generar nuevos candidatos, los cuales reemplazan a otros individuos de la población.

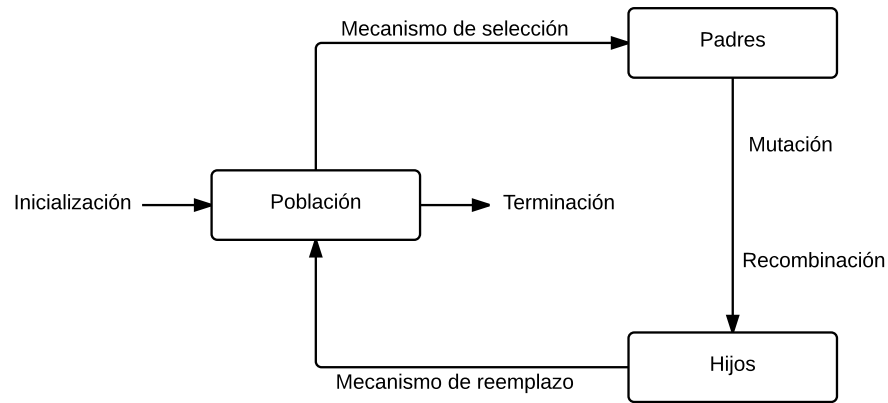


Figura 2.3: Esquema general del funcionamiento de un algoritmo evolutivo

Como se puede observar, la computación evolutiva sigue un proceso de generación y prueba, buscando en el espacio de soluciones con un equilibrio entre exploración y explotación. Además, al emplearse un método poblacional, se obtienen del proceso no solo la mejor solución, sino un conjunto de soluciones que han ido mejorando a lo largo del proceso y que puede ocurrir que muchas de ellas tengan la calidad suficiente como para ser consideradas correctas. Es por ello que se ha optado por la computación evolutiva como base para el generador automático de mapas. A continuación se realiza una breve reseña histórica sobre estas técnicas así como una introducción a las vertientes más comunes que han ido surgiendo dentro de la computación evolutiva.

2.2.1 Reseña histórica

Orígenes

Una de las primeras referencias que se tienen del uso de un proceso evolutivo para la resolución de un problema se remonta a mediados de los 50, en los trabajos de Fraser [50] (1957), Friedberg [52] (1958) y Friedberg *et al.* [53] (1959). El primero está considerado como un trabajo influyente dentro del campo de la computación evolutiva, mientras que los otros dos suponen una primera aproximación a lo que los autores denominaron como *programación automática*, es decir, calcular la función que representa a un conjunto de datos de entrada y salida. Unos años más tarde, Bremermann [17] presentó los resultados de los primeros intentos de aplicar evolución simulada a problemas de optimización numérica, tanto lineal como convexa, además de aplicarla para obtener la solución de ecuaciones simultáneas no lineales.

Aunque estos trabajos se pueden considerar como fuente de inspiración para la computación evolutiva, no es hasta mediados de los 60 cuando se establecen las bases de lo que todavía hoy se conocen como las principales formas de AE. Las estrategias de evolución fue-

ron desarrolladas inicialmente en Berlín por los estudiantes Bienert, Rechenberg y Schwefel [131]. Por su parte, Fogel *et al.* [44] estableció los fundamentos de la programación evolutiva mientras que Holland [69] hizo lo mismo para los algoritmos genéticos. Cada una de estas vertientes se fueron desarrollando de forma independiente a lo largo de los años, hasta que, a principios de los 90, un nuevo enfoque se incorporó al trío gracias a Koza: la programación genética [81].

2.2.2 Componentes de un algoritmo evolutivo

Como se ha comentado anteriormente, las distintas variantes dentro de los algoritmos evolutivos tienen en común su esquema general, del cual se pueden extraer los distintos componentes que caracterizan y configuran el algoritmo evolutivo en sí mismo.

En primer lugar es necesario definir la representación que tendrán los individuos, o lo que es lo mismo, cómo se van a representar las soluciones candidatas. Es evidente que la representación de los individuos está directamente relacionada con el problema que se desea resolver. Puede ocurrir que la representación elegida para los individuos no puede utilizarse tal cual como solución candidata del problema a resolver, por lo que el proceso evolutivo se realiza en base a la representación (genotipo) y, una vez finalizado, los candidatos vencedores se decodifican en soluciones reales del problema (fenotipo). Un ejemplo clásico de esta situación es un problema de optimización en el dominio de los enteros, donde el fenotipo estaría formado por el conjunto de enteros que forman la solución y el genotipo es la representación en binario de dichos números.

El segundo componente relacionado intrínsecamente con el problema a resolver es la función de aptitud, también conocida como función de evaluación, que es una forma de encapsular los requisitos que deben cumplir las soluciones deseadas y por tanto hacia donde deben evolucionar los individuos. La función de aptitud asigna una medida de calidad a los individuos y se utiliza tanto en el proceso de selección de progenitores como en la política de reemplazo empleada, esto es a qué individuos reemplazan los descendientes que se han creado. La función de evaluación puede ser tan inmediata como resolver un sencillo cálculo matemático o, por el contrario, cuantificar el resultado de una compleja simulación.

Por su parte, los individuos se agrupan en lo que se denomina población del algoritmo, o lo que es lo mismo, un conjunto que permite individuos repetidos. Realmente, durante la ejecución del algoritmo los individuos como tales no evolucionan ya que son estáticos, sino que es la población la que lo hace, reemplazando alguno de sus componentes en cada iteración. A la hora de definir un algoritmo evolutivo simple, basta con establecer el tamaño que tendrá la población. Sin embargo, existen algoritmos más sofisticados que establecen caracte-

Representación de los individuos

Función de aptitud

Población de individuos

rísticas adicionales a la población, como, por ejemplo, una relación de vecindad o distancia entre individuos.

Mecanismo de selección

El mecanismo de selección define la forma de seleccionar algunos individuos de entre toda la población atendiendo a sus valores de aptitud, por lo que generalmente aquellos individuos mejor adaptados tienen más posibilidades de ser seleccionados para engendrar nuevos candidatos, fomentando de esta forma la mejora en la calidad global de la población. Los individuos seleccionados son sometidos a los mecanismos de mutación y recombinación.

Operadores de variación

Por su parte, los operadores de variación tienen como objetivo fundamental crear nuevos individuos a partir de los antiguos, explorando por tanto nuevas posibles soluciones al problema. Los operadores de variación se categorizan en operadores de mutación y recombinación, según se apliquen sobre uno o varios individuos, respectivamente (este número se conoce como aridad del operador). El operador de mutación se aplica sobre un individuo, obteniendo una versión ligeramente modificada del mismo. Este operador es siempre estocástico y depende de una serie de procesos aleatorios que deciden, por ejemplo, la parte del individuo que será modificada y cuánto será esa modificación. Por su parte, el operador de recombinación o cruce funciona obteniendo información de dos o más individuos, combinándola en nuevos candidatos que compartirán algunas de las características de los individuos progenitores. Al igual que la mutación, la recombinación es un proceso estocástico.

Mecanismo de reemplazo

Por último, el mecanismo de reemplazo se encarga de diferenciar los individuos de la población en base a su valor de aptitud, configurando la forma de seleccionar qué individuos pasan a la población de la siguiente iteración y cuales son eliminados. Este reemplazo es necesario puesto que, generalmente, el tamaño de la población permanece constante a lo largo de las iteraciones y después de aplicar los operadores de variación nos encontramos con más individuos de los que teníamos originalmente.

Algoritmo 1: Funcionamiento general de un algoritmo evolutivo

```

inicializar(población);           //soluciones aleatorias
evaluar(población);
repetir
|   padres ← selección(población)
|   descendientes ← cruce(padres)
|   descendientes ← mutación(descendientes)
|   evaluar(descendientes)
|   población ← reemplazo(población,descendientes)
hasta condición de parada;
devolver población;

```

En el Algoritmo 1 se detalla el funcionamiento general de un algoritmo evolutivo. Los algoritmos evolutivos encajan a la perfección con el esquema de generación y evaluación y, por ello, forman la base del generador de mapas que se propone en el Capítulo 4.

2.2.3 Algoritmos genéticos

Dentro de la familia de la computación evolutiva existen distintas clases de algoritmos que difieren en sus detalles de funcionamiento, aun estando inspirados en el mismo patrón general. En primer lugar encontramos la familia de algoritmos más conocida por la comunidad científica: los algoritmos genéticos [67].

Estos algoritmos son considerados comúnmente como métodos de optimización de funciones. En su versión canónica, la representación de las soluciones que se utiliza son las cadenas binarias, mientras que exhiben una baja probabilidad de mutación, la cual se lleva a cabo invirtiendo el valor de determinadas posiciones de la cadena. Por su parte, la selección de los individuos se realiza de forma proporcional a sus valores de aptitud, de forma que aquellos individuos más aptos serán seleccionados con una mayor probabilidad. Ya que la probabilidad de mutación es baja, los algoritmos genéticos se basan en el mecanismo de recombinación para generar nuevas soluciones, siendo ésta la característica más distintiva de este tipo de algoritmos.

Aunque la versión canónica del algoritmo genético cuenta con una representación binaria de los individuos, es posible usar otras representaciones que se adapten mejor a las características de las soluciones que se desean encontrar. Además de la binaria, las representaciones más frecuentes dentro de los algoritmos genéticos son la representación entera, en coma flotante, y las permutaciones para aquellos problemas en los que hay que decidir el orden en el cual debe ocurrir una secuencia de eventos. Dependiendo de la representación elegida, los operadores de mutación y recombinación varían. Por ejemplo, en el caso de la representación entera o real, la mutación podría realizarse asignando un valor aleatorio permitido al gen que se desea mutar, o sumar o restar un determinado valor al valor actual del gen. En el caso de las permutaciones, una posible mutación sería intercambiar dos genes de posiciones aleatorias, mover uno de ellos hasta colocarlo inmediatamente anterior a otro, o incluso “barajar” un subconjunto de la permutación.

Con respecto a la recombinación, para las representaciones numéricas (binario, enteros y reales) existen los cruces de uno o varios puntos, explicado en la sección 4.2, mientras que para las permutaciones destaca el operador *PMX* (*Partially mapped crossover*), presentado por primera vez en el trabajo de Goldberg y Lingle [57].

2.2.4 Estrategias de evolución

Los algoritmos que se detallan a continuación son las denominadas estrategias de evolución [135], las cuales se usan normalmente en la optimización de problemas con parámetros continuos.

Esquema general de una estrategia de evolución

Generalmente, la mutación que se le aplica a los individuos en este algoritmo es una perturbación aleatoria que sigue una distribución normal $N(\mu, \sigma)$. La recombinación, por su parte, elige un elemento al azar de cada individuo e intercambia sus valores. La selección de estos individuos es determinista y no está sesgada por los valores de aptitud, sino por el orden que ocupa cada individuo en la ordenación de los mismos según su valor de aptitud, mecanismo similar al que se sigue para la política de selección de supervivientes y que puede descartar o tener en cuenta a la población antigua. Con respecto a la representación de los individuos, se suele utilizar un vector de número reales, al cual se incorporan los parámetros μ y σ que definen la mutación, concretamente la distribución normal de la que se obtiene la perturbación aleatoria como se ha descrito anteriormente, que van coevolucionando a la vez que las soluciones, dando lugar a lo que se ha denominado la auto-adaptación.

Auto-adaptación

Esta auto-adaptación es la contribución más importante de las estrategias de evolución al campo de la computación evolutiva, ya que aunque inicialmente se incluyera en las primeras, sus ventajas han hecho que también se use la auto-adaptación en otras familias de algoritmos evolutivos [104].

2.2.5 Programación evolutiva

La programación evolutiva [43, 44] es una familia de los algoritmos evolutivos cuya característica distintiva es el énfasis que realiza en la relación entre los progenitores y sus descendientes, en lugar de la relación genética como ocurría con los algoritmos genéticos. Para ello usa los operadores de mutación para generar diversidad en las poblaciones de soluciones a la vez que se mantiene un alto grado de correlación entre los progenitores y sus descendientes.

En su variante canónica, la programación evolutiva hace uso del clásico flujo de inicialización, variación, evaluación y selección. La inicialización de la población puede ser aleatoria o siguiendo alguna heurística. Cada individuo de la población se convierte en progenitor, generando λ descendientes cada uno de ellos mediante el operador de mutación. El mecanismo de selección de los que serán los progenitores en la siguiente generación puede ser elitista, mediante el cual las mejores μ soluciones se mantienen en la población como progenitores, una selección mediante torneo, donde las soluciones enfrentan sus valores de aptitud, o una selección proporcional a los valores de aptitud similar al mecanismo de los algoritmos genéticos. Como ya se

ha comentado, la programación evolutiva no efectúa recombinación a los individuos, ya que el enfoque de optimización del algoritmo es global y rechaza la hipótesis de los bloques constructivos [68] en la que se asume que la aptitud es una función separable de las partes que forman el genoma.

2.2.6 Programación genética

La principal característica de la programación genética [81] es que los individuos que se someten a evolución son programas ejecutables. Ya que estos programas pueden tener diferente tamaño, la representación de los individuos se realiza con estructura de tamaño variable, principalmente árboles y listas. Estas estructuras están formadas por funciones básicas y operaciones simples, además de valores constantes o variables. Además, las operaciones y funciones pueden tener una aridad determinada, es decir, requieren de un número determinado de parámetros, por lo que los mecanismos de inicialización y los operadores de variación tienen que lidiar con el problema de generar individuos que no sean válidos.

La inicialización de los individuos se realiza seleccionando aleatoriamente las operaciones y funciones de un conjunto y encadenándolas. La evaluación de los individuos es tan sencilla como ejecutar el programa que codifica y analizar el resultado del mismo para determinar cómo de bien lo ha hecho. Por su parte, el mecanismo de selección tiene en cuenta el valor de aptitud de los individuos de la población. El operador de recombinación intercambia sub-árboles aleatorios de dos padres, mientras que el de mutación introduce un nuevo sub-árbol en un nodo aleatorio del objetivo.

2.2.7 Optimización multiobjetivo

Todo lo referenciado hasta ahora sobre computación evolutiva partía de la premisa de que únicamente había un objetivo a optimizar, cuantificado por la función de aptitud. Sin embargo, la realidad está repleta de problemas multiobjetivo, que son aquellos donde la calidad de una determinada solución está definida por su rendimiento con respecto a varios objetivos distintos, los cuales pueden ser complementarios entre ellos.

Una posible solución para enfrentarse a estos problemas es asignar un valor numérico de calidad de la solución para cada objetivo y agruparlos en un único valor mediante una suma ponderada, lo que se conoce como escalarización [22]. La otra opción es tratar conjuntamente todos los valores de calidad de la solución para cada uno de los objetivos y establecer un orden de calidad entre ellas, para lo cual es necesario definir el concepto de *dominancia*. Dadas dos soluciones, se dice que una de ellas *domina* a la otra si sus valores de calidad de

cada objetivo son al menos tan altos como los de la segunda, y es estrictamente superior en al menos un caso. De la misma forma, una solución *no dominada* es aquella que, como su propio nombre indica, no es dominada por ninguna otra solución. El conjunto de todas las soluciones *no dominadas*, el cual se denomina como *conjunto o frente de Pareto*, tiene la propiedad de que no es posible mejorar la calidad de una solución con respecto a algún objetivo sin que se reduzca en otros. Por tanto, el objetivo final de los algoritmos evolutivos multi-objetivo, como el *NSGA-II* [30] o el *SPEA-2* [167] por citar algunos, es acabar con un conjunto de soluciones que se aproxima o coincide con el frente de Pareto del problema.

2.2.8 Otras técnicas

Además de las técnicas reseñadas anteriormente, existen otras técnicas de evolución a las cuales se le añade algún tipo de influencia externa para mejorar el proceso evolutivo. Además, el lector puede encontrar en la literatura multitud de algoritmos bio-inspirados más allá de los evolutivos, como la optimización basada en colonias de hormigas [33] o abejas [150], por citar algunos. A continuación se reseñan algunas técnicas del primer grupo, es decir, aquellas que están relacionadas directamente con los algoritmos evolutivos.

En los algoritmos coevolutivos [124], la influencia externa a la que se hacía referencia anteriormente proviene de una población adicional, cuyos miembros afectan al valor de aptitud de la población que se está evolucionando, la cual, a su vez, influye en la aptitud de la primera. Se puede decir, por tanto, que ambas poblaciones evolucionan al mismo tiempo. Dependiendo de la forma de influencia entre las poblaciones se distinguen dos familias de algoritmos coevolutivos: los cooperativos, en los cuales varias poblaciones distintas que representan partes de un problema cooperan entre sí para alcanzar una solución global, y los competitivos, en los que, como su propio nombre indica, los individuos compiten entre sí para incrementar su valor de aptitud a costa del contrincante.

Por su parte, en los algoritmos evolutivos interactivos [149] es el usuario el que entra a formar parte del mismo, guiando el proceso evolutivo. La interacción ocurre principalmente durante la fase de selección, en la cual el usuario puede seleccionar directamente los individuos que pasan a la fase de reproducción o, de forma más indirecta, asignando valores de aptitud a dichos individuos u ordenándolos en base a algún criterio subjetivo. La introducción del usuario en el proceso evolutivo tiene numerosas ventajas entre las que se incluye la posibilidad de tratar problemas en los que la función de aptitud no está clara, una capacidad de búsqueda mejorada ya que el usuario puede guiar al algoritmo si éste se queda atascado y un incremento en la diversidad de los individuos. Sin embargo también hay que te-

ner en cuenta algunas desventajas como el incremento en el tiempo de ejecución del algoritmo, ya que el usuario es bastante más lento evaluando los individuos, la inconsistencia inherente a los humanos, que cambian de opinión con facilidad y la fatiga que se provoca al usuario si el número de evaluaciones que se le solicita es muy alto.

Como ya se ha comentado en la introducción (capítulo 1), la IC va a tener un gran impacto sobre el desarrollo de videojuegos y por esta razón, la aplicación de estas técnicas en el campo de los videojuegos está fomentando un creciente interés tanto por parte del mundo académico como de la industria, encontrándose este campo en plena ebullición tras pasar los últimos años en un período de profunda transformación. Para estudiar y analizar esta transformación se ha realizado un estudio exhaustivo de la red social formada por aquellos investigadores que han trabajado en la combinación de IC y videojuegos.

El análisis de redes sociales [136] ha adquirido notoriedad recientemente gracias a la incorporación de nuevas herramientas provenientes del campo de las redes complejas [115]. El estudio de todo tipo de redes ha sido un campo de investigación extremadamente activo en los últimos años, motivado por la introducción de modelos para las redes que siguen una ley de potencias [10] así como las de escala libre [160], lo cual, de hecho, ha inducido el estudio de nuevos fenómenos relacionados con este nuevo ámbito.

Este estudio se centra en la red de co-autoría [99], cuyos nodos representan autores de artículos científicos mientras que una arista entre dos nodos representa al menos un artículo escrito en común entre ambos autores (nodos). Esta red se considera social ya que la colaboración en un trabajo de investigación implica que los autores se conozcan a nivel personal y, por tanto, estudiar sus enlaces, estructuras y evolución permite adquirir un conocimiento más profundo sobre los factores que influyen en la colaboración científica. En concreto, el estudio se ha centrado en el análisis de esta red social en tres niveles diferentes (macroscópico, mesoscópico y microscópico) para determinar las características distintivas de la comunidad, las reglas que gobiernan la cooperación científica y la dirección hacia la que se dirige la comunidad de IC en videojuegos, identificando los temas más activos en el área.

En este sentido, este tipo de análisis de red se ha realizado en otros campos de investigación tales como en el de las matemáticas [9, 59], computación evolutiva [25, 27, 154], trabajo colaborativo [70], ciencias sociales [71] y física, bio-medicina y ciencias de la computación [113, 114], por citar algunos.

Redes sociales y complejas

3.1 OBTENCIÓN DE LOS DATOS BIBLIOGRÁFICOS

Origen de los datos

Los datos bibliográficos usados para la elaboración de la red de colaboración han sido obtenidos del servidor del proyecto Digital Bibliography and Library Project (DBLP¹). Esta base de datos está gestionada por la universidad de Trier, y proporciona información bibliográfica de la mayoría de las revistas y actas de congresos relacionadas con la ciencia de la computación. Además, ofrece una interfaz de uso sencillo para obtener resultados filtrados mediante una aplicación. Los resultados se codifican en ficheros XML o JSON, lo que facilita su posterior análisis mediante otras herramientas software.

La herramienta

Se ha programado una herramienta que lanza consultas a la base de datos y va construyendo la red de co-autoría. El proceso para la obtención de datos es el siguiente. El programa lee un conjunto de términos clave de búsqueda de un fichero externo. Para cada una de estas palabras clave, lanza una consulta a la DBLP y procesa los resultados devueltos por la base de datos, es decir, un listado de artículos con la información asociada a ellos, tal como los autores o el año de publicación. Para cada artículo, la herramienta establece como co-autores a aquellos que aparecen como autores, dos a dos. La herramienta evita procesar el mismo artículo más de una vez, pues la base de datos asigna un identificador único a cada uno. Una vez procesados todos los términos de búsqueda, la herramienta genera varios ficheros que representan el grafo de co-autoría. Con respecto al análisis, se ha usado *Cytoscape* [139] además de la biblioteca *igraph* [28] para dibujar y analizar los grafos obtenidos.

Términos de búsqueda

Para obtener un conjunto de datos representativo se han seleccionado términos de búsqueda que obtienen los artículos que han sido publicados en conferencias y revistas con una temática que está directamente relacionadas con la IC en videojuegos (*CIG*, *TCAIG*, *AIIDE*, *GAMEON*, etc.). Además, se han analizado también artículos de otras revistas y congresos de IC en los cuales pueden aparecer artículos relacionados con esta temática (*GECCO*, *PPSN*, *IJCAI*, etc.), filtrados mediante palabras clave relacionadas con los videojuegos (ver Apéndice A.1. Por último, se han empleado algunas palabras clave para obtener aquellos artículos que no hayan sido publicados en las conferencias y revistas descritas previamente y, sin embargo, estén dentro del ámbito de este estudio.

Ventana temporal

El conjunto de datos analizado se recogió a finales de abril de 2012 usando el método que acaba de ser descrito. La fecha de publicación de los artículos obtenidos va desde 1971 hasta 2012, aunque el número de ellos es bastante disperso en los primeros años, motivo por el cual se ha centrado el análisis en el período que va desde 1997 hasta 2011. La ventana temporal, por tanto, abarca 15 años de la investigación reciente sobre IC en videojuegos y, aunque no se pretendió obte-

¹ <http://dblp.org>

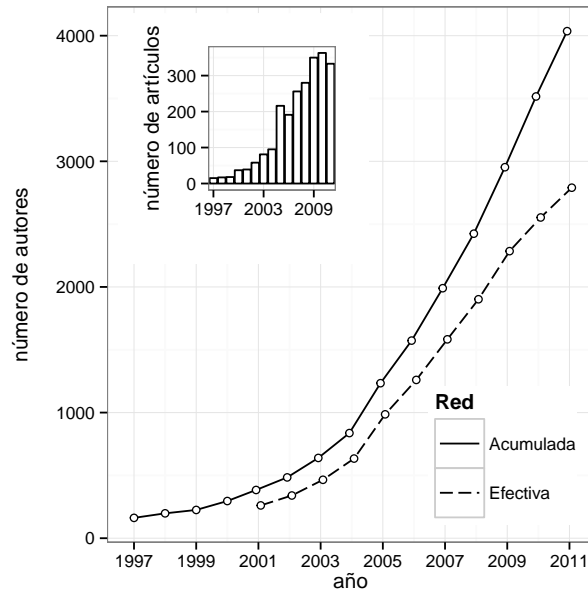


Figura 3.1: Evolución del número total de autores en la red. La gráfica interina muestra el número de artículos publicados cada año.

ner el grafo completo de este campo, se obtuvo un grafo lo bastante representativo como para suponer que el comportamiento del mismo se puede extrapolar al grafo completo. Sobra decir que se han tenido los problemas típicos con respecto a distinguir autores con el mismo apellido, nombres propios e intermedios, o nombres con caracteres especiales. Aunque en la mayoría de los casos estos problemas son inevitables, es cierto que también son muy infrecuentes, por lo que se ha supuesto que tuvieron un impacto insignificante en los resultados obtenidos.

3.2 ANÁLISIS DE LA RED DE COLABORACIÓN

Usando el conjunto de datos obtenidos de la DBLP es posible construir la red de colaboración como un grafo no dirigido con pesos $G(V, E, W)$ en el cual cada vértice $v \in V$ representa un autor y cada arista $(v, w) \in E \subseteq V \times V$ indica que los autores v y w comparten la autoría (posiblemente con otros autores) de, al menos, un artículo. Precisamente, el número de artículos compartidos entre dos autores se representa con el peso w de las aristas, en concreto, W es una función $W : E \rightarrow \mathbb{N}^+$ tal que $W((v, w))$ es el número de artículos que los autores v y w tienen en común. Ya que el estudio tenía como interés la evolución temporal de la red, se han definido una serie de grafos con peso G_t , cada uno de ellos agrupando la información de co-autoría obtenida hasta el año t (incluido).

Anteriormente se ha definido la ventana temporal de los datos analizados, por lo que $1997 \leq t \leq 2011$ (hay que tener en cuenta que

Representación de la red

Red acumulada y red efectiva

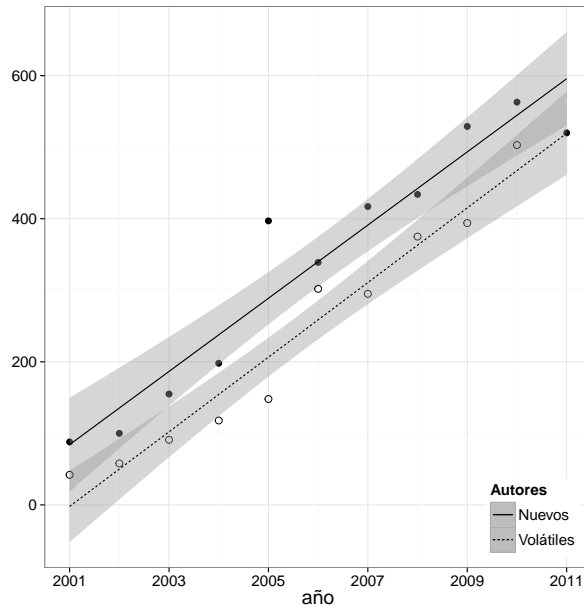


Figura 3.2: Evolución temporal del número de autores nuevos y volátiles (datos anuales).

en el momento de realización del estudio, los datos para el año 2012 estaban incompletos, por lo que se dejaron fuera del mismo). Esta ventana temporal es lo suficientemente amplia para obtener observaciones interesantes en la evolución de la red. Es, además, un período de tiempo suficiente para que algunos autores abandonen el campo de investigación y pasen a estar inactivos. Aunque la información que se obtiene de estos nodos “fósiles” puede ser interesante, también distorsionan en algunos casos el comportamiento a corto plazo de la red. Para evitarlo, se ha considerado una ventana móvil de tiempo de X años y se han construido los grafos G_t^X que agrupan la información sobre los artículos publicados en los años t' tales que $t - X < t' \leq t$. De esta forma, autores y los enlaces entre ellos pueden desaparecer de la red si se vuelven inactivos durante X o más años (en este estudio, concretamente, se ha usado una ventana temporal de $X = 5$ años). A lo largo del capítulo se hará referencia a G_t y G_t^X como las redes acumuladas y efectivas, respectivamente.

A continuación se analizan las propiedades y estructuras de dichas redes.

3.2.1 Propiedades macroscópicas básicas

El comportamiento con respecto a las publicaciones de los autores de un determinado campo de investigación, es decir, la lógica interna de la red de colaboración científica, marcan la tendencia a seguir por las propiedades macroscópicas de la misma. Para empezar, la Figura 3.1 muestra un resumen del tamaño de la red, así como de su

*Crecimiento,
volatilidad y número
de artículos*

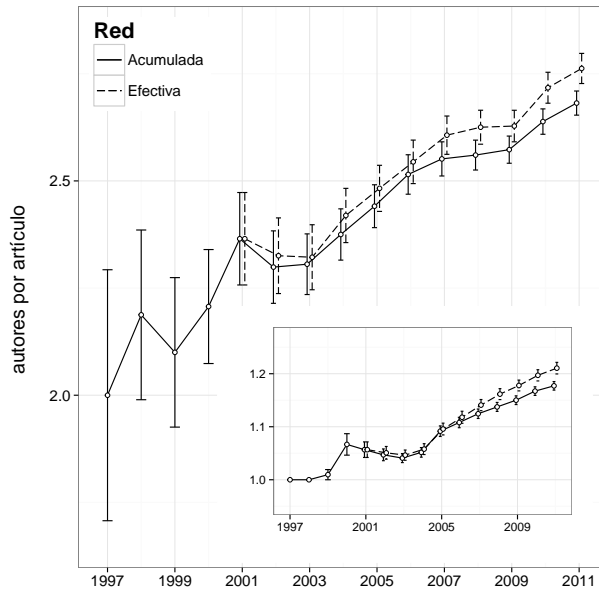


Figura 3.3: Número medio de autores por artículo y artículos por autor (figura interior). Las barras verticales marcan el error estándar de la media.

evolución temporal. La curva que muestra el número de autores en la red muestra una tendencia creciente en los primeros años, lo que denota un crecimiento acelerado en este número hasta mediados de los años 2000, que es cuando el crecimiento se estabiliza, fenómeno que queda reflejado con mayor detalle en la gráfica de la red efectiva. Esto sugiere que el campo de la IC y videojuegos es vibrante y activo, atrayendo nuevos investigadores y produciendo nuevos trabajos. El número de trabajos también muestra una tendencia creciente a partir del 2005, sosteniendo el crecimiento de la red, como se puede observar en la gráfica interior de la Figura 3.1. Por su parte, la trayectoria lineal que sigue el tamaño de la red indica que, además de existir un número constante de nuevos autores que llegan a formar parte de la red año a año también existe otro número de ellos que la abandonan.

Como se observa en la Figura 3.2, la tasa de abandono de la red por parte de los autores volátiles, es decir, aquellos que no publican más allá de un año t , sigue una tendencia similar a los nuevos autores que pasan a formar parte de la comunidad. Aunque ambas magnitudes crecen linealmente, el número de nuevos autores supera al número de autores que abandonan la red. En realidad, el coeficiente lineal que define el crecimiento del número de autores volátiles es ligeramente superior al de nuevos autores, lo que podría indicar una leve desaceleración de la comunidad. Sin embargo, la estimación de la volatilidad es inherentemente pesimista conforme el análisis se va acercando al presente, ya que los autores tienen menos tiempo para volver a publicar. La tendencia real de la volatilidad será, por tanto,

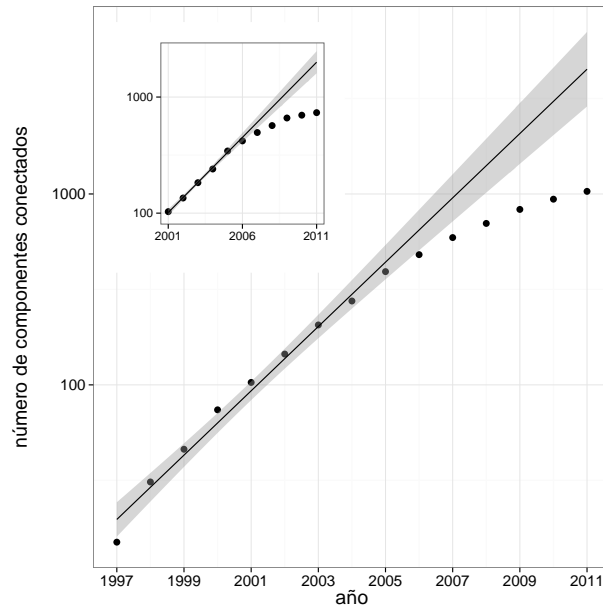


Figura 3.4: Evolución del número de componentes conectadas (escala log) en la red acumulada. La línea sólida muestra el mejor ajuste ($y = \alpha(1 + \beta)^x$, $\beta = 0.47$, $R^2 = 0.98$ con $1997 \leq x \leq 2005$). La figura interna muestra el mismo ajuste para la red efectiva ($\beta = 0.35$, $R^2 = 0.99$ con $2001 \leq x \leq 2005$).

más baja y similar al número de nuevos autores, apoyando la idea de un crecimiento constante de la red.

*Evolución de la
productividad*

Si se atiende a la Figura 3.3, se puede observar otro patrón macroscópico destacable en la evolución temporal de la comunidad: el incremento en el número medio de autores por artículo. Este patrón es consistente con lo observado por [27] en un análisis del campo de la Computación Evolutiva, pero no se puede atribuir a las mismas razones que en ese trabajo se detallan, como son la madurez que la comunidad va adquiriendo y la creciente complejidad de los artículos de investigación. Este razonamiento tiene sentido en el caso de la Computación Evolutiva dada su larga trayectoria temporal y la tendencia prácticamente plana de la productividad de los autores. Sin embargo, el número medio de artículos por autor se ha incrementado aproximadamente un 10% en la red efectiva desde 2005 a un ritmo prácticamente similar al número de autores por artículo en esos mismos años, lo que sugiere que el incremento en la productividad de los autores se debe a que forman parte de los co-autores de muchos artículos. Esta estrategia parece clara si se asume que los autores intentan optimizar su productividad con un esfuerzo constante, al menos como una forma por parte de los autores para seguir el ritmo de la comunidad científica. No cabe duda de que el problema es mucho más complejo dadas las diferentes mecánicas de la producción académica y científica (por ejemplo, el hecho de que el esfuerzo combinado

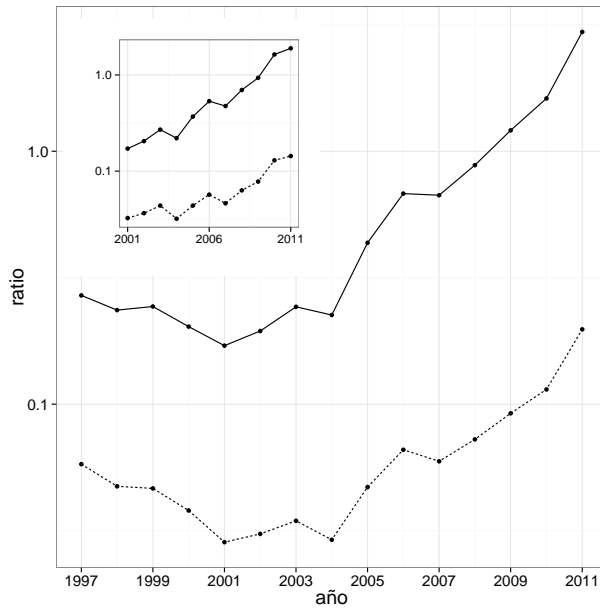


Figura 3.5: Ratio entre el tamaño de la componente más grande de la red acumulada y $N^{2/3}$ (línea sólida). Ratio entre ese mismo tamaño y el de la componente gigante de un grafo aleatorio con el mismo número de nodos y aristas (línea discontinua). La figura interior muestra los mismos datos para la red efectiva.

de dos artículos escritos por los mismos dos co-autores es mayor o menor que el esfuerzo de dos artículos escritos individualmente por ellos está abierto al debate). Independientemente de si el motivo se debe a una estrategia social o a la creciente dificultad en la complejidad de los artículos, este fenómeno requiere de (i) la disponibilidad de una red científica en expansión y (ii) una confluencia real de los temas de investigación que justifiquen el trabajo conjunto. Con estos ingredientes pueden surgir diferentes escenarios que dependerán de las reglas que definan la forma que tienen los autores de agruparse en torno a los artículos que desarrollan. Estas reglas se pueden analizar mediante la observación del efecto global que tienen sobre la red al completo, que es precisamente lo que se ha realizado en este estudio y se detalla a continuación.

3.2.2 Estructura global de la red

Una de las características más distintivas de las redes complejas es la aparición de una *componente gigante*, es decir, un subconjunto conectado de la red que aglutina una fracción no trivial de todos los nodos. Esta componente surge de la unión de múltiples componentes más pequeñas conforme se van añadiendo nuevos enlaces a la red. Está claro que se añaden continuamente nuevos nodos a la red y que pueden enlazarse, o no, a los nodos ya existente, por lo que

Componente gigante de la red

puede ser que se creen nuevas componentes, pero conforme crece la componente gigante, también crece la probabilidad de que los nuevos nodos pasen a formar parte también de esta componente. La Figura 3.4 muestra la evolución del número de componentes conectadas tanto en la red acumulada como en la efectiva. Como era de esperar, este número crece rápidamente, aumentando un 47% y 35% por año en la red acumulada y efectiva, respectivamente, hasta mediados de los 2000. Alrededor de 2005, la tendencia se frena, correspondiéndose con las fases iniciales de la formación de una *componente gigante*. Sin embargo, esta componente no se hace realmente grande hasta hace relativamente poco tiempo (2009 para la red acumulada y 2010 para la efectiva), como se muestra en la Figura 3.5. De hecho, esta *componente gigante* es más pequeña que la homónima de una red aleatoria Erdős-Rényi (ER) [38] con la misma densidad y número de nodos. Además, su tamaño es relativamente modesto (alrededor de un 18.6% de la red acumulada y un 13.5% de la efectiva en 2011) en comparación con otras redes similares como la de programación genética [154], con un 36.5%, o la computación evolutiva [27] con un 62.8%, por citar algunas. Esto sugiere que la red en cuestión está todavía en una fase inicial de desarrollo, formando enlaces y mejorando su cohesión pero sin llegar a estar completamente formada.

Con fines ilustrativos se incluye la Figura 3.6, que muestra la evolución de la componente gigante de la red presentando su estructura en dos momentos concretos: finales de 2006 y 2011.

3.2.3 Mecánicas de crecimiento de la red: vinculación preferente

Vinculación preferente en las redes de colaboración científica

Habiendo mostrado cómo evoluciona la estructura general de la red, este estudio centra su atención a las mecánicas de crecimiento de la misma. Para entender cómo crece la red, hay que empezar observando la distribución $P(k)$ del grado de los nodos, es decir, la fracción de los vecinos del nodo que son vecinos entre sí, que se muestra en la Figura 3.7. La cola de la distribución sigue una ley de potencias $P(k) \sim k^{-\gamma}$ y, por tanto, es una red de escala libre [10], lo cual es bastante común en otras redes de colaboración científica y sugiere que el crecimiento de las mismas está guiado por la vinculación preferente. Al contrario que con las redes Erdős-Rényi (ER) [38] en las cuales dos vértices pueden enlazarse con probabilidad uniforme constante p (lo cual da cabida a una distribución de Poisson del grado de los nodos $P(k) \sim e^{-k}\lambda^k/k!$), en presencia de la vinculación preferente un nuevo vértice se asociará con mayor probabilidad a nodos con un grado elevado (es decir, que están conectados a muchos otros nodos) que a aquellos con un grado bajo. En concreto, sea la probabilidad de que un nuevo vértice se asocie con el vértice v

$$\Pi_v = \frac{k_v^\alpha}{\sum_{w \in V} k_w^\alpha}, \quad (3.1)$$

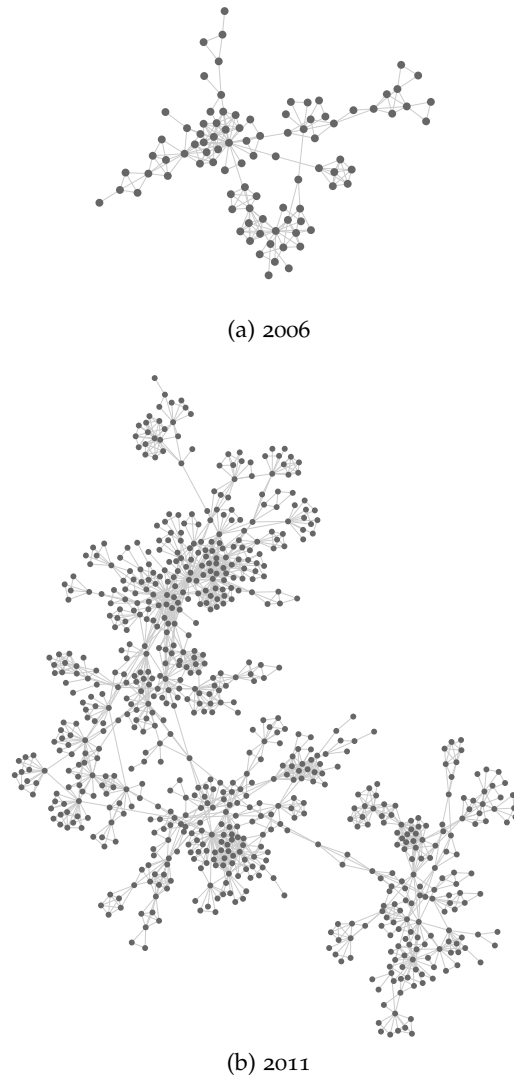


Figura 3.6: Componente gigante de la red de IC y videojuegos en (a) 2006 y (b) 2011. Los nodos representan a los autores y las aristas, al menos una colaboración entre ellos.

donde k_w es el grado de un vértice $w \in V$ y $\alpha \geq 0$ es alguna constante. Para $\alpha = 1$ se da una vinculación preferente lineal que da pie a una red de escala libre. Si además de esto, los nodos entran en la red de forma constante y se asocian a un número determinado m de nodos ya existentes bajo una vinculación preferente, se obtiene un exponente $\gamma = 3$ [9]. En este caso, el exponente que se obtiene es algo mayor ($\gamma = 3.27$). Además, aunque el ajuste a la ley de potencias es bueno ($R^2 = 0.96$), la distribución se adapta mejor a una distribución log-normal, mientras que su cola se ajusta a una pendiente de $\approx -3 \ln k$ en una escala log-log.

Para analizar si existe vinculación preferente en esta red, se ha seguido la metodología sugerida por [112] cuya idea es medir la desviación del ratio de vinculación de los nuevos nodos con respecto a la dis-

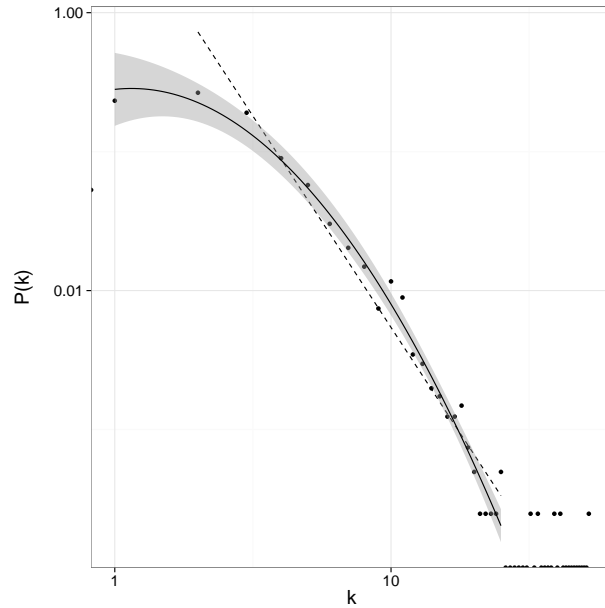


Figura 3.7: Grados de los nodos para la red acumulada G_{2011} (escala log-log). La línea con guiones es un ajuste a la ley de potencias $P(k) \sim k^{-\gamma}$ con $\gamma = 3.27$ y $R^2 = 0.96$ ($2 \leq k \leq 25$). La línea sólida es un ajuste log-normal $P(k) \sim \exp(\alpha \ln k - \beta (\ln k)^2)$ con $\alpha = -10.5908$ y $\beta = -2.9923$ ($R^2 = 0.98$).

tribución normal. En este sentido, si se denota $n_{k'} = |\{v \in V | k_v = k'\}|$ como el número de nodos de la red con exactamente k' vecinos, la probabilidad $P_{k'}$ de que un nuevo nodo v se vincule a un vértice $w \in V$ con grado $d_w = k'$ sería $P_{k'} = n_{k'}/N$ si los enlaces se distribuyen uniformemente. Si la vinculación preferente está presente, los vértices con mayor grado recibirán más enlaces que los indicados por estas expresiones, siendo necesario introducir el término $R_{k'}$ como medida de dicha desviación, dando como resultado la expresión:

$$P_{k'} = R_{k'} \frac{n_{k'}}{N} \quad (3.2)$$

Además, $R_{k'}$ se puede interpretar como la probabilidad relativa de vinculación de un vértice con grado k' . Se puede calcular determinando las probabilidades reales $P_{k'}$ a partir de los datos empíricos y resolviendo la Ecuación 3.2 usando los valores conocidos n_k y N (nótese que todas estas cantidades deben ser entendidas en función del tiempo t). La Figura 3.8 muestra los resultados para la red IC y videojuegos. Dado el tamaño moderado de G_t y el hecho de que puede que no todos los valores altos de grado estén correctamente representados, se han usado datos acumulados ($R'_k = \sum_1^k R_k$) para mejorar las estadísticas. Los datos se ajustan a una ley de potencias con exponente $\alpha + 1 = 1.64$, lo cual indica una tasa sub-lineal de vinculación preferente similar a la encontrada por [154] para la red de programación genética ($\alpha = 0.76$), y por Barabási et al. [9] para

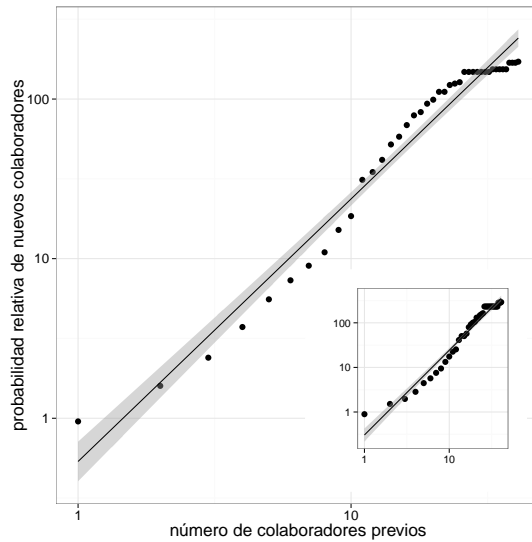


Figura 3.8: Probabilidad relativa acumulada (para 2007-2011) de que los nuevos autores colaboren con los que ya están en la red dependiendo del número de colaboraciones previas que tienen los segundos. Los datos siguen una ley de potencia ($\alpha + 1 = 1.64$ y $R^2 = 0.97$). Figura interna: probabilidad relativa acumulada de nuevas colaboraciones (autores ya existentes colaboran por primera vez entre ellos). Los datos siguen una ley de potencia ($\alpha + 1 = 2.01$ and $R^2 = 0.91$). En ambos casos el número de colaboradores previos se calcula usando la red acumulada.

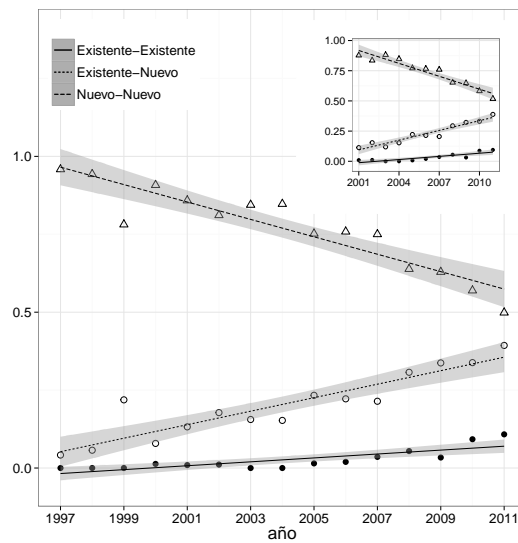


Figura 3.9: Porcentaje de cada tipo de enlace para los datos acumulados. Las líneas sólida, punteada y discontinua muestran un ajuste lineal al número de enlaces entre nodos existentes ($\Delta_x = 0.0036, R^2 = 0.68$), entre un nodo existente y otro nuevo ($\Delta_x = 0.0217, R^2 = 0.83$) y entre nodos nuevos ($\Delta_x = -0.0279, R^2 = 0.85$), respectivamente. Figura interna: los mismos datos para la red efectiva. Los parámetros de los ajustes son $\Delta_x = 0.0087, R^2 = 0.75$, $\Delta_x = 0.0269, R^2 = 0.0268$ y $\Delta_x = -0.0356, R^2 = 0.92$ respectivamente.

las matemáticas ($\alpha = 0.8$) y neuro-ciencia ($\alpha = 0.75$). Una posible hipótesis para este valor bajo es el hecho de que la red se encuentra aún en una fase temprana de su desarrollo en comparación con las otras redes mencionadas, y por tanto se encuentra más descentralizada, de forma que están emergiendo figuras destacadas pero todavía no son responsables de la aglutinación macroscópica de los nuevos autores. Este fenómeno aparece también si se inspecciona el perfil de los nuevos enlaces de la red, mostrado en la Figura 3.9. Nótese como la mayoría de nuevos enlaces se corresponde con los llamados enlaces externos, es decir, enlaces entre nuevos autores que se incorporan a la red. Sin embargo, se puede observar una tendencia a la baja en la proporción de este tipo de enlaces, mientras que tanto los enlaces entre nuevos autores y aquellos que ya están en la red como los denominados enlaces internos (entre autores que ya pertenecen a la red) están tomando importancia.

Enlaces internos de la red

Los enlaces internos tienen una gran influencia sobre la topología y evolución de la red [34]. Como ya se señaló en [9], se ha observado que estos enlaces internos están sujetos a vinculación preferente en la red estudiada, como puede observarse en la gráfica interna de la Figura 3.8. De hecho, esta vinculación preferente es lineal, ya que los datos acumulados siguen una ley de potencias con exponente $\alpha + 1 = 2.01$. De esta forma, los autores consolidados son más propicios a colaborar entre sí por primera vez cuantos más co-autores previos tengan. Sin embargo, esto no lo explica todo, ya que, en principio, los autores que trabajan en distintas sub-áreas no están dispuestos a colaborar entre ellos debido a la diferencia existente entre sus intereses. Además, una vez se establece una colaboración, es interesante analizar su duración o, más concretamente, cómo de productiva es. La siguiente sección arroja cierta luz sobre cómo se seleccionan los colaboradores y la duración de las colaboraciones.

3.2.4 *Patrones de colaboración y clustering*

Como ya se ha mencionado, la selección de co-autores para artículos de investigación es un proceso en el cual el grado de autoridad de los actores, medido por el número de colaboradores previos, tiene una gran influencia, pero no es el único factor, es necesario una coincidencia en el tema a investigar. En algunos casos, esta similitud en los temas se puede inferir indirectamente de la existencia de co-autores en común: si se asume que cada autor se caracteriza por una colección de temas de investigación y una colaboración implica una intersección no vacía con la colección de temas del segundo investigador, cuanto más co-autores tengan en común dos de ellos, mayor probabilidad habrá de encontrar un tema en común de entre todos los tratados por ambos. Por lo tanto, es más probable que dos autores colaboren si ambos han colaborado con la misma gente en el pasado.

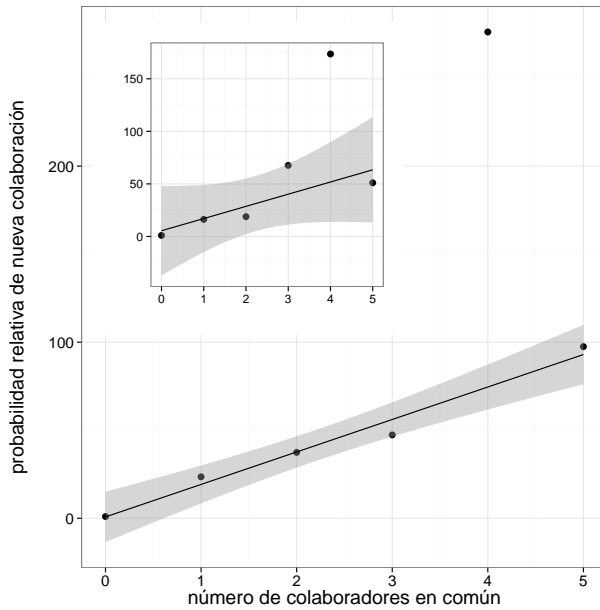


Figura 3.10: Probabilidad relativa de nueva colaboración dependiendo del número de colaboradores en común. La línea sólida es el mejor ajuste lineal ($\Delta_x = 18.44, R^2 = 0.98$). La figura interior muestra los mismos datos para la red efectiva ($\Delta_x = 11.59, R^2 = 0.66$). En ambos casos se excluye el punto $x = 4$, cuya condición de valor atípico se atribuye al pequeño tamaño de la red)

Al igual que en [112], este fenómeno se ha medido de la misma forma que en 3.2.3, es decir, calculando la probabilidad relativa R_m de que dos autores con m colaboradores previos empiecen a colaborar entre ellos. Si se denota n_m como el número de pares de autores no conectados con exactamente m vecinos, se tiene que la probabilidad P_m de un nuevo enlace interno que conecte a esos dos autores es

$$P_m = R_m \frac{2n_m}{N(N-1)} \tag{3.3}$$

Las probabilidades relativas R_m para la red IC y videojuegos se muestran en la Figura 3.10. Como se puede observar, hay un incremento lineal en la probabilidad de colaboración conforme aumenta el número de colaboradores en común. Este patrón de colaboración da pie al fenómeno conocido como *clustering*, que no es otra cosa que el hecho de que es más probable que los vecinos de un determinado nodo sean vecinos entre sí, lo cual queda capturado por el coeficiente de *clustering*. Matemáticamente, el coeficiente de *clustering* local C_i de un nodo i se define como $C_i = 2E_i / [k_i(k_i - 1)]$ donde E_i es el número de aristas que conectan a los vecinos inmediatos con el nodo i , y k_i es el grado del nodo i . Una vez se ha calculado el coeficiente de *clustering* local para cada nodo, se puede promediar para obtener el coeficiente de la red completa. La evolución de este coeficiente se muestra en la Figura 3.11. Se observa que hay una tendencia crecien-

Clustering en redes científicas

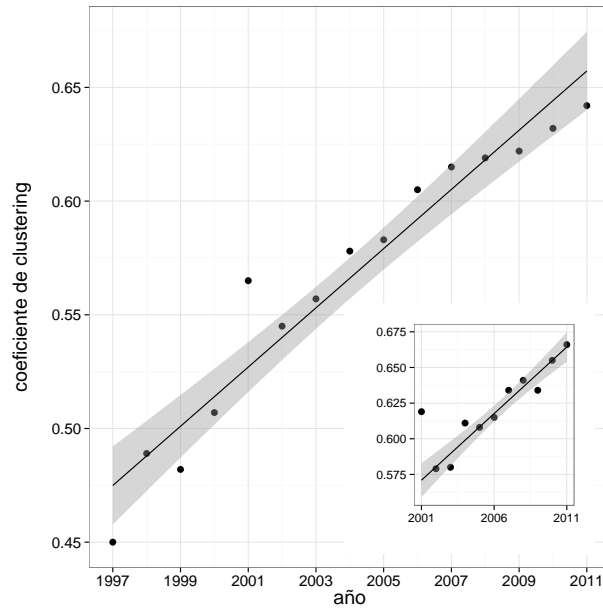


Figura 3.11: Evolución del coeficiente de clustering. La línea sólida es el mejor ajuste lineal ($\Delta_x = 0.013$, $R^2 = 0.93$). La figura interior muestra la evolución de este coeficiente para la red efectiva y su ajuste lineal ($\Delta_x = 0.0093$, $R^2 = 0.94$).

te lineal en el *clustering* de la red. Esto contrasta con las evidencias empíricas halladas en la comunidad científica de Eslovenia [126], pero es coherente con los resultados sobre programación genética de [154]. De la misma forma, [9] han analizado un modelo simple para la evolución de redes de co-autoría en los cuales hay un crecimiento neto constante y la vinculación preferente gobierna la creación de enlaces externos e internos (tal y como se ha observado en la red de IC y videojuegos) y han observado que para una tasa positiva de creación de enlaces internos debería haber un incremento asintótico en el coeficiente de *clustering*.

Duración de las colaboraciones

Una vez se ha analizado como los enlaces internos se crean, es el momento de centrar la atención en la duración de las colaboraciones. Para ello, es útil analizar la distribución de los pesos de las artistas de la red. Es preciso recordar que cada arista de la red tiene un peso indicando el número de colaboraciones entre dos autores cualesquiera. Por tanto, esta distribución de los pesos da una indicación sobre si la colaboración alcanza una determinada productividad. La distribución puede observarse en la Figura 3.12. Nótese como el número de artículos que tienen en común dos autores se distribuye exponencialmente. Esto indica un proceso sin memoria en el cual la probabilidad de obtener un nuevo artículo en común es constante (alrededor del 43 % según los datos empíricos obtenidos). El ratio de caída puede ser atribuido a la dificultad intrínseca de conseguir que un artículo sea aceptado y/o al hecho de que las colaboraciones se disipan tras un

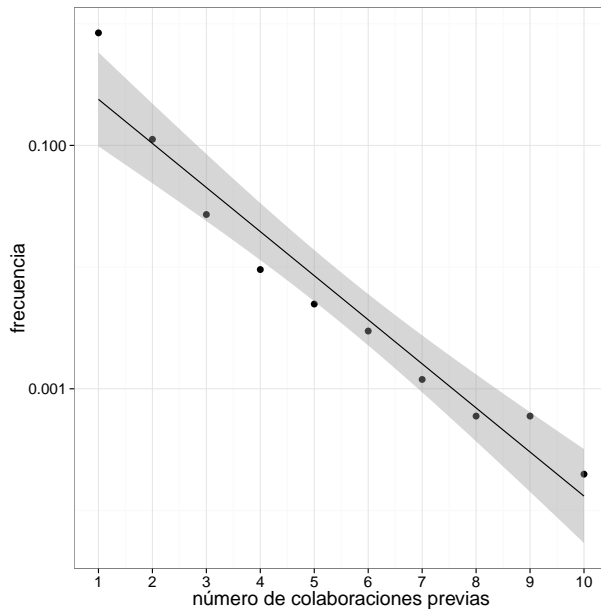


Figura 3.12: Distribución del número de colaboraciones previas. El eje Y está en escala logarítmica. La línea sólida muestra el mejor ajuste exponencial ($P(y) \sim a^x$, $a = 0.4343$, $R^2 = 0.94$).

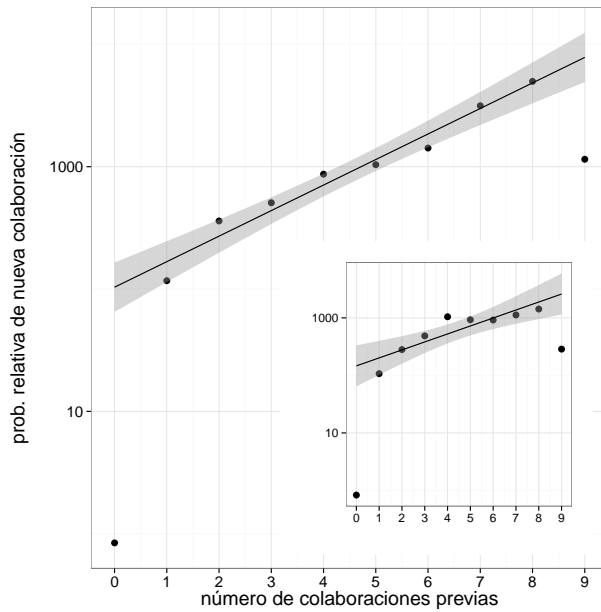


Figura 3.13: Probabilidad relativa de nueva colaboración en base al número de colaboraciones previas entre dos autores en 2011. Eje Y en escala logarítmica. La línea sólida muestra el mejor ajuste exponencial ($a = 1.6318$, $R^2 = 0.97$). La figura interior muestra los mismos datos en grupos de 5 años y su ajuste exponencial ($a = 1.3763$, $R^2 = 0.80$). Los datos se corresponden en ambos casos a la red acumulada y para el cálculo del ajuste se han omitido el primer y último punto de los datos.

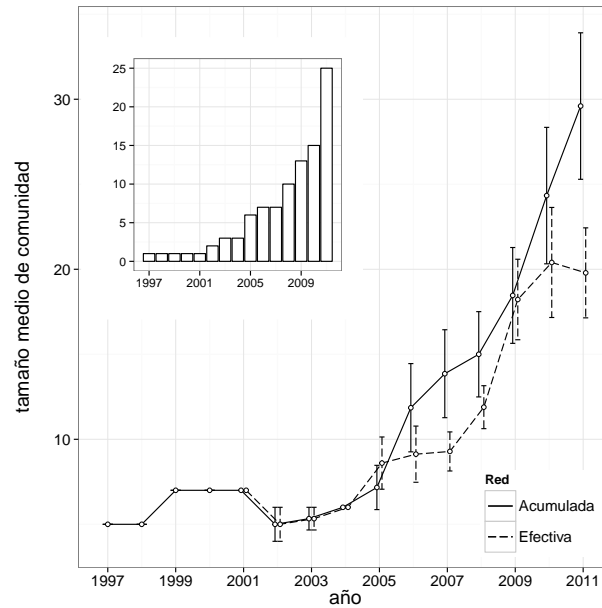


Figura 3.14: Evolución del tamaño medio de comunidad en la componente gigante de la red acumulada. La figura interior muestra el número de comunidades en la componente gigante.

determinado período de tiempo. La primera razón es más importante que la segunda, a razón de los datos mostrados en la Figura 3.13. La figura muestra la probabilidad relativa de colaboraciones adicionales en base al número de colaboraciones previas que ya se han realizado. Excluyendo el primer punto (que se corresponde con una situación donde no existe colaboración previa) y el último punto (que representa un decremento debido al tamaño finito de la red), la probabilidad relativa crece geoméricamente. A modo de comparación, [112] encontró un incremento lineal en esta probabilidad relativa para dos bases de datos bibliográficas correspondiente al campo de la Física y la Medicina. Aunque el factor de crecimiento no es mucho mayor que 1, considerando los últimos cinco años hay una probabilidad un 38% mayor de escribir un artículo con un co-autor con el cual ya se han escrito anteriormente m artículos que con otro con el que se hayan escrito $m - 1$, indicando esto una cierta fidelidad en las colaboraciones científicas, algo que parece ser muy duradero en el área de la IC y videojuegos.

3.2.5 Estructura mesoscópica de la red

La forma de elegir co-autor así como las dinámicas de enlace descritas anteriormente dotan a la red de una particular estructura cuando se la analiza a gran escala. La red de IC y videojuegos no es homogénea, ya que algunos grupos de autores están densamente conectados entre sí, pero vagamente con otros autores fuera del grupo. Estos gru-

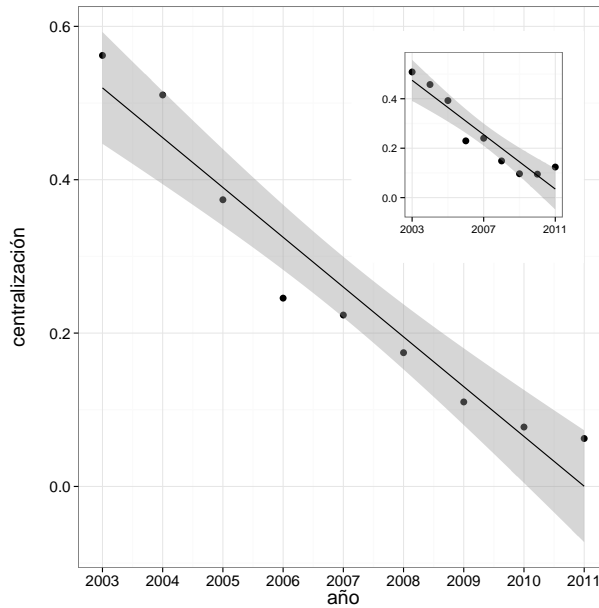


Figura 3.15: Evolución de la centralización por grado en la componente gigante de la red acumulada. Se incluye un ajuste lineal ($\Delta_x = -0.065$, $R^2 = 0.93$). La figura interior muestra estos datos para la red efectiva.

pos son las *comunidades* dentro de la red [56]. Existen varios métodos para identificar la estructura de las comunidades de una red. En este estudio se ha optado por un método basado en la optimización voraz de la modularidad [116], que es una medida de la capacidad de la red para dividirse en módulos, también llamados clústeres. El método consiste en mezclar los nodos en comunidades de forma que se maximice la modularidad de la red. Intuitivamente, la modularidad es un índice de calidad de una partición de la red. Una buena puntuación es aquella en la cual la cohesión interna, medida como el número de conexiones internas dentro de los grupos es alta y radicalmente diferente de la densidad media de la red.

Se ha aplicado el anterior análisis para obtener información relevante sobre cómo las comunidades han evolucionado a lo largo del tiempo en el campo de estudio. La Figura 3.14 muestra la evolución del número de comunidades y sus respectivos tamaños. Además de crecer el número de comunidades a lo largo del tiempo (acentuado a partir de 2003), también lo hace el tamaño de las mismas. Las comunidades aumentan su tamaño por la anexión de nuevos autores y la incorporación de comunidades más pequeñas, aunque también pueden dividirse en comunidades más pequeñas más adelante en el tiempo. Como muestra la Figura 3.15, el incremento en el número de comunidades también se refleja en el ámbito macroscópico con un descenso en la centralización de grado de la *componente gigante* (intuitivamente, la centralización de grado mide el grado de parecido

*Evolución temporal
de las comunidades*

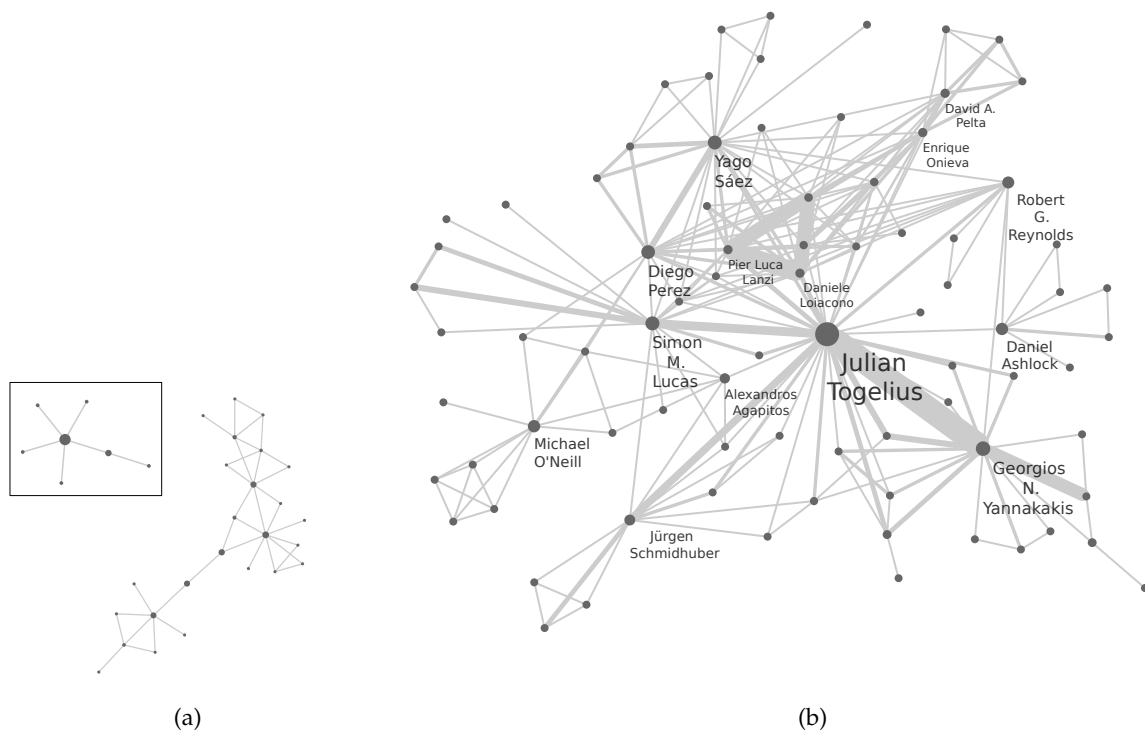


Figura 3.16: (a) Grafo de interacción entre comunidades de la componente gigante hasta 2006 (figura interior) y 2011, usando datos acumulados. Cada nodo representa una comunidad. El tamaño del nodo es proporcional al grado de intermediación del nodo. (b) Comunidad más grande y conectada de la componente gigante hasta 2011 usando datos acumulados. El tamaño del nodo es proporcional a su grado de intermediación mientras que el de la arista se corresponde con el número de colaboraciones entre los autores.

de la red con respecto a la red más central posible, una con forma de estrella con un nodo central y el resto de nodos conectados directamente a éste). Este patrón, que también ha sido observado en la red de la computación evolutiva [27], no debe atribuirse únicamente a la agrupación de diferentes comunidades en la *componente gigante*, sino también a la edad de los nodos (conforme un científico evoluciona de estudiante de doctorado a investigador senior, empieza a desarrollar nuevas conexiones y a tener alumnos bajo su supervisión, dando lugar a nuevos nodos centrales dentro de la red). Un análisis preliminar del lugar de publicación de los artículos elegidos por cada comunidad muestra que hay un solapamiento significativo entre ellos, sugiriendo que la segregación por sitio de publicación no es el factor determinante para la formación de las comunidades.

Una vez se han identificado las comunidades es posible construir el grafo de interacción de las mismas, es decir, un grafo donde todos los autores que pertenecen a la misma comunidad se agrupan

Interacción entre las
comunidades

en un único nodo y las aristas indican que existe colaboración entre autores de dos comunidades. La Figura 3.16a muestra el grafo de interacción de comunidades para la componente gigante en G_{2011} . Dentro de la figura hay otro gráfico que muestra el mismo grafo de interacción para G_{2006} . Se observa claramente como la componente gigante ha crecido no solo en tamaño, sino que también ha desarrollado una estructura compleja interna con multitud de comunidades conectadas entre sí. Nótese como este grafo de interacción tiene, sin embargo, una forma estirada, lo cual no es típico de una red de escala libre. Además, se puede identificar una comunidad que juega un papel importante tanto en el grafo de interacción (es la que tiene más conexiones) como en la red en sí (es la comunidad más grande). Se puede ver dicha comunidad en la Figura 3.16b donde se usa el tamaño del nodo para indicar la centralidad del mismo (tema que se volverá a tratar más adelante) y el tamaño de la arista como la fuerza de la colaboración entre dos autores.

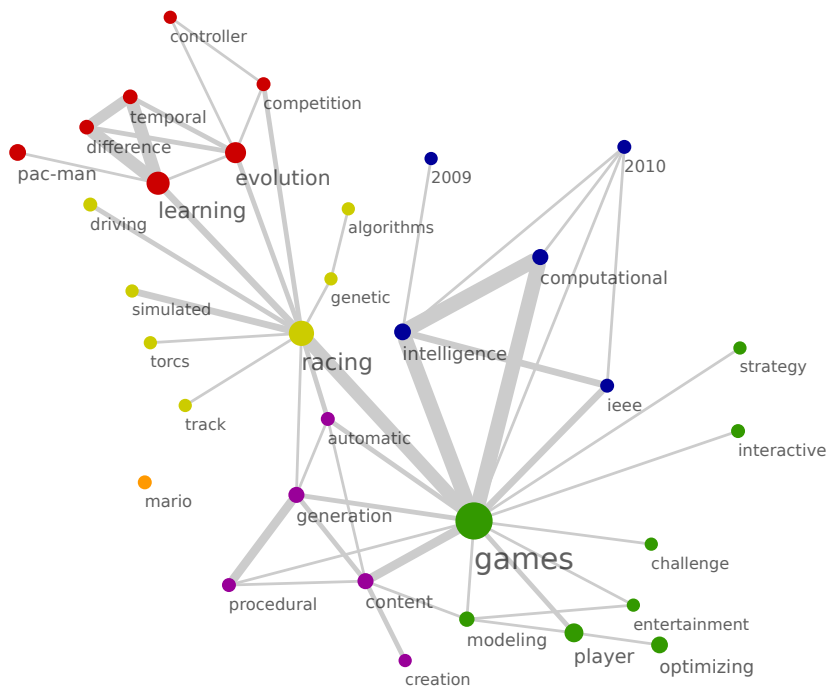


Figura 3.17: Grafo que representa las palabras más frecuentes en el título de los artículos escritos por al menos un miembro de la comunidad más grande y central. Cada nodo representa una palabra y una arista entre dos nodos indica que esas palabras aparecen en el mismo título. El tamaño de los nodos representa la frecuencia de la palabra, mientras que el de las aristas representa la frecuencia de la relación entre ambas palabras. El color de cada nodo indica a que comunidad pertenece dentro del grafo de palabras

Para analizar el ámbito y el área de investigación de esta comuni-

Principales temas de investigación de la comunidad

dad destacable hay que analizar no solo a los autores que la forman, sino también el texto de las publicaciones. Precisando, se ha generado un grafo que representa las palabras usadas en los títulos de los artículos escritos por al menos uno de sus miembros (ver Figura 3.17). En este grafo, cada nodo representa una palabra mientras que una arista entre dos nodos significa que esas dos palabras aparecen en un mismo título de artículo. El tamaño del nodo representa la frecuencia de la palabra mientras que el de las aristas se corresponde con la frecuencia de la relación. Los artículos y las preposiciones se han filtrado ya que no ofrecen información útil. Además, se han incluido únicamente aquellas palabras que representan el 10% superior del ranking de frecuencias. Por la misma razón, las aristas entre palabras que solo aparecen una única vez en el mismo título han sido eliminadas. Una vez se ha obtenido el grafo, se ha lanzado el algoritmo de detección de comunidades usado anteriormente, obteniendo 6 áreas de investigación en las cuales está trabajando la comunidad. Además de un grupo de palabras clave relacionadas con la edición de actas de congresos, hay tres áreas relacionadas directamente con videojuegos: Super Mario (nodo aislado), Pac-Man (en el contexto de aprendizaje evolutivo y por diferencia temporal) y juegos de conducción (TORCS y técnicas como los algoritmos genéticos). Además aparecen dos áreas adicionales, una centrada en la generación automática de contenido y la otra en la interacción y el modelado del jugador. Estas áreas son aquellas en las que está trabajando el núcleo de la comunidad de IC y videojuegos. Para obtener una perspectiva más amplia de las áreas es necesario una caracterización más profunda de quienes son los investigadores centrales de la comunidad, estudio que se describe a continuación.

3.2.6 Análisis de centralidad

Medidas de centralidad

Las medidas de centralidad son indicadores de la importancia de un nodo dentro de la red. Esta importancia puede ser adquirida de diferentes formas dependiendo del contexto y el significado de las conexiones, por lo que se han definido diferentes medidas de centralidad en la literatura. Borgatti and Everett [13] proporcionan una clasificación cruzada de medidas de centralidad a lo largo de dos ejes: medidas radiales ante promediadas, y basadas en volumen comparadas con aquellas basadas en longitud. Para analizar las diferentes perspectivas de centralidad propuestas en esta clasificación se han considerado cuatro medidas de centralidad, una por cada celda de clasificación. La intermediación (*betweenness* en inglés) [15] se ha elegido como medida basada en volumen. Básicamente, los nodos con un valor alto de intermediación son aquellos que aparecen en muchos de los caminos más cortos entre nodos (y, por tanto, son intermediarios en el intercambio de información). Como medida radial basada

en volumen, se ha seleccionado la centralidad de Kleinberg [78], definida como el autovector de $A' \times A$ donde A es la matriz de adyacencia del grafo. Intuitivamente, un nodo se considerará influyente si está conectado a otros nodos influyentes. Con respecto a la dimensión basada en la longitud, se ha seleccionado la proximidad (*closeness* en inglés) [51] como medida radial. Los nodos con una proximidad alta tienen una baja distancia media al resto de nodos de la red (medida como el número de pasos requeridos para alcanzar dichos nodos). Por tanto, estos nodos con alta proximidad pueden ser considerados como emisores privilegiados, ya que la información que se genera en ellos alcanza rápidamente al resto de nodos de la red. Por último, la distancia de fragmentación de Borgatti [13] se ha seleccionado como medida promedio basada en la longitud. Este índice tiene en cuenta cómo afecta a la red el hecho de eliminar un determinado nodo. Los nodos que se definen como muy centrales con esta medida son aquellos que provocan un gran incremento en la distancia media entre nodos si son eliminados de la red, manteniendo, por tanto, la cohesión de la misma.

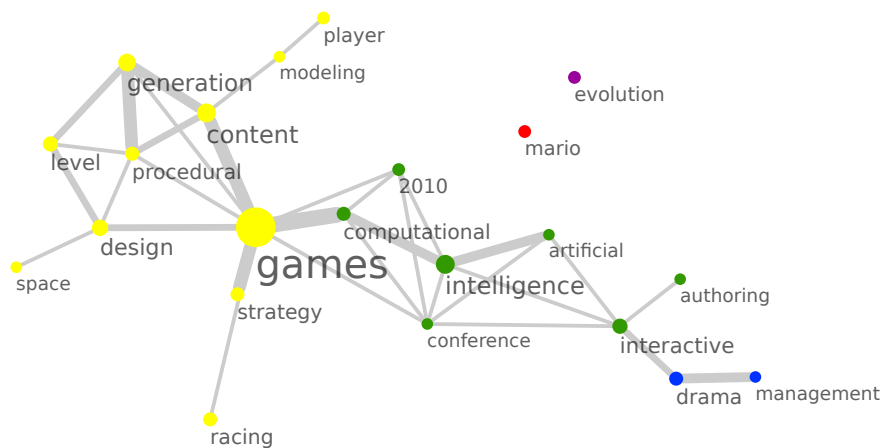


Figura 3.18: Grafo que representa las palabras más frecuentes en el título de los artículos escritos por al menos un miembro de los autores más centrales de la red efectiva de 2011. Cada nodo representa una palabra y una arista entre dos nodos indica que esas palabras aparecen en el mismo título. El tamaño de los nodos representa la frecuencia de la palabra, mientras que el de las aristas representa la frecuencia de la relación entre ambas palabras. El color de cada nodo indica a que comunidad pertenece dentro del grafo de palabras

Se han calculado estas medidas de centralidad para la red efectiva de 2001 para cada nodo, y se han tratado estos datos como un problema de optimización multiobjetivo. El frente de Pareto está formado por cuatro científicos. De la misma forma, se ha realizado un

Análisis de centralidad de la red

análisis de palabras clave similar al realizado en la sección anterior, es decir, se han seleccionado los artículos escritos por alguno de estos cuatro autores y se han analizado sus títulos, generando el grafo de palabras que se puede observar en la Figura 3.18. En esta ocasión aparece una única comunidad que contiene a la generación automática de contenido, los juegos de conducción y el modelado del jugador. Dos nodos aislados revelan la importancia de Super-Mario y evolución. Por último, un tema relacionado con una edición determinada de un congreso agrupa algunos otros temas como gestión del drama, interacción y creación de videojuegos. Este grafo proporciona una visión general sobre algunos de los temas más relevantes que se tratan en la comunidad de IC y videojuegos.

3.3 RESULTADOS Y CONTRIBUCIONES

Resultados generales

En este estudio se ha realizado una aproximación inicial al estudio de las dinámicas que rigen la red de co-autoría de IC y videojuegos, así como su modelado. Se ha observado que el campo en cuestión es vibrante y activo, atrayendo a nuevos investigadores y produciendo nuevos artículos, con un crecimiento estable en el número de autores que se vio acelerado a mediados de los 2000. El número de artículos publicados por año se ha ido incrementando desde el año 2005, facilitando el crecimiento estable de la red. Además de un flujo constante de nuevos autores también se ha observado un flujo constante en sentido inverso, es decir, autores que abandonan el campo, lo cual puede indicar una ligera desaceleración de la comunidad. Sin embargo, la tendencia real de la volatilidad es muy parecida a la de nuevos autores, definiendo pues el crecimiento constante. La red se encuentra aún en plena fase de desarrollo, construyendo enlaces y ganando cohesión, pero no está completamente definida, a tenor del tamaño de la *componente gigante* de la misma.

Crecimiento de la red

El crecimiento de la red está influenciado por lo que se conoce como vinculación preferente, concretamente con una tasa sub-lineal, que se debe principalmente al hecho de que la red se encuentra aún en fase de desarrollo y, por tanto, descentralizada. La vinculación preferente defiende el hecho de que los nuevos autores son más propensos a entrar en este campo colaborando con autores ya establecidos que tienen a su vez un número elevado de co-autores. También se ha observado un grado significativo de fidelidad en las colaboraciones científicas, debido al crecimiento geométrico de la probabilidad relativa de colaboración entre autores que depende del número de colaboraciones comunes previas.

Estructura mesoscópica

Con respecto a la estructura mesoscópica de la red, el incremento en el número de comunidades se ve reflejado en el descenso del grado de centralización de la *componente gigante*. La incorporación a ésta de comunidades más pequeñas sugiere que la red está todavía en fase

temprana de su desarrollo. Este hecho y la heterogeneidad del campo de investigación animan a los investigadores a formar parte de la comunidad.

Parte II

LA GENERACIÓN AUTOMÁTICA DE MAPAS

En los siguientes capítulos, el lector encontrará una definición del algoritmo generador de mapas, cómo se mejora la experiencia de juego haciendo que los mapas generados tengan que cumplir algunas propiedades para que, de esta forma, las partidas que se desarrollen en dichos mapas sean equilibradas y dinámicas. También se estudian propiedades relacionadas con el aspecto de los mapas generados, modificando el mecanismo de evaluación de los mismos para que la estética del contenido generado sea más agradable al usuario.

UN GENERADOR AUTOMÁTICO DE MAPAS

Dentro del campo de la **GAC** existen diferentes perspectivas que son tenidas en cuenta a la hora del proceso de generación. Una de ellas es la generación de contenido basado en búsquedas, que, como su propio nombre indica, basa su funcionamiento en la búsqueda del contenido dentro del espacio de posibles soluciones usando un esquema de generación y prueba. Las técnicas enmarcadas dentro de esta filosofía tienen en común un esquema de funcionamiento en dos fases: una primera en la cual se generan diferentes soluciones candidatas (por ejemplo, diferentes niveles de un determinado juego) y una fase de evaluación y selección, que selecciona de entre todas las soluciones candidatas aquellas que cumplen con los criterios de calidad deseados. Este capítulo trata sobre el diseño de un algoritmo de generación automática de mapas basado en búsquedas para el juego de estrategia en tiempo real *Planet Wars*, descrito ampliamente en la sección 2.1.4. Los mapas generados con este algoritmo cumplen con determinadas propiedades que hacen al juego más interesante y atractivo.

4.1 DESCRIPCIÓN DEL ALGORITMO

Se ha optado por un **AE** como base para el diseño del algoritmo, concretamente una estrategia de evolución (ver Sección 2.2) para seguir un esquema de generación y prueba como mecanismo de creación de los mapas, esto es generar un conjunto de mapas y posteriormente evaluar la aptitud de cada uno de ellos con respecto a los objetivos o propiedades deseadas. Se ha intentado modular el algoritmo todo lo posible, de forma que con un mismo algoritmo es posible generar mapas que intentan satisfacer diferentes propiedades, cambiando únicamente el mecanismo de evaluación de los mismos.

Como ya se dijo en la sección 2.2.2, uno de los componentes que definen cualquier **AE** es cómo se representan las soluciones candidatas. En este caso, las soluciones se corresponden con mapas del juego en cuestión, los cuales pueden definirse como un número n_p de planetas situados en un plano bidimensional. Cada uno de estos planetas, por tanto, posee unas coordenadas (x_i, y_i) que determinan su situación sobre el plano y cuyos valores se encuentran en el dominio de los números reales. Además, cada planeta del juego tiene dos propiedades adicionales: un tamaño que a su vez se corresponde con la tasa de creación de nuevas naves en el planeta (s_i) y el número de naves que ocupan el planeta en cuestión al comienzo de la parti-

Representación de las soluciones

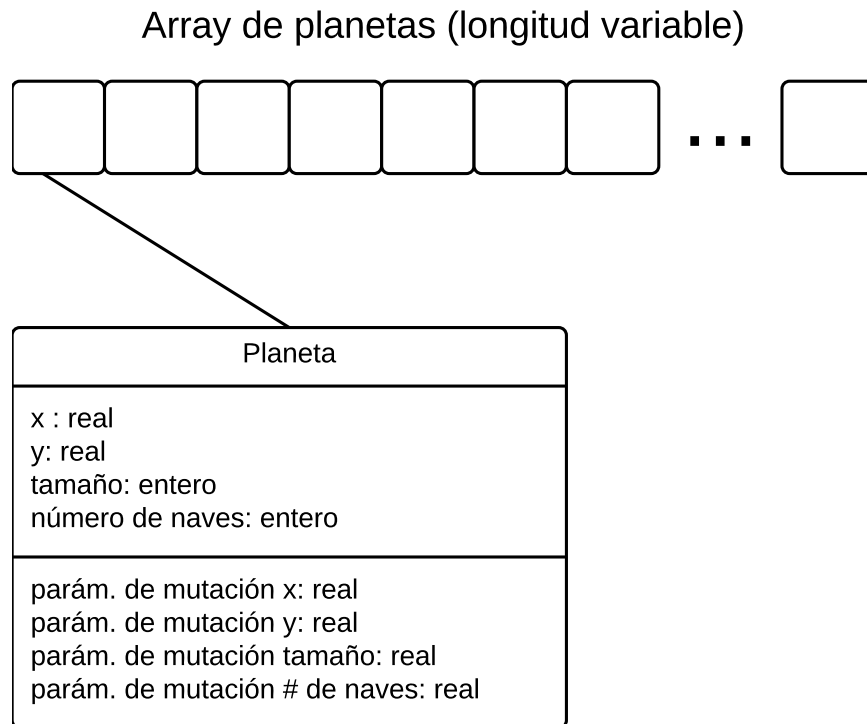


Figura 4.1: Esquema sobre la representación de un mapa de Planet Wars como individuo del algoritmo evolutivo

da, ambos parámetros tienen valores enteros. En principio se podría definir un mapa como una lista de vectores de 4 componentes, uno por cada propiedad del planeta, sin embargo, ya que se ha optado por una estrategia de evolución que contempla la auto-adaptación (ver Sección 2.2.4, ha sido necesario agregar 4 parámetros adicionales a la representación de la solución que actúan como parámetros para el operador de mutación (ver Figura 4.1), el cual se define en la siguiente sección. Resumiendo, una solución del algoritmo es una lista $[\vec{\rho}_1, \vec{\rho}_2, \dots, \vec{\rho}_{n_p}]$ donde cada $\vec{\rho}_i$ es una tupla $\langle x_i, y_i, s_i, w_i \rangle$, con $x_i, y_i \in [0.0, 15.0]$, $s_i \in [1, 5]$ y $w_i \in [1, 100]$, además de 4 parámetros $\sigma_x, \sigma_y, \sigma_s, \sigma_w$ que controlan la mutación de los planetas.

Restricciones de los
mapas

Además de lo anterior, a la hora de generar mapas para *Planet Wars* también hay que tener en cuenta ciertas restricciones que las propiedades de los mapas deben cumplir, y que se definen según las reglas de la *Google AI Competiton 2010*, a saber: el número de planetas que puede haber en un mapa variará entre 15 y 30, las posiciones de los mismos estarán comprendidas entre las coordenadas 0.0 y 15.0 mientras que la tasa de crecimiento y el número de naves podrá tomar valores con un mínimo de 1 y un máximo de 5 y 100, respectivamente. Durante el proceso de evolución de los mapas puede ocurrir que se generen mapas que no cumplan estas restricciones. Una posible solución a estas soluciones no válidas es asignarles el mínimo valor posible de aptitud, de manera que no formen parte de la siguiente ge-

neración de soluciones. Otra opción, que es la que se ha utilizado en este caso, es la de reparar la solución para que vuelva a ser válida. En el caso del número de planetas de un mapa, valor determinado por el operador de cruce que se define en la siguiente sección, se seleccionan los puntos de corte de manera que los dos nuevos mapas tengan un número de planetas en el rango válido. Las coordenadas, tasa de crecimiento y número de naves de cada planeta varían conforme a la mutación aplicada, por lo que se ha controlado dicha mutación para no aplicarse en el caso de que el valor ya mutado se salga del rango de validez.

Algoritmo 2: Estrategia de evolución para la generación automática de mapas

Entrada: $\mu, \lambda, N_{iter}, p_{cruce}, p_{mut}$

Salida: población final de mapas

```

inicializar(población);
evaluar(población);
para iter  $\leftarrow$  1 hasta  $N_{iter}$  hacer
    descendientes  $\leftarrow$   $\emptyset$ ;
    para j  $\leftarrow$  1 hasta  $\lambda$  hacer
        padres  $\leftarrow$  muestraAleatoria(población, 2);
        hijos  $\leftarrow$  clonar(padres);
        si valorAleatorio  $<$   $p_{cruce}$  entonces
            hijos  $\leftarrow$  cruce(hijos);
        fin si
        si valorAleatorio  $<$   $p_{mut}$  entonces
            hijos  $\leftarrow$  mutación(hijos);
        fin si
        descendientes  $\leftarrow$  descendientes  $\cup$  hijosj;
    fin para
    evaluar(población);
    población  $\leftarrow$  selección(población  $\cup$  descendientes,  $\mu$ );
fin para
devolver población;

```

Para el correcto funcionamiento del generador automático de mapas, que se detalla en el Algoritmo 2, es necesario especificar algunos parámetros, concretamente el tamaño de la población μ , el número de descendientes que se generan en cada iteración λ , el número máximo de iteraciones N_{iter} y las probabilidades de cruce p_{cruce} y mutación p_{mut} . Inicialmente se genera una población aleatoria de μ mapas, los cuales cumplen con los requisitos para que sean considerados como jugables. Antes de empezar con el proceso iterativo, se evalúan las soluciones de la población según la función de aptitud que se haya definido. Como ya se ha dicho anteriormente, esta función se ha

Funcionamiento del algoritmo

ido cambiando a lo largo del desarrollo del trabajo que se presenta para estudiar diferentes propiedades, por lo que el lector podrá encontrar la descripción completa de cada una de las funciones de aptitud en los capítulos y secciones siguientes. El proceso iterativo continúa mientras no se haya alcanzado el número máximo de iteraciones N_{iter} . Cada iteración se corresponde con una generación del AE, en la cual se realizan los procesos de selección de padres, cruce y mutación de los mismos (si se da el caso, pues son procesos estocásticos) para obtener λ descendientes, re-evaluación de estos y selección de las soluciones supervivientes que formarán parte de la población para la siguiente generación (ver la sección 2.2.2 para más detalles sobre estos procesos). Una vez concluye el proceso iterativo, se obtiene una población de μ mapas los cuales, tras pasar por el proceso evolutivo, tendrán un valor de aptitud elevado, es decir, cumplirán en gran medida con los requisitos capturados por la función de evaluación.

4.2 OPERADORES

En la sección anterior ya se ha definido la forma de representar los mapas dentro de la estrategia de evolución, así como el funcionamiento del algoritmo. En esta sección se definen los operadores de cruce y mutación que se utilizan en el generador automático de mapas.

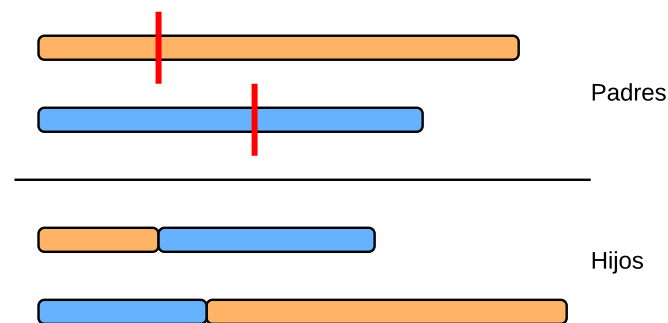


Figura 4.2: Ejemplo del operador de cruce “corte y empalme”. La línea roja indica el punto de corte elegido para cada padre. Nótese que los hijos tienen un tamaño diferente al de sus padres.

Operador de cruce

El operador de cruce que se ha utilizado en el algoritmo es el denominado como “corte y empalme” (del inglés *cut and splice*), que recombina dos individuos intercambiando piezas cortadas de diferente tamaño. De esta forma, los nuevos mapas generados con este tipo de recombinación tienen un número de planetas diferentes, razón por la que se ha utilizado este operador ya que la representación de los mapas son vectores de planetas. En el caso de que el número de planetas del mapa resultante tras el cruce exceda el máximo (30 planetas), se trunca la tupla de planetas para que el mapa cumpla con dicha restricción. En el otro sentido, a la hora de seleccionar el punto de corte, el operador tiene en cuenta el número mínimo de planetas de forma

que los mapas resultantes tampoco incumplen la restricción de contener, al menos, 15 planetas. Se puede observar una representación esquematizada de este operador de cruce en 4.2.

Debido a la heterogeneidad de los parámetros que definen los planetas (valores enteros y reales) se ha optado por un operador de mutación híbrido que utiliza diferentes métodos según el tipo del parámetro. Las coordenadas del planeta, que son valores reales, junto con sus correspondientes tamaños de perturbación necesarios para la auto-adaptación del algoritmo $\langle x_1, x_2, \sigma_1, \sigma_2 \rangle$ se mutan de la siguiente manera:

$$\begin{aligned}\sigma'_i &= \sigma_i \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)} \\ x'_i &= x_i + \sigma_i \cdot N_i(0,1)\end{aligned}$$

donde $\tau' \propto 1/\sqrt{2n}$, $\tau \propto 1/\sqrt{2\sqrt{n}}$, y $N(0,1)$, $N_i(0,1)$ son valores aleatorios que siguen una distribución normal de media 0 y varianza 1. Se aplica una regla adicional a los tamaños de perturbación para evitar desviaciones estándar cercanas a cero: $\sigma'_i < \epsilon_0 \Rightarrow \sigma'_i = \epsilon_0$. El valor de σ_0 utilizado para este algoritmo se corresponde con un 1% del rango del parámetro.

La mutación Gaussiana utilizada para los parámetros reales presenta un problema al aplicarla a parámetros enteros: la perturbación, que es un valor real, tiene que ser redondeada a un valor entero para poder aplicarla al parámetro en sí. Para evitar esto, se ha optado por un método de mutación de enteros que genera perturbaciones adecuadas para este tipo de parámetros [94, 132]. El método es muy parecido a la mutación auto-adaptativa para parámetros reales definida anteriormente, con la diferencia de que en este caso se usa la diferencia entre dos variables aleatorias geoméricamente distribuidas para generar las perturbaciones en lugar de usar variables aleatorias normalmente distribuidas como en el caso de parámetros reales. Sean $\langle z_1, \dots, z_m, \sigma_1, \dots, \sigma_m \rangle$ los parámetros enteros junto con sus tamaños de perturbación σ , el mecanismo de mutación para enteros se define como sigue:

$$\begin{aligned}\sigma'_i &= \text{máx}(1, \sigma_i \cdot e^{\tau \cdot N(0,1) + \tau' \cdot N(0,1)}) \\ \psi_i &= 1 - (\sigma'_i/m) \left(1 + \sqrt{1 + \left(\frac{\sigma'_i}{m} \right)^2} \right)^{-1} \\ z'_i &= z_i + \left\lfloor \frac{\ln(1 - U(0,1))}{\ln(1 - \psi_i)} \right\rfloor - \left\lfloor \frac{\ln(1 - U(0,1))}{\ln(1 - \psi_i)} \right\rfloor\end{aligned}$$

donde $\tau = 1/\sqrt{2m}$, $\tau' = 1/\sqrt{2\sqrt{m}}$, y $N(0,1)$ y $U(0,1)$ son valores aleatorios que siguen una distribución normal (con media 0 y varianza 1) y uniforme entre 0 y 1, respectivamente. Se puede observar que la principal diferencia con el otro método es la distribución usada para generar la perturbación.

*Operador de
mutación*

Los dos siguientes capítulos presentan una metodología para la generación automática de mapas con el objetivo de incrementar la satisfacción del jugador ya sea mediante la imposición de requisitos como el equilibrio o dinamismo al contenido generado (Capítulo 5) así como mejorando el aspecto y la estética de dicho contenido (Capítulo 6). Ambos enfoques hacen uso del algoritmo definido en el presente capítulo.

MEJORA DE LA EXPERIENCIA DE JUEGO

Como se ha comprobado anteriormente (ver Capítulo 2), existen multitud de casos de éxito del uso de la GAC para juegos, abarcando una gran variedad de tipos de contenido como mapas, niveles y música. Sin embargo, el contenido generado se limita a cumplir unas restricciones básicas de jugabilidad, esto es, que sea apto para el juego objetivo. Este capítulo trata de mejorar la experiencia de juego haciendo que los mapas creados con el generador automático cumplan unas determinadas restricciones de forma que las partidas jugadas en ellos exhiban propiedades deseables tales como el dinamismo de las mismas o el equilibrio entre los jugadores. Así mismo, profundiza en el estudio sobre la relación existente entre ambas propiedades, planteando la generación automática de mapas balanceados y dinámicos bajo una perspectiva de optimización multiobjetivo.

5.1 EQUILIBRIO

Una de las características deseables en los juegos multijugador es el equilibrio entre los jugadores y su nivel de juego, ya que, de lo contrario, el jugador de menor nivel puede sentirse frustrado al ser incapaz de vencer al de mayor nivel, mientras que éste puede perder rápidamente interés en el juego debido a lo fácil que le resulta ganar. Este equilibrio entre jugadores puede alcanzarse a diferentes niveles como, por ejemplo, en la fase de emparejamiento en un juego multijugador se intenta seleccionar a dos jugadores con puntuación similar. En este caso, se va a fomentar el equilibrio entre jugadores directamente en el diseño del mapa en el cual va a transcurrir la partida, generando mapas que no ofrecen ninguna ventaja a algún jugador sobre los demás.

Un primer enfoque para fomentar partidas equilibradas en mapas generados de forma automática fue el usado en [85], donde la aptitud de los mapas generados se midió usando un sistema de torneo con jugadores artificiales que medían el nivel de equilibrio que había existido entre estos jugadores a lo largo de la partida. En concreto, para evaluar la calidad de cada individuo, el algoritmo llevaba a cabo un torneo que se disputaba en el mapa en cuestión. Una vez terminado el torneo, el algoritmo recopilaba estadísticas individuales de cada partida para computar el valor de aptitud total del mapa, según la siguiente ecuación:

Midiendo el equilibrio, un primer enfoque

$$\bar{P}_i = \frac{\sum_{j=1}^{t_i} |P_{ij}^{(1)} - P_{ij}^{(2)}|}{t_i} \quad (5.1)$$

$$\bar{S}_i = \frac{\sum_{j=1}^{t_i} |S_{ij}^{(1)} - S_{ij}^{(2)}|}{t_i} \quad (5.2)$$

$$\text{aptitud} = \left(\frac{1}{N_m} \sum_{i=1}^{N_m} \frac{K_i \cdot t_i}{\bar{P}_i + \bar{S}_i + 1} \right)^2 \quad (5.3)$$

donde N_m representa el número de partidas jugadas en el torneo, t_i es el número de turnos jugados en la partida i , K_i es el porcentaje acumulado de planetas ocupados al final de la partida, $P_{ij}^{(1)}$, $P_{ij}^{(2)}$ son los porcentajes de planetas conquistados por el jugador 1 y 2, respectivamente, en la partida i y turno j y $S_{ij}^{(1)}$, $S_{ij}^{(2)}$ son los porcentajes del total de naves que pertenecen al jugador 1 y 2, respectivamente, en la partida i y turno j . Como se puede observar, las componentes \bar{P}_i y \bar{S}_i asignan puntuaciones mas altas cuando los jugadores tienen un número similar de planetas conquistados y naves controladas (el hecho de sumar 1 a esta cantidad es para evitar la división por cero), mientras que t_i promueve que las partidas sean largas (es decir, se disputen a lo largo de un número elevado de turnos), ya que esto significa que no ha habido un jugador ganador superior al otro o que éste se ha decidido casi al final del juego. Por último, con la componente K_i se bonifica aquellos mapas en los que ha habido una alta actividad, concretamente un número elevado de planetas conquistados.

Experimentación

Se llevaron a cabo dos experimentos, uno donde el número de planetas en cada mapa se iba auto-adaptando para mejorar la aptitud de los mapas mientras que en el otro se fijó el número de planetas de cada mapa en 23, ya que según las reglas del juego, este número varía entre 15 y 30. Tanto en uno como en el otro, el algoritmo empleado fue una estrategia de evolución con los parámetros que se describen en la siguiente tabla:

| | |
|-----------------------------|---|
| Representación | Vector de planetas |
| Recombinación | Corte y empalme |
| Mutación | Perturbación gaussiana (reales) y diferencia geométrica (enteros) |
| Selección de progenitores | Torneo binario |
| Selección de supervivientes | $(\mu + \lambda)$ con $\mu = 10$ y $\lambda = 100$ |

Tabla 5.1: Parámetros de los algoritmos para generar mapas equilibrados

El sistema de torneos

El sistema de torneos usado para evaluar cada mapa es una herramienta software, se diseñó explícitamente para este trabajo, la cual

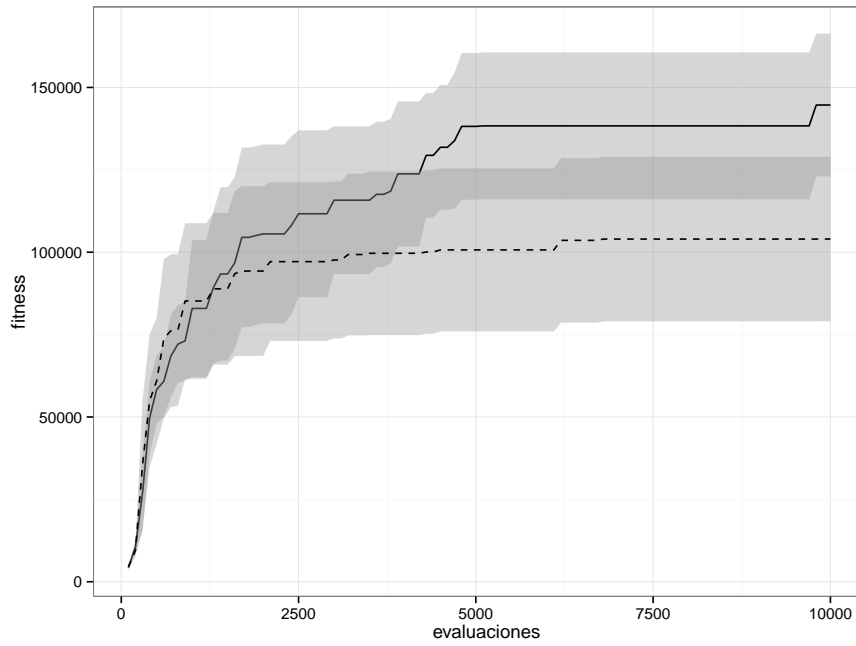


Figura 5.1: Evolución del valor de aptitud medio para la versión auto-adaptativa (línea sólida) y con número fijo de planetas.

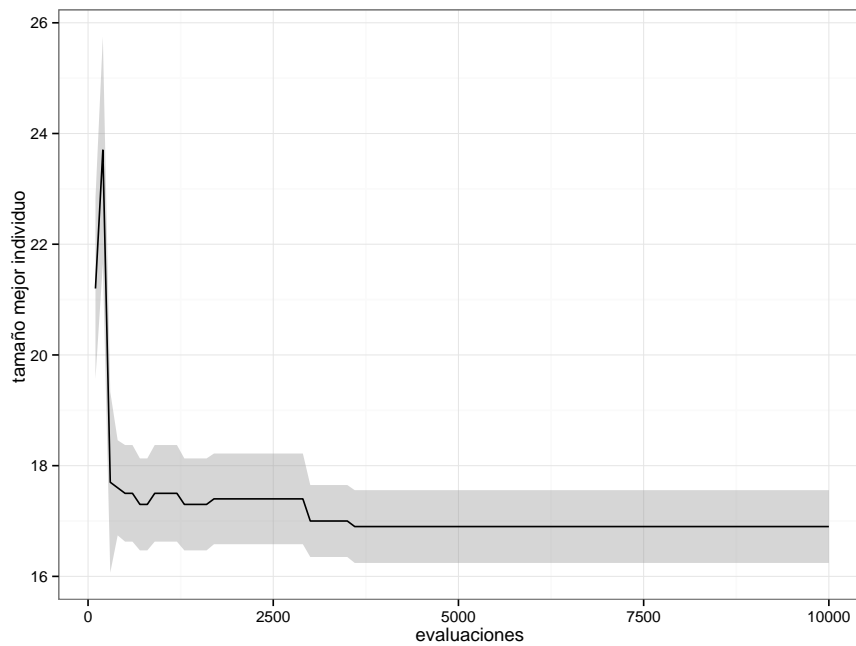


Figura 5.2: Evolución del número de planetas del mapa más equilibrado. En ambos casos, el área sombreada indica el error estándar de la media.

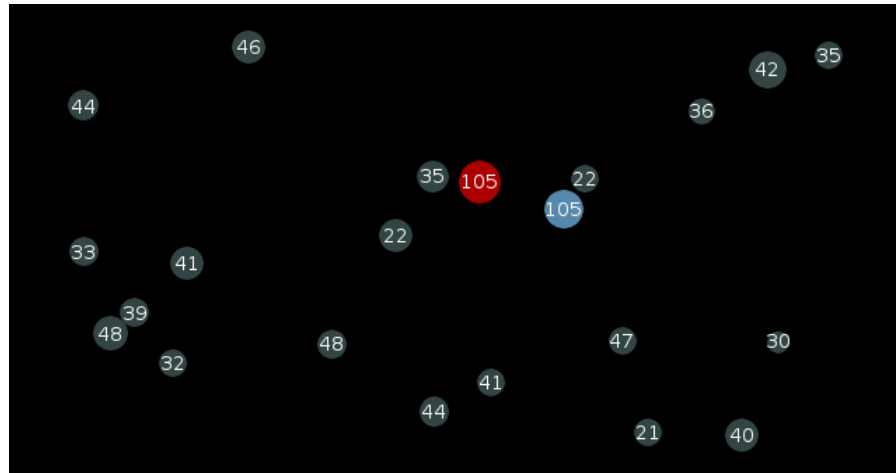


Figura 5.3: Un mapa equilibrado creado de forma automática por el generador. Los números sobre los planetas indican el número inicial de naves mientras que el color define si el planeta es neutral al inicio (gris) o pertenece a alguno de los dos jugadores (rojo y azul).

ejecuta un conjunto de partidas de *Planet Wars* entre un número arbitrario de oponentes, los cuales juegan al menos una vez contra cada uno de los demás jugadores. Básicamente, la herramienta genera los emparejamientos entre los jugadores artificiales atendiendo a los parámetros de configuración que se le proporcionan, tales como el número de partidas que se jugarán entre cada par de jugadores, el tiempo máximo que tiene cada jugador para calcular su movimiento y la localización al mapa donde se disputarán las partidas. Además, la herramienta analiza los archivos históricos de cada partida para obtener las estadísticas necesarias para el cálculo de la función de aptitud. Con respecto a los oponentes artificiales se optó por utilizar tres bots participantes del *Google AI Challenge 2010* que hubiesen quedado entre las cien mejores posiciones de los más de 4600 participantes en total (*Manwe56*, en la posición 31, *fglider* en la 61 y *FlagCapper* en la 91) y que estuviesen disponibles para descargar.

Primeros resultados

La Figura 5.1 muestra la evolución del valor de aptitud medio para ambos experimentos. El área sombreada muestra el error estándar de la media. Como se puede observar, ambos experimentos han tenido un comportamiento similar en las primeras evaluaciones, aunque el algoritmo auto-adaptativo obtiene un mejor valor de aptitud para el resto de evaluaciones. En la Figura 5.2 se puede observar la evolución del número medio de planetas en el mejor mapa (es decir, el individuo con el valor de aptitud más alto), que converge a 17 tras algunas evaluaciones. Esto hace pensar que los mapas con 17 planetas son más equilibrados que otros mapas con un número de planetas más elevado. La Figura 5.3 muestra un ejemplo de un mapa generado automáticamente que ha sido el que ha obtenido mayor valor de aptitud

en su experimento, y por tanto se considera el más equilibrado de la población.

Este primer enfoque descrito anteriormente define su función de aptitud como una fórmula matemática que agrupa varios estadísticos en un único valor. A pesar de ser una práctica frecuente también hay que tener en cuenta que hay un cierto grado de arbitrariedad en la forma en la que se combinan los valores y cómo se promedian para obtener un único valor. De la misma forma, aunque la función de agregación se especifique según el conocimiento de los expertos, es preferible expresar este conocimiento al más alto nivel, haciendo más amigable su interpretación así como la optimización y el ajuste de los parámetros. Es por ello que en un trabajo posterior sobre este mismo tema [89] se optó por usar un conjunto de reglas difusas que capturasen el conocimiento de los expertos sobre qué determina el equilibrio de una partida cualquiera (ver Figura 5.4). Además, se reformularon las medidas anteriores y se definieron unas nuevas para capturar con mayor precisión el nivel de equilibrio de las mismas.

Función agregada y lógica difusa

Al igual que las medidas de equilibrio definidas anteriormente, éstas se obtienen para cada i -ésima partida de las N_g totales, con el objetivo de medir cómo de equilibrada ha sido dicha partida:

Nuevas medidas de equilibrio

- Desequilibrio territorial: Sea τ_i el número de turnos jugados en la partida actual, y sea $\pi_{ij}^{(a)}$ el porcentaje de planetas (sobre el total de n_p planetas) controlados por el jugador a (donde $a \in \{1, 2\}$), en el j -ésimo turno de esta i -ésima partida. Entonces, se define

$$\Pi_i = \frac{1}{\tau_i} \sum_{j=1}^{\tau_i} \left| \pi_{ij}^{(1)} - \pi_{ij}^{(2)} \right| \quad (5.4)$$

es decir, el desequilibrio medio en el número de planetas conquistados a lo largo del juego. Esta variable puede tomar valores desde 0 hasta 1.

- Desequilibrio en el crecimiento: los planetas tienen diferentes tamaños y por tanto producen nuevas naves con diferente frecuencia. Debido a esto, se puede definir el desequilibrio en la incapacidad de producir nuevas naves de forma análoga a 5.4, es decir:

$$\Gamma_i = \frac{1}{\tau_i} \sum_{j=1}^{\tau_i} \left| \gamma_{ij}^{(1)} - \gamma_{ij}^{(2)} \right|, \quad (5.5)$$

donde $\gamma_{ij}^{(a)}$ es el porcentaje del total de capacidad de crecimiento del mapa acumulado por el jugador a en el j -ésimo turno de la partida i -ésima. Aunque ese valor está íntimamente relacionado con el desequilibrio territorial, no son lo mismo, ya que éste proporciona una perspectiva diferente sobre si las fuerzas

1. si Π es LO y Γ es LO entonces bal es HI
2. si Π es HI y Γ es LO y Ξ es LO entonces bal es MED
3. si (Π es LO y Γ es HI) o T es LO entonces bal es LO

Figura 5.4: Reglas difusas para el equilibrio (bal). HI, MED y LO representan un nivel alto, medio y bajo, respectivamente.

de los jugadores están equilibradas o no. Al igual que antes, esta variable toma valores entre 0 y 1.

- Desequilibrio en naves: siguiendo un razonamiento similar, es posible medir si la cantidad de naves controladas por los jugadores está equilibrada o no con la siguiente ecuación:

$$\Xi_i = \frac{1}{\tau_i} \sum_{j=1}^{\tau_i} \left| \xi_{ij}^{(1)} - \xi_{ij}^{(2)} \right|, \quad (5.6)$$

donde $\xi_{ij}^{(a)}$ es el porcentaje del total de naves del turno j -ésimo de la partida i -ésima que posee el jugador a . Hay que destacar que un jugador puede conseguir desequilibrio territorial a base de involucrarse en multitud de batallas, perdiendo por tanto una gran cantidad de naves. De esta forma, la variable está relacionada con las anteriores pero no es cualitativamente idéntica. Su valor varía entre 0 y 1.

- Duración de la partida: esta variable es el porcentaje de turnos con respecto al máximo permitido $\tau_{\text{máx}}$ que se han jugado en la partida actual (tomando valores, obviamente, entre 0 y 1):

$$T_i = \tau_i / \tau_{\text{máx}} \quad (5.7)$$

Todas estas variables se promedian para las N_g partidas jugadas en cada evaluación del mapa, de forma que se obtienen valores agregados. Por simplicidad, se ha optado por omitir el sub-índice i para indicar estos valores promediados (Π , Γ , etc.), los cuales forman parte de las reglas difusas definidas a continuación:

La primera regla establece que un mapa tendrá un equilibrio alto si la diferencia entre el número de planetas de ambos jugadores durante la partida (es decir, la variable Π) ha sido baja al igual que la diferencia en el crecimiento Γ . Si ésta última es baja y, por tanto, ninguno de los dos jugadores ha tenido una ventaja con respecto al otro en la tasa de crecimiento, se determina que el equilibrio ha sido medio (ver regla 2) si, además, hay un desequilibrio en el número de planetas y naves (que se traduce en que uno de los jugadores ha tenido una ventaja territorial). Por último, la tercera regla establece que hay un equilibrio bajo si ambos jugadores han controlado un número

Interpretación de las reglas difusas

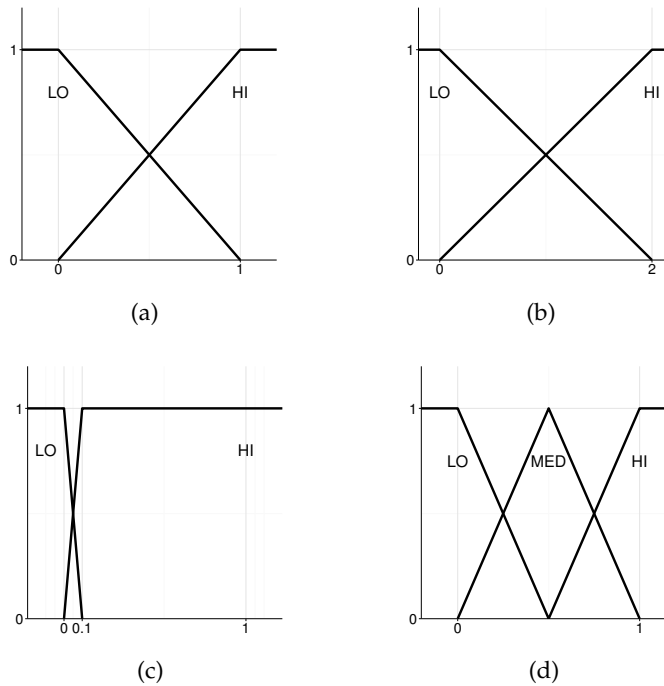


Figura 5.5: Funciones de pertenencia a los conjuntos difusos. Se corresponden con las variables (a) Π, Γ, Ξ, T, K (b) $\Delta^\pi, \Delta^\gamma, \Delta^\xi$ (c) Z y (d) bal y dyn .

similar de planetas pero con tamaños dispares, o si la partida ha durado pocos turnos (y por tanto uno de los jugadores ha podido vencer con facilidad). Nótese que no es necesario tener en cuenta todas las posibles combinaciones de las variables de entrada. Por ejemplo, si hay un desequilibrio en el número de planetas y en el tamaño de los mismos no sería posible establecer si la partida ha sido globalmente equilibrada o no, ya que esto dependería en gran medida de si ambos desequilibrios se asignan a un mismo jugador o se asigna uno a cada uno. Esta situación, sin embargo, no supone un problema, pues las reglas pueden activarse parcialmente en el caso de ocurrir, ya que hay dos conjuntos difusos de entrada que se solapan (ver Figura 5.5). A la hora de evaluar las reglas, se ha usado la función mín como t-norma, el máx como t-conorma, y el centro de masas como método para seleccionar el valor de salida.

5.2 DINAMISMO

Aunque el hecho de que una partida entre dos jugadores sea equilibrada supone una mejora en la experiencia de los jugadores, puede suceder que, al mismo tiempo, la experiencia se vea arruinada por un exceso de equilibrio. Por ejemplo, una partida en la que los jugadores estén sumamente igualados se convierte en algo monótono y carente de sentido, pues ninguno de ellos es capaz de destacar sobre el

otro, con la consecuente incapacidad de crear situaciones divertidas o que supongan un desafío. Una situación extrema puede darse por la completa inactividad de los jugadores, lo cual técnicamente se traduce en una partida perfectamente equilibrada, pero que dista mucho de ser algo divertido de jugar. Es por ello que se ha decidido estudiar otra propiedad que deberían fomentar los mapas generados de forma automática: el dinamismo. Cabe por tanto preguntarse cómo definir este dinamismo en el contexto del juego *Planet Wars* y cómo capturarlo para utilizarlo como medida de aptitud en el generador propuesto.

Capturando el dinamismo

A la hora de evaluar el dinamismo, se han considerado dos grupos de indicadores [90]. El primero refleja el dinamismo de la partida desde una perspectiva basada en los recursos (es decir, se ha relacionado el dinamismo con la variación en la cantidad de recursos controlados por cada jugador). El segundo grupo, por su parte, se centra en las confrontaciones entre los jugadores (o lo que es lo mismo, el dinamismo se intenta capturar por el alcance de las batallas entre jugadores).

Indicadores de dinamismo

De forma más detallada, estos son los indicadores de dinamismo para una determinada partida i :

■ **Dinamismo basado en recursos:**

- *Longitud de la partida* T_i : este es el ratio de turnos que se han jugado con respecto al máximo número de turnos $\tau_{\text{máx}}$ permitido para cada partida (toma valores entre 0 y 1):

$$T_i = \tau_i / \tau_{\text{máx}} \quad (5.8)$$

- *Ratio de conquista* K_i : este es el ratio de planetas que no son neutrales al final de la partida, es decir, que han sido conquistados por algún jugador. Se puede calcular fácilmente usando valores $\pi_{i\tau_{\text{máx}}}^{(a)}$ tales que

$$K_i = \pi_{i\tau_{\text{máx}}}^{(1)} + \pi_{i\tau_{\text{máx}}}^{(2)} \quad (5.9)$$

tomando, de nuevo, valores entre 0 y 1.

- *Ratio de reconquista* Z_i : Sea ζ_{ij} el número de planetas controlados por un jugador en el turno $j - 1$ y conquistados por otro jugador en el turno j (sin importar quién era el propietario original, o lo que es lo mismo, se cuentan cuantos planetas han cambiado de propietario en cada turno. Entonces

$$Z_i = \frac{1}{\tau_i} \sum_{j=1}^{\tau_i} \zeta_{ij} / n_p \quad (5.10)$$

donde n_p es el número de planetas del mapa. Aunque esta variable teóricamente toma valores entre 0 y 1, en la práctica es más común encontrar valores cercanos al límite inferior.

- *Pico de diferencia*: este es un conjunto de variables que miden la máxima amplitud de la variación en alguno de los recursos que se tienen en cuenta, es decir, los planetas (π), la capacidad de crecimiento (γ) y el número de naves (ξ). Sea $\phi_{ij}^{(a)}$ la cantidad de recursos ϕ controlados por el jugador a en el turno j -ésimo de la partida i -ésima, se registran los dos puntos en los cuales la diferencia relativa es mayor para un jugador como para su oponente y se suman ambas cantidades:

$$\Delta_i^\phi = \max_{1 \leq j \leq \tau_i} \left\{ \frac{\phi_{ij}^{(1)} - \phi_{ij}^{(2)}}{\phi_{ij}^{(1)} + \phi_{ij}^{(2)}} \right\} - \min_{1 \leq j \leq \tau_i} \left\{ \frac{\phi_{ij}^{(1)} - \phi_{ij}^{(2)}}{\phi_{ij}^{(1)} + \phi_{ij}^{(2)}} \right\} \quad (5.11)$$

En esta ocasión, las tres variables $\Delta_i^\pi, \Delta_i^\gamma$ and Δ_i^ξ toman valores entre 0 y 2.

■ **Dinamismo basado en la confrontación:**

- *Ratio de batallas* B_i : este es el ratio de planetas bajo ataque a lo largo del juego. Sea β_{ij} el número de planetas que estaban siendo atacados durante el turno j -ésimo, entonces $B_i = \frac{1}{\tau_i} \sum_{j=1}^{\tau_i} \beta_{ij} / n_p$.
- *Naves destruidas* S_i : este es el ratio de naves generadas que han sido destruidas a lo largo de la partida. Sea χ_i el número de naves destruidas y ψ_i el número de naves creadas, entonces $S_i = \chi_i / \psi_i$.

Al igual que para el equilibrio, se ha usado un sistema de torneo para la evaluación de los mapas, por lo que se ha considerado el valor medio de los indicadores para las N_g partidas (se ha eliminado el sub-índice para indicar este valor promedio). Además, se ha definido un conjunto de reglas difusas para expresar el valor de dinamismo en función de estos indicadores. El conjunto de reglas se detalla en la Figura 5.6. Los conjuntos difusos (HI y LO) se han definido de forma que alcancen su máximo en el correspondiente extremo del rango de valores posibles para la variable en consideración, y decrece linealmente hacia el extremo opuesto. Hay dos excepciones, Z y B cuyos valores normales se encuentran lejos del máximo teórico 1.0. En este caso, se ha saturado el valor HI a 1 cuando se alcanzan los valores 0.1 y 0.35, respectivamente (estos valores se han determinado empíricamente). Para la variable de salida dyn se ha definido un conjunto intermedio triangular MED, que alcanza su máximo en 0.5 y decrece linealmente hacia ambos extremos. En este caso, tanto HI como LO alcanzan su mínimo en este valor 0.5.

De la misma forma que para el equilibrio, se ha usado la función mín como t-norma, el máx como t-conorma, la función x^2 para el mo-

Descripción de las reglas de dinamismo

Dinamismo basado en recursos:

1. **si K es HI y Z es HI entonces dyn es HI**
2. **si Δ^π es HI y Δ^γ es HI y Δ^ξ es HI entonces dyn es HI**
3. **si Δ^π es HI y (Δ^γ es LO o Δ^ξ es LO) entonces dyn es MED**
4. **si Δ^γ es HI y (Δ^π es LO o Δ^ξ es LO) entonces dyn es MED**
5. **si Δ^ξ es HI y (Δ^γ es LO o Δ^π es LO) entonces dyn es MED**
6. **si Δ^π es LO y Δ^γ es LO y Δ^ξ es LO entonces dyn es LO**
7. **si K es LO o Z es LO o T es muy LO entonces dyn es LO**

Dinamismo basado en la confrontación:

1. **si B es HI y S es HI entonces dyn es HI**
2. **si (B es HI y S es LO) o (B es LO y S es HI) entonces dyn es MED**
3. **si B es LO y S es LO entonces dyn es LO**

Figura 5.6: Reglas difusas para el dinamismo.

dificador difuso *very*, y el centro de masas como método para seleccionar el valor de salida. Las funciones de pertenencia a los conjuntos difusos pueden verse en la Figura 5.5. El razonamiento que subyace en las reglas difusas basadas en los recursos es el de considerar una partida como muy dinámica si se han conquistado muchos planetas y éstos han cambiado de mano con frecuencia, como se observa en la regla (1). De la misma forma, si los picos de diferencia de todos los recursos que se han tenido en cuenta son altos, entonces la partida también es dinámica ya que uno de los jugadores ha logrado cambiar las tornas de la partida cuando se encontraba en una posición difícil, que es lo que se encapsula en la regla (2). Si el pico de diferencia es alto solo en uno de los tres recursos entonces se considera la partida como medianamente dinámica (reglas (3)-(5)). Siguiendo con el mismo razonamiento, si todos los picos de diferencia son bajos, también lo será el dinamismo de la partida, como establece la regla (6). No obstante, hay que tener en cuenta que esta regla no cubriría una situación en la que ambos jugadores estuviesen constantemente luchando entre ellos pero manteniendo sus fuerzas equilibradas. Esta situación debería considerarse dinámica, y de cualquier forma se considerará así, pues la regla (1) entra en juego aumentando el nivel de dinamismo. Por último, la regla (7) expresa que si se han conquistado pocos planetas o éstos cambian de propietario con baja frecuencia o la partida ha sido muy corta, entonces ésta debe considerarse poco dinámica.

Con respecto a las reglas difusas para el dinamismo basado en la confrontación, éstas establecen que la partida es dinámica si el ratio

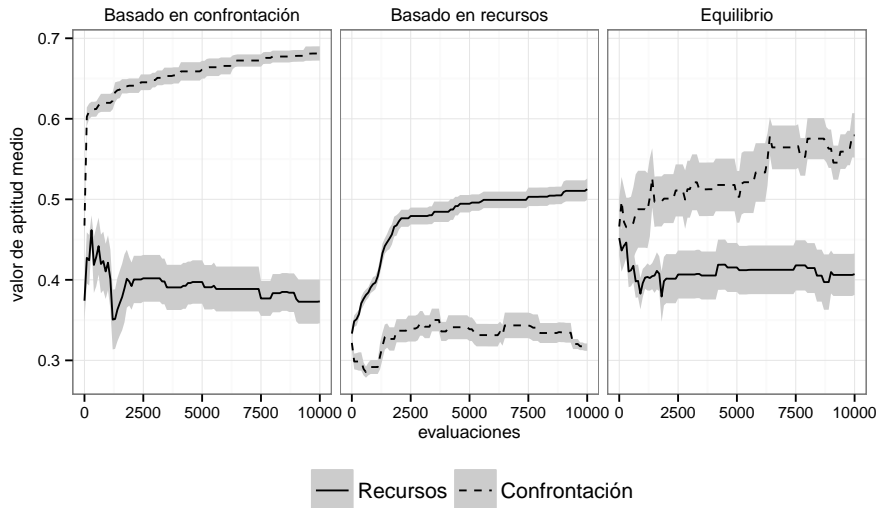


Figura 5.7: Evolución del valor de aptitud medio para ambas funciones objetivo. El área sombreada muestra el error estándar de la media. Las figuras izquierda y central muestran la evolución del dinamismo medido según ambos enfoques (recursos y confrontación) cuando se está optimizando uno de ellos, indicado por el título de la figura. En la derecha se muestra la evolución del equilibrio cuando se optimiza el dinamismo según ambos enfoques. Cada línea muestra la media de 10 ejecuciones del algoritmo.

de batallas y naves destruidas han sido altos, y, consecuentemente, se establece un dinamismo bajo para aquellas partidas con pocas batallas y naves destruidas. La regla (2), por su parte, asigna un dinamismo de nivel medio a las partidas donde, a pesar de haberse producido un número elevado de batallas, el número de naves destruidas no ha sido elevado.

Se han realizado dos experimentos centrándose cada uno de ellos en el comportamiento del generador cuando se intenta optimizar el dinamismo basado en la confrontación o en los recursos. El algoritmo en cuestión ha sido el mismo que se ha utilizado para el equilibrio en la Sección 5.1, mientras que los resultados se pueden observar en la Figura 5.7. Dichas figuras muestran la correlación de ambos objetivos cuando se está optimizando el otro, y se puede observar que ambas medidas son ortogonales, es decir, cuando se está optimizando una de ellas la otra medida sigue una trayectoria plana, por lo que ambas medidas se pueden ver como formas complementarias de medir el dinamismo. Nótese también que el dinamismo basado en la confrontación parece converger con mayor rapidez, indicando que es más sencillo motivar la confrontación entre los jugadores que provocar grandes fluctuaciones en sus recursos mediante el ajuste de los mapas.

Experimentación

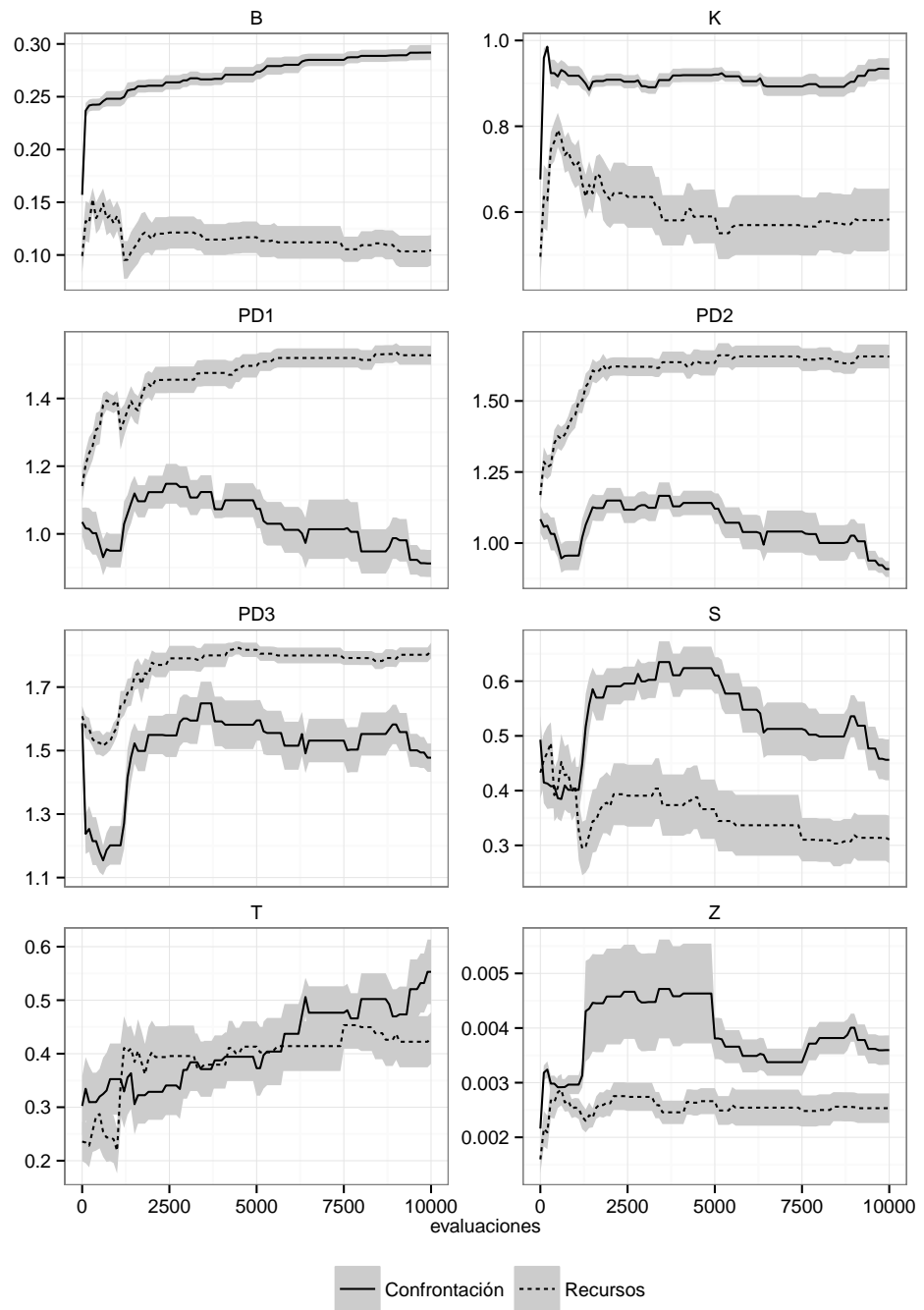


Figura 5.8: Evolución de las diferentes variables recogidas en las reglas de los conjuntos difusos. B y K son los ratios de batallas y conquistas, respectivamente, las variables PD representan los picos de diferencia entre los recursos de cada jugador, S se corresponde con las naves destruidas, T es la longitud de la partida y Z se corresponde con el ratio de reconquista.

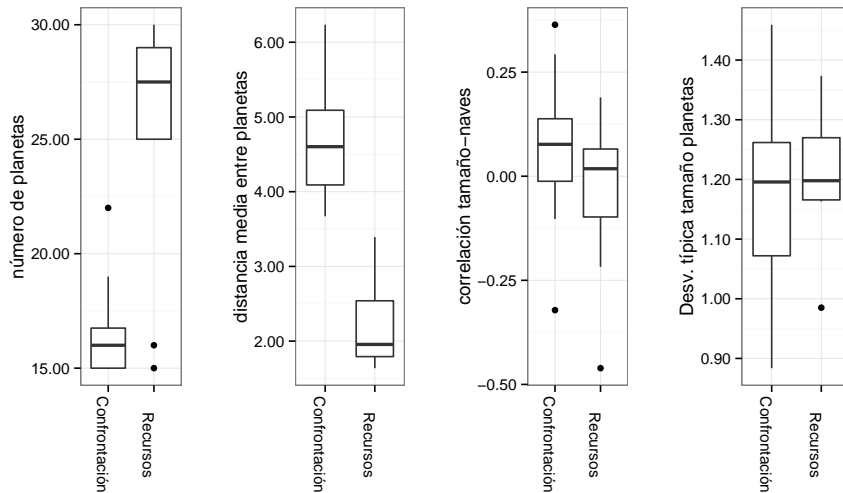
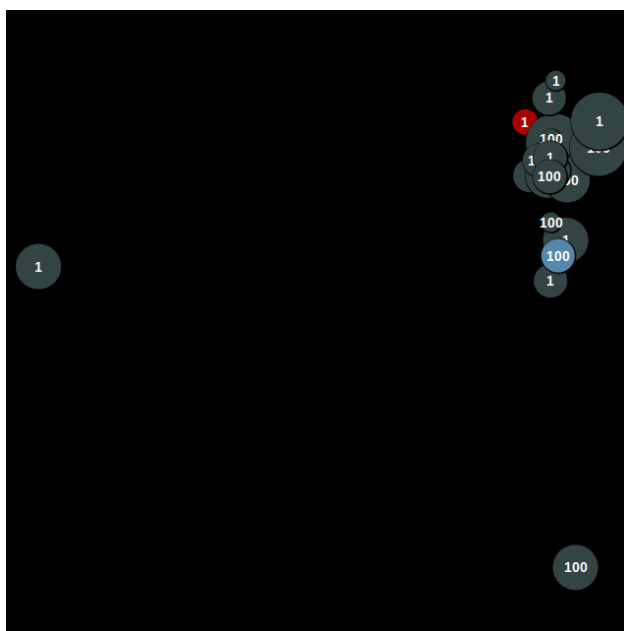
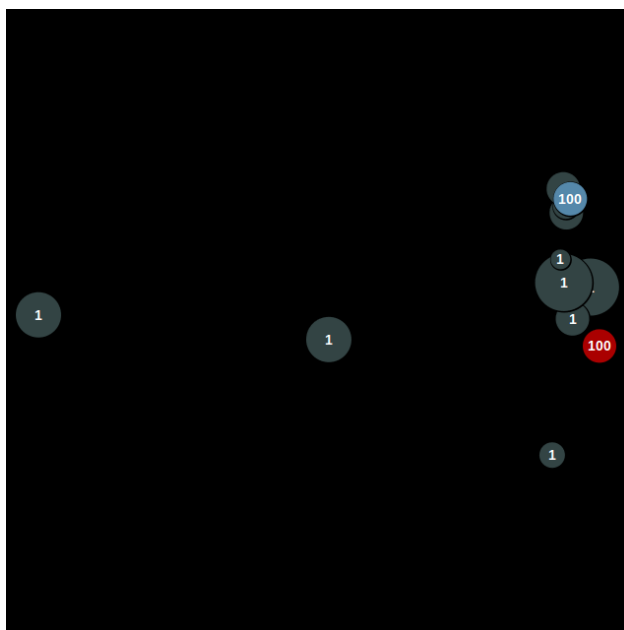


Figura 5.9: Varias características de los mejores mapas generados en cada ejecución del algoritmo y para cada función objetivo: número de planetas en el mapa, distancia media entre esos planetas, correlación entre el tamaño de los planetas y el número inicial de naves de cada uno y la desviación típica del tamaño de los mismos.

La Figura 5.8 muestra la evolución de las diferentes variables usadas para medir el dinamismo de los mapas durante la evolución de los mismos. Como era de esperar, las variables usadas durante la optimización de ambos enfoques (confrontación y recursos) muestran una tendencia creciente in general. Lo que resulta más interesante son las interrelaciones entre ellas. En primer lugar, la tasa de conquista K crece de forma más rápida en el caso del dinamismo orientado a la confrontación que en el orientado a los recursos, a pesar del hecho de que solo se la incluye explícitamente en el segundo caso. Esto afecta a la optimización del ratio de batallas B : hay que atacar un planeta para conquistarlo por primera vez, por lo que un incremento en el número de planetas conquistados implica un incremento en el número de batallas, lo cual es explotado por el algoritmo evolutivo para llevar a cabo la optimización. De manera similar, la tasa de reconquista es también más alta en la optimización basada en la confrontación, ya que un número elevado de batallas puede provocar que numerosos planetas cambien de mano. Ambas funciones objetivo tienden a generar partidas más largas (o lo que es lo mismo, valores más altos de T): mientras que esto se recoge explícitamente en la función objetivo basada en los recursos, también aparece de forma implícita en la que está basada en la confrontación ya que en partidas más largas se llevan a cabo más batallas. Además incrementa el número de naves producidas lo cual, de nuevo, provoca un incremento en el número de batallas para mantener una alta tasa de naves destruidas.



(a)



(b)

Figura 5.10: Dos ejemplos de mapas dinámicos.

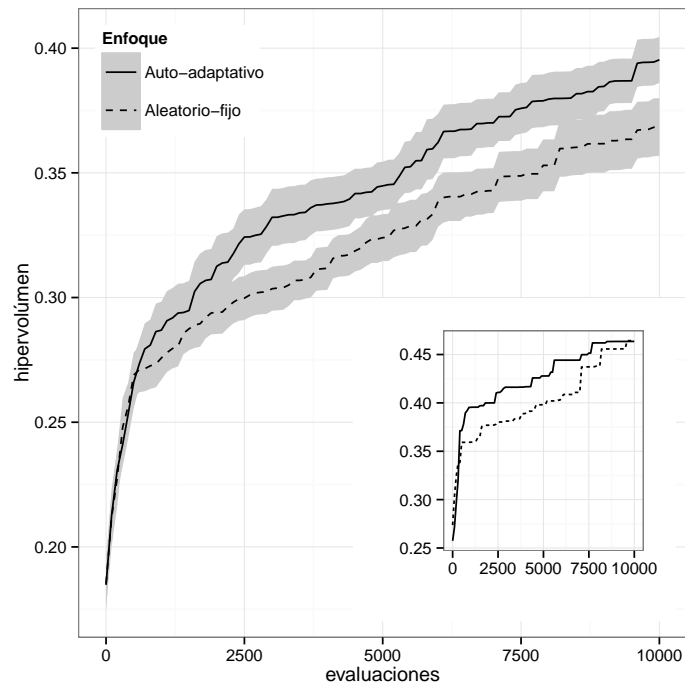
Por último, en la Figura 5.9 se pueden observar las características de los mapas generados. Los mapas obtenidos mediante ambos enfoques tienen características similares como demuestra la correlación entre el tamaño de los planetas y el número de naves que se sitúan inicialmente en ellos, así como en términos de variabilidad del tamaño de los planetas. Sin embargo difieren en el número de planetas y en la distancia media entre ellos. El bajo número de planetas en los mapas dinámicos basados en la confrontación se puede explicar por el hecho de que tener pocos planetas reduce las posibilidades de expansión de los jugadores, forzándolos a que batallen por los mismos planetas y por tanto incrementen el número de batallas. Por su parte, el hecho de tener una mayor distancia entre los planetas tiene el efecto de incrementar el tiempo de traslado de las naves de uno a otro. Una posible hipótesis es que este tiempo de traslado introduce un factor de inestabilidad haciendo más complicado mantener posiciones e incrementando el tiempo de reacción a los ataques, provocando entonces más batallas para volver a ganar el control de los planetas. La Figura 5.10 muestra dos ejemplos de mapas dinámicos generados automáticamente.

5.3 EQUILIBRIO VS. DINAMISMO

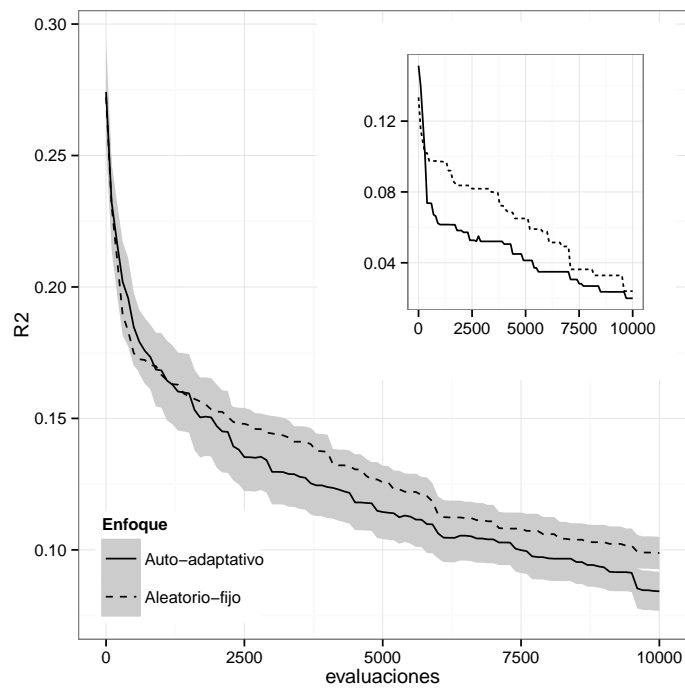
Una vez estudiadas ambas características deseables, el paso lógico era establecer una comparación entre ambas, para ver las posibles relaciones así como el comportamiento de una con respecto a la otra. Mediante el análisis de los resultados obtenidos durante la generación automática de mapas equilibrados (sección 5.1) y dinámicos (sección 5.2) se han observado varias diferencias cualitativas entre ambos enfoques con respecto a su optimización.

En primer lugar, la convergencia del algoritmo evolutivo para el caso del dinamismo es más lenta, sugiriendo que queda margen para la mejora si se lanzan ejecuciones más largas. Además, el valor medio de aptitud de las soluciones se queda en la mitad alta del rango posible, lo cual unido a la anterior observación, indica una gran dificultad a la hora de encontrar mapas que sean muy dinámicos. Al contrario que en el caso de los mapas equilibrados, en el cual el algoritmo es capaz de encontrar soluciones perfectas (valor de aptitud 1.0) en bastantes ocasiones (de hecho, el valor de aptitud mediano es, precisamente, 1.0).

En segundo lugar, la trayectoria que sigue el equilibrio de los mapas que están siendo optimizados para que sean dinámicos no decrece de forma significativa conforme aumenta el dinamismo, a diferencia de lo que ocurre en el caso contrario, donde el dinamismo decrece considerablemente conforme aumenta el equilibrio de los mapas. Una posible explicación a este efecto es el hecho de que la definición de dinamismo incluye de forma implícita un componente de equilibrio



(a)



(b)

Figura 5.11: Evolución de los indicadores de rendimiento para el frente no dominado en cada una de las ejecuciones del algoritmo. La figura interior muestra la misma medida para el frente acumulado de ambos algoritmos. (a) Hipervolumen (b) R_2 .

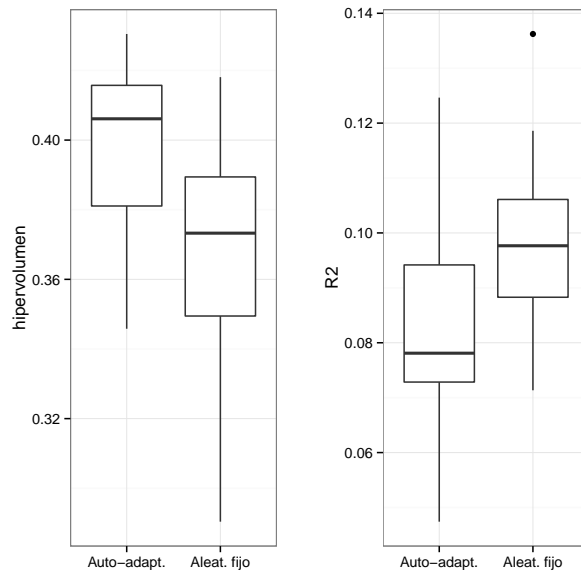


Figura 5.12: Diagrama de cajas de los indicadores de rendimiento para los dos enfoques de algoritmo multiobjetivo.

al medir la diferencia pico en el valor de las variables, de forma que si el juego está muy desequilibrado y un jugador domina al otro, estas variables tomarían valores más bajos que en el caso de que el jugador dominado se recupera en mitad de la partida y termina venciendo, pasando la partida por un estado equilibrado intermedio. El hecho de que en ese tipo de partidas muy desequilibradas la victoria del jugador dominante puede ocurrir en los primeros turnos de la partida penaliza al dinamismo de la misma, pues hay una regla para el dinamismo que penaliza las partidas de corta duración.

La observación de los resultados anteriores y más concretamente la diferencia sustancial en las trayectorias de los valores de aptitud de ambas propiedades saca a la luz algunas cuestiones interesantes como, por ejemplo, cuáles son las contrapartidas existentes entre el equilibrio y el dinamismo. Para analizarlo se requiere abordar el problema desde una perspectiva de optimización multiobjetivo, que es precisamente el tema de la siguiente sección.

5.4 EQUILIBRIO Y DINAMISMO: UN ENFOQUE MULTIOBJETIVO

Para el enfoque de optimización multiobjetivo se ha optado por usar el algoritmo *Non-dominated Sorting Generic Algorithm II – NSGA-II* [30]. Los motivos de esta elección han sido principalmente la eficiencia computacional del algoritmo en la librería software utilizada para la experimentación [45] así como el hecho de ser un algoritmo ampliamente utilizado y testeado en la literatura. Al igual que con los algoritmos de objetivo único para equilibrio y dinamismo, se ha

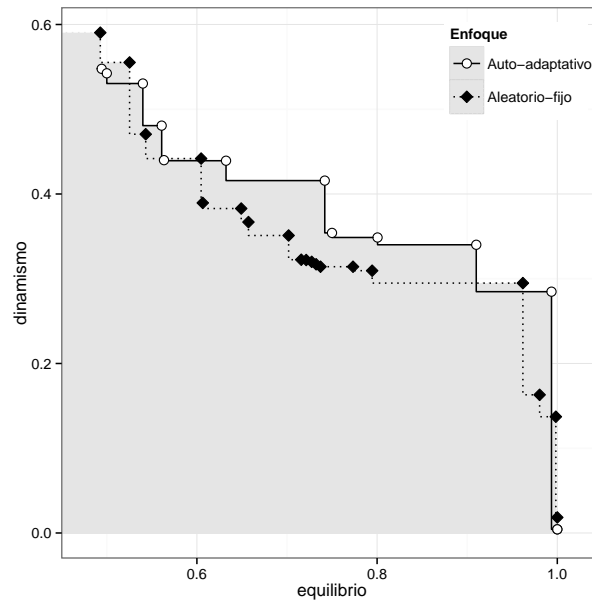
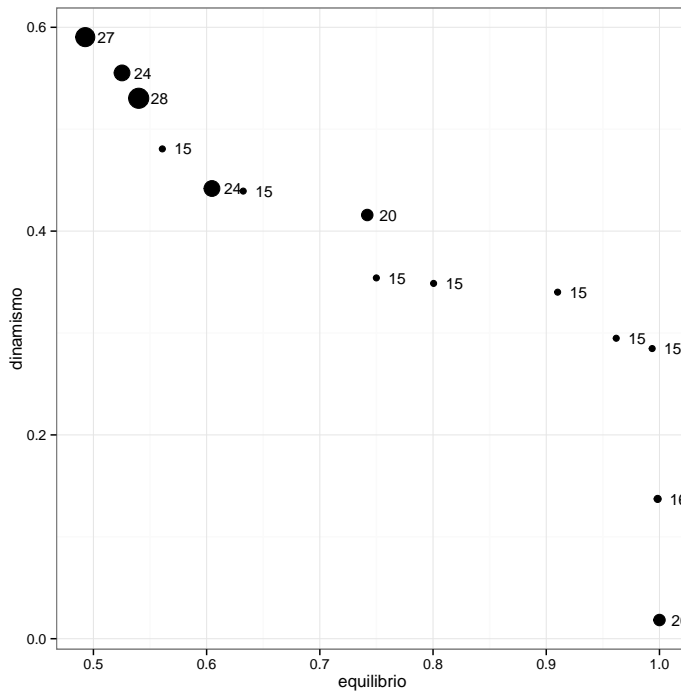


Figura 5.13: Comparativa de los frentes no dominados globales obtenidos por ambos algoritmos.

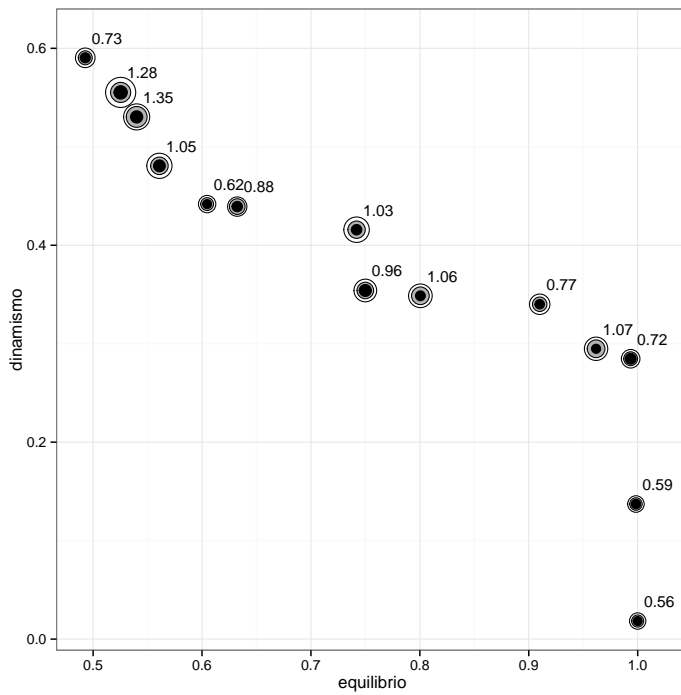
evaluado la efectividad de la auto-adaptación en el número de planetas contra otra versión del algoritmo en la cual el número de planetas se fijó aleatoriamente al principio de cada ejecución. La comparación se basa en dos indicadores de rendimiento: el hipervolumen y la métrica R_2 . El primero de ellos proporciona una medida de la región del espacio de soluciones que está dominado. Esto requiere un punto de referencia, el cual se ha decidido que sea $(0,0)$, es decir, el valor de aptitud multiobjetivo de la peor solución hipotética. Con respecto al segundo indicador, es una estimación sobre cuánto se aproxima un determinado frente a otro dado. El frente sobre el que comparar, en este caso, se ha construido uniendo los frentes no dominados que se descubrieron en todas las ejecuciones y, posteriormente, eliminando las soluciones dominadas. Este frente agrupado se usó como frente de referencia sobre el que calcular la medida R_2 . Queda claro, entonces, que el objetivo es minimizar el valor de R_2 pues ello significa que el frente de soluciones no dominadas del algoritmo se parece mucho al frente en conjunto de las soluciones encontradas en todas las ejecuciones del algoritmo.

Como se puede ver en la Figura 5.11, el algoritmo auto-adaptativo tiene una convergencia más rápida hacia buenos valores de ambos indicadores. La distribución real de los valores finales de estos indicadores se muestra en la Figura 5.12. En ambos casos la diferencia es favorable al algoritmo auto-adaptativo (estadísticamente significativo al nivel de $\alpha = .1$).

Por otro lado, la comparación de los frentes acumulados encontrados para cada algoritmo se muestra en la Figura 5.13. Es interesante



(a)



(b)

Figura 5.14: (a) Número de planetas en cada solución del frente no dominado. (b) Distribución de las distancias interplanetarias. El círculo negro se corresponde con el percentil 25 %, el gris con la mediana y el blanco con el percentil 75 %

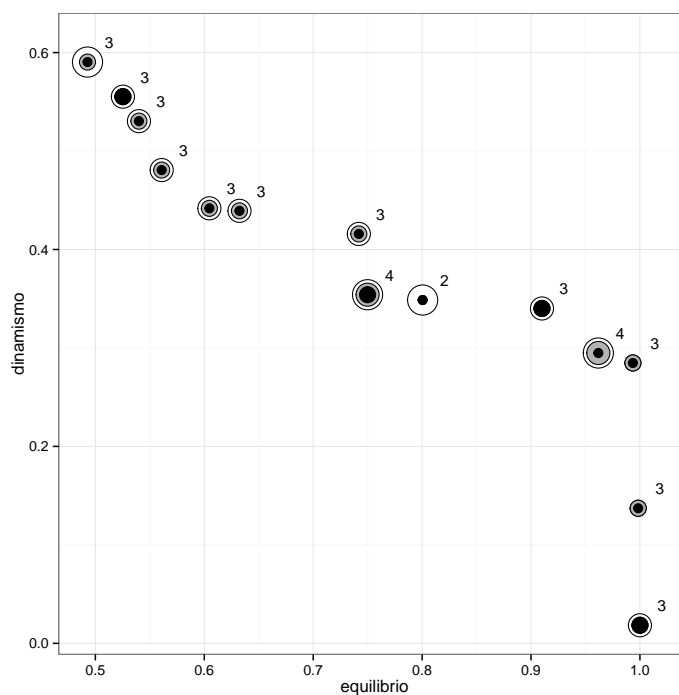


Figura 5.15: Distribución de los tamaños de los planetas en cada solución del frente no dominado. El círculo negro se corresponde con el percentil 25 %, el gris con la mediana y el blanco con el percentil 75 %.

el hecho de que la variante con un número fijo de planetas parece comportarse mejor en los extremos del frente, mientras que el auto-adaptativo tiene un mejor comportamiento a la hora de explorar la parte central del mismo, donde se dan cita un amplio espectro de contrapartidas entre equilibrio y dinamismo.

Poniendo el foco en las soluciones que conforman el frente no dominado, se han analizado las relaciones que hay en las propiedades de los mapas para cada objetivo considerado, es decir, dinamismo y equilibrio. Si se analiza el número de planetas en cada mapa (ver 5.14a) se observa que, por norma general, este valor tiende a ser mayor en los mapas dinámicos. Además, el número se acerca al límite superior de 30 planetas en la parte izquierda del frente y alcanza el límite inferior de 15 varias veces conforme se avanza hacia la derecha. Esto se debe al elevado número de opciones de las que disponen los jugadores en los mapas grandes y la facilidad que aportan para construir grandes imperios o librar más batallas por el control de los planetas. Más allá de la cantidad de planetas, su distribución en el mapa también es diferente dependiendo de la zona del frente no dominado que se analice, como muestra la Figura 5.14b. Aunque la tendencia no es tan marcada como en el caso anterior, se puede observar que los planetas están generalmente más separados en los mapas con alto dinamismo que en los equilibrados. Además, la localización de éstos parece estar

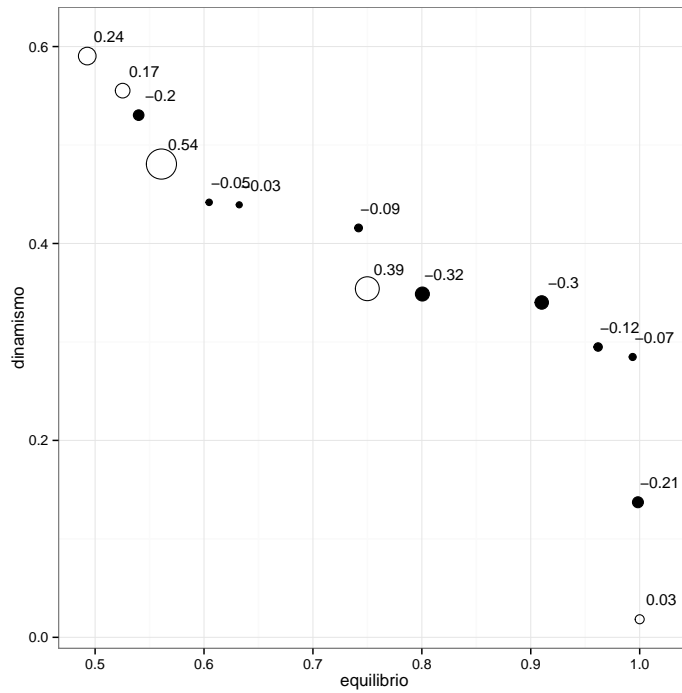


Figura 5.16: Coeficiente de correlación de Pearson entre el tamaño del planeta y el número inicial de naves que se encuentran en él al principio de la partida. Un círculo blanco indica correlación positiva mientras que uno negro indica una correlación negativa.

menos estructurada (como indica la elevada distancia inter-cuartilica). Esta irregularidad alta contribuye al dinamismo ya que hace que los mapas sean más anisotrópicos y, por tanto, proporcionando diferentes estímulos a los jugadores que terminarán convirtiéndose en un mayor número de acciones de juego.

Con respecto a los tamaños de los planetas (Figura 5.15) el valor mediano en los extremos del frente se comporta de forma análoga, pero parecen tener una mayor dispersión en el extremo correspondiente al dinamismo alto. De nuevo, el efecto de la irregularidad juega en favor del dinamismo, aunque la tendencia no está muy marcada. Sin embargo, un hecho interesante que aparece al intentar correlacionar el tamaño de cada planeta con el número inicial de naves que aparecen en él (Figura 5.16): la correlación tiende a ser positiva en el extremo de alto dinamismo mientras que es prácticamente cero cerca del extremo opuesto. Esto puede verse como una guarda contra escenarios desequilibrados en los cuales planetas de gran tamaño puedan ser conquistados sin apenas esfuerzo, pudiéndose convertir en una ventaja decisiva para uno de los jugadores.

5.5 RESULTADOS Y CONTRIBUCIONES

A lo largo del presente capítulo se han presentado varios enfoques diferentes sobre la generación automática de mapas para un juego de estrategia en tiempo real que buscan incrementar la satisfacción del jugador exigiendo que el contenido generado, es decir los mapas, muestren determinadas características deseables que ayudan a dicho objetivo.

En primer lugar se ha presentado el concepto de equilibrio entre jugadores, y se ha descrito un método de generación enfocado en esta característica que es capaz de crear mapas que no proporcionan ventaja alguna a los jugadores, independientemente de su habilidad o estrategia elegida. De esta forma, las partidas se hacen más interesantes tanto para los jugadores novatos, que no pierden con facilidad las partidas, así como para los más experimentados, que experimentan un incremento en su nivel de competitividad gracias a los nuevos retos que estos mapas equilibrados les suponen. Aunque los mapas generados son totalmente jugables y no incumplen ninguna regla del juego, se ha detectado que en muchos casos el equilibrio en las partidas se consigue mediante la completa inacción de los jugadores, por lo que ha sido necesario proponer una nueva característica deseable que evita que las partidas se conviertan en algo aburrido y monótono: el dinamismo.

A la hora de buscar el dinamismo se ha optado por dos enfoques distintos, uno basado en los recursos que mantienen los jugadores a lo largo de la partida y otro basado en las confrontaciones entre ellos, que se centra en las batallas y el nivel de naves perdidas. Ambos enfoques han resultado ser ortogonales y, por su parte, se ha detectado una posible correlación entre el dinamismo basado en la confrontación y el equilibrio, lo que ha motivado el posterior análisis en conjunto de ambas medidas así como el desarrollo de una optimización multi-objetivo para generar mapas que sean tanto dinámicos como equilibrados.

Con respecto al análisis conjunto de ambas características, se ha detectado que la definición de dinamismo que se ha usado como objetivo de la optimización incluye implícitamente una componente de equilibrio, que se debe a que un mapa extremadamente desequilibrado hace que la partida termine muy pronto, reduciendo así el dinamismo. De esta forma se ha llegado a la conclusión de que el equilibrio es compatible con un nivel medio de dinamismo.

Por último se ha estudiado un enfoque multi-objetivo del problema que ha sacado a la luz otras relaciones entre equilibrio y dinamismo y mediante el cual se ha establecido que conforme aumenta el equilibrio se produce un leve descenso en el dinamismo, acabando con un descenso brusco en la frontera superior del equilibrio. Además se han estudiado las características de los mapas generados por este enfo-

que multi-objetivo, y, en general, las partidas dinámicas parecen estar relacionadas con mapas que tienen un número elevado de planetas distribuidos de forma dispersa y cuyos tamaños están positivamente correlacionados con el número inicial de naves en ellos.

Desde el punto de vista algorítmico, se ha presentado un AE que optimiza mapas auto-adaptando los parámetros de mutación así como el número de planetas. Se ha analizado el rendimiento de la versión auto-adaptativa del algoritmo comparando sus resultados con una variante que fija de forma aleatoria el número de planetas en los mapas en el momento de inicio del algoritmo. Se ha observado que la versión auto-adaptativa supera a la segunda tanto en la optimización de un único objetivo como en el caso multi-objetivo.

Los resultados obtenidos puedan parecer específicos para *Planet Wars*, el juego empleado en la experimentación. Sin embargo, el método utilizado podría ser adaptado con facilidad a otros juegos de ETR, siempre que en sus mecánicas de juego esté incluida la confrontación de los jugadores por la obtención de recursos que se distribuyen espacialmente en el mapa. Las medidas de equilibrio y dinamismo se adaptarían a los recursos y tipos de unidades del juego en cuestión, pero los conceptos serían los mismos: desequilibrio territorial, en el crecimiento y en las unidades, y dinamismo basado en los recursos y en la confrontación.

En el capítulo 5 se han propuesto diversos métodos para mejorar la experiencia de juego mediante la generación automática de mapas que hacen que las partidas tengan ciertas características deseables por los jugadores, como son un equilibrio en el nivel de juego de los mismos así como un alto grado de dinamismo en las partidas. Un problema de los mapas obtenidos de esta forma es su mala apariencia estética (por ejemplo, la mayoría de los planetas agrupados muy juntos en una zona pequeña del mapa), lo cual puede provocar rechazo o desorientación a los jugadores. Es por ello que el presente capítulo también se centra en mejorar la experiencia de juego pero desde un punto de vista diferente: la estética.

Para evaluar el nivel estético de los mapas es necesario caracterizar sus propiedades y compararlas con aquellos mapas que se sabe tienen buena o mala estética. Para este propósito se han considerado diferentes tipos de propiedades. Desde un punto de vista general se ha establecido una distinción entre las características *geométricas*, que se basan en la geometría espacial del mapa (coordenadas y distancias), y *topológicas*, que se basan a su vez en propiedades cualitativas de los mapas que no se ven afectadas por transformaciones *geométricas* simples. Además, se han tenido en cuenta propiedades morfológicas en combinación con los otros dos tipos, basadas en las características individuales de los planetas de cada mapa, como el tamaño o el número inicial de naves. En conjunto se obtienen dos grupos de características a efectos de clasificación: medidas *geométricas* y *topológicas*.

Caracterización de los mapas

6.1 MEDIDAS GEOMÉTRICAS

Las medidas *geométricas* son aquellas que se basan en la geometría espacial del mapa:

- Distribución espacial de los planetas: Sean $\vec{p}_i = (x_i, y_i)$ las coordenadas del i -ésimo planeta y n_p el número total de ellos. La distancia media entre planetas μ_d y la desviación típica de estas distancias σ_d se define de la siguiente forma:

$$\mu_d = \frac{1}{n_p^2} \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \|\vec{p}_i - \vec{p}_j\| \quad (6.1)$$

$$\sigma_d = \sqrt{\frac{1}{n_p^2} \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} (\|\vec{p}_i - \vec{p}_j\| - \mu_d)^2} \quad (6.2)$$

- Propiedades de los planetas: Sean s_i y w_i el tamaño y número inicial de naves, respectivamente, del i -ésimo planeta. La media y desviación típica de estos tamaños (μ_s y σ_s respectivamente) y la correlación de Pearson ρ entre el tamaño del planeta y el número inicial de naves se definen de la siguiente forma:

$$\mu_s = \frac{1}{n_p} \sum_{i=1}^{n_p} s_i \quad (6.3)$$

$$\sigma_s = \sqrt{\frac{1}{n_p} \sum_{i=1}^{n_p} (s_i - \mu_s)^2} \quad (6.4)$$

$$\rho = \frac{\sum_{i=1}^{n_p} s_i w_i - n_p \mu_s \mu_w}{n_p \sigma_s \sigma_w} \quad (6.5)$$

donde μ_w y σ_w son la media y desviación típica del número de naves, respectivamente.

Estas medidas se han aplicado para comparar el grado de similitud entre mapas de la siguiente manera: cada mapa se ha caracterizado como una tupla $\langle \mu_d, \sigma_d, \mu_s, \sigma_s, \rho \rangle$, de forma que la distancia euclídea entre dos de estas tuplas define el grado de similitud entre los mapas que representan. Además, se han definido dos conjuntos de mapas, el primero de ellos conteniendo 10 mapas con una buena estética, mientras que el segundo contiene 10 mapas con una estética mala (en la Figura 6.6 hay ejemplos de mapas pertenecientes a estos conjuntos). Un experto ha sido el encargado de evaluar la estética de los mapas para asignarlos a un conjunto o al otro siguiendo su propia percepción de lo que es estético y lo que no, obteniéndose la muestra tanto de los mapas oficiales del *Google AI Challenge* como de los obtenidos por el generador automático presentado en el capítulo 5. Estos conjuntos sirven como punto de referencia de forma que el objetivo de generar mapas estéticos se ha convertido en un problema de optimización donde se minimiza la distancia a los mapas del conjunto estético, mientras se maximiza a los mapas del no estético. Esto último es necesario para introducir diversidad en el conjunto de mapas generados, evitando así la creación de mapas que sean prácticamente copias de los mapas del conjunto estético.

6.2 MEDIDAS TOPOLÓGICAS

Las medidas *topológicas* de cada mapa se han obtenido a partir del Grafo de Esferas de Influencia (GEI) [155]. Este grafo establece una relación entre los puntos de un determinado conjunto basada en la distribución espacial de los mismos (ver Figura 6.1). Esta relación no se ve afectada por operaciones de escalado, traslación o rotación sobre los puntos, lo cual es una característica deseable como ya se ha dicho anteriormente. Los GEI se han usado para multitud de aplicaciones

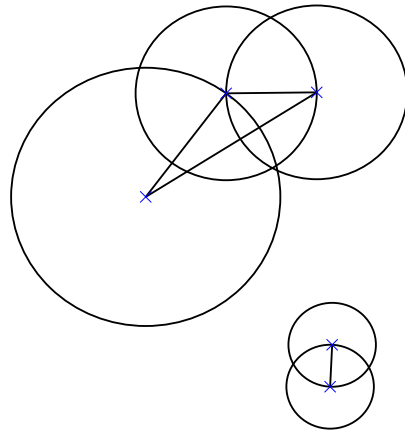


Figura 6.1: Un grafo de esferas de influencia. Hay una arista entre dos puntos solo si sus esferas de influencia intersecan. Estas esferas están centradas en los puntos y sus radios son la distancia mínima desde el punto central hasta el resto de puntos.

y áreas de investigación tales como la minería de datos (*clustering*), visión por computador (reconocimiento de objetos a partir de patrones de puntos) y gráficos por computador (definición de superficies sobre nubes de puntos). En este caso, para calcular el GEI de un determinado mapa se ha seguido el siguiente procedimiento:

1. Para cada planeta p_i situado en las coordenadas (x_i, y_i) , se calcula la distancia d_i al planeta más cercano como $d_i = \min_j \{ \|(x_i - x_j, y_i - y_j)\| \mid j \neq i \}$ donde $\|v\|$ es la norma Euclídea del vector v .
2. Para cada planeta p_i se dibuja una circunferencia con centro en (x_i, y_i) y radio d_i .
3. Se construye el grafo de esferas de influencia $G(V, E)$ donde $V = \{p_1, \dots, p_{n_p}\}$ y $(u, v) \in E$ si y solo si las circunferencias con centros en u y v intersecan.

Una vez que se ha construido el GEI, se calculan las siguientes medidas para caracterizar los mapas:

- Número de componentes conectadas (sub-grafos en los cuales cualquier par de vértices están conectados por caminos).
- Grado medio de los nodos (número de aristas incidentes al nodo).

| | |
|---|--|
| <i>Número de estimadores</i> | 50 |
| <i>Medida de calidad</i> | Impureza Gini |
| <i>Máximo de atributos para partición</i> | \sqrt{n} , n = número de atributos |
| <i>Mínimo de muestras para partición</i> | 2 |

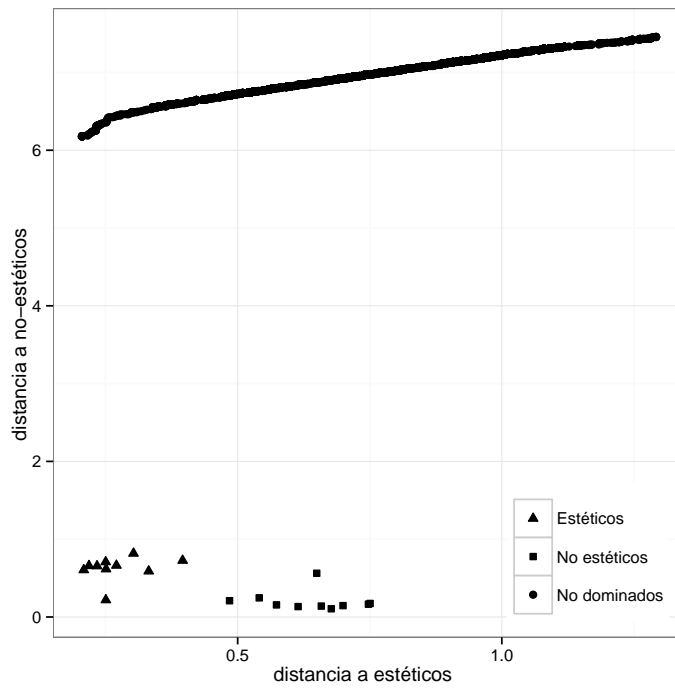
Tabla 6.1: Parámetros del clasificador de bosques aleatorios

- Densidad del grafo, es decir, la proporción entre el número de aristas y nodos. En concreto, para los grafos no dirigidos $\delta = 2m/(n_p(n_p - 1))$, donde n_p y m son el número de nodos (planetas) y vértices, respectivamente.
- Coeficiente medio de *clustering*. El coeficiente de *clustering* c_i de un nodo cuantifica cómo de interconectado está con sus vecinos. Matemáticamente, $c_i = 2E_i/[k_i(k_i - 1)]$, donde E_i es el número de aristas que conectan dos vecinos cualesquiera del nodo i mientras que k_i es el grado del nodo.
- Coeficiente de correlación de Pearson entre el tamaño de los nodos y su centralidad por interrelación. Esto último es una medida de la importancia de un nodo como paso intermedio en un grafo, y se calcula como la fracción media de caminos más cortos entre cualquier par de vértices que pasan a través del nodo en cuestión.
- Coeficiente de correlación de Pearson entre el tamaño de los nodos y su grado.
- Asortatividad por tamaño, o lo que es lo mismo, coeficiente de correlación de Pearson entre el tamaño de los nodos conectados en el grafo.

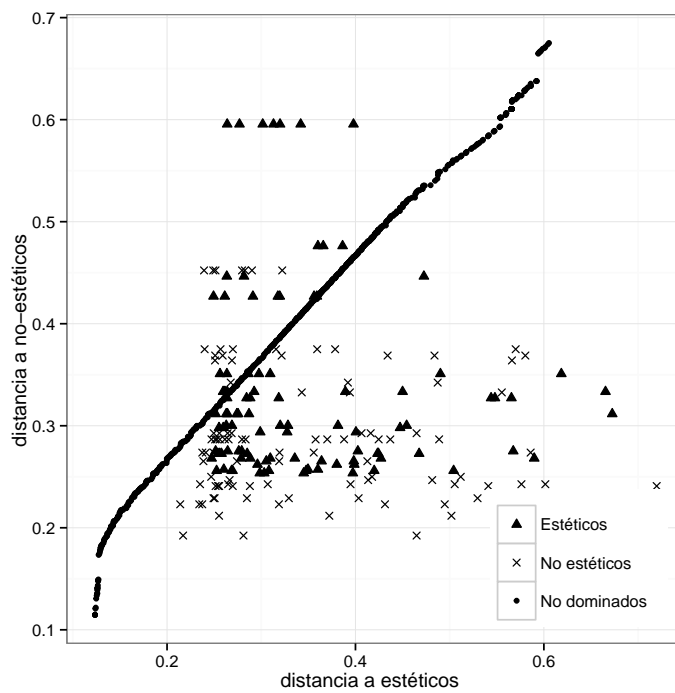
El número de componentes conectadas da una medida sobre el nivel de agrupación de los planetas, concretamente el número de grupos interconectados de planetas, mientras que el coeficiente medio de *clustering* define la forma del *clúster*. El grado medio de los nodos así como la densidad del grafo cuantifica cómo de regular es la distribución de los planetas, ya que de acuerdo con la definición de un GEI, si los planetas están muy interconectados entonces las distancias entre ellos son bastante similares. Por último, los coeficientes de correlación agregan el tamaño de los planetas al proceso de caracterización de los mapas, estableciendo relaciones entre este parámetro y otras características de los planetas tales como la centralidad por interrelación, el grado del nodo y el tamaño de los planetas vecinos.

Para reducir el número de medidas y elegir un subconjunto representativo de las mismas se ha diseñado un clasificador de bosques aleatorios [16] con 50 estimadores cada uno y muestreo *bootstrapping*

Subconjunto
representativo de las
medidas

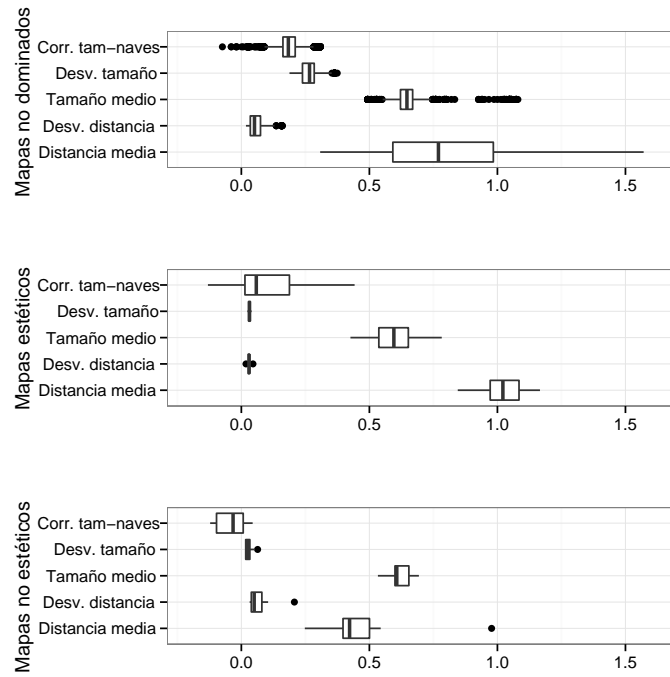


(a)

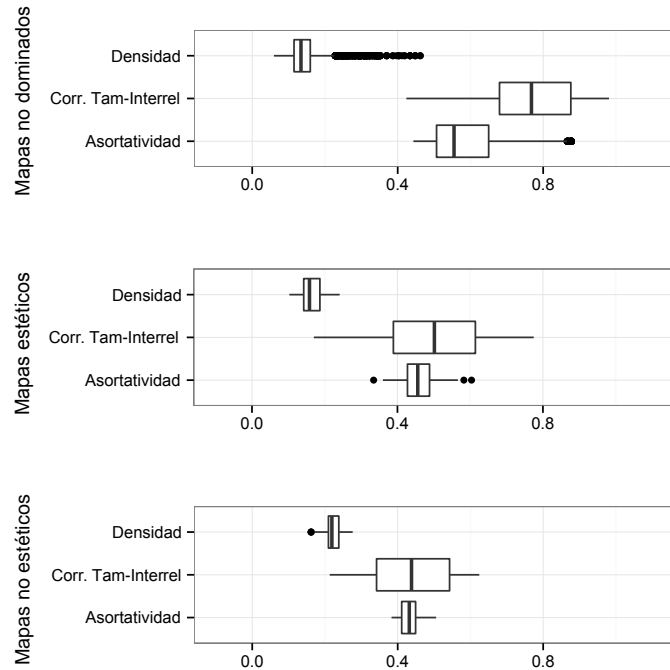


(b)

Figura 6.2: Conjunto acumulado de soluciones no dominadas (círculo) y mapas estéticos (triángulo) y no estéticos (cuadrado) pertenecientes a los conjuntos de entrenamiento tanto para el enfoque geométrico (a) como el topológico (b)



(a)



(b)

Figura 6.3: Variables de caracterización para los mapas no dominados y entrenamiento, tanto para el enfoque geométrico (a) como para el topológico (b)

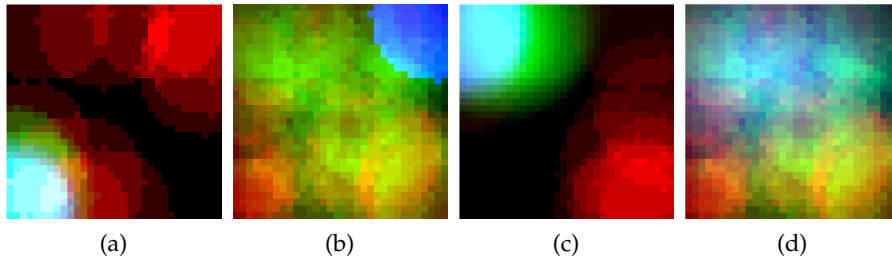


Figura 6.4: Distribución de los mapas sobre el mapa auto-organizado tanto para el enfoque geométrico (a) como el topológico (b). El rojo para los no estéticos, el verde para los estéticos y el azul para los no dominados. (c) y (d) muestran las soluciones del enfoque topológico proyectadas sobre el mapa auto-organizado geométrico y viceversa.

para cada una de las posibles combinaciones de las medidas anteriores (la Tabla 6.1 resume los parámetros utilizados). Éste método de clasificación construye un conjunto de árboles de decisión seleccionando grupos de atributos de forma aleatoria, mitigando el elevado sobreajuste que se suele dar en la construcción de los árboles de decisión de forma individual. Se ha evaluado el rendimiento de los clasificadores usando validación cruzada *Leave-one-out*. Atendiendo al área bajo la curva ROC de las diferentes combinaciones de variables se decide escoger las siguientes variables como caracterización *topológica* de los mapas: densidad del grafo (δ), correlación entre tamaño de nodo e interrelación (ρ_{SB}), y la asortatividad por tamaño (ρ_{Size}), con un valor de área bajo curva ROC de $AUC = 0.9916$. La curva ROC, del inglés *Receiver Operating Characteristic*, ilustra el rendimiento de un clasificador binario en función del umbral de discriminación, es decir, el valor de salida del clasificador que establece la categoría (si la salida del clasificador es menor o mayor al umbral, la muestra pertenece a una categoría u otra, respectivamente).

De la misma forma que con las medidas *geométricas*, se caracteriza cada mapa según la tupla $\langle \delta, \rho_{SB}, \rho_{Size} \rangle$ y, por tanto, la distancia Euclídea entre estas tuplas establece la similitud entre los mapas que representan.

6.3 RESULTADOS Y CONTRIBUCIONES

Para la generación de mapas estéticos se ha seguido un enfoque multiobjetivo similar al utilizado en el Capítulo 5, es decir, una estrategia de evolución auto-adaptativa ($\mu + \lambda$) con $\mu = 10$ y $\lambda = 100$ multiobjetivo NSGA-II con dos objetivos: reducir e incrementar, respectivamente, la distancia entre los mapas generados y los que pertenecen al conjunto de entrenamiento que se consideraron estéticos y no-estéticos. Al establecer estos objetivos se evita que los mapas

Generación de los mapas estéticos

generados sean prácticamente similares a los del conjunto de entrenamiento estético, ya que además de reducir la distancia con ellos, se fomenta el distanciamiento a los no estéticos. En caso contrario, el generador alcanzaría el valor de aptitud óptimo copiando exactamente estos mapas estéticos de entrenamiento. Los detalles del generador pueden ser consultados en el Capítulo 4.

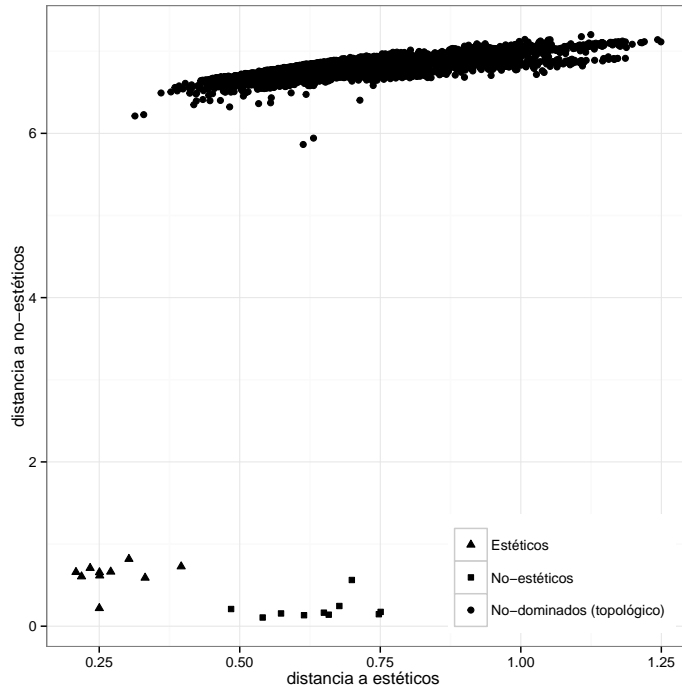
Se han realizado dos conjuntos de 20 ejecuciones del algoritmo en total durante 100 generaciones, uno por cada forma de caracterización estética de los mapas, es decir, usando medidas *geométricas* y *topológicas*. Además se ha calculado el conjunto de soluciones no dominadas a partir de los resultados de cada una de las ejecuciones individuales.

Como se puede observar en la Figura 6.2, existe una relación lineal entre la distancia a los mapas estéticos y a los no estéticos por parte de los mapas solución que están en el conjunto de soluciones no dominadas, con independencia del enfoque (*geométrico* o *topológico*) que se emplee. Esto da una idea de la densidad del espacio de búsqueda y la dificultad de reducir la distancia a los mapas estéticos al mismo tiempo que se incrementa hacia los no estéticos.

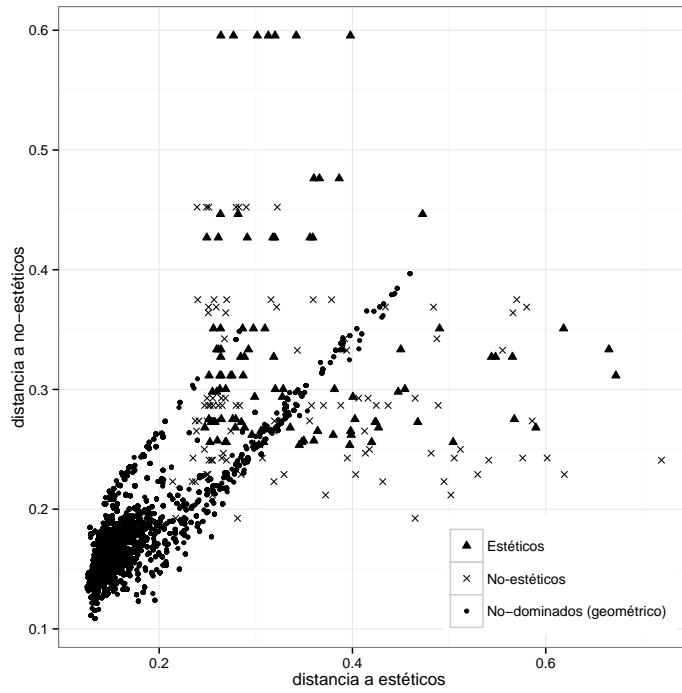
Por su parte, la Figura 6.3 muestra cómo se distribuyen los valores de las diferentes medidas que se han utilizado para la caracterización de los mapas. Se puede observar que algunas de las medidas son similares tanto en los mapas estéticos como en los que no lo son, tales como la desviación típica de la distancia σ_d , la desviación del número inicial de naves σ_s y la densidad δ . Sin embargo, otras medidas como la correlación entre el tamaño y la interrelación ρ_{SB} y la asortatividad por tamaño ρ_{size} muestran un valor más alto en el caso de los mapas no dominados, lo que explicaría la elevada distancia entre muchas de las soluciones y los mapas de referencia, como se observa en la Figura 6.2. Otra observación interesante son los valores medios de la distancia μ_d tan repartidos en los mapas no dominados, lo cual puede significar que esta variable tiene un efecto dudoso sobre el valor de aptitud y, por tanto, el espacio de búsqueda para esta medida es más amplio que para el resto de variables.

Análisis cualitativo

Con el objetivo de analizar cualitativamente los mapas generados, se han creado dos mapas auto-organizados con 32×32 unidades de proceso siguiendo una distribución rectangular no toroidal, uno para cada enfoque estudiado. Un mapa auto-organizado [79] es una arquitectura de red neuronal que tiene la propiedad de representar espacialmente las características de las señales de entrada así como sus abstracciones, descubriendo relaciones no triviales a lo largo del proceso de auto-adaptación. Como se observa en la figura 6.4a, el mapa auto-organizado del enfoque *geométrico* consiguió establecer una separación entre los mapas no estéticos (zonas rojas en la parte superior derecha) y los estéticos (zonas verdes en la parte inferior izquierda). Además, los mapas generados (zonas azules) comparten la misma re-



(a)



(b)

Figura 6.5: Soluciones topológicas no dominadas sobre el espacio de aptitud de las soluciones geométricas (a) y a la inversa (b).

gión que los mapas estéticos, por lo que deberían considerarse como estéticos también. Con respecto al enfoque *topológico*, el mapa auto-organizado no fue capaz de establecer una distinción clara entre los mapas que son estéticos y los que no, lo cual se observa en la figura 6.4b con el solapamiento de las zonas rojas y verdes.

Se ha realizado un análisis cruzado entre las soluciones no dominadas obtenidas por ambos enfoques para comparar las respectivas caracterizaciones. Como se observa en la Figura 6.5a, las soluciones tanto del enfoque *topológico* como del *geométrico* se distribuyen sobre la misma región del espacio de soluciones, manifestando una calidad similar entre ambos conjuntos de soluciones según las medidas *geométricas*. Se han obtenido unos resultados similares al proyectar las soluciones de ambos enfoques sobre el mapa auto-organizado *geométrico* (ver Figura 6.4c). Con respecto a las soluciones del enfoque *geométrico*, éstas se distribuyen principalmente sobre la parte inferior izquierda del espacio de aptitud *topológico* (Figura 6.5b), indicando que dichos mapas están más cerca de los mapas estéticos y no estéticos del conjunto de entrenamiento que muchas de las soluciones *topológicas*. Si se observa la proyección de las soluciones sobre el mapa auto-organizado *topológico* (Figura 6.4d) se puede constatar que están distribuidas de forma más dispersa que las correspondientes soluciones *topológicas*, aunque comparten la misma zona que los mapas estéticos, lo que se puede interpretar como una mayor diversidad con respecto a las soluciones del enfoque *topológico*. Con independencia de esto, ambos enfoques son capaces de generar mapas adecuados, como puede observarse en la Figura 6.6.

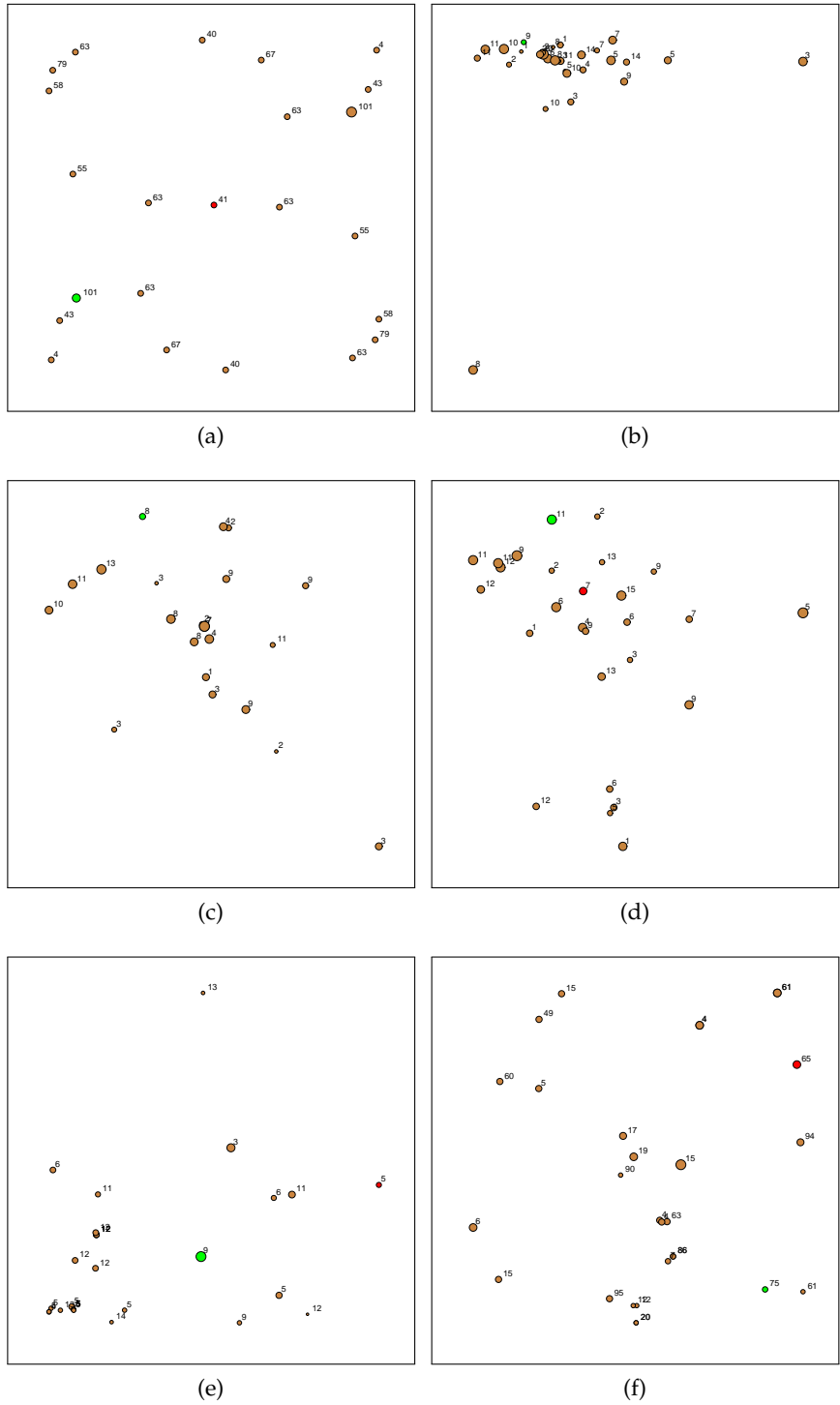


Figura 6.6: Algunos mapas de ejemplo: pertenecientes al conjunto de entrenamiento con buena (a) y mala (b) estética, generados mediante el enfoque geométrico (c) (d) y el topológico (e) (f). El color de los planetas representa quién los controla: rojo y verde para los jugadores 1 y 2 respectivamente, marrón para indicar que el planeta es neutral. El número junto a los planetas indica el total de naves al inicio de la partida en cada planeta.

CONCLUSIONES Y TRABAJO FUTURO

Este trabajo define una metodología para la generación automática de mapas para juegos de ETR a los cuales se les exige que cumplan ciertas características con el objetivo de incrementar la satisfacción del jugador. En concreto, los objetivos estudiados han sido el equilibrio de los mapas de forma que éstos no otorgan ventaja a ninguno de los jugadores sobre el resto, el dinamismo de las partidas para que éstas sean interesantes y divertidas y eviten que los jugadores pierdan interés por el juego, y la estética de los mapas, de forma que no produzcan rechazo o desconcierto a los jugadores. Aunque el juego utilizado para la experimentación ha sido el *Planet Wars*, es posible extrapolar la metodología a otros juegos de ETR con muy poco esfuerzo, ya que se han empleado características comunes a este tipo de juego, tales como la distribución de los recursos a lo largo del mapa del juego o el combate entre unidades, por nombrar algún ejemplo.

Durante el desarrollo de esta tesis se ha comprobado la viabilidad de usar algoritmos evolutivos y, en particular, estrategias de evolución como núcleo de un generador automático de mapas para juegos de ETR. Se ha comprobado la versatilidad del generador creando mapas equilibrados, dinámicos y estéticos.

El desarrollo de este trabajo ha seguido un proceso secuencial que comenzó con un estudio del estado de la cuestión y un análisis de la red de co-autoría de la comunidad de investigación de IC y videojuegos. Este análisis ha arrojado datos interesantes relacionados con las dinámicas que rigen la red. Una conclusión obtenida del estudio es que este campo de investigación es activo y vibrante, como se observa en el alto número de artículos enmarcados en este campo, el cual ha ido creciendo desde el 2005, y al crecimiento estable en el número de autores, sobre todo a partir del año 2000. Aunque se ha detectado un flujo constante de autores que abandonan la comunidad, lo cual podría indicar una ligera desaceleración de la misma, la tendencia real de la volatilidad es muy parecida a la de nuevos autores, con lo cual el crecimiento es prácticamente constante. Dicho crecimiento se rige por lo que se denomina vinculación preferente con una tasa sub-lineal debida a que la red se encuentra aún descentralizada debido a su nivel de desarrollo. El hecho de existir vinculación preferente significa que los nuevos autores entran en la comunidad generalmente colaborando con autores ya establecidos quienes a su vez tienen un elevado número de colaboradores. Además, se ha observado un grado significativo en la fidelidad de las colaboraciones. Analizando el tamaño de la componente gigante de la red, se ha llegado a la conclusión de

que ésta no está completamente formada aun, sino que se encuentra en plena fase de desarrollo aumentando su cohesión e incorporando nuevos enlaces. Del análisis de las comunidades formadas en el campo se obtiene que su crecimiento se ve reflejado en el decremento de la centralización del componente gigante y que la entrada de comunidades más pequeñas en dicha componente gigante indica la poca madurez de la red. Con respecto a los principales temas de investigación tratados por la comunidad obtenidos mediante el análisis de los títulos de los artículos se han obtenido 6 temas muy activos en la comunidad: tres áreas relacionadas directamente con videojuegos concretos que son el juego *Super Mario* (nodo aislado), *Pac-Man* (en el contexto de aprendizaje evolutivo y por diferencia temporal) y juegos de conducción (*TORCS* y técnicas como los algoritmos genéticos), y otras dos áreas aplicables a todos los géneros que son la generación automática de contenido y la interacción y el modelado del jugador.

Los trabajos relacionados con la GAC para videojuegos que se pueden encontrar en la literatura se centran principalmente en que el contenido generado sea válido, es decir, encaje correctamente con las mecánicas de juego y cumpla con sus reglas, dejando de lado otras propiedades del contenido que pueden suponer un incremento en la satisfacción del jugador. Tras plantear un generador automático de mapas basado en algoritmos evolutivos, se ha presentado el concepto de equilibrio entre jugadores y se ha diseñado un método de creación para generar mapas que presenten esta característica, es decir, el mapa no de ventaja a alguno de los jugadores sobre el resto. Se ha comprobado que el método es capaz de generar mapas equilibrados completamente jugables ya que no incumplen ninguna regla del *Planet Wars*. Sin embargo, se ha detectado que en muchos casos el equilibrio obtenido en las partidas viene propiciado por la pasividad total de los jugadores, que pasan la partida sin hacer nada. Esto puede llevar al aburrimiento de los jugadores, por lo que se ha planteado una nueva característica, el dinamismo, para evitar esto.

Se han empleado dos enfoques diferentes a la hora de buscar el dinamismo en los mapas generados, uno basado en los recursos de los jugadores durante la partida y otro que se ha basado en las batallas y confrontaciones que ha habido entre ellos. Los dos enfoques han resultado ser ortogonales y, además, se ha detectado una correlación entre el dinamismo basado en la confrontación y el equilibrio. Es por esto por lo que se ha llevado a cabo un análisis conjunto de ambas medidas, el cual ha arrojado el siguiente resultado: la definición de dinamismo usada incluye implícitamente un componente de equilibrio, ya que si un mapa es extremadamente desequilibrado la partida acaba pronto, influyendo en la medida de dinamismo (la partida es poco dinámica). La conclusión obtenida es que un nivel medio de dinamismo es compatible con el equilibrio.

Debido a la interrelación descubierta entre ambas medidas, se ha optado por utilizar un enfoque multi-objetivo a la hora de generar los mapas, intentando conseguir de esta forma que los mapas sean equilibrados y dinámicos. A la luz de los resultados obtenidos se han descubierto nuevas relaciones entre ambas medidas de manera que conforme aumenta el equilibrio, el dinamismo sufre un leve descenso, acabando con un descenso brusco cuando se alcanza el límite superior de equilibrio. Con respecto a las características de los mapas generados, el dinamismo parece estar relacionado con mapas formados por un elevado número de planetas cuyos tamaños están positivamente correlacionados con el número inicial de naves en ellos y que se distribuyen de forma dispersa. Otra de las características observadas en los mapas generados es la falta de estética de los mismos, con planetas agrupados muy juntos en grupos que están muy distantes unos de otros. Dada esta situación, se ha planteado un nuevo objetivo a conseguir y no es otro que mejorar la estética de los mapas que se generen.

Al igual que con el equilibrio y el dinamismo, se ha abordado el problema de mejorar la estética de los mapas siguiendo dos enfoques, uno geométrico que se basan en la geometría espacial del mapa (coordenadas y distancias), y otro topológico que se basan en propiedades cualitativas de los mapas que no se ven afectadas por transformaciones geométricas simples y que han sido obtenidas a partir del grafo de esferas de influencia. Se ha seguido un enfoque multiobjetivo que intenta maximizar la diferencia con un conjunto de mapas no estéticos a la vez que se reduce con otro conjunto de mapas considerados estéticos (ambos conjuntos han sido etiquetados por un experto). Los resultados indican que existe una relación lineal entre la diferencia con los mapas estéticos y con los que no lo son, lo que da una idea de la densidad del espacio de búsqueda. Con respecto a las medidas utilizadas, se observa que hay algunas de ellas que tienen una mayor influencia sobre el valor de aptitud que el resto de variables. Por último, tras realizar un análisis cruzado de las soluciones obtenidas con ambos enfoques se observa que ambos enfoques son capaces de generar mapas adecuados, aunque hay que destacar que existe una mayor diversidad en las soluciones del enfoque geométrico que en las del topológico, además de que las soluciones geométricas obtenidas quedan más cerca del conjunto de mapas estéticos que las propias soluciones topológicas según la medida de aptitud de éstas últimas.

7.1 TRABAJO FUTURO

Los resultados obtenidos en esta tesis forman el punto de inicio de una nueva línea de investigación en generación automática de contenido con diversas propiedades enfocadas al incremento de la satisfacción del jugador.

Una primera línea de trabajo futuro es la incorporación del humano en el proceso de evaluación de los mapas generados, dando lugar a la denominada evolución interactiva [8, 26]. Para ello se podría construir una aplicación web en la que distintas personas juegan los mapas obtenidos por el generador y evalúan su experiencia de juego, influyendo en el valor de aptitud de los mapas y guiando el proceso evolutivo de los mismos, aunque también habría que tratar el problema de la fatiga del usuario.

Por otra parte, se ha comprobado que el grafo de esferas de influencia es una herramienta muy útil para caracterizar los mapas generados. Sería interesante invertir el proceso de forma que lo que se evoluciona son los grafos de esferas de influencia, que posteriormente se transforman en mapas. Para ello sería necesario estudiar si dicha transformación es posible.

Por último, aunque las estrategias de evolución han funcionado correctamente, sería interesante estudiar el impacto en el rendimiento del generador que tendría usar otras vertientes de la computación evolutiva como, por ejemplo, la programación genética, y la selección de otros operadores de variación o un refinamiento de los parámetros de los algoritmos.

Parte III
APÉNDICES

APÉNDICES

A.1 CONFERENCIAS, REVISTAS Y PALABRAS CLAVE PARA EL ANÁLISIS DE LA RED

- **Inteligencia Artificial y Computacional en videojuegos:**
 - IEEE Trans. on Computational Intelligence and AI in games (TCIAIG)
 - Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE)
 - IEEE congress on Computational Intelligence and Games (CIG)
 - Agents for Games and Simulations (AGS)
 - Advanced Intelligent Paradigms in Computer Games
- **Inteligencia artificial y computación evolutiva:**
 - IEEE congress on evolutionary computation (CEC)
 - Genetic and Evolutionary Computation Conference (GECCO)
 - Parallel Problem solving from Nature (PPSN)
 - National Conference on Artificial Intelligence
 - Applications of Evolutionary Computation (EvoApplications)
 - International Conference on Artificial Neural Networks (ICANN)
 - Canadian Conference on Artificial Intelligence
 - Artificial Intelligence and Applications (AIA)
 - Artificial Intelligence Applications and Innovations (AIAI)
 - Artificial Intelligence and Computational Intelligence (AICI)
 - Artificial Intelligence: Methodology, Systems, Applications (AIMSA)
 - Artificial Intelligence and Simulations (AIS)
 - Computational Intelligence
 - European Conference on Artificial Intelligence (ECAI)
 - International Conference on Agents and Artificial Intelligence (ICAART)

- International Conference on Artificial Intelligence (IC-AI)
 - International Conference on Artificial Intelligence and Soft Computing (ICAISC)
 - International Conference on Evolutionary Computation (ICEC)
 - International Conference on Genetic Algorithms (ICGA)
 - International Conference on Genetic and Evolutionary Computing (ICGEC)
 - International Conference on Intelligent Computing (ICIC)
 - International Joint Conference on Artificial Intelligence (IJ-CAI)
 - International Joint Conference on Computational Intelligence (IJCCI)
 - International Joint Conference on Neural Network(IJCNN)
 - International Symposium on Computational Intelligence and Design
 - International Symposium on Neural Networks
 - International Work-Conference on Artificial and Natural Neural Networks (IWANN)
 - IITA International Joint Conference on Artificial Intelligence
 - Modeling Decisions for Artificial Intelligence
 - IEEE Transactions on Evolutionary Computation (TEC)
 - International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC)
 - International Conference on Adaptive and Natural Computing Algorithms (ICANNGA)
 - Artificial Life (ALIFE)
 - Simulation of Adaptive Behavior (SAB)
 - International Conference on Artificial Neural Networks (ICANN)
 - Simulated Evolution and Learning (SEAL)
 - European Conference on Advances in Artificial Life (ECAL)
 - International conference on Swarm Intelligence (ICSI)
 - Hybrid Intelligent Systems (HIS)
 - European Symposium on Artificial Neural Networks (ESANN)
 - IEEE Transactions on Neural Networks
 - IEEE Transactions on Fuzzy Systems
- **Videojuegos:**
- GAME-ON Conference (GAMEON)

- Advances in Computer Games (ACG)
- Computers and Games
- Computers Games Conference (CGAMES)
- Conference of the Digital Games Research Association
- International Conference on Entertainment Computing
- Computers in Entertainment
- Advances in computer entertainment technology

■ **Palabras clave:**

- Inteligencia Artificial y Computacional en videojuegos: sin palabras clave. Todos los artículos seleccionados.
- Inteligencia artificial y computación evolutiva: game/s, puzzle, player.
- Videojuegos: fuzzy, evol*, genetic, swarm, agent, local search, neural, ant.
- Otros: todas las combinaciones de {fuzzy, evol*, genetic} y {game/s, puzzle, player}

REFERENCIAS

- [1] *i3D '15: Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*. New York, NY, USA: ACM, 2015.
- [2] D. W. Aha, M. Molineaux, y M. J. V. Ponsen, "Learning to win: Case-based plan selection in a real-time strategy game," en *International Conference on Case-Based Reasoning*, ser. Lecture Notes in Computer Science, H. Muñoz-Avila y F. Ricci, Eds., vol. 3620. Springer, 2005, pp. 5–20.
- [3] V. Alcázar, D. Borrajo, y C. Linares López, "Modelling a RTS planning domain with cost conversion and rewards," en *Artificial Intelligence in Games. Workshop of the Eighteenth European Conference on Artificial Intelligence*, Patras, Greece, 2008, pp. 50–54.
- [4] J. M. L. Asensio, J. Peralta, R. Arrabales, M. G. Bedia, P. Cortez, y A. L. Peña, "Artificial intelligence approaches for the generation and assessment of believable human-like behaviour in virtual characters," *Expert Systems with Applications*, vol. 41, num. 16, pp. 7281–7290, 2014.
- [5] D. Ashlock, S. Gent, y K. Bryden, "Embryogenesis of artificial landscapes," en *Design by Evolution*, ser. Natural Computing Series, P. Hingston, L. Barone, y Z. Michalewicz, Eds. Springer Berlin Heidelberg, 2008, pp. 203–221.
- [6] P. Avery y S. J. Louis, "Coevolving team tactics for a real-time strategy game," en *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–8.
- [7] T. Back, D. B. Fogel, y Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*. Bristol, UK, UK: IOP Publishing Ltd., 1997.
- [8] A. Badillo, J. Ruiz, C. Cotta, y A. Fernández-Leiva, "On user-centric memetic algorithms," *Soft Computing*, vol. 17, num. 2, pp. 285–300, 2013.
- [9] A.-L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, y T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A: Statistical Mechanics and its Applications*, vol. 311, num. 3-4, pp. 590–614, 2002.
- [10] A.-L. Barabási y R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, num. 5439, pp. 509–512, 1999.

- [11] Blizzard Entertainment. (1994) Warcraft: Orcs & Humans. Blizzard Entertainment.
- [12] Blizzard Entertainment. (2010) StarCraft II: Wings of Liberty. Blizzard Entertainment.
- [13] S. P. Borgatti y M. G. Everett, "A Graph-theoretic perspective on centrality," *Social Networks*, vol. 28, num. 4, pp. 466–484, 2006.
- [14] A. Botea, B. Bouzy, M. Buro, C. Bauckhage, y D. Nau, "Path-finding in Games," en *Artificial and Computational Intelligence in Games*, ser. Dagstuhl Follow-Ups, S. M. Lucas, M. Mateas, M. Preuss, P. Spronck, y J. Togelius, Eds. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013, vol. 6, pp. 21–31.
- [15] U. Brandes, "A faster algorithm for betweenness centrality," *The Journal of Mathematical Sociology*, vol. 25, num. 2, pp. 163–177, 2001.
- [16] L. Breiman, "Random forests," *Machine Learning*, vol. 45, num. 1, pp. 5–32, 2001.
- [17] H. J. Bremermann, "Optimization through evolution and recombination," *Self-Organizing Systems*, pp. 93–106, 1962.
- [18] M. Buro, "ORTS: A hack-free RTS game environment," en *Computers and Games*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, vol. 2883, pp. 280–291.
- [19] M. Buro y T. M. Furtak, "RTS games and real-time AI research," en *Proceedings of the Behavior Representation in Modeling and Simulation Conference (BRIMS)*, 2004, pp. 51–58.
- [20] L. Cardamone, P. L. Lanzi, y D. Loiacono, "Trackgen: An interactive track generator for TORCS and Speed-Dreams," *Applied Soft Computing*, vol. 28, num. 0, pp. 550–558, 2015.
- [21] D. Cheng y R. Thawonmas, "Case-based plan recognition for real-time strategy games," en *GameOn Conference*, A. El-Rhalibi y D. van Welden, Eds. EUROSIS, 2004, pp. 36–40.
- [22] C. A. C. Coello, D. A. Van Veldhuizen, y G. B. Lamont, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2002, vol. 242.
- [23] K. Collins, "An introduction to procedural music in video games," *Contemporary Music Review*, vol. 28, num. 1, pp. 5–15, 2009.

- [24] K. Compton y M. Mateas, "Procedural level design for platform games," en *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment International Conference (AIIDE)*, 2006, pp. 109–111.
- [25] C. Cotta y J. J. Merelo, "The complex network of EC authors," *ACM SIGEVOlution*, vol. 1, num. 2, pp. 2–9, 2006.
- [26] C. Cotta y A. J. F. Leiva, "Bio-inspired combinatorial optimization: Notes on reactive and proactive interaction," en *Advances in Computational Intelligence - 11th International Work-Conference on Artificial Neural Networks, IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011, Proceedings, Part II*, 2011, pp. 348–355.
- [27] C. Cotta y J. J. Merelo, "Where is Evolutionary Computation going? a temporal analysis of the EC community," *Genetic Programming and Evolvable Machines*, vol. 8, num. 3, pp. 239–253, 2007.
- [28] G. Csárdi y T. Nepusz, "The igraph software package for complex network research," *InterJournal, Complex Systems*, vol. 1695, num. 5, pp. 1–9, 2006.
- [29] A. Dahlbom y L. Niklasson, "Goal-directed hierarchical dynamic scripting for RTS games," en *Artificial Intelligence and Interactive Digital Entertainment*, J. E. Laird y J. Schaeffer, Eds. The AAAI Press, 2006, pp. 21–28.
- [30] K. Deb, S. Agrawal, A. Pratap, y T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," en *Parallel Problem Solving from Nature PPSN VI*, ser. Lectures Notes in Computer Science. Springer Berlin Heidelberg, 2000, vol. 1917, pp. 849–858.
- [31] E. W. Dereszynski, J. Hostetler, A. Fern, T. G. Dietterich, T.-T. Hoang, y M. Udarbe, "Learning probabilistic behavior models in real-time strategy games," en *Artificial Intelligence and Interactive Digital Entertainment Conference*. The AAAI Press, 2011, pp. 20–25.
- [32] J. Doran y I. Parberry, "Controlled procedural terrain generation using software agents," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, num. 2, pp. 111–119, 2010.
- [33] M. Dorigo, M. Birattari, y T. Stutzle, "Ant colony optimization," *Computational Intelligence Magazine, IEEE*, vol. 1, num. 4, pp. 28–39, 2006.
- [34] S. N. Dorogovtsev y J. F. F. Mendes, "Scaling behaviour of developing and decaying networks," *Europhysics Letters*, vol. 52, pp. 33–39, 2000.

- [35] A. E. Eiben y J. E. Smith, *Introduction to Evolutionary Computing*, ser. Natural Computing, G. Rozenberg y A. Eiben, Eds. Springer-Verlag Berlin Heidelberg, 2003.
- [36] Ensemble Studios. (1997) Age of Empires. Microsoft Studios.
- [37] Entertainment Software Association, "Essential facts about the computer and video game industry," 2015.
- [38] P. Erdős y A. Rényi, "On the evolution of random graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, pp. 17–61, 1960.
- [39] A. Fernández-Ares, A. Mora, J. Merelo, P. García-Sánchez, y C. Fernandes, "Optimizing player behavior in a real-time strategy game using evolutionary algorithms," en *Evolutionary Computation (CEC), 2011 IEEE Congress on*, 2011, pp. 2017–2024.
- [40] A. Fernández-Ares, A. Mora, J. Merelo, P. García-Sánchez, y C. Fernandes, "Optimizing strategy parameters in a game bot," en *Advances in Computational Intelligence*, ser. Lecture Notes in Computer Science, J. Cabestany, I. Rojas, y G. Joya, Eds. Springer Berlin Heidelberg, 2011, vol. 6692, pp. 325–332.
- [41] A. Fernández-Ares, P. García-Sánchez, A. Mora, y J. Merelo, "Adaptive bots for real-time strategy games via map characterization," en *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, 2012, pp. 417–423.
- [42] A. Fernández-Ares, A. Mora, M. García-Arenas, J. Guervós, P. García-Sánchez, y P. Castillo, "Co-evolutionary optimization of autonomous agents in a real-time strategy game," en *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science, A. I. Esparcia-Alcázar y A. M. Mora, Eds. Springer Berlin Heidelberg, 2014, vol. 8602, pp. 374–385.
- [43] L. J. Fogel, "On the organization of intellect," Tesis Doctoral, University of California, 1964.
- [44] L. J. Fogel, A. J. Owens, y M. J. Walsh, *Artificial Intelligence through Simulated Evolution*. New York, USA: John Wiley, 1966.
- [45] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, y C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.
- [46] M. Frade, F. Fernández de Vega, y C. Cotta, "Modelling video games' landscapes by means of genetic terrain programming - a new approach for improving users' experience," en *Applications of Evolutionary Computing*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, vol. 4974, pp. 485–490.

- [47] M. Frade, F. Fernández de Vega, y C. Cotta, "Breeding terrains with genetic terrain programming: The evolution of terrain generators," *International Journal of Computer Games Technology*, vol. 2009, pp. 1–8, 2009.
- [48] M. Frade, F. Fernández de Vega, y C. Cotta, "Evolution of artificial terrains for video games based on accessibility," en *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6024, pp. 90–99.
- [49] M. Frade, F. Fernández de Vega, y C. Cotta, "Evolution of artificial terrains for video games based on obstacles edge length," en *Evolutionary Computation (CEC), 2010 IEEE Congress on*, 2010, pp. 1–8.
- [50] A. Fraser, "Simulation of genetic systems by automatic digital computers," *Aust. J. Biol. Sci.*, vol. 10, pp. 484–499, 1957.
- [51] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, num. 3, pp. 215–239, 1978.
- [52] R. Friedberg, "A learning machine: Part I," *IBM Journal of Research and Development*, vol. 2, num. 1, pp. 2–13, 1958.
- [53] R. Friedberg, B. Dunham, y J. North, "A learning machine: Part II," *IBM Journal of Research and Development*, vol. 3, num. 3, pp. 282–287, 1959.
- [54] Gas Powered Games. (2010) *Supreme Commander 2*. Square Enix.
- [55] Gearbox Software. (2009) *Borderlands*. 2K Games.
- [56] M. Girvan y M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, num. 12, pp. 7821–7826, 2002.
- [57] D. E. Goldberg y R. Lingle, Jr., "Alleles, Loci, and the Traveling Salesman Problem," en *Proceedings of the 1st International Conference on Genetic Algorithms*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1985, pp. 154–159.
- [58] W. Gong, E.-P. Lim, P. Achananuparp, F. Zhu, D. Lo, y F. Chua, "In-game action list segmentation and labeling in real-time strategy games," en *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, Sept 2012, pp. 147–154.
- [59] J. W. Grossman, "The evolution of the mathematical research collaboration graph," *Congressus Numerantium*, pp. 201–212, 2002.

- [60] M. Gunnerud, "A CBR/RL system for learning micromanagement in real-time strategy games," Trabajo fin de Máster, Norwegian University of Science and Technology, 2009.
- [61] J. Hagelbäck, "Potential-field based navigation in StarCraft," en *Computational Intelligence and Games*. IEEE, 2012, pp. 388–393.
- [62] J. Hagelbäck y S. J. Johansson, "The rise of potential fields in real time strategy bots," en *Artificial Intelligence and Interactive Digital Entertainment Conference*, C. Darken y M. Mateas, Eds. The AAAI Press, 2008, pp. 42–47.
- [63] J. Hagelbäck y S. J. Johansson, "Using multi-agent potential fields in real-time strategy games," en *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, ser. AAMAS '08, 2008, pp. 631–638.
- [64] J. Hagelbäck y S. J. Johansson, "A multi-agent potential field-based bot for a full RTS game scenario," en *Artificial Intelligence and Interactive Digital Entertainment Conference*, C. Darken y G. M. Youngblood, Eds. The AAAI Press, 2009, pp. 28–33.
- [65] J. Hamari, J. Koivisto, y H. Sarsa, "Does gamification work?—a literature review of empirical studies on gamification," en *System Sciences (HICSS), 2014 47th Hawaii International Conference on*. IEEE, 2014, pp. 3025–3034.
- [66] P. Hart, N. Nilsson, y B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, num. 2, pp. 100–107, 1968.
- [67] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA, USA: MIT Press, 1992.
- [68] J. H. Holland, "Building blocks, cohort genetic algorithms, and hyperplane-defined functions," *Evolutionary computation*, vol. 8, num. 4, pp. 373–391, 2000.
- [69] J. H. Holland, "Nonlinear environments permitting efficient adaptation," *Computer and Information Sciences II*, 1967.
- [70] D. B. Horn, T. A. Finholt, J. P. Birnholtz, D. Motwani, y S. Jayaraman, "Six degrees of jonathan grudin: a social network analysis of the evolution and impact of CSCW research," en *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, ser. CSCW '04. New York, NY, USA: ACM, 2004, pp. 582–591.
- [71] H. Hou, H. Kretschmer, y Z. Liu, "The structure of scientific collaboration networks in Scientometrics," *Scientometrics*, vol. 75, num. 2, pp. 189–202, 2008.

- [72] J.-L. Hsieh y C.-T. Sun, "Building a player strategy model by analyzing replays of real-time strategy games," en *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, June 2008, pp. 3106–3111.
- [73] S.-H. Jang, J. Yoon, y S.-B. Cho, "Optimal strategy selection of non-player character on real time strategy game using a speciated evolutionary algorithm," en *IEEE Symposium on Computational Intelligence and Games*, P. L. Lanzi, Ed. IEEE, 2009, pp. 75–79.
- [74] M. Jennings-Teats, G. Smith, y N. Wardrip-Fruin, "Polymorph: A model for dynamic level generation," en *Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2010, pp. 138–143.
- [75] N. Justesen, B. Tillman, J. Togelius, y S. Risi, "Script- and cluster-based UCT for StarCraft," en *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, 2014, pp. 1–8.
- [76] D. Keaveney y C. O’Riordan, "Evolving robust strategies for an abstract real-time strategy game," en *IEEE Symposium on Computational Intelligence and Games*, P. L. Lanzi, Ed. IEEE, 2009, pp. 371–378.
- [77] D. Keaveney y C. O’Riordan, "Evolving coordination for real-time strategy games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, num. 2, pp. 155–167, 2011.
- [78] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, num. 5, pp. 604–632, 1999.
- [79] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, num. 9, pp. 1464–1480, 1990.
- [80] A. Kovarsky y M. Buro, "A First Look at Build-Order Optimization in Real-Time Strategy Games," en *GameOn Conference*, L. Wolf y M. Magnor, Eds. EUROSIS, 2006, pp. 18–22.
- [81] J. R. Koza, *Genetic Programming*. MIT Press, 1992.
- [82] R. Lara-Cabrera, C. Cotta, y A. Fernández-Leiva, "A self-adaptive evolutionary approach to the evolution of aesthetic maps for a rts game," en *Evolutionary Computation (CEC), 2014 IEEE Congress on*, 2014, pp. 298–304.
- [83] R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, "Procedural map generation for a RTS game," en *13th International Conference on Intelligent Games and Simulation - GAMEON 2012*, 2012, pp. 53–58.

- [84] R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, "Evolving aesthetic maps for a real time strategy game," en *I Simposio Español de Entretenimiento Digital*. Universidad Complutense de Madrid, 2013, pp. 61–71.
- [85] R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, "A procedural balanced map generator with self-adaptive complexity for the real-time strategy game Planet Wars," en *Applications of evolutionary computation*, ser. Lecture Notes in Computer Science, A. Esparcia-Alcázar, Ed. Springer Berlin Heidelberg, 2013, vol. 7835, pp. 274–283.
- [86] R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, "A review of computational intelligence in RTS games," en *Foundations of Computational Intelligence (FOCI), 2013 IEEE Symposium on*, 2013, pp. 114–121.
- [87] R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, "An analysis of the structure and evolution of the scientific collaboration network of computer intelligence in games," *Physica A: Statistical Mechanics and its Applications*, vol. 395, num. 0, pp. 523–536, 2014.
- [88] R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, "Geometrical vs topological measures for the evolution of aesthetic maps in a RTS game," *Entertainment Computing*, vol. 5, num. 4, pp. 251–258, 2014.
- [89] R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, "On balance and dynamism in procedural content generation with self-adaptive evolutionary algorithms," *Natural Computing*, vol. 13, num. 2, pp. 157–168, 2014.
- [90] R. Lara-Cabrera, C. Cotta, y A. J. Fernández-Leiva, "Using self-adaptive evolutionary algorithms to evolve dynamism-oriented maps for a real time strategy game," en *Large-Scale Scientific Computing*, ser. Lecture Notes in Computer Science, I. Lirkov, S. Margenov, y J. Waśniewski, Eds. Springer Berlin Heidelberg, 2014, vol. 8353, pp. 256–263.
- [91] R. Lara-Cabrera, M. Nogueira, C. Cotta, y A. J. Fernández-Leiva, "Game artificial intelligence: Challenges for the scientific community," en *Actas del II Congreso de la Sociedad Española para las Ciencias del Videojuego, Barcelona, España*, 2015, pp. 1–12.
- [92] R. Lara-Cabrera, M. Nogueira, C. Cotta, y A. J. Fernández-Leiva, "Optimización en videojuegos: retos para la comunidad científica," en *Actas del X Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, Mérida, España.*, F. Chávez

- de la O, R. Luque Baena, F. Luna, y F. Fernández de Vega, Eds., 2015, pp. 463–470.
- [93] R. Lara-Cabrera, M. Nogueira, C. Cotta, y A. J. Fernández-Leiva, “Procedural content generation for real-time strategy games,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 3, num. 2, pp. 40–48, 2015.
- [94] R. Li, “Mixed-integer evolution strategies for parameter optimization and their applications to medical image analysis,” Tesis Doctoral, Leiden Institute of Advanced Computer Science (LIACS), Faculty of Science, Leiden University, 2009.
- [95] A. Liapis, G. Yannakakis, y J. Togelius, “Generating map sketches for strategy games,” en *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 7835, pp. 264–273.
- [96] P. Lichocki, K. Krawiec, y W. Jaskowski, “Evolving teams of cooperating agents for real-time strategy game,” en *EvoWorkshops*, ser. Lecture Notes in Computer Science, M. Giacobini *et al.*, Eds., vol. 5484. Springer, 2009, pp. 333–342.
- [97] S. Lucas, “Computational intelligence and games: Challenges and opportunities,” *International Journal of Automation and Computing*, vol. 5, num. 1, pp. 45–57, 2008.
- [98] J. Ludwig y A. Farley, “Examining extended dynamic scripting in a tactical game framework,” en *Artificial Intelligence and Interactive Digital Entertainment*, C. Darken y G. M. Youngblood, Eds. The AAAI Press, 2009, pp. 76–81.
- [99] T. Luukkonen, O. Persson, y G. Sivertsen, “Understanding Patterns of International Scientific Collaboration,” *Science, Technology & Human Values*, vol. 17, num. 1, pp. 101–126, 1992.
- [100] T. Mahlmann, J. Togelius, y G. Yannakakis, “Modelling and evaluation of complex scenarios with the strategy game description language,” en *Computational Intelligence and Games (CIG)*, 2011 IEEE Conference on, 2011, pp. 174–181.
- [101] T. Mahlmann, J. Togelius, y G. Yannakakis, “Towards procedural strategy game generation: Evolving complementary unit types,” en *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6624, pp. 93–102.
- [102] T. Mahlmann, J. Togelius, y G. Yannakakis, “Spicing up map generation,” en *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7248, pp. 224–233.

- [103] Maxis. (2008) Spore. Electronic Arts.
- [104] S. Meyer-Nieberg y H.-G. Beyer, "Self-adaptation in evolutionary algorithms," en *Parameter setting in evolutionary algorithms*. Springer, 2007, pp. 47–75.
- [105] D. R. Michael y S. L. Chen, *Serious games: Games that educate, train, and inform*. Muska & Lipman/Premier-Trade, 2005.
- [106] C. Miles, S. Louis, N. Cole, y J. McDonnell, "Learning to play like a human: case injected genetic algorithms for strategic computer gaming," en *Congress on Evolutionary Computation*, vol. 2, 2004, pp. 1441–1448.
- [107] K. Mishra, S. Ontañón, y A. Ram, "Situation assessment for plan retrieval in real-time strategy games," en *European Conference on Advances in Case-Based Reasoning*, ser. Lecture Notes in Computer Science, K.-D. Althoff *et al.*, Eds., vol. 5239. Springer, 2008, pp. 355–369.
- [108] Mojang. (2011) Minecraft. Mojang.
- [109] M. Molineaux y D. Aha, "Defeating novel opponents in a real-time strategy game," en *International Joint Conference on Artificial Intelligence Workshop on Reasoning, Representation, and Learning in Computer Games*, D. W. Aha *et al.*, Eds. The AAAI Press, 2005, pp. 72–77.
- [110] M. Molineaux, D. W. Aha, y P. Moore, "Learning continuous action models in a real-time strategy environment," en *Florida Artificial Intelligence Research Society Conference*, D. Wilson y H. C. Lane, Eds. The AAAI Press, 2008, pp. 257–262.
- [111] A. Mora, A. Fernández-Ares, J. Merelo, P. García-Sánchez, y C. Fernandes, "Effect of noisy fitness in real-time strategy games player behaviour optimisation using evolutionary algorithms," *Journal of Computer Science and Technology*, vol. 27, num. 5, pp. 1007–1023, 2012.
- [112] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 64, num. 2, 2001.
- [113] M. E. J. Newman, "Scientific collaboration networks. I. Network construction and fundamental results," *Phys. Rev. E*, vol. 64, num. 1, p. 16131, 2001.
- [114] M. E. J. Newman, "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality," *Phys. Rev. E*, vol. 64, num. 1, p. 16132, 2001.

- [115] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, num. 2, pp. 167–256, 2003.
- [116] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 74, num. 3, 2006.
- [117] P. H. F. Ng, Y. J. Li, y S. C. K. Shiu, "Unit formation planning in RTS game by using potential field and fuzzy integral," en *Fuzzy Systems*. IEEE, 2011, pp. 178–184.
- [118] M. Nogueira, C. Cotta, y A. Fernández-Leiva, "Hall-of-fame competitive coevolutionary algorithms for optimizing opponent strategies in a new game," en *13th International Conference on Intelligent Games and Simulation. GAME-ON 2012.*, A. Fernández-Leiva, C. Cotta, y R. Lara-Cabrera, Eds. EUROSIS-ETI, 2012, pp. 61–63.
- [119] M. Nogueira, C. Cotta, y A. Fernández-Leiva, "On modeling, evaluating and increasing players' satisfaction quantitatively: Steps towards a taxonomy," en *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science, C. Di Chio, A. Agapitos, S. Cagnoni, C. Cotta, F. Fernández de Vega, G. Di Caro, R. Drechsler, A. Ekárt, A. Esparcia-Alcázar, M. Farooq, W. Langdon, J. Merelo-Guervós, M. Preuss, H. Richter, S. Silva, A. Simões, G. Squillero, E. Tarantino, A. Tettamanzi, J. Togelius, N. Urquhart, A. Uyar, y G. Yannakakis, Eds. Springer Berlin Heidelberg, 2012, vol. 7248, pp. 245–254.
- [120] M. Nogueira, C. Cotta, y A. Fernández-Leiva, "Virtual player design using self-learning via competitive coevolutionary algorithms," *Natural Computing*, vol. 13, num. 2, pp. 131–144, 2014.
- [121] I.-S. Oh, H.-C. Cho, y K.-J. Kim, "Imitation learning for combat system in RTS games with application to Starcraft," en *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, 2014, pp. 1–2.
- [122] S. Ontañón, K. Mishra, N. Sugandh, y A. Ram, "Case-based planning and execution for real-time strategy games," en *International Conference on Case-Based Reasoning*, ser. Lecture Notes in Computer Science, R. Weber y M. M. Richter, Eds., vol. 4626. Springer, 2007, pp. 164–178.
- [123] S. Ontañón, K. Mishra, N. Sugandh, y A. Ram, "Learning from demonstration and case-based planning for real-time strategy games," en *Soft Computing Applications in Industry*, ser. Studies in Fuzziness and Soft Computing, B. Prasad, Ed., vol. 226. Springer, 2008, pp. 293–310.

- [124] J. Paredis, "Coevolutionary algorithms," *Evolutionary computation*, vol. 2, pp. 224–238, 1998.
- [125] C. Pedersen, J. Togelius, y G. Yannakakis, "Modeling player experience in Super Mario Bros," en *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, 2009, pp. 132–139.
- [126] M. Perc, "Growth and structure of Slovenia's scientific collaboration network," *Journal of Informetrics*, vol. 4, pp. 475–482, 2010.
- [127] D. Plans y D. Morelli, "Experience-driven procedural music generation for games," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 4, num. 3, pp. 192–198, 2012.
- [128] M. J. V. Ponsen, H. Muñoz-Avila, P. Spronck, y D. W. Aha, "Automatically generating game tactics through evolutionary learning," *AI Magazine*, vol. 27, num. 3, pp. 75–84, 2006.
- [129] M. Preuss, N. Beume, H. Danielsiek, T. Hein, B. Naujoks, N. Piatkowski, R. Stür, A. Thom, y S. Wessing, "Towards intelligent team composition and maneuvering in real-time strategy games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, num. 2, pp. 82–98, 2010.
- [130] Re-Logic. (2011) *Terraria*. 505 Games.
- [131] I. Rechenberg, "Cybernetic solution path of an experimental problem," en *Royal Aircraft Establishment Translation No. 1122, B. F. Toms, Trans.* Farnborough Hants: Ministry of Aviation, Royal Aircraft Establishment, 1965.
- [132] G. Rudolph, "An evolutionary algorithm for integer programming," en *Parallel Problem Solving from Nature — PPSN III*, ser. Lecture Notes in Computer Science, Y. Davidor, H.-P. Schwefel, y R. Männer, Eds., vol. 866. Springer-Verlag Berlin Heidelberg, 1994, pp. 139–148.
- [133] F. Sailer, M. Buro, y M. Lanctot, "Adversarial planning through strategy simulation," en *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, 2007, pp. 80–87.
- [134] F. Schadd, S. Bakkes, y P. Spronck, "Opponent modeling in real-time strategy games," en *GameOn Conference*, M. Rocetti, Ed. EUROSIS, 2011, pp. 61–68.
- [135] H.-P. Schwefel, *Numerical Optimization of Computer Models*. New York, NY, USA: John Wiley & Sons, Inc., 1981.
- [136] J. Scott, "Social network analysis," *Sociology*, vol. 22, num. 1, pp. 109–127, February 1988.

- [137] M. Shaker, M. Sarhan, O. Al Naameh, N. Shaker, y J. Togelius, "Automatic generation and analysis of physics-based puzzle games," en *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, 2013, pp. 1–8.
- [138] N. Shaker, M. Shaker, y J. Togelius, "Evolving playable content for Cut the Rope through a simulation-based approach," en *Artificial Intelligence and Interactive Digital Entertainment*. The AAAI Press, 2013, pp. 72–78.
- [139] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, y T. Ideker, "Cytoscape: A software environment for integrated models of biomolecular interaction networks," *Genome Research*, vol. 13, num. 11, pp. 2498–2504, 2003.
- [140] M. Sharma, M. P. Holmes, J. C. Santamaría, A. Irani, C. L. Isbell Jr., y A. Ram, "Transfer learning in real-time strategy games using hybrid CBR/RL," en *International Joint Conference on Artificial Intelligence*, M. M. Veloso, Ed., 2007, pp. 1041–1046.
- [141] R. M. Smelik, K. J. D. Kraker, S. A. Groenewegen, T. Tutenel, y R. Bidarra, "A survey of procedural methods for terrain modelling," en *Proc. of the CASA Workshop on 3D Advanced Media In Gaming And Simulation (3AMIGAS)*, 2009, pp. 25–34.
- [142] G. Smith, M. Treanor, J. Whitehead, y M. Mateas, "Rhythm-based level generation for 2D platformers," en *Proceedings of the 4th International Conference on Foundations of Digital Games*. ACM, 2009, pp. 175–182.
- [143] S. Snodgrass y S. Ontañón, "Experiments in map generation using Markov chains," en *Proceedings of the 9th International Conference on Foundations of Digital Games*. ACM, 2014, pp. 1–8.
- [144] N. Sorenson y P. Pasquier, "Towards a generic framework for automated video game level creation," en *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science, C. Di Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekárt, A. Esparcia-Alcazar, C.-K. Goh, J. Merelo, F. Neri, M. Preuß, J. Togelius, y G. Yannakakis, Eds. Springer Berlin Heidelberg, 2010, vol. 6024, pp. 131–140.
- [145] P. Spronck, I. G. Sprinkhuizen-Kuyper, y E. O. Postma, "On-line adaptation of game opponent AI with dynamic scripting," *Int. J. Intell. Games & Simulation*, vol. 3, num. 1, pp. 80–88, 2004.
- [146] G. Synnaeve y P. Bessiere, "A bayesian model for opening prediction in RTS games with application to StarCraft," en *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, Aug 2011, pp. 281–288.

- [147] G. Synnaeve y P. Bessiere, "A bayesian model for RTS units control applied to StarCraft," en *IEEE Conference on Computational Intelligence and Games*. IEEE, 2011, pp. 190–196.
- [148] G. Synnaeve y P. Bessiere, "Special tactics: A bayesian approach to tactical decision-making," en *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, Sept 2012, pp. 409–416.
- [149] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, num. 9, pp. 1275–1296, 2001.
- [150] V. Tereshko, "Reaction-diffusion model of a honeybee colony's foraging behaviour," en *Parallel Problem Solving from Nature PPSN VI*, ser. Lecture Notes in Computer Science, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, y H.-P. Schwefel, Eds. Springer Berlin Heidelberg, 2000, vol. 1917, pp. 807–816.
- [151] J. Togelius, R. De Nardi, y S. Lucas, "Towards automatic personalised content creation for racing games," en *Computational Intelligence and Games (CIG), 2007. CIG 2007. IEEE Symposium on*, 2007, pp. 252–259.
- [152] J. Togelius, M. Preuss, N. Beume, S. Wessing, J. Hagelback, y G. Yannakakis, "Multiobjective exploration of the StarCraft map space," en *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, 2010, pp. 265–272.
- [153] J. Togelius, G. Yannakakis, K. Stanley, y C. Browne, "Search-based procedural content generation: A taxonomy and survey," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, num. 3, pp. 172–186, 2011.
- [154] M. Tomassini y L. Luthi, "Empirical analysis of the evolution of a scientific collaboration network," *Physica A: Statistical Mechanics and its Applications*, vol. 385, num. 2, pp. 750–764, 2007.
- [155] G. T. Toussaint, "A graph-theoretic primal sketch," *Computational Morphology*, pp. 229–260, 1988.
- [156] J. M. Traish y J. R. Tulip, "Towards adaptive online RTS AI with NEAT," en *Computational Intelligence and Games*. IEEE, 2012, pp. 430–437.
- [157] A. Uriarte y S. Ontanon, "PSMAGE: Balanced map generation for StarCraft," en *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, 2013, pp. 1–8.
- [158] M. van der Heijden, S. Bakkes, y P. Spronck, "Dynamic formations in real-time strategy games," en *IEEE Symposium on*

- Computational Intelligence and Games*, P. Hingston y L. Barone, Eds. IEEE, 2008, pp. 47–54.
- [159] W. van Eck y M. Lamers, “Biological content generation: Evolving game terrains through living organisms,” en *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2015, vol. 9027, pp. 224–235.
- [160] D. J. Watts, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, num. 6684, pp. 440–442, 1998.
- [161] B. G. Weber, M. Mateas, y A. Jhala, “Applying goal-driven autonomy to StarCraft,” en *Artificial Intelligence and Interactive Digital Entertainment Conference*, G. M. Youngblood y V. Bulitko, Eds. The AAAI Press, 2010, pp. 101–106.
- [162] B. Weber y M. Mateas, “A data mining approach to strategy prediction,” en *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, 2009, pp. 140–147.
- [163] S. Wender y I. Watson, “Applying reinforcement learning to small scale combat in the real-time strategy game StarCraft:Broodwar,” en *Computational Intelligence and Games*. IEEE, 2012, pp. 402–408.
- [164] Westwood Studios. (1992) *Dune II: The Building of a Dynasty*. Virgin Interactive.
- [165] Westwood Studios. (1995) *Command & Conquer*. Electronic Arts.
- [166] G. Yannakakis y J. Togelius, “A panorama of artificial and computational intelligence in games,” *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. PP, num. 99, pp. 1–1, 2014.
- [167] E. Zitzler, M. Laumanns, y L. Thiele, “SPEA2: Improving the strength pareto evolutionary algorithm,” en *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, 2001, pp. 95–100.