



# UNIVERSIDAD DE GRANADA

---

## Uso de tecnologías de Ciencia de Datos y de algoritmos genéticos para la optimización de planes de marketing en simulaciones basadas en agentes

---

Memoria presentada por

**Juan Francisco Robles Fuentes**

Para optar al grado de Doctor

PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA INFORMACIÓN  
Y LA COMUNICACIÓN

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA  
ARTIFICIAL

Directores

OSCAR CORDÓN GARCÍA

MANUEL CHICA SERRANO

Granada, Marzo de 2023

Editor: Universidad de Granada. Tesis Doctorales  
Autor: Juan Francisco Robles Fuentes  
ISBN: 978-84-1117-860-0  
URI: <https://hdl.handle.net/10481/82090>

La perseverancia es el secreto de todos  
los triunfos.

---

Victor Hugo

## Agradecimientos

Ahora que veo el final, empiezo a valorar el camino que dejo atrás.

Estos siete años de esfuerzo, altibajos, decepciones y satisfacciones me han hecho crecer como ser humano y me han demostrado el valor de las personas que me acompañan. Aprovecho estas líneas para dar mi más profundo agradecimiento y mi pequeño homenaje a todas aquellos que han formado parte de mi camino y que me han llevado donde estoy.

En primer lugar, quiero dar las gracias a mis padres Teodoro y Conchi, por mostrarme lo que es el sacrificio y la persistencia y por apoyarme en los buenos y en los malos momentos. A mi hermano José Carlos, por aguantarme y acompañarme en cualquier momento y lugar. Valoro mucho tu compañía. A mis abuelos Francisco y María, y a mi tío José, por enseñarme el valor de la humildad y por quererme tanto como me quisisteis. Sé que no sería nada de lo que soy sin vosotros. Gracias a mi compañera de viaje Tatti, por su cariño, paciencia y confianza, y por alentarme a seguir cuando menos ganas tenía de hacerlo. Te quiero mucho y estoy muy orgulloso de tenerte a mi lado.

Mi más sincero agradecimiento a mis directores de tesis, Oscar y Manuel, por su ayuda, apoyo y amistad. Por enseñarme el valor de la investigación y de la excelencia. Os admiro profundamente y me siento privilegiado por vuestra cercanía y familiaridad. Gracias al resto de los que son y los que fueron parte del grupo de investigación SOCCER: Sergio Damas, Óscar Ibañez, Óscar Gómez, Enrique Bermejo, Andrea Valsecchi, Jesús Giraldez, Pablo Mesejo, etc., por hacerme formar parte no sólo de un maravilloso equipo de investigación sino de una gran familia de la que me siento muy orgulloso.

A Ramón Rueda, mi amigo y mi compañero de fatigas durante gran parte de mi etapa académica. Juntos conseguimos cumplir muchas de nuestras metas. Agradecer a Ángel Garrido, por su compañía y tu paciencia ante mis bromas. A David Aguilera, por compartir equipos y rivalidades durante esos partidos de los jueves tan necesarios para despejar la mente.

Y, finalmente, gracias a mi amigo Nacho Moya, al que conocí recién llegado a SOCCER y con el que compartí y seguiré compartiendo muchos buenos momentos y muchas charlas sobre baloncesto.

Gracias a todos, por todo.

## **Acrónimos**

**ABM** Agent-based model

**AI** Artificial Intelligence

**BA** Barabási-Albert

**CGPSOA** Convergence-guaranteed Particle Swarm Optimization Algorithm

**DE** Differential Evolution

**DEB** Dynamic Energy Budget

**EMO** Evolutive Multiobjective Optimization

**GA** Genetic Algorithm

**GRP** Gross Rating Point

**IM** Influence Maximization

**KPI** Key Performance Indicator

**MAPE** Mean Absolute Percentage Error

**MMEA** Multimodal Evolutionary Algorithm

**MMM** Marketing Mix Modeling

**MMOP** Multimodal Optimization Problem

**MOP** Multiobjective Optimization Problem

**MRE** Mean Relative Error

**NPV** Net Present Value

**PSO** Particle Swarm Optimization

**SMSE** Symmetric Mean Squared Error

**SN** Social Network

**SNA** Social Network Analysis

**VM** Viral Marketing

**WOM** Word-of-Mouth

# Índice general

<b>Resumen</b>	<b>13</b>
<b>Abstract</b>	<b>17</b>
<b>1 Introducción</b>	<b>19</b>
1.1 Justificación . . . . .	21
1.2 Objetivos . . . . .	22
1.3 Estructura . . . . .	23
<b>I Fundamentos</b>	<b>27</b>
<b>2 Simulación social</b>	<b>29</b>
2.1 Sistemas complejos . . . . .	29
2.2 Modelos basados en agentes para simulación social . . . . .	31
2.3 Ciencia de Redes y redes sociales . . . . .	34
2.3.1 Las redes sociales como base de los ABMs . . . . .	35
2.3.2 Topologías de redes sociales . . . . .	36
2.3.3 Medidas en la Ciencia de Redes . . . . .	38
2.3.4 ABMs aplicados a estrategias de marketing . . . . .	42
<b>3 Metaheurísticas para calibración y optimización de ABMs</b>	<b>45</b>
3.1 Validación y calibración de ABMs . . . . .	45
3.2 Optimización de estrategias usando ABMs . . . . .	48
3.3 Metaheurísticas para calibración y optimización de estrategias . . . . .	50
3.3.1 Optimización multiobjetivo . . . . .	51
3.3.2 Metaheurísticas multiobjetivo para optimización de ABMs . . . . .	56
3.3.3 Indicadores de calidad multiobjetivo . . . . .	58
3.3.4 Optimización multimodal . . . . .	60
3.3.5 Algoritmos evolutivos multimodales . . . . .	62
<b>II Propuesta</b>	<b>69</b>
<b>4 Modelo CONSUMAT extendido</b>	<b>71</b>
4.1 Introducción . . . . .	71
4.2 El modelo <i>CONSUMAT</i> extendido . . . . .	72
4.2.1 Consumidores en el modelo <i>CONSUMAT</i> . . . . .	73
4.2.2 Influencia de la SN en los agentes . . . . .	77
4.2.3 <i>Awareness</i> y difusión de opiniones en la SN . . . . .	77

4.3	Comparación entre los modelos <i>CONSUMAT</i> base y extendido . . . . .	79
4.3.1	Configuración experimental . . . . .	80
4.3.2	Impacto en redes libres de escala . . . . .	82
4.3.3	Impacto en redes de mundo pequeño . . . . .	83
4.3.4	Discusión de los resultados . . . . .	85
<b>5</b>	<b>Selección de <i>influencers</i> para campañas de marketing viral</b>	<b>87</b>
5.1	Introducción . . . . .	87
5.2	Estado del arte . . . . .	91
5.3	Método automático basado en GAs para campañas de VM . . . . .	93
5.3.1	Variables de decisión . . . . .	94
5.3.2	Funciones objetivo . . . . .	95
5.3.3	Inicialización y evaluación de los cromosomas . . . . .	97
5.3.4	Mecanismo de selección de los padres . . . . .	97
5.3.5	Operadores de cruce y mutación . . . . .	97
5.3.6	Estrategia de reemplazo . . . . .	98
5.4	Experimentos y análisis de resultados sobre la propuesta monoobjetivo . . . . .	98
5.4.1	Configuración experimental . . . . .	99
5.4.2	Experimentos sobre el modelo <i>CONSUMAT</i> extendido . . . . .	99
5.5	Método automático basado en algoritmos EMO para campañas de VM . . . . .	101
5.5.1	Diseño de los métodos de selección de <i>influencers</i> empleados . . . . .	101
5.5.2	Adaptación multiobjetivo de un algoritmo genético estacionario monoobj- jetivo . . . . .	103
5.5.3	Adaptación multiobjetivo de un algoritmo voraz . . . . .	104
5.6	Experimentos y análisis de resultados sobre la propuesta multiobjetivo . . . . .	105
5.6.1	Configuración experimental . . . . .	105
5.6.2	Análisis de resultados . . . . .	107
5.6.3	Análisis de las soluciones obtenidas . . . . .	114
<b>6</b>	<b>Optimización multimodal para calibración de ABMs</b>	<b>119</b>
6.1	Introducción . . . . .	119
6.2	Estado del arte . . . . .	122
6.3	Descripción del ABM de inversión publicitaria en un mercado competitivo . . . . .	124
6.3.1	Estructura del modelo . . . . .	124
6.3.2	Estados de los agentes y toma de decisiones . . . . .	125
6.3.3	Red social subyacente e interacciones WOM . . . . .	126
6.3.4	Descripción de los medios publicitarios . . . . .	127
6.4	Método automático basado en MMEAs para calibración de ABMs . . . . .	129
6.4.1	Parámetros y objetivos de calibración . . . . .	130
6.4.2	Algoritmo de calibración basado en MMEAs . . . . .	132
6.4.3	Análisis estadístico y visual de resultados de calibración . . . . .	133
6.5	Diseño experimental . . . . .	134
6.5.1	Definición de escenarios . . . . .	134
6.5.2	Parámetros de calibración . . . . .	137
6.5.3	Configuración experimental . . . . .	138
6.6	Análisis de resultados . . . . .	140
6.6.1	Comparación de resultados . . . . .	141
6.6.2	Análisis de los mejores resultados de calibración . . . . .	146



<b>7 Optimización multimodal para calibración de modelos ecológicos</b>	<b>153</b>
7.1 Introducción . . . . .	153
7.2 Descripción de la herramienta . . . . .	156
7.2.1 Funcionalidades principales e integración con <i>DEBtool</i> . . . . .	156
7.2.2 Calibración de parámetros con <i>MultiCalib4DEB</i> . . . . .	157
7.2.3 Módulo de calibración multimodal . . . . .	163
7.2.4 Análisis y visualización de resultados . . . . .	165
7.3 Evaluación del rendimiento del módulo de calibración . . . . .	169
<b>III Observaciones finales</b>	<b>175</b>
<b>8 Conclusiones y trabajos futuros</b>	<b>177</b>
8.1 Conclusiones . . . . .	177
8.2 Trabajos futuros . . . . .	180
8.3 Publicaciones . . . . .	181
8.4 Agradecimientos . . . . .	182
<b>Bibliografía</b>	<b>183</b>

# Índice de tablas

4.1	Lista de parámetros para el modelo <i>CONSUMAT</i> base y extendido. . . . .	81
5.1	Lista de parámetros para el GA monoobjetivo. . . . .	99
5.2	Resultados obtenidos por el GA para las redes libre de escala y de mundo pequeño. . . . .	100
5.3	Principales características de las topologías de SNs utilizadas. . . . .	106
5.4	Valores medios y desviación estándar de los indicadores <i>HVR</i> y cardinalidad (los mejores resultados se muestran en negrita). . . . .	108
5.5	Valores medios y desviación estándar del indicador de cobertura <i>C</i> (los mejores resultados se muestran en negrita). . . . .	108
5.6	Valores de la media y la desviación estándar de $I_\epsilon$ multiplicativo (los mejores resultados se muestran en negrita). . . . .	108
5.7	Valores medios y desviación estándar de los indicadores de rendimiento multiobjetivo para el caso real. . . . .	113
5.8	Valores del espacio de decisión obtenidos por las soluciones de MOEA/D, NSGA-II y GA en las SN <i>email</i> y regular. . . . .	116
6.1	Lista y configuración de parámetros fijos de la instancia base de nuestro modelo de mercado (instancia P1(24)). . . . .	136
6.2	Nombre, número de medios publicitarios y variables de decisión asociados a cada instancia del problema. . . . .	136
6.3	Resumen de las transformaciones de parámetros realizadas en la generación de P2(40) a partir de la instancia P1(24). Las modificaciones de inversión se muestran como un porcentaje de incremento/decremento del medio original al nuevo. También se incluye la modificación de los valores históricos objetivo ( <i>awareness</i> y volumen de WOM). . . . .	137
6.4	Lista de variables de decisión asociadas a la instancia P1(24). . . . .	138
6.5	Ejemplo de una solución de calibración para la instancia P2(39) donde las variables de decisión coinciden con los parámetros codificados reales del modelo (que se limitan a $[0, 1]$ ) y el valor final representa el <i>fitness</i> de la solución. . . . .	138
6.6	Promedio, desviación estándar y mejores valores de ajuste encontrados por los MMEAs para cada instancia del problema (los mejores resultados aparecen resaltados en negrita). . . . .	142
6.7	Clasificación media de los MMEAs y los <i>p</i> -valores resultantes utilizando SHADE como método de control. . . . .	146
7.1	Descripción de los módulos de <i>MultiCalib4DEB</i> . . . . .	157
7.2	Descripción de las opciones de calibración de <i>MultiCalib4DEB</i> . . . . .	160
7.3	Descripción de la información proporcionada tras el proceso de calibración en <i>MultiCalib4DEB</i> . . . . .	165
7.4	Opciones para trazar resultados y gráficos de la salida de <i>MultiCalib4DEB</i> . . . . .	167

7.5 Comparación entre los resultados de calibración de Nelder-Mead y SHADE para diferentes especies. La tabla contiene el número de parámetros a calibrar y la función de ajuste inicial para cada especie. También se incluyen los valores de la función de ajuste mínima (mejor), SMSE y MRE tanto para el algoritmo Nelder-Mead como para el SHADE. El promedio de la función de ajuste de las 200 soluciones que devuelve SHADE se muestra a la derecha de los mejores valores de la función de ajuste. Los mejores valores de la función de ajuste, el SMSE y el MRE aparecen en negrita. . . . . 174

# Índice de figuras

3.1	Esquema clásico para calibración de ABMs. . . . .	47
3.2	Ejemplo de relaciones de dominancia entre soluciones dadas dos funciones a minimizar $f_1$ y $f_2$ en un problema multiobjetivo . . . . .	54
3.3	Tipos de frentes óptimos de Pareto para el mismo espacio de soluciones. La forma de cada frente es distinta dependiendo de si los objetivos del problema $f_1$ y $f_2$ son maximizados o minimizados. . . . .	55
3.4	Ejemplo de un espacio de soluciones multimodal generado por la función Alpine 1, ampliamente utilizada para testar algoritmos de optimización. Las regiones con valores más bajos (mínimos) para Alpine 1 aparecen coloreadas en azul mientras que las regiones con valores más altos (máximos) aparecen en rojo. . . . .	61
4.1	Mecanismos involucrados en las dinámicas de compra de los consumidores en el modelo <i>CONSUMAT</i> extendido. . . . .	78
4.2	Uso de heurísticas de decisión y consumo de productos (tanto temporal como total) en los modelos <i>CONSUMAT</i> base y extendido en una topología de red aleatoria. . . . .	80
4.3	Adquisición de productos (total y paso a paso) en los modelos <i>CONSUMAT</i> base y extendido en la red libre de escala. . . . .	82
4.4	Uso de heurísticas de decisión en los modelos <i>CONSUMAT</i> base (líneas continuas) y extendido (líneas discontinuas) en la red libre de escala. . . . .	83
4.5	Consumo de productos (total y paso a paso) en los modelos <i>CONSUMAT</i> base y extendido en la red de mundos pequeño. . . . .	84
4.6	Uso de heurísticas de decisión en en los modelos <i>CONSUMAT</i> base (líneas continuas) y extendido (líneas discontinuas) en la red de mundo pequeño. . . . .	85
5.1	Metodología y módulos incluidos en los métodos avanzados de VM y relación con el modelo <i>CONSUMAT</i> extendido. . . . .	89
5.2	Codificación de cromosomas en el GA. . . . .	94
5.3	Resultado de la estrategia de VM propuesta por el GA en redes libre de escala (izquierda) y de mundo pequeño (derecha). Los <i>influencers</i> seleccionados aparecen delimitados con círculos negros con borde punteado blanco. Cada color representa un producto consumido. . . . .	101
5.4	Aproximaciones del frente de Pareto obtenidas para la red de <i>emails</i> . En el gráfico vemos cómo el frente de Pareto agregado obtenido por MOEA/D domina claramente a los obtenidos por el resto de algoritmos EMO. . . . .	111
5.5	Aproximaciones del frente de Pareto obtenidas para la red SF. De nuevo, el frente de Pareto agregado obtenido por MOEA/D domina a los obtenidos por el resto de algoritmos EMO. . . . .	112

5.6	Aproximaciones del frente de Pareto obtenidas para la red regular. En este caso, vemos cómo es el frente de Pareto agregado obtenido por NSGA-II el que domina, en la mayoría de los casos, a los obtenidos por MOEA/D y las adaptaciones multiobjetivo del GA y del algoritmo <i>greedy</i> . . . . .	113
5.7	Aproximaciones del frente de Pareto obtenidas para la red APP. Como ocurría en el caso de los resultados obtenidos para las redes libre de escala y de mundos pequeños, el frente de Pareto agregado obtenido por MOEA/D domina a los frentes obtenidos por el resto de algoritmos EMO en la mayor parte de los casos.	114
6.1	Esquema general del modelo ABM tomado de [MCC19]. Los agentes expuestos a la publicidad de los medios publicitarios pueden conocer la marca que se promociona y hablar de ella con sus contactos directos (aumentando el volumen de WOM). .	125
6.2	Diagrama que ilustra los componentes y el flujo de trabajo de nuestro marco global para la calibración de ABMs utilizando MMEAs. . . . .	129
6.3	Ejemplo de método de análisis visual sobre soluciones obtenidas por un MMEA durante la calibración de dos modelos tomado de [Chi+17]. A la izquierda de la figura se presenta un análisis de sensibilidad realizado sobre dos parámetros, $p_M$ y $X_k$ , con respecto a un tercer parámetro $P_{X_m}$ en un modelo ecológico poblacional. A la derecha, se realiza el mismo análisis de sensibilidad sobre los parámetros $P_{15}$ y $P_6$ respecto a $P_{12}$ en un modelo de dinámica de sistemas. . . . .	135
6.4	Distribución de <i>fitness</i> para 200,000 soluciones aleatorias para las instancias P1, P6, P10 y P12. Los histogramas ilustran las diferencias en el espacio objetivo para las instancias consideradas. . . . .	145
6.5	Rangos de <i>fitness</i> obtenidos por SHADE en las 12 instancias del problema. . . .	145
6.6	Representación de los conjuntos de soluciones de calibración obtenidos por SHADE, NichePSO y Crowding GA en los gráficos 2D del espacio de búsqueda multimodal de la instancia P1 correspondiente a los parámetros $d$ y $\alpha^{WOM}$ . Las soluciones encontradas por SHADE están todas en la región de menor valor de <i>fitness</i> (mejor calidad), mientras que las soluciones encontradas por NichePSO y Crowding GA sólo logran situar unas pocas (o ninguna) soluciones en la región de mejor rendimiento, respectivamente. . . . .	148
6.7	Representación de los conjuntos de soluciones de calibración obtenidos por SHADE, NichePSO y Crowding GA en los gráficos 2D del espacio de búsqueda multimodal de la instancia P6 ABM correspondiente a los parámetros $\alpha_{15}$ y $\alpha_7$ . Las soluciones encontradas por SHADE se encuentran todas en las regiones de mejor <i>fitness</i> , mientras que las soluciones de NichePSO están fuera de ellas. Las soluciones de Crowding GA cubren una mayor parte del espacio de búsqueda, pero no se encuentran en las mejores regiones. . . . .	149
6.8	Detalle de los conjuntos de soluciones de calibración SHADE en los gráficos 2D del espacio de búsqueda multimodal de las instancias P1 y P6 mostradas en las Figuras 6.6 y 6.7. . . . .	149
6.9	Mejores combinaciones de parámetros devueltas por SHADE en las instancias P1, P5, P10, and P12. . . . .	151
7.1	Diagrama que ilustra el funcionamiento de <i>MultiCalib4DEB</i> y sus etapas durante un proceso de calibración de un modelo DEB. . . . .	159

7.2	Ejemplo de los gráficos disponibles en el módulo <code>charts</code> . La figura de la parte superior muestra la salida del método <code>textttscatter</code> mientras que la figura de la parte inferior muestra la salida del método <code>density_hm_scatter</code> . Los pares de parámetros comparados en los gráficos son el factor de zoom ( $z$ ) y $f_tW$ . $z$ se define como la longitud volumétrica máxima de un organismo en centímetros mientras que $f_tW$ es una respuesta funcional escalada para el peso húmedo en la pubertad de una especie. . . . .	168
7.3	Ejemplo ilustrativo de dos procesos de calibración de diez minutos (izquierda) y una hora (derecha) sobre la especie <i>Clarias gariepinus</i> . El eje X contiene el tiempo medido en días mientras que el eje Y contiene los valores del “peso húmedo” (en gramos). . . . .	169
7.4	Ejemplo de la salida del módulo <code>charts</code> de <i>MultiCalib4DEB</i> usando como datos de entrada los resultados del Ejemplo 3. . . . .	169
7.5	Ejemplo de la salida del módulo <code>plots</code> de <i>MultiCalib4DEB</i> usando los resultados Ejemplo 4. . . . .	170

# Resumen

## Introducción al problema

Los procesos de modelado y simulación computacionales son herramientas cada vez más empleadas en distintas disciplinas como economía [Boe+15; Tad16], ingeniería civil [NM12], sociología [MW02; BS15] o marketing [MCC19; NY14].

En los últimos años, la aplicación de simulaciones para el estudio de sistemas complejos utilizando el paradigma del modelado basado en agentes ha gozado de gran popularidad [FF09; Wal18; Coa+19]. Esto se debe a la dificultad de replicar el funcionamiento de este tipo de sistemas mediante reglas matemáticas o agregando el comportamiento de sus partes.

Los modelos basados en agentes han demostrado gran capacidad para modelar interacciones locales en sistemas complejos que, posteriormente, generan patrones globales tales como la difusión de información, reglas universales emergentes y acciones colectivas [MW02]. En particular, estos modelos son apropiados para mimetizar y responder a preguntas relacionadas con grupos de consumidores heterogéneos que operan en un entorno complejo con una medida de interés que proviene de un resultado emergente [RR11]. Estas técnicas de modelado son especialmente útiles en marketing y en particular representando a los consumidores, en los que aparecen diversos criterios de carácter objetivo y subjetivo asociados al consumidor final [VB10], así como distintos canales y vías de modificación de sus percepciones y de su posicionamiento respecto a las marcas y productos del mercado. Evaluar y emular las características sociales de cada consumidor, sus preferencias y sus decisiones de compra es, de facto, una tarea compleja que no obedece a reglas globales [Chi+16].

Los modelos basados en agentes se han aplicado específicamente en marketing tanto para comprender cómo las relaciones entre los clientes impulsan la dinámica del mercado [Del+10] como para estudiar cómo afecta la presión publicitaria de distintos medios a las dinámicas de compra de sus consumidores [MCC21].

Durante la fase de diseño de un modelo basado en agentes, el modelador codifica una serie de microrreglas del comportamiento de los agentes para luego medir los resultados emergentes a nivel macro buscando patrones de comportamiento generales. Sin embargo, la construcción de modelos es una tarea compleja, ya que los modeladores normalmente tienen que establecer los valores de un gran número de parámetros que suelen ser desconocidos.

El proceso de ajustar los valores de los parámetros de un modelo basado en agentes para replicar correctamente la dinámica deseada se conoce como calibración y es un paso crucial durante la validación del modelo [Oli03]. Un enfoque comúnmente utilizado es la calibración automática, la cual requiere normalmente un método de optimización para modificar los parámetros de un modelo basado en agentes de forma sistemática y una medida de error para comparar los datos históricos del sistema con la salida del modelo simulado. Durante el proceso de calibración automática se ajusta un conjunto de parámetros del modelo, minimizando la medida de error considerada para que se ajuste a los datos reales [Oli03; Sar10].

Mediante la modificación de ciertos parámetros e información de los modelos virtuales construidos, calibrados y validados es posible plantear escenarios hipotéticos “what-if” para realizar simulaciones y estudiar los efectos de distintas políticas de toma de decisiones en las dinámicas del sistema. De esta forma se pueden evaluar estrategias previas a su implementación en el mundo real, reduciendo el riesgo de aplicarlas sin haberlas testado fuera del mercado real. Un ejemplo es el lanzamiento de estrategias de marketing en mercados competitivos.

No obstante, la aplicación de estas herramientas se enfrenta con varias dificultades, generalmente de índole metodológica. La principal es que para poder simular un modelo basado en agentes se debe identificar cuáles sus atributos y sus comportamientos, especialmente respecto a las interacciones agente-agente y agente-entorno [MN14]. En los mercados virtuales (un modelo de simulación basada en agentes que representa un mercado real) asociados al marketing esto supone un reto dado que las motivaciones y acciones de los consumidores de un determinado producto o servicio ni son observables de forma explícita ni se pueden representar fácilmente como un conjunto de funciones sobre variables discretas. Por otro lado, un mercado virtual se trata de un concepto profundamente abierto, sujeto a múltiples dinámicas y criterios, y con fronteras



---

generalmente difusas. Por ello, definir procesos de ingeniería y computación de software que permitan construir modelos basados en agentes rigurosos, descriptivos y realistas es actualmente uno de los principales retos de investigación tanto a nivel científico como industrial [RR11; CR17].

Esta tesis doctoral se enfoca, en primer lugar, en la construcción un modelo basado en agentes para marketing en el que se representan distintas características de comportamiento de consumidor. Las dinámicas de consumo, producto de las relaciones entre los agentes, serán analizadas para medir la robustez del modelo. Sobre el modelo construido se diseñarán, en una segunda fase, técnicas de ciencia de datos para optimizar estrategias de marketing viral en distintos entornos de consumo basadas en algoritmos evolutivos mono y multiobjetivo. Por último, se propondrán técnicas de optimización basadas en algoritmos evolutivos multimodales con un doble objetivo: i) facilitar a los modeladores la tarea de validar los resultados obtenidos por un modelo basados en agentes, simplificando por tanto la validación del mismo, y ii) ofrecerles distintas soluciones de calibración de similar calidad dentro del espacio de parámetros del modelo, permitiendo así analizar su robustez. De este modo, los modeladores pueden seleccionar las soluciones que mejor se adaptan a sus necesidades o aumentar las alternativas que tienen para apoyar sus decisiones. Adicionalmente, se ha desarrollado una herramienta también basada en algoritmos evolutivos multimodales para calibración de modelos ecológicos poblacionales para ayudar los investigadores del área de biología a la hora de analizar y estudiar los parámetros de las especies con las que trabajan.

## Desarrollo

El trabajo desarrollado a lo largo de esta tesis doctoral se divide en tres bloques principales que abordan los objetivos planteados en la misma:

- I El desarrollo de un modelo basado en agentes para marketing al que incorporamos reconocimiento de marca e incertidumbre en la toma de decisiones de los agentes con el que

pretendemos obtener modelos más realistas que permitan reproducir las dinámicas de consumo de los mercados y aprovecharlas para descubrir patrones globales de comportamiento en los mismos.

- II El diseño de un método automático de selección de individuos semilla para la difusión de productos en campañas de marketing viral sobre el modelo que representa un mercado virtual desarrollado anteriormente basado en algoritmos evolutivos mono y multiobjetivo.
- III La propuesta de métodos de calibración de modelos basados en agentes que permitan obtener distintos conjuntos de parámetros de rendimiento óptimo haciendo uso de algoritmos evolutivos multimodales, para facilitar a los modeladores la validación del modelo e incrementar las alternativas disponibles para apoyar sus decisiones.

Las siguientes secciones resumen los bloques mencionados con anterioridad. En primer lugar, se define el diseño e implementación de un modelo basado en agentes que extiende el modelo clásico *CONSUMAT* [JJ03] incorporando el manejo de la incertidumbre para hacerlo más cercano al mundo real. A continuación, se contempla el diseño, desarrollo, validación y aplicación de nuestras técnicas automáticas propuestas para abordar la selección de individuos semilla para generar campañas de marketing viral basadas en algoritmos genéticos monoobjetivo y en algoritmos evolutivos multiobjetivo. El modelo *CONSUMAT* extendido se utilizará para simular y testar las estrategias generadas. Por su parte, las técnicas propuestas se encargan, por un lado, de seleccionar el número idóneo de individuos influyentes a través de los cuales difundir un producto en un mercado competitivo y, por otro, de optimizar dos objetivos en conflicto: El coste de seleccionar a dichos individuos y el beneficio obtenido en la campaña. Finalmente, se considera el uso de algoritmos evolutivos multimodales para abordar dos problemas: i) la calibración de parámetros en modelos basados en agentes y ii) la calibración de modelos ecológicos poblacionales. Con la primera propuesta pretendemos aportar a los modeladores distintas alternativas de calidad con las que mejorar el conocimiento que tienen de sus modelos y reforzar su toma de decisiones. Con la segunda, proveemos al área de biología una potente herramienta software con la que enriquecer el conocimiento obtenido durante sus estudios.

## Incorporación de reconocimiento de marca al modelo *CONSUMAT*

Iniciamos la presente tesis doctoral con la incorporación de reconocimiento de marca (*awareness* [Aak97], en inglés) a uno de los modelos basados en agentes más asentados y conocidos: *CONSUMAT* [JJ03]. El objetivo perseguido fue triple: Primero, trabajar con un modelo basado en agentes que fuese robusto y que permitiese representar las relaciones entre los agentes durante los procesos de toma de decisión de compra en el mercado. Segundo, incorporar incertidumbre en el modelo para representar de una forma más realista los procesos de interacción y las decisiones de compra llevados a cabo por los consumidores. Incorporando un filtro basado en *awareness* [Cao99] a los consumidores, éstos no disponen de información completa acerca de los productos ofertados en el mercado sino que necesitan obtener dicha información por medio de la comunicación con sus contactos en la red. Tercero, incluir de un mecanismo de difusión de opiniones basado en el boca a boca (*word-of-mouth*, en inglés) en el modelo basado en agentes, dotándolo de un mayor realismo y permitiendo obtener diversos comportamientos emergentes avanzados gracias a la combinación con el filtro basado en *awareness* [Duz+07; RR11].

Al incorporar *awareness* y procesos de difusión de información al modelo, los consumidores son capaces de difundir sus preferencias a través de su red social por medio de la comunicación con otros individuos y ganar y perder *awareness* sobre los productos disponibles en el mercado. En este modelo *CONSUMAT* mejorado los agentes no disponen de una información completa e inmutable sobre el mercado sino que dependen de su entorno para obtener dicha información y para tomar decisiones, tal y como ocurre en los mercados reales. De esta forma, se aprovechan de mejor manera las características de las redes sociales que soportan el sistema de agentes ya que los mismos agentes dependen de sus conexiones con otros individuos, del grado de conexiones que poseen dentro de la red y de la comunidad a la que pertenecen para difundir con mayor o menor éxito sus opiniones.

Una vez extendido el comportamiento del modelo *CONSUMAT* original, se procedió a validar el comportamiento del nuevo modelo mediante una extensa batería de pruebas que comprendió variaciones en los principales parámetros del modelo, en los parámetros asociados al *awareness*

y a la difusión de conocimiento de la marca a través del *word-of-mouth*, y simulaciones sobre distintas topologías de redes sociales [ZSM12] para estudiar su influencia sobre las dinámicas y resultados del modelo basado en agentes.

## **Método de selección de *influencers* en campañas de marketing viral basado en algoritmos evolutivos mono y multiobjetivo**

Tras la consecución del primer objetivo de esta tesis doctoral, procedimos a la explotación del modelo basado en agentes desarrollado. Para ello, elaboramos un entorno de trabajo para la identificación y selección de usuarios influyentes que tienen un gran impacto en las dinámicas de mercado (más conocidos como *influencers*, en inglés) con el propósito de optimizar la difusión de productos, opiniones y/o servicios en comunidades de consumidores por medio de campañas de marketing viral [LAH07; Wat+07]. El entorno de trabajo mencionado anteriormente combina dos elementos: El modelo basado en agentes para marketing desarrollado en la sección anterior y los algoritmos evolutivos monoobjetivo y multiobjetivo. Por un lado, el modelo *CONSUMAT* extendido permite simular los procesos de compra que tienen lugar dentro de un entorno de mercado formado por agentes independientes y extraer comportamiento de forma agregada. Por otro lado, los algoritmos evolutivos permiten identificar el número de individuos a seleccionar como difusores del producto ofertado por una determinada marca en campañas de marketing viral al mismo tiempo que se optimizan los beneficios de la campaña (propuesta monoobjetivo) o dos objetivos en conflicto dentro de dichas campañas: El coste de la selección y contratación de *influencers* y el beneficio generado por la campaña de marketing viral (propuesta mutiobjetivo).

La optimización de campañas de marketing viral con algoritmos evolutivos necesita de una estrategia para identificar a los *influencers* que participan en las campañas de marketing viral [SRW10]. Para ello, utilizamos diferentes métricas de redes sociales tales como el coeficiente de clustering, el grado y el número de vecinos en el entorno de cada consumidor para medir de forma cuantitativa su importancia dentro de la red social que representa las conexiones entre los agentes del modelo. Junto a estas métricas, añadimos el número total de individuos semilla a seleccionar que, como veremos posteriormente, estaba estrechamente relacionado con el coste

de las campañas.

La propuesta monoobjetivo considera un algoritmo genético estacionario para optimizar el número de individuos *influencer* a seleccionar y agrupa en un único objetivo el beneficio y coste de las campañas de marketing generadas. Por su parte, los algoritmos evolutivos multiobjetivo fueron los encargados de buscar las mejores combinaciones de parámetros (métricas de redes sociales asociadas a los consumidores y número total de *influencers* seleccionado) de una función de selección de individuos semilla que maximizara el beneficio de las campañas de marketing viral realizadas mientras minimiza su coste. Los algoritmos evolutivos multiobjetivo son metaheurísticas poblacionales capaces de explorar entornos de soluciones con objetivos en conflicto y encontrar distintas soluciones Pareto-optimales en una única ejecución. Siguiendo el esquema planteado para la identificación de *influencers*, cada solución al problema se codifica con distintas variables de decisión que representan el valor de cada una de las métricas de red social y el número total de individuos semilla seleccionado. Del amplio abanico de métodos enmarcados dentro del campo de la optimización multiobjetivo seleccionamos los dos más extendidos para optimizar nuestro problema: NSGA-II [Deb+02] y MOEA/D [ZL07]. Además, para validar el desempeño de estos dos métodos y medir la calidad de sus resultados realizamos una batería de pruebas en la que los comparamos con un algoritmo genético y con un algoritmo voraz (*greedy*, en inglés). Las soluciones devueltas por los dos algoritmos evolutivos multiobjetivo superaron claramente a las ofrecidas por los algoritmos genético y voraz, siendo MOEA/D el algoritmo con mejor desempeño en la mayoría de las pruebas realizadas.

## **Métodos para facilitar la calibración de modelos basados en agentes mediante algoritmos evolutivos multimodales**

Como ya mencionamos en la introducción al problema, un paso crucial en la construcción y validación de un modelo basado en agentes es el ajuste de los parámetros del modelo por medio de la calibración. La calibración es un procedimiento por el cual un modelador o un mecanismo automatizado lanza simulaciones sobre un modelo basado en agentes probando diferentes combinaciones de sus parámetros para ajustarlos a un valor real. El ajuste del resultado

devuelto por el modelo respecto a los valores de los datos históricos disponibles del sistema podrá ayudar a confirmar que las dinámicas simuladas en el modelo basados en agentes se corresponden a las experimentadas en el mundo real. Un mejor ajuste reforzará la confianza sobre el modelo simulado y aumentará la seguridad en las respuestas que pueda ofrecer.

Aunque la calibración de un modelo podría efectuarse de forma manual, la mayoría de los modelos de simulación suelen considerar un número alto de parámetros susceptibles de ser calibrados. El volumen de parámetros a calibrar y las interacciones complejas que aparecen entre ellos hace inviable para un modelador calibrar el modelo por sí mismo. Ante esta situación, los algoritmos evolutivos [BFM97] proporcionan modelos computacionales mono [Gol89] y multiobjetivo [Deb01; CLVV+07] para la optimización, la búsqueda y el aprendizaje automático basados en la evolución y la selección natural Darwiniana. En concreto, permiten realizar búsquedas robustas en espacios complejos y que ofrecen soluciones de alta calidad en un tiempo razonable.

Se han propuesto una gran cantidad de trabajos enfocados en la calibración de modelos basados en agentes utilizando tanto algoritmos mono [CH05; Fab13; CR17; Moy+21b; Moy+21a] como multiobjetivo [RT04; NMM06; Far+16; OEBY20; MCC21]. Sin embargo, el estudio de las distintas configuraciones de parámetros que resultan de la exploración del espacio de soluciones de la calibración de los modelos basados en agentes no se ha desarrollado [Chi+17]. La naturaleza multidimensional del problema de calibración y la presencia de varias soluciones subóptimas sugieren el uso de algoritmos evolutivos multimodales [Das+11] como una metaheurística recomendada para la calibración de modelos basados en agentes. Los algoritmos evolutivos multimodales son también un enfoque útil cuando se requieren múltiples configuraciones óptimas de parámetros para apoyar mejor las decisiones que un experto podría tomar a partir de los resultados del método de optimización [Chi+17].

Esta carencia nos lleva a diseñar e implementar un entorno de trabajo que utiliza distintos tipos de algoritmos evolutivos multimodales para llevar a cabo el proceso de calibración para estimar los parámetros de un modelo basado en agentes para marketing utilizando un esquema de codificación real. De la amplia familia de algoritmos evolutivos, aplicados con éxito a problemas de

---

optimización multimodal [TI20], consideramos tres métodos clásicos de optimización multimodal basados en algoritmos genéticos y nichos [SD06] (como Crowding [DJ75], Clearing [Pet96; Pét97], y Sharing [GR+87]); un algoritmo evolutivo multimodal basado en enjambres de partículas (NichePSO) [BEB02]; una extensión asistida por nichos de NSGA-II [Deb+02] para optimización multimodal (PNA-NSGA-II) [BD13]; y cuatro extensiones multimodales del conocido algoritmo de evolución diferencial [SP97], DE/NRAND/2 [EPV11], MOBiDE [BDT12], SHADE [TF13], y L-SHADE [TF14].

Los algoritmos evolutivos multimodales seleccionados se compararon utilizando 12 escenarios, cada uno de los cuales estaba definido por diferentes instancias de un modelo basado en agentes para marketing, teniendo cada instancia entre 24 y 129 parámetros a calibrar, lo que nos permitió analizar el comportamiento de los diferentes algoritmos al calibrar escenarios con más de 100 parámetros. Los resultados fueron evidentes, SHADE, L-SHADE y NichePSO mostraron un claro dominio sobre los demás algoritmos. Además, los conjuntos de soluciones devueltos por estos tres algoritmos evolutivos multimodales mostraron una alta diversidad y un ajuste de calidad, lo que confirmó la utilidad de estos métodos para, por una parte, dar a los modeladores un amplio abanico de soluciones de igual preferencia en vez de una única solución global con la que soportar mejor sus decisiones y, por otra, estudiar el entorno de soluciones del modelo basado en agentes que calibran.

### ***MultiCalib4DEB: Un toolbox que explota la optimización multimodal en la calibración de parámetros en modelos ecológicos poblacionales***

Por último, y como objetivo adicional a esta tesis doctoral, decidimos aprovechar el potencial de los algoritmos evolutivos multimodales para ofrecer soluciones diversas y de calidad en la calibración de modelos basados en agentes y aplicarlo a otro área de conocimiento: Los modelos ecológicos poblacionales. Concretamente, desarrollamos un producto software llamado *MultiCalib4DEB* para la calibración multimodal de modelos basados en la teoría del balance energético dinámico. Estos modelos estudian comportamientos biológicos y metabólicos de diferentes especies [Sar+13; Sma+17] usando la teoría del balance energético dinámico, una

teoría formal basada en principios químicos y físicos de la captación y uso de sustratos (alimentos, nutrientes, luz) por microorganismos, animales y plantas para su mantenimiento, crecimiento, maduración y reproducción. Durante el desarrollo de este software colaboramos con distintos investigadores de la universidad de Dalhousie en Canadá, del Instituto Superior Técnico de Lisboa y del Intitituto de Investigación Marina de Noruega.

*MultiCalib4DEB* se desarrolló con el objetivo de ser fácilmente aplicable sobre los modelos y especies existentes dentro de la teoría del balance energético dinámico tanto para calibrar los parámetros de sus modelos como para evaluar su solidez. El *toolbox* proporciona una amplia variedad de módulos, diferentes alternativas de calibración y herramientas estadísticas y visuales para realizar análisis de sensibilidad sobre los resultados de calibración obtenidos. Además, la herramienta también es interactiva y flexible, y permite que los usuarios puedan personalizar los ajustes de sus calibraciones (tiempo total de calibración, origen y rangos de los parámetros a calibrar, etc).

Además de mejorar el ajuste de los modelos, una de las principales ventajas de *MultiCalib4DEB* es que es capaz de devolver conjuntos óptimos y diversos de soluciones de calibración igualmente deseables que pueden ser utilizadas por el modelador para realizar análisis de sensibilidad más robustos sobre sus resultados o para lanzar nuevas calibraciones con las que profundizar en la exploración de una solución específica.

Dentro de la amplia familia de los algoritmos evolutivos multimodales seleccionamos dos extensiones de DE [SP97], uno de los algoritmos de búsqueda basados en poblaciones más versátiles y robustos del campo de la optimización multimodal [DMS16]: Success-history based adaptive differential evolution (SHADE) [TF13] y su extensión L-SHADE [TF14] como motores de calibración multimodal dentro de *MultiCalib4DEB*. Nuestro desarrollo, se integró directamente con la principal herramienta existente para el análisis de modelos basados en la teoría del balance energético dinámico (DEBtool) y con su marco de calibración<sup>1</sup>. La creación del proyecto Add-my-Pet<sup>2</sup> y la introducción de DEBtool [Koo21; Mar+12] proporcionaron un marco de calibración cómodo y accesible y una amplia biblioteca de funciones y ejemplos para utilizar en la calibración

---

<sup>1</sup>[https://github.com/add-my-pet/DEBtool\\_M/tree/master/lib/MultiCalib4DEB](https://github.com/add-my-pet/DEBtool_M/tree/master/lib/MultiCalib4DEB)

<sup>2</sup>[http://www.bio.vu.nl/thb/deb/deblab/add\\_my\\_pet/](http://www.bio.vu.nl/thb/deb/deblab/add_my_pet/)



---

de parámetros y en las aplicaciones de modelos ecológicos basados en la teoría del balance energético dinámico [Koo00; Koo09]. DEBtool fue una de las razones del éxito de la aplicación de la teoría del balance energético dinámico para simular las dinámicas del metabolismo (balances de materia y energía) de organismos vivos a nivel individual, ya que se actualiza constantemente para mejorar la experiencia de modelado [Lik+11; Lik+14; Mar+18; Mar+19]. Por este motivo, varios autores han utilizado el *toolbox* como base para la calibración de los parámetros de las especies con las que trabajan [Sar+13], han propuesto varios métodos para mejorar tanto los resultados de calibración de la DEBtool [MCM19] como su validación [Acc+20], y han contribuido al proyecto Add-my-Pet. A finales de 2022, hay unas 3.000 especies disponibles en la base de datos del proyecto.

Para comprobar la eficacia de *MultiCalib4DEB* a la hora de calibrar modelos basados en la teoría del balance energético dinámico realizamos la calibración de 13 especies con diferentes complejidades. De esta forma comparamos los algoritmos evolutivos multimodales incluidos en el software (SHADE y L-SHADE) con el método de ajuste de parámetros disponible en DEBtool: Nelder-Mead. Los resultados de la experimentación mostraron un ajuste superior de los algoritmos evolutivos multimodales con respecto al Nelder-Mead al tiempo que devolvieron alrededor de 200 soluciones óptimas y diversas para cada especie en lugar de una única solución. De este conjunto de soluciones, el usuario puede elegir cuáles visualizar, analizar o utilizar para otro proceso de calibración. Esta última afirmación refuerza las ventajas de utilizar los algoritmos evolutivos multimodales para la calibración de los modelos basados en la teoría del balance energético dinámico.

Nuestra aportación marca un antes y un después en el desarrollo de DEBtool y en el estudio de modelos basados en la teoría del balance energético dinámico al introducir un potente marco de calibración, análisis y validación de especies. Además, la aplicación de los algoritmos evolutivos multimodales a la calibración de modelos ecológicos y biológicos no se limita a la teoría del balance energético dinámico y, por tanto, los resultados de este estudio y el *toolbox* presentados pueden ser utilizados para técnicas de modelización similares.

## Conclusiones y trabajos futuros

En conclusión, en esta tesis doctoral hemos diseñado, desarrollado y aplicado métodos basados en algoritmos evolutivos multiobjetivo y multimodales para abordar y resolver problemas en el área de los modelos basados en agentes para marketing. Primero, extendimos un modelo basado en agentes clásico, diseñando e incorporando mecanismos de difusión de información y reconocimiento de marca al mismo para crear un modelo de consumo más realista. A continuación, propusimos métodos de optimización mono y multiobjetivo para la selección de *influencers* en campañas de marketing viral sobre el modelo extendido. Por último, implementamos métodos de calibración multimodal extensibles a cualquier modelo basado en agentes y a otros modelos computacionales como los modelos ecológicos poblacionales.

Tanto los métodos desarrollados para la selección de *influencers* en campañas de marketing viral como los propuestos para la calibración de modelos basados en agentes son trabajos pioneros y novedosos en sus respectivas áreas de investigación. Los trabajos futuros a abordar relacionados con las dos tareas anteriormente mencionadas son comunes y se centran en el escalado tanto de los modelos basados en agentes como de los algoritmos de optimización sobre ellos aplicados para poder realizar simulaciones de modelos más cercanos al mundo real en los que interactúan millones de agentes virtuales.

En lo referente a la aplicación de algoritmos evolutivos multimodales para calibración de modelos basados en agentes, ya sean para mercados de consumo o biológicos, queda como tarea futura el proponer nuevas herramientas cuantitativas y cualitativas para facilitar la selección de soluciones devueltas tras la calibración de los modelos. En esta última tarea se abordaría una de las principales debilidades del área de optimización multimodal. Además, la continua aparición de nuevas propuestas de algoritmos evolutivos multimodales hace necesario una nueva comparación de los algoritmos evolutivos multimodales aplicados a la calibración de modelos basados en agentes con objeto de diseñar el método de calibración de mejor rendimiento.

Los resultados obtenidos a partir de los métodos propuestos en esta tesis doctoral demuestran tanto la utilidad de la aplicación de los algoritmos multiobjetivo para la optimización de objetivos

---

conflictivos en campañas de marketing viral como el uso de algoritmos multimodales para la calibración de modelos basados en agentes, siendo MOEA/D y SHADE los algoritmos con mejores resultados en sus respectivos problemas. Además, encontramos algunos resultados interesantes durante las fases de experimentación de cada propuesta, como la capacidad de las técnicas utilizadas para adaptarse a las dinámicas de las redes sociales subyacentes de cada modelo basado en agentes y al aumentos en la complejidad (número de parámetros a calibrar) de los mismos.

La aplicación de técnicas de optimización multimodal para la calibración de modelos basados en la teoría del balance energético dinámico y su integración en el *MultiCalib4DEB* también fueron satisfactorias. Encontramos resultados muy prometedores y obtuvimos un magnífico desempeño del *toolbox* y de los módulos de optimización desarrollados a la hora de calibrar especies con distintas dinámicas metabólicas y diferentes complejidades. Asimismo, los paquetes para el análisis visual y cuantitativo integrados en *MultiCalib4DEB* no sólo facilitaron y enriquecieron el análisis de los resultados devueltos por los algoritmos de optimización multimodal sino que también hicieron más cercano su uso a los investigadores del área.



# Abstract

Modeling and decision making are common aspects in fields such as different as business, marketing, social sciences, engineering and medicine, among others. Information management from a decision-making point of view involves being able to capture and represent meaningful information to human modelers for them to make effective and efficient decisions.

We can state that agent-based models are one of the most prominent modeling techniques for accurately representing emergent phenomena in complex systems. However, these techniques require robust construction and thorough validation of the models to be successfully exploited. In addition to the problems involved in the development of an agent-based model there are certain problems or obstacles both in the validation and exploitation of agent-based models that could be solved using approaches such as multi-criteria decision making, multiobjective and multimodal optimization. On the one hand, problems such as targeting of *influencers* in viral marketing campaigns or the calibration of agent-based models parameters considering multiple key performance indicators include the existence of several conflicting objectives that must be optimized simultaneously with or without the expert information of the decision-maker. On the other hand, the calibration of agent-based models also involves the existence of different sets of parameter values that often produce similar calibration settings and that are intractable for exact approaches.

In this doctoral thesis we propose methods based on multi-objective and multi-modal evolutionary algorithms to address and solve problems in the area of agent-based models in applications such as virtual markets. First, we extend a classic agent-based model named *CONSUMAT*, designing and incorporating information diffusion and brand awareness mechanisms into it to create a more realistic virtual market model. Then, we propose a viral marketing method based on multi-objective evolutionary algorithms to deal with conflicting objectives during the optimization of marketing campaigns. The method uses two well-established multi-objective evolutionary algorithms (NSGA-II and MOEA/D) for indentifying and targeting the optimal number of *influencers* for increasing the revenue while decreasing the costs of the media campaigns and

is applied and validated over the previously extended model. Finally, we propose multimodal calibration methods for fitting the virtual agent-based model dynamics to the real market ones. We apply and compare nine multimodal evolutionary algorithms (both classical such as Crowding, Clearing, and Sharing, and novel such as NichePSO, PNA-NSGA-II, DE/NRAND/2, MOBiDE, SHADE, and L-SHADE) on 12 instances of an agent-based model for marketing with different complexities. These methods can be extended both to other agent-based models and to other computational model such as those based on ecological populations.

Both the methods developed for the selection of *influencers* in viral marketing campaigns and those proposed for the calibration of agent-based models are pioneering and innovative works in their respective research areas. The results obtained from our methods show the suitability of the application of the multiobjective algorithms for the optimization of conflicting objectives in marketing campaigns and also the use of multimodal algorithms for the calibration of agent-based models with MOEA/D and SHADE being the best performing algorithms in their respective problems. Moreover, we found some interesting results during the experimentation phases of each task such as the ability of the techniques to adapt to the underlying social networks of each agent-based model and to increases in their complexity (number of parameters to calibrate).

The application of multimodal optimization techniques for the calibration of ecological population models and their integration into the *MultiCalib4DEB toolbox* were also satisfactory. We found very promising results and obtained a remarkable performance of the *toolbox* and its optimization modules when calibrating species with different metabolic dynamics and complexities. Also, the visual and quantitative analysis packages integrated in *MultiCalib4DEB* not only made the analysis of the results returned by the multimodal optimization algorithms easier, but also made their use closer to the researchers in the area (for whom advanced optimization approaches are often unfamiliar).

Future works to be addressed are related to scaling both the agent-based models and the optimization algorithms to perform simulations closer to the real world where millions of virtual agents interact. Regarding the application of multimodal evolutionary algorithms for the calibration of agent-based models, it remains as a future task to propose new quantitative

and qualitative tools to ease the selection of solutions returned after model calibration. This last task would address one of the main weaknesses in the area of multimodal optimization. In addition, the continuous emergence of new proposals for multimodal evolutionary algorithms makes necessary a new comparison of multimodal evolutionary algorithms applied both to the calibration of agent-based models and ecological population models to design the calibration method with the best performance.





## Introducción

El comienzo es la parte más importante del recorrido.

---

Platón

La sociología (del latín *socius*, ‘socio, compañero’, y el sufijo griego *-λογία*- *loguía*, ‘tratado’, ‘estudio’) y el análisis de las relaciones interpersonales han sido un fértil campo de estudio desde que Comte<sup>1</sup> acuñase el término en 1824. En pleno siglo XXI, en la era de la globalización y las nuevas tecnologías, la evolución de las redes de comunicación y el crecimiento de la conectividad entre individuos ha puesto de manifiesto la importancia del estudio de las relaciones sociales y su utilidad para entender los mecanismos de influencia social y aprovecharlos para fines biológicos, socio-económicos o políticos.

La simulación social y, en particular, los modelos basados en agentes (ABMs, por sus siglas en inglés) han demostrado su capacidad para modelar interacciones locales con el fin de generar patrones globales como la difusión de información, la aparición de reglas universales, la transmisión de enfermedades virales o la participación en acciones colectivas [MW02]. Estos patrones sociales emergentes pueden aparecer y desaparecer de forma inesperada en distintos entornos, como en las revoluciones, las pandemias o las crisis de mercado. Los ABMs [MW02; Bon02] proporcionan un valioso marco donde los patrones globales de comportamiento aportan mucho más conocimiento que la simple agregación de atributos individuales, habiéndose aplicado a un gran número de problemas entre los que se encuentran diferentes problemas económicos y de comportamiento del consumidor.

---

<sup>1</sup>[https://es.wikipedia.org/wiki/Auguste\\_Comte](https://es.wikipedia.org/wiki/Auguste_Comte)

El hecho de que los ABMs se hayan aplicado a entornos económicos y de consumo se debe a que las economías de mercado descentralizadas comprenden sistemas complejos y adaptativos formados por un gran número de elementos individuales implicados en muchas interacciones locales que se producen en paralelo. Estas interacciones dan lugar a regularidades macroeconómicas y normas de comportamiento que, a su vez, se retroalimentan y determinan dichas interacciones locales. El resultado es un sistema dinámico formado por una cadena casual y recurrente que conecta comportamientos individuales que interactúan a través de redes sociales [Wat+07].

La metodología básica utilizada en la aplicación de los ABMs a los problemas de comportamiento de los consumidores consiste en el modelado de un conjunto de agentes sociales que representan a consumidores con comportamientos y propiedades propias que interactúan con el entorno y sus conexiones sociales de forma intensa. La descripción del comportamiento de cada uno de los agentes individuales del sistema viene dada por reglas simples como sus necesidades de compra, la presión social o sus características socio-demográficas. Además, los agentes están conectados mediante una red social que permite la aplicación de herramientas de análisis de redes sociales y Ciencia de Datos. La red social complementa la simulación de comportamientos emergentes y complejos que ocurren en los ABMs [CSM10] (por ejemplo, la difusión de información y la comunicación viral) y que generan procesos de difusión como el *word-of-mouth* (WOM).

Por esta razón, los ABMs se han aplicado con gran éxito en el marketing y la simulación de mercados [ZWD09; RR11] para entender cómo las relaciones entre los clientes impulsan la dinámica del mercado y también, junto con metaheurísticas como los algoritmos genéticos (GAs, por sus siglas en inglés), para calibrar y validar los modelos obtenidos con el fin de simular escenarios hipotéticos de mercado (escenarios *what-if*, en inglés). Sin embargo, la falta de realismo (no se maneja de forma adecuada la información de los agentes ni de su entorno), por una parte, y la calibración, validación y optimización (no se modelan adecuadamente los problemas existentes en entornos reales de toma de decisiones para marketing), por otro, impiden que los modelos teóricos y los de investigación básica sean ampliamente utilizados y aceptados como sistemas de apoyo a la toma de decisiones en las empresas.

Para hacer frente a los problemas mencionados, se deben emplear técnicas de inteligencia

computacional como los GAs para la optimización [Gol89; Tal09], Big Data [LD08; BL13], y Ciencias de Datos, o técnicas de validación de sistemas complejos para diseñar modelos de simulación social más avanzados y ABMs más realistas que proporcionen decisiones válidas y útiles para el personal responsable de la toma de decisiones en empresas y organismos.

El objetivo de esta tesis doctoral es combinar ABMs, técnicas de Ciencia de Redes y algoritmos avanzados de optimización heurística para construir herramientas de simulación de mercados más potentes, útiles y robustas que permitan a los modeladores realizar simulaciones sociales de forma eficiente y eficaz. Por ello, desarrollamos nuevas funcionalidades que permitan aumentar el potencial de la tecnología de modelado basado en agentes existente y proponer herramientas computacionales que supongan un avance en el estado actual de la técnica. Entre los métodos desarrollados se encuentran i) la selección de *influencers* multiobjetivo de agentes en campañas de marketing viral (VM, por sus siglas en inglés), que permite lidiar con restricciones económicas dentro de entornos de mercados y ii) la calibración multimodal para aumentar la capacidad de validación de los modelos y asegurar su fiabilidad.

## 1.1. Justificación

Como se ha comentado previamente, los ABMs de consumo existentes necesitan de mejores técnicas para su calibración y validación. Sin una mejora sustancial que incremente la confianza en las salidas de los modelos es prácticamente inviable su aplicación y su uso para la toma de decisiones en entornos tan competitivos y donde se mueve un volumen de inversión tan alto como los de marketing. Es necesario, por tanto, desarrollar modelos capaces de representar y simular de forma fidedigna las dinámicas de los mercados reales e incrementar la confianza que en ellos tienen los responsables en la toma de decisiones de las empresas. Aparte de esto, los problemas del mundo real exigen la exploración de entornos de alta dimensionalidad y la optimización conjunta de varios objetivos en conflicto en tiempo real para ser aplicables y rentable para las empresas del sector.

En lo referente a las técnicas de VM que se han aplicado sobre ABMs, éstas son exclusivamente

monoobjetivo [SRW10] y, por tanto, están muy limitadas a la hora de encontrar soluciones aplicables en un entorno real. De esta forma, se hace necesario proponer métodos avanzados capaces de lidiar con objetivos en conflicto. Estos nuevos métodos cubrirán dos requisitos de robustez como son los de obtener soluciones válidas para diferentes escenarios y conseguir una o un conjunto de soluciones de calidad en un entorno incierto.

La calibración y validación de ABMs es un proceso costoso y complejo que también requiere de la exploración de grandes conjuntos de parámetros que interactúan de forma no lineales entre sí y que hace inviable su resolución manual o usando métodos exactos. Los algoritmos de optimización evolutiva multiobjetivo y multimodal han demostrado un gran rendimiento a la hora de encontrar soluciones a problemas complejos, en los que los espacios de búsqueda son extensos y no lineales. Por un lado, los algoritmos evolutivos de optimización multiobjetivo (EMO, por sus siglas en inglés), han sido aplicados con éxito sobre problemas que requieren del tratamiento de varios criterios conflictivos como la calibración de ABMs [MCC19; MCC21], resolviéndolos con éxito [CLVV+07]. Esta es una de las razones del gran desarrollo de los campos del modelado y la toma de decisiones multicriterio [Cha+85]. Por otro lado, la naturaleza multidimensional del problema de calibración de ABMs y la presencia de varias soluciones subóptimas sugieren el uso de algoritmos evolutivos multimodales (MMEAs, por sus siglas en inglés) para su resolución. Los MMEAs son un enfoque útil cuando se buscan múltiples configuraciones óptimas de parámetros para reforzar la validación de los modelos y las decisiones que un experto podría tomar a partir de los resultados obtenidos [Chi+17].

El buen desempeño de los algoritmos EMO y de los MMEAs para la resolución de problemas reales refuerzan su uso sobre los problemas planteados en esta tesis doctoral.

## 1.2. Objetivos

A modo de resumen, el principal objetivo de esta tesis consiste en el diseño de ABMs realistas para el campo del marketing que ayuden a los expertos en su toma de decisiones. Para lograrlo, hemos de cumplir los siguientes tres objetivos específicos:

- En nuestro **primer objetivo** pretendemos ampliar y mejorar el funcionamiento de un reconocido ABM para representación de un mercado virtual de consumidores añadiéndole incertidumbre a los agentes del modelo de tal forma que necesitan de procesos de difusión de conocimiento en redes para adquirir información acerca de los productos ofertados en el mercado cuando son adquiridos.
- Un **segundo objetivo** centrado en el diseño de un método avanzado de VM para la selección de *influencers* en campañas de VM basado en algoritmos EMO que permita maximizar la difusión de productos y/o noticias a través de redes sociales. Actualmente, la literatura relacionada con este tema se enfoca en el uso de algoritmos monoobjetivo (GAs, por ejemplo[SRW10]) para maximizar el beneficio obtenido en las campañas de marketing. Nuestro método considera la optimización simultánea de varios objetivos en conflicto como el beneficio y el coste de las campañas obteniendo así resultados más robustos y realistas.
- Un **tercer objetivo** en el que proponemos un método avanzado de calibración basado en MMEAs para calibrar ABMs para marketing. El método desarrollado es capaz de buscar soluciones en un entorno complejo con parámetros relacionados de forma no lineal y de devolver diferentes configuraciones óptimas con ajustes similares. De esta forma, los modeladores pueden seleccionar las que mejor se adaptan a sus necesidades o aumentar las alternativas que tienen para apoyar sus decisiones, mejorando a su vez la validación y la robustez del modelo calibrado.

### 1.3. Estructura

Esta tesis doctoral se divide en tres partes: Fundamentos, Propuesta y Observaciones finales, además de la presente introducción. La primera parte introduce al lector en los fundamentos de varias teorías y metodologías comunes y necesarias para el resto de partes de esta tesis doctoral. En el capítulo 2 se revisan los fundamentos, la historia y el estado del arte del modelado basado en agentes, de los sistemas complejos y de la Ciencia de Redes, haciendo un breve repaso de las topologías de red más conocidas y de las métricas asociadas. A continuación, en el capítulo 3, se revisan los fundamentos teóricos de las técnicas heurísticas de optimización multiobjetivo y

multimodales basadas en algoritmos evolutivos utilizadas en esta tesis doctoral para optimizar campañas de VM (en el caso de los algoritmos EMO) y para calibrar ABMs y modelos basados en la teoría de balance energético (en el caso de los MMEAs).

La segunda parte (Propuesta) desarrolla todos los trabajos realizados durante la tesis doctoral y está formada por cuatro capítulos. Por un lado, el capítulo 4 describe las mejoras realizadas sobre el modelo *CONSUMAT*, un ABM establecido y con una base sólida en el área de la simulación social y el marketing, al que se le proporciona un *awareness* o reconocimiento de marca, incertidumbre en la decisión de compra y difusión de información vía red social. Seguidamente, el capítulo 5 expone nuestro método de VM basado en algoritmos EMO. Este capítulo trata la aplicación de técnicas de inteligencia artificial para, por una parte, identificar *influencers* como semillas o difusores de innovaciones a través de la red social que interconecta a los consumidores del modelo *CONSUMAT* que extendimos en el capítulo 4 y, por otra, expone cómo hacerlo de forma realista y efectiva mediante la optimización conjunta de objetivos enfrentados tales como el beneficio y el coste derivados de la selección de los *influencers* como canal de publicidad. Este capítulo incluye, además, una sección donde se aborda el uso de GAs para la optimización monoobjetivo de campañas de VM.

Por su parte, el capítulo 6, está dedicado al desarrollo del método propuesto para la calibración de ABMs basado en MMEAs. En él analizamos el potencial de los MMEAs para buscar soluciones óptimas y diversas durante la calibración de un ABM para marketing poniendo de manifiesto las ventajas que la optimización multimodal puede brindar a los modeladores durante el análisis y exploración del entorno de soluciones de sus modelos y a la hora de aumentar la capacidad de validación y la fiabilidad de dichos modelos.

Finalmente, en el capítulo 7, exponemos un trabajo adicional en el que proponemos una adaptación de la calibración basada en MMEAs para modelos ecológicos poblacionales (basados en la teoría del balance energético dinámico, concretamente), que simulan dinámicas biológicas en especies animales y microbianas y que representan una metodología de simulación basada en el individuo (*individual-based*, en inglés) muy similar a la utilizada en los ABMs. En este capítulo demostramos la aplicabilidad de la calibración multimodal fuera de los ABMs y sus

beneficios a la hora de encontrar soluciones igualmente preferibles y de alta calidad en problemas multidisciplinares. Además, hacemos una introducción a la herramienta *MultiCalib4DEB* que desarrollamos con el objetivo de que los modeladores de la teoría del balance energético pudiesen utilizar para la calibración de sus especies.

Por último, la tercera parte (Observaciones finales), consta de un único capítulo (capítulo 8) en el que exponemos las conclusiones obtenidas en los trabajos científicos realizados durante la presente tesis doctoral. También se presentan los posibles trabajos futuros y se muestran las aportaciones científicas realizadas en el transcurso de la formación doctoral.





# Parte I

## Fundamentos



# Simulación social

“Puede que llegue el día en el que el valor de los hombres falle cuando abandonemos a nuestros amigos y rompamos todos los lazos de compañerismo. Pero hoy no es este día.”

---

Viggo Mortensen (Aragorn),  
The Lord of The Rings

## 2.1. Sistemas complejos

Las sociedades son agrupaciones de individuos que se relacionan entre sí y que conviven siguiendo reglas de conducta y organización. Estos individuos forman redes de relaciones interpersonales basadas en patrones de conducta y reglas de organización y cooperación que les permiten desarrollarse y cumplir objetivos de distinta índole. Los millones de interacciones simples que se producen entre ellos son capaces de generar dinámicas y comportamientos complejos. [Saw05]. La cuestión fundamental del enfoque de los sistemas complejos es que, desde un punto de vista macroscópico, el desarrollo de órdenes políticos, sociales o culturales no es sólo la suma de interacciones individuales, sino el resultado colectivo de interacciones no lineales.

Dado el importante papel que desempeñan los sistemas sociales en nuestra vida, su comprensión, cuantificación, predicción y eventual control ha cobrado una importancia capital, convirtiéndose en uno de los principales retos científicos e intelectuales del siglo XXI. La imposibilidad de

abordarlos con los enfoques tradicionales ha dado lugar a la ciencia de la complejidad [MT10], una materia interdisciplinar dedicada al diseño de nuevos protocolos, modelos y formalismos para la reconstrucción de los fenómenos emergentes y dinámicas que caracterizan a los sistemas complejos y a las ciencias sociales computacionales [Laz+09], sistemas del mundo real cuyos componentes están fuertemente interrelacionados, que se autoorganizan y que son dinámicos [MT10].

Sin embargo, los problemas reales son cada vez más complicados y aumentan su dimensionalidad, lo que dificulta enormemente su resolución. Estos problemas se caracterizan generalmente por las siguientes características, independientemente de su dominio concreto:

1. **La incertidumbre:** La información disponible es a menudo imprecisa, incierta, ruidosa, o puede haber una aguda falta de información. El objetivo del problema de modelado o de decisión es ambiguo y la estructura del problema puede estar poco especificada y el entorno del problema puede no ser estacionario, sino cambiante.
2. **La dimensión del problema:** Que exige tiempos computacionales prohibitivos para lograr la trazabilidad del problema.
3. **La interpretabilidad de los resultados:** Cada vez es más necesario comprender los procesos mentales humanos subyacentes para gestionar la información en un estilo más orientado al ser humano [GP14].

Entre los formalismos y teorías nacidos de la Ciencia de la Complejidad podemos encontrar la Ciencia de Redes [Str01], una disciplina de la Ciencia de Datos basada en el procesamiento de información en el que no solo el contenido es relevante sino también las relaciones entre los elementos constituyentes del sistema (*linked data*). La Ciencia de la Complejidad se ha aplicado en ámbitos específicos como las finanzas [Bae07], las ciencias sociales [GT00; Byr02], la política [Lav05], la administración [MK03; Hay15], la economía [Art09], el turismo [CHBG21], el marketing [HS08; Ove+19], las dinámicas de opinión [XWX11] y los fenómenos migratorios [Dha+20].

## 2.2. Modelos basados en agentes para simulación social

Una de las técnicas de modelado más destacadas para representar con precisión los fenómenos emergentes que se producen dentro de los sistemas complejos es el modelado basado en agentes [Bon02]. El modelado basado en agentes se centra en el modelado de sistemas sociales como, por ejemplo, los procesos epidemiológicos, las redes sociales, los sistemas políticos, las actividades comerciales o la difusión de información, y es más apropiado que otras herramientas como el modelado analítico o estadístico o el modelado basado en dinámica de sistemas [VB10] cuando se trata de un entorno complejo y dinámico y la medida de interés es un resultado emergente de las interacciones sociales [RR11].

Los ABMs [Eps99; GT05; Eps07; MN14; WR15] permiten la simulación de acciones e interacciones entre individuos autónomos dentro de un entorno para estudiar y determinar qué efectos producen en el conjunto del sistema. Como técnica, combinan propiedades de la teoría de juegos, sistemas complejos, sociología computacional, sistemas multi-agente y programación evolutiva.

Esta técnica, al igual que otras técnicas como el modelado de sistemas, tiene el potencial de complementar modelos no formales de sistemas complejos y modelos matemáticos abstractos que dependen de múltiples variables. Muy a menudo, la representación de sistemas complejos carece del rigor formal requerido para valorar su coherencia lógica e impide el poder generalizar a partir de ellos. Por su parte, los modelos matemáticos suelen ser poco realistas debido a las fuertes restricciones que son necesarias para poder resolverlos e interpretarlos. El uso de técnicas de modelado computacional como los ABMs reúnen las características necesarias para construir modelos capaces de combinar la parte descriptiva de los modelos verbales con el rigor formal de los modelos matemáticos más abstractos.

A muy alto nivel, los ABMs simulan la interacción de entidades autónomas [Bon02; Lib+10] (agentes) en un intento de recrear y predecir las acciones por las que se producen fenómenos complejos. Las interacciones entre los distintos elementos del sistema juegan un papel esencial en la evolución de éste, tanto desde un punto de vista “macro” (comportamiento observable del sistema como un todo), como desde un punto de vista “micro” (cada una de las entidades

del sistema que se adaptan a los cambios que perciben a través de esas interacciones con otras entidades). Estas interacciones pueden incluir una estructura espacial para que los agentes sólo interactúen con los agentes de su entorno. Por lo tanto, los ABMs suelen considerar una red social artificial (SN, por sus siglas en inglés) [WS98; BA99; SZM13] que permite las interacciones entre los agentes vecinos. Esto permite a los ABMs replicar de forma realista la dinámica de los procesos epidemiológicos [FMWFM11; Sil+20; Hoe+20], de difusión de información [BCZ10; Kie+12] o los mecanismos de WOM [ZGG11; CR17]. Además, los agentes de la población pueden estar expuestos a influencias externas que representan las interacciones con su entorno o su exposición a acciones externas.

La construcción de ABMs para simular dinámicas reales puede seguir varios propósitos como el verificar y probar las propiedades, beneficios o consecuencias de un determinado producto, plan o acción antes de su aplicación en un entorno real [CHBG21]. Para ello, las simulaciones de ABMs permiten a los modeladores definir escenarios *what-if* [Lin92; An12; HAN13; LJS16; TSN20; Sil+20] a partir de modificaciones de los parámetros de un modelo calibrado y validado previamente. A través de estos escenarios, los modeladores pueden obtener información basada en el comportamiento del sistema real que sería muy complicada de obtener por otras vías. Este hecho ha incrementado el uso de los ABMs en los últimos años [MCC19; Sil+20; CHBG21].

La correcta implementación de un modelo requiere de un análisis pormenorizado del sistema que se pretende simular con tal de evitar simplificaciones en las entidades que lo componen y en sus interacciones. Estas reglas diferencian a esta técnica de otros modelos que también buscan estudiar sistemas complejos [BBC08]. En el modelado basado en agentes se deben identificar claramente los tipos de agentes que representan las entidades autónomas del sistema, sus relaciones e interacciones y el entorno en el que están inmersos. Los agentes actúan siguiendo percepciones u objetivos con un conocimiento limitado del medio. Este conocimiento sumado al hecho de que se encuentren en un medio limitado les proporcionan la capacidad de experimentar procesos de aprendizaje, adaptación al medio y reproducción en el mismo.

Las características de los agentes de un ABM son muy diversas y están estrechamente ligadas al sistema modelado, aunque destacan de forma genérica las siguientes:

- **Independencia:** Los agentes representan entidades identificables con significado y funciones dentro del sistema, con un conjunto de características y reglas que gobiernan tanto su comportamiento como su capacidad de toma de decisiones.
- **Autonomía:** Los agentes deciden independientemente sus acciones teniendo en cuenta los objetivos que persiguen y en conocimiento del que disponen en cada momento. Normalmente un agente no posee un conocimiento completo del sistema, lo que añade un grado de incertidumbre al mismo, además de que sus acciones pueden no surtir el efecto deseado por la acción concurrente de los demás agentes.
- **Adaptatividad:** Los agentes tienen la capacidad de adaptar su comportamiento a las circunstancias basándose en su conocimiento y su experiencia. Por tanto, deben estar dotados de una memoria.
- **Heterogeneidad:** Los agentes tienen una serie de rasgos que les sirven para reconocer y distinguir a otros agentes (grado de importancia en la red social a la que pertenecen, tener distintos rangos dentro del modelo, etc). Estas características diferenciadoras permiten que los agentes puedan interactuar de distinta forma con unos agentes u otros dependiendo de cómo los categorizan por sus rasgos o afinidades, lo que da lugar a diferentes comportamientos dentro del modelo.
- **Interactividad:** Los agentes están situados en un entorno en el que pueden moverse e interactuar con otros agentes y con el entorno en sí.

La complejidad de los modelos que se desarrollan en la actualidad requiere de la colaboración de amplios equipos interdisciplinarios que sean capaces de abordar el problema evitando restar realismo a los modelos. De esta forma, se puede obtener un modelo bien fundado y sólido sobre el que poder inferir resultados por medio de la simulación del modelo. No obstante, comprender los comportamientos emergentes necesita de la ejecución de simulaciones masivas para comprender con el menor error posible cómo surgen en base a las decisiones tomadas por una mezcla heterogénea de factores humanos. Para ello es necesario el uso de ABMs a gran escala con los que hacer frente a la dimensión de sistemas complejos para la obtención de teorías válidas derivadas de la simulación.

Afortunadamente, la potencia de la computación avanza a un ritmo vertiginoso y el desarrollo de protocolos de modelado basado en agentes en entornos informáticos de alto rendimiento es una técnica en auge en los últimos años [LD08; BL13; Kan15; Tei+17] que, combinada con técnicas tales como la Ciencia de Datos, la Ciencia de Redes, el aprendizaje automático o la teoría de grafos es capaz de generar aplicaciones con las que sacar partido a los ABMs, aprovecharlos para resolver problemas y cubrir objetivos empresariales.

### 2.3. Ciencia de Redes y redes sociales

La Ciencia de Redes es un campo académico interdisciplinar que estudia las estructuras y dinámicas (variables e imprevisibles) que representan fenómenos de redes complejas, como las redes de telecomunicaciones, las redes informáticas, las redes biológicas, las redes cognitivas y las redes sociales, que conduce a modelos predictivos de dichos fenómenos [BA99; NBW06].

Las SN son un tipo concreto de redes complejas [WF94] formadas por un conjunto de actores y uno o más lazos o relaciones definidos entre ellos. Su análisis y estudio originaron el campo del análisis de redes sociales (SNA, por sus siglas en inglés), que se centra en el modelado y el estudio de fenómenos sociales, dando una importancia preponderante a las interrelaciones existentes entre las entidades que interactúan dentro de las redes. En el SNA convergen diversas disciplinas tales como las ciencias sociales y del comportamiento, las matemáticas, la biología, la estadística y ciencias de la computación.

El auge en la adopción y uso de las nuevas tecnologías, unido a la rápida implantación de Internet a nivel mundial, a la aparición de aplicaciones de interacción social y al aumento en la capacidad de computación han situado al SNA como un campo de estudio potente, con proyección y con capacidad para influir notablemente en las dinámicas sociales y políticas de nuestros tiempos. Dado que el foco de esta tesis doctoral es trabajar con simulación social, las SNs son idóneas como base para nuestros modelos y nos centraremos en ellas a lo largo de este capítulo.



### 2.3.1. Las redes sociales como base de los ABMs

Las SNs son una parte fundamental del modelado basado en agentes ya que representan de una forma estructurada a las entidades de un sistema complejo junto con las conexiones e influencia entre los entes que lo conforman. Así, una SN es un grafo [Tru13] o red donde los nodos son entes sociales y los enlaces son las conexiones entre ellos. Estas relaciones pueden ser dirigidas entre pares de nodos o entre un mismo nodo y también pueden estar ponderadas siguiendo alguna medida de importancia/dependencia entre dichos nodos.

Las SNs son tremendamente útiles para estudiar comportamientos emergentes en los ABMs, para comprobar cambios en las dinámicas del modelo en función de la estructura de la red o para medir el impacto que tiene ésta en los resultados obtenidos [NC16; Wil+20]. Por ejemplo, algunos autores han tratado de encontrar un nexo de unión entre la sociología experimental y la sociología computacional mediante la unión del modelado basado en agentes y las SNs [MW02]. Sus trabajos se enfocaron en la construcción de ABMs empíricos donde los agentes computacionales se comportaban de manera significativamente similar a como lo hacen los agentes reales en los experimentos realizados en laboratorio [RR11; WR15; Boe+15; BS15]. Estos modelos exploran estructuras complejas de interacción entre los agentes que difícilmente pueden llevar a cabo en los estudios de campo para analizar los efectos de tales estructuras sobre diferentes variables sociales.

La estructura de las SNs influye poderosamente en el funcionamiento de los sistemas modelados [Zha+15] y su incorporación a las simulación basada en agentes permite analizar la aparición de distintos fenómenos biológicos [ES+12], organizativos [Du+17], sociales [HG09; Wil+20] o cognitivos [Mom22]. Por tanto, es necesario el uso de estructuras complejas de redes [WS98; BA99; ZSM12] para infundir más realismo a los modelos empíricos basados en agentes. A partir de estos modelos, se puede investigar cómo el comportamiento de los agentes y la dinámica de la red social artificial subyacente se afectan mutuamente.

### 2.3.2. Topologías de redes sociales

Las SNs están caracterizadas por la forma en la que se interconectan los nodos que las integran. Los patrones observados en la conexión entre los nodos de una SN forman la denominada topología de red. Entre las topologías de red más destacadas, estudiadas y aplicadas de la literatura se encuentran las siguientes:

#### 2.3.2.1. Redes aleatorias: Modelo Erdős-Rényi

Las redes aleatorias [ER+60] representan uno de los métodos empleados en la generación de grafos aleatorios. Este modelo basa su funcionamiento en la creación iterativa de nodos que son enlazados con igual probabilidad con el resto de nodos de la red, manteniendo una independencia estadística con ellos.

El modelo conceptual diseñado por Erdős y Rényi considera  $N$  nodos distribuidos de forma aleatoria en una red sin enlaces. Seguidamente, se crean enlaces entre pares de nodos que, en pasos posteriores, se enlazarán de forma aleatoria con otras parejas de nodos (descartando aquellos nodos que ya se encuentren enlazados). Repitiendo el proceso un número  $M$  de veces se habrán establecido como máximo  $M$  enlaces entre las parejas de nodos. De esta forma, si  $M$  es un valor pequeño con respecto a  $N$ , muchos de los nodos se encontrarán aislados (desconectados del resto) y otros estarán formando pequeñas islas. Por el contrario, si  $M$  es grande en comparación con  $N$ , es altamente probable que la mayor parte de los nodos de la red se encuentren conectados entre sí formando una única componente gigante [Jan+93]. Cuando se da esta forma de conexión entre los nodos, aparecen propiedades específicas en la distribución de grado  $P(k)$  como, por ejemplo, que sigue la forma de una distribución de Poisson<sup>1</sup>. Durante los años 60, 70 y 80 se pensó que las redes con esta característica eran las más adecuadas para describir ciertas redes complejas reales. Con el nacimiento de Internet y el desarrollo de la tecnología, pronto se vio que esto no era del todo cierto.

Las medidas básicas asociadas a las redes aleatorias son  $N_p = \binom{N}{2} = \frac{N(N-1)}{2}$  y  $p_c = \frac{M}{N_p} = \frac{2M}{N(N-1)}$ ,

<sup>1</sup>[https://en.wikipedia.org/wiki/Poisson\\_distribution](https://en.wikipedia.org/wiki/Poisson_distribution)

donde  $N_p$  es el número de enlaces de la red y  $p_c$  la probabilidad de conexión entre pares de nodos.

### 2.3.2.2. Redes libres de escala: Modelo Barabási-Albert

Las redes libres de escala [BA99] son un tipo específico de red compleja en las que las conexiones entre sus nodos siguen la ley de la potencia. Es decir, la red está formada por una minoría de *hubs* y por una mayoría de nodos con un número de conexiones bajo.

El modelo Barabási-Albert (BA) es un algoritmo para generar redes aleatorias complejas libres de escala empleando una regla o mecanismo denominado conexión preferencial. El modelo BA ha sido utilizado para explicar el crecimiento y evolución de innumerables tipos de SNs, tecnológicas o biológicas, con independencia de la naturaleza de las mismas.

En el modelo BA, la probabilidad  $P(k)$  de que un nodo de la red esté conectado a  $k$  nodos es proporcional a  $k^{-\gamma}$ , es decir, a la ley de la potencia en la que el exponente *gamma* depende del tipo específico de red. Para la mayor parte de los sistemas se ha medido que  $\gamma$  se encuentra comprendido en el rango  $2 < \gamma \leq 3$ . Cuando  $\gamma \leq 2$ , la varianza de la distribución del número de enlaces por nodo es infinita. Las redes libres de escala se generan con el algoritmo de unión preferencial (*preferential attachment* en inglés) descrito en [BA99].

### 2.3.2.3. Redes de mundo pequeños: Modelo Watts-Strogatz

Nacidas en los años 60 tras el “experimento de mundo pequeño” realizado por el psicólogo Stanley Milgram [Mil67] y formalizadas por Duncan Watts y Steven Strogatz en 1998 [WS98], son un tipo de grafo en el que cualesquier par de nodos de la red se comunican por un camino de nodos intermedios relativamente pequeño. Conforme se añaden nuevos nodos a la red, la distancia máxima entre pares de nodos crece logarítmicamente. Estas redes se caracterizan por presentar valores altos de agrupamiento entre nodos, existiendo una alta probabilidad de conexión entre dos nodos no conectados directamente entre sí mediante conexiones entre otros nodos. Las redes de mundo pequeño aparecen con frecuencia en sistemas reales como es el caso,

por ejemplo, de la topología de Internet. Este hecho ha favorecido la aplicación de este tipo de redes en diferentes áreas de la ciencia como el modelado de SNs, biología, epidemiología, etc.

La generación de redes de mundo pequeño se centra en dos propiedades:

- **El fenómeno del mundo pequeño:** Cualquiera dos nodos de la red se comunican por un camino de nodos intermedios de longitud relativamente pequeña. Se comprueba que la distancia máxima entre dos nodos crece logarítmicamente con el número de nodos en la red.
- **Alto coeficiente de clustering:** Si dos nodos no están conectados directamente entre sí existe una alta probabilidad de que lo estén mediante la intervención de otros nodos vecinos.

El efecto del mundo pequeño es sencillo de medir en cualquier red, basta con buscar las distancias entre cada par de nodos en la red y calcular su distancia media. En un grafo aleatorio el grado medio es  $\langle k \rangle$  y coincide con el valor medio de vecinos, el de vecinos a distancia 2 es de  $\langle k \rangle^2$  y el de vecinos a distancia 3  $\langle k \rangle^3$ , etc. La generación de redes siguiendo el modelo Watts-Strogatz comienza con una red formada por  $N$  nodos en forma de anillo en la que cada nodo está conectado a sus primeros  $k$  vecinos ( $k/2$  por cada lado). Cada vez que se añade un nuevo nodo se enlaza con el resto empleando una probabilidad  $p$  para cualquier nodo de la red. Una de las características principales de las redes de mundo pequeño es que su distribución de grado debía corresponderse con una distribución de Poisson, pero pronto se vio que esta topología de red puede tener distribuciones de grado que siguen una distribución exponencial (como las redes libres de escala).

### 2.3.3. Medidas en la Ciencia de Redes

El análisis de SNs es el proceso de investigación de las estructuras sociales mediante el uso de la Ciencia de Redes y la teoría de grafos y se centra en analizar los patrones de relación entre los nodos en los grupos que forman en la topología de una red. Son útiles para examinar la

estructura social y las interdependencias (o patrones de comportamiento/colaboración) de los individuos o las organizaciones. Existen varias medidas asociadas tanto a la Ciencia como al análisis de redes que pueden dividirse en dos niveles en cuanto su nivel en la red: Métricas a nivel de nodo (locales) y métricas a nivel de grupo (globales).

### 2.3.3.1. Medidas locales

Las medidas locales se centran en el análisis de relaciones a nivel de nodo por lo que se mide la importancia de dicho nodo en su entorno más cercano. Todas ellas están basadas en el concepto general de centralidad (redes no dirigidas) o prestigio (redes dirigidas), una medida general de la posición de un actor en la estructura global de la red social. Entre las medidas locales más utilizadas dentro del análisis de SNs se encuentran el grado, la intermediación (betwenness) y la cercanía (closeness) [FRM79a; FRM79b; OAS10], la excentricidad (eccentricity) [HH95], la centralidad de vector propio [Bon87] y el coeficiente de agrupamiento (clustering [HL71]). Su importancia dentro del análisis de SNs queda resumida en los siguientes puntos:

- **Centralidad de grado:** Mide el número de relaciones entrantes o salientes (o ambas) de un nodo, según la orientación de una proyección de relación. Es una medida importante ya que permite identificar nodos importantes de la red en base a su número de contactos. La centralidad de grado mide la importancia de las relaciones directas entre un nodo y sus vecinos más cercanos asumiendo que las conexiones entre sus nodos vecinos no importan.
- **Centralidad de intermediación:** Es una forma de detectar la cantidad de influencia que tiene un nodo sobre el flujo de información en un grafo. Suele utilizarse para encontrar nodos que sirvan de puente entre partes de un grafo. El algoritmo calcula los caminos más cortos no ponderados entre todos los pares de nodos de un grafo. Cada nodo recibe una puntuación, basada en el número de caminos más cortos que pasan por él.
- **Centralidad de cercanía:** Es una forma de detectar los nodos que son capaces de difundir información de forma muy eficiente a través de un grafo. La centralidad de cercanía de un nodo mide su distancia geodésida (distancias de los caminos mínimos) los demás nodos. Los nodos con un alto valor de cercanía tienen las distancias más cortas a todos los demás

nodos. Para cada nodo del grafo, el algoritmo de centralidad de cercanía calcula la suma de sus distancias a todos los demás nodos, basándose en el cálculo de los caminos más cortos entre todos los pares de nodos. La suma resultante se invierte para determinar la puntuación de centralidad de proximidad de ese nodo.

- **Centralidad de excentricidad:** La excentricidad de un nodo es una medida análoga a la cercanía que mide la distancia máxima (el camino mínimo más largo) desde dicho nodo a cualquier otro nodo dentro de una red o grafo. El valor inverso de la excentricidad (la máxima distancia geodésica) se utiliza como medida de la importancia de un nodo dentro de una red. Los nodos con un mayor valor de excentricidad se denominan nodos periféricos, los de menor valor forman el centro de la red.
- **Centralidad de vector propio:** Mide la influencia transitiva de los nodos de un grafo. La idea principal de esta medida es calcular el poder y el status de un nodo recursivamente a partir del poder y el status de sus vecinos. Así, una puntuación alta de la centralidad de vector propio significa que un nodo está conectado a muchos nodos que a su vez tienen muchas conexiones a otros nodos. Es una versión más elaborada de la centralidad de grado al asumir que no todas las conexiones tienen la misma importancia. No se tiene en cuenta la cantidad sino la calidad de las mismas.
- **Coefficiente de clustering (agrupamiento):** Describe la tendencia de las personas a formar vínculos dentro de los grupos sociales, donde los contactos de uno también están conectados entre sí (propiedad de transitividad de una red [Blo15]). Esto significa que las conexiones indirectas entre individuos -es decir, dos nodos no conectados que están vinculados a un tercer nodo, o que están conectados por un camino doble- tienden a convertirse en conexiones directas con el tiempo. El coeficiente de clustering puede ser global, a nivel de toda la red, o local, a nivel de cada individuo.

### 2.3.3.2. Medidas globales

Al contrario que las medidas locales, las medidas globales se aplican a nivel de toda la SN para estudiar la red a un nivel agregado. Existen varias medidas globales en análisis de SNs, la mayoría de ellas empleadas para analizar cualquier tipo de red compleja tales como el diámetro

y radio, la distancia y el grado medios, la densidad, el coeficiente de agrupamiento global y la reciprocidad. El objetivo de cada una de estas medidas queda resumido en los siguientes puntos:

- **Diámetro:** Longitud del camino mínimo más largo de la red. Equivale al valor máximo de excentricidad para todos los nodos de una red. En el contexto del análisis de SNs, esta métrica da una idea de la proximidad entre pares de actores en la red, indicando cómo de lejos están en el peor de los casos. Las redes más dispersas suelen tener un mayor diámetro que las más densas al existir menos caminos entre cada par de nodos.
- **Radio:** Es el valor mínimo de excentricidad para una red. Representa el mejor caso de un proceso de transferencia sobre la red.
- **Distancia media:** Es la distancia geodésica entre pares de nodos de un grafo. Esta medida muestra cómo de lejos están los distintos nodos en promedio. En el análisis de SNs representa la eficiencia del flujo de información en una red.
- **Grado medio:** Es el número medio de aristas por nodo en el grafo y mide la conectividad total en la red.
- **Densidad:** Es la proporción de enlaces existentes comparada con los enlaces posibles en una red social y mide el grado de conectividad en dicha red social a nivel global. En las redes desigualmente conectadas, la densidad hace posible medir las áreas más o menos enlazadas de la red y detectar el vecindario de un nodo en base a su cohesión de acuerdo con la proporción de enlaces entre los nodos.
- **Coeficiente de clustering medio** Como comentamos anteriormente, el coeficiente de agrupamiento o de clustering es una medida de la tendencia a agruparse que tienen los nodos de una red. En la mayoría de redes reales, y en particular las SNs, los nodos tienden a crear grupos compactos que se caracterizan por una densidad relativamente alta que tiende a ser mayor que la probabilidad media de un enlace aleatorio entre dos nodos. El coeficiente de agrupamiento medio es una medida global que indica la probabilidad de que dos vecinos de un nodo de la red escogido aleatoriamente estén conectados entre sí. Las SNs reales son redes de mundo pequeño (ver Sección 2.3.2.3) y suelen tener valores de coeficiente de clustering medio altos lo que implica que la transitividad entre nodos

aparece más y con más fuerza, incrementando la probabilidad de que se formen cliques<sup>2</sup>.

- **Reciprocidad:** La reciprocidad en un grafo dirigido mide la tendencia de pares de nodos a tener conexiones mutuas entre ellos [Blo15]. La forma más habitual de calcularla es como un ratio entre el número de conexiones (díadas) mutuas (conexiones tanto de entrada como de salida en un nodo) y el número de conexiones totales en la red, las mutuas y las asimétricas (conexiones que presentan un arco en un único sentido). La reciprocidad mide la probabilidad de que dos actores de la red apunten el uno al otro.

Además de las medidas anteriores, existen otras medidas denominadas medidas de centralización que son agregaciones a nivel de grupo de las medidas locales de centralidad (ver Sección 2.3.3.1) que permiten comparar redes tales como la centralización de grado, la centralización de intermediación o la centralización de cercanía [FRM79a; FRM79b].

### 2.3.4. ABMs aplicados a estrategias de marketing

El marketing es un sector en constante evolución [CI06; VT10; MBA14; SV18] que siempre ha tenido un objetivo bien definido: Identificar y satisfacer las necesidades sociales humanas de forma rentable [KS19]. La clave para poder cumplir este objetivo es conocer cuál será la reacción de un cliente ante determinados estímulos publicitarios para definir estrategias adecuadas que permitan beneficiar a la(s) marca(s) que las realizan no sólo en cuanto al volumen de venta generado sino en cuanto a un refuerzo o incremento en la notoriedad e imagen de marca. La proliferación de los nuevos medios de comunicación, la explosión de datos disponibles y la creciente influencia de los medios digitales y sociales ha cambiado la forma en la que los consumidores reciben y procesan la información. Asimismo, una competencia cada vez más agresiva ha supuesto un cambio de paradigma en el que los tradicionales modelos de ayuda a la toma de decisiones de marketing basados en la econometría y las matemáticas han perdido parte de su eficacia.

Para poder explorar, comprender y aprovechar el comportamiento de consumidor se han

---

<sup>2</sup>Un clique es un subgrafo en el que cada vértice está conectado a todos los demás vértices del subgrafo. El tamaño de un clique es el número de vértices que contiene.



desarrollado diversas técnicas entre las que destacan el Marketing Mix Modeling (MMM) [Gat93; NRW05] y los modelos de Atribución Digital [GT15]. El MMM es una técnica de modelado estadístico que comenzó a usarse en los años 90 para evaluar la relación entre el gasto en marketing en distintos medios offline (TV, radio y prensa, entre otros) y los resultados obtenidos por el mismo (ya sea en términos ventas, adquisición de clientes, viralidad u otros indicadores de rendimiento). Pese a haber sido el estándar de la industria durante la última década, los modelos basados en MMM muestran serias limitaciones [Cai08] a la hora de representar dinámicas de consumo fiables y replicables. Es por ello que carecen de la granularidad suficiente para bajar al nivel básico de decisión (los consumidores) y son incapaces de adaptarse al ritmo frenético que exige el escenario actual caracterizado por el *real-time marketing*. Además, otra de las limitaciones de los modelos basados en MMM es su incapacidad para representar dinámicas a nivel local o individual para generar comportamientos agregados. Los modelos de MMM permiten medir interacciones entre distintos factores a nivel global (siguiendo un enfoque macro o *top-down*) lo que impide simular las micro-relaciones existentes entre los consumidores [RR11] de un modelo complejo y agregarlas posteriormente para estudiar su aportación a la dinámica global del sistema (enfoque micro o *bottom-up*).

En el extremo contrario, se encuentran los modelos de Atribución Digital que, a pesar de solventar algunas de las limitaciones de los modelos de MMM, son incapaces de integrar cualquier variable que afecte al proceso de comprar de los consumidores (su Customer Journey<sup>3</sup>) fuera del entorno online (inversiones en medios tradicionales, precio, promociones o presencia en la distribución), y que además ignoran el impacto que tienen en los consumidores aspectos tangibles tan relevantes como la oferta de los competidores que operan en el mercado o intangibles tan importantes como las imágenes de las propias marcas.

Tal y como se expuso en la Sección 2.2, los ABMs aparecen como una técnica idónea para representar las micro-relaciones entre individuos frente a las limitaciones de los modelos de MMM y de Atribución Digital. Los ABM se nutren de la información que los clientes obtienen de cuestionarios o encuestas a nivel de consumidor. Esta información aporta un conocimiento cuantitativo de las preferencias de los clientes o de la influencia que ejercen sobre ellos medios

---

<sup>3</sup><https://www.inboundcycle.com/blog-de-inbound-marketing/customer-journey>

como la web, las SNs electrónicas o la televisión. Mediante la representación de los consumidores a partir de sus datos individualizados usando ABMs es posible encontrar patrones de conducta y relaciones entre las variables que resultan de mucha utilidad para entender y aprovechar los mecanismos de consumo de un mercado.

Han sido muchas las aplicaciones de los ABMs a marketing [NY14]. Una de las aportaciones más conocidas fue la propuesta de modelo CONSUMAT por Marco Janssen y Wander Jager [JJ03; JJ12]. CONSUMAT considera y relaciona las preferencias personales de los consumidores de un mercado y la utilidad social de un producto (el número de consumidores que lo consumen dentro de dicho mercado) a la hora de representar la toma de decisiones y, por ende, la compra. El modelo CONSUMAT ha sido utilizado en diferentes aplicaciones como el estudio de dinámicas del hogar [JJV01], la gestión de inundaciones [BV03], la implantación del coche eléctrico [Kan14], la eficiencia energética [MPM18] o la difusión de los esfuerzos de conservación del suelo [VO+19].

Otros trabajos como [WMA13] estudiaron la mejora en la predicción de las dinámicas de difusión incorporando información acerca de la estructura de la red social subyacente de forma que futuros productos involucrarían subconjuntos de la misma red. Otros autores como [TRJ13] propusieron un enfoque para identificar las condiciones sistemáticas que son estables a través de las dinámicas difusión y, por tanto, “transferibles” al lanzamiento de nuevos productos dentro de una red determinada. Utilizando datos de aplicaciones de Facebook, los autores demostraron que la incorporación de esas condiciones sistemáticas mejora las previsiones previas al lanzamiento. Este trabajo tiende un puente entre las disciplinas de estadística bayesiana y el modelado basado en agentes al demostrar cómo los investigadores pueden utilizar las relaciones estocásticas simuladas dentro de sistemas complejos como entradas significativas para los modelos de inferencia bayesiana. La predicción de la difusión de nuevos productos ha sido abordada por otros autores como [XH16]. En su trabajo, se desarrolla un ABM construido sobre la red de influencia oculta sobre la que se difunde la innovación. El autor presenta un método eficaz para estimar los parámetros no estructurales de la red y formula un modelo logístico multinomial para identificar el tipo de red de influencia oculta para los datos de difusión.

---

# Metaheurísticas para calibración y optimización de ABMs

Quien vive sin pensar, no puede decir  
que vive.

---

Pedro Calderón de la Barca

## 3.1. Validación y calibración de ABMs

La validación de un modelo es el proceso de determinar hasta qué punto dicho modelo representa con exactitud el mundo real desde la perspectiva de su uso [Mac05]. El proceso de validación puede ser muy complejo, especialmente si la metodología de modelado implica la definición y configuración de muchos parámetros. A su vez, crear y configurar un modelo para un problema específico desde cero puede ser difícil para los diseñadores y los responsables de la toma de decisiones. Si algunos de los valores de los parámetros del modelo no pueden especificarse con la información y los conocimientos disponibles, el modelador debe estimarlos manualmente para simular correctamente la dinámica del sistema subyacente.

La calibración de ABMs implica la tarea de realizar cambios en el modelo (por ejemplo, los valores de los parámetros y las condiciones iniciales) para que refleje con precisión el comportamiento del sistema objetivo del mundo real. La calibración es una etapa imprescindible antes de la explotación de modelos computacionales en la que se requiere del ajuste de la salida de estos modelos con datos históricos reales para generar modelos que repliquen lo ocurrido en el pasado antes de generar salidas para escenarios *what-if*. El proceso de calibración es una

etapa crucial y muy compleja en la validación de ABMs [Oli03; Mac05; Sar10; RR11; Chi+17]. La calibración de un modelo puede ser realizada de forma manual por parte de un modelador mediante la simulación iterativa del modelo creado y el ajuste de sus parámetros en base a los resultados observados [BVL16a]. Sin embargo, este enfoque resulta impracticable para muchos modelos realistas que se caracterizan por considerar un amplio conjunto de parámetros con y sin interdependencias entre sí.

Las limitaciones operativas de la calibración manual se resuelven parcialmente aplicando un proceso de calibración automático en el que los valores de los parámetros del modelo son ajustados para que sus resultados casen con la realidad de los datos del escenario modelado [Oli03; Sar10]. De esta forma, cada configuración de parámetros  $X = (x_1, \dots, x_n)$  puede definirse como un vector de  $n$  variables de decisión. Sin embargo, los métodos de calibración automáticos tienen que lidiar con otro problema adicional: La presencia de interacciones no lineales entre los parámetros de los modelos computacionales. Frente a este problema, las técnicas de optimización no lineales como las metaheurísticas [Tal09] destacan como el mejor enfoque cuando se busca a través de la amplia gama de parámetros de los ABMs [SR14; Chi+17; MCC19; MCC21; Moy+21a]

La Figura 3.1 muestra los principales componentes para ejecutar un proceso de calibración automática. La calibración automática requiere un conjunto de datos históricos que reflejen el comportamiento del sistema real, una medida de error y un método de optimización para buscar el espacio de valores de los parámetros de forma sistemática minimizando la medida de error. Los ABMs se caracterizan por su naturaleza estocástica, por lo que se requieren resultados de varias ejecuciones independientes para poder analizarlos (simulaciones de Monte Carlo) [Eps07]. Esto dificulta la comparación de su comportamiento a lo largo del proceso de calibración ya que, a veces, considerar los valores de error medio no es suficiente para determinar la mejor opción en términos de robustez (por ejemplo, buen comportamiento puntual pero mal promediado frente a mal comportamiento puntual pero bien promediado). La comparación de las distribuciones de probabilidad (juzgar las distribuciones de error en lugar de simplemente el error medio) es un reto en el modelado e identificación de sistemas [SW10].

El modelador debe seleccionar cuidadosamente los parámetros del modelo que se ajustarán

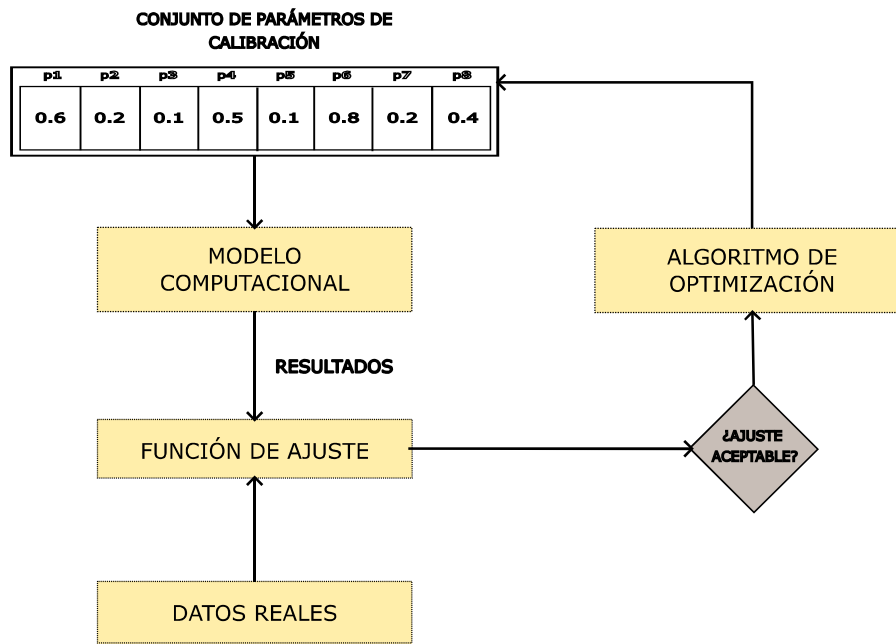


Figura 3.1: Esquema clásico para calibración de ABMs.

mediante la calibración automática puesto que la dificultad de validar las configuraciones calibradas (es decir, la dimensión del espacio de búsqueda) se incrementa con el número de parámetros calibrados. Por un lado, el modelador debe considerar aquellos parámetros más inciertos y difíciles de definir según la información disponible y, por otro lado, los parámetros sensibles para los que pequeños cambios en sus valores pueden afectar significativamente a la respuesta del modelo y a la salida del mismo. Los parámetros a calibrar suelen codificarse como valores enteros o reales y deben ser manejados adecuadamente por el método de optimización seleccionado. A su vez, la selección del método de optimización que lleva a cabo el proceso de calibración determina en gran medida la calidad de la configuración de los parámetros resultantes, ya que la precisión del modelo depende de la capacidad del método para explorar y explotar el espacio de búsqueda de los parámetros [MCC19; Moy+21a]. Tras la aplicación del proceso de calibración, los valores de los parámetros resultantes deben ser cuidadosamente revisados y validados, ya que un buen ajuste de los datos históricos no garantiza la validez del modelo [RR11; Fag+19].

## 3.2. Optimización de estrategias usando ABMs

Una vez construido, calibrado y validado un ABM, los modeladores pueden proceder a su aprovechamiento y explotación. Como comentamos en la sección anterior, un mercado rigurosamente validado es capaz de generar salidas sobre escenarios alternativos. Estos escenarios *what-if* se construyen a partir de modificaciones de parámetros sobre el mercado original calibrado y permiten medir cómo afectan dichos cambios a las dinámicas del mercado, tanto para el KPI calibrado (decremento o aumento de las ventas conseguidas por una determinada marca, por ejemplo) como para la eficiencia reportada por la presión realizada sobre un medio de comunicación específico. Además de responder preguntas sobre escenarios hipotéticos, un ABM sólido permite el acoplamiento de otros módulos y la aplicación de otras técnicas con las que investigar más en profundidad las dinámicas de los mercados simulados.

Las metaheurísticas aparecen de nuevo como un enfoque apropiado para abordar estos problemas dado que todos ellos requieren de la búsqueda de soluciones en grandes espacios de búsqueda en un tiempo limitado. Así pues, la hibridación de técnicas de simulación como los ABMs con metaheurísticas (conocida como *simheuristics* [Chi+20]), proporcionan a los profesionales una herramienta flexible y eficaz a la hora de abordar problemas de optimización en dominios inciertos. Cuando se diseñan debidamente, las *simheuristics* son capaces de resolver problemas de optimización en los que el componente de simulación se ocupa de la incertidumbre del modelo e interactúa con el componente metaheurístico que, por su parte, busca en el espacio de soluciones una solución próxima a la óptima. No obstante, la aplicación de *simheuristics* conlleva un grado de dificultad añadido ya que no sólo se buscan soluciones en un entorno complejo sino que dicha búsqueda requiere de un tiempo y de una carga computacional normalmente altos asociada a la complejidad y el volumen de los ABMs utilizados.

Entre los ejemplos de la aplicación de *simheuristics* usando ABMs se encuentran la combinación de diversas métricas de análisis de SNs (ver Sección 2.3.3) para la identificación y selección de consumidores para fomentar la adopción de productos dentro de un mercado competitivo [SRW10] o el análisis de políticas de recompensa de usuarios en aplicaciones *Freemium* [Kum14] y su

posterior uso para la búsqueda y selección de estos usuarios para maximizar las suscripciones *Premium* vía WOM en una aplicación real [CR17]. La presente tesis doctoral se centra en la selección de *influencers* [Fre+11] para lanzar campañas de VM [TBP08] utilizando GAs. Para ello, el GA busca la mejor combinación de pesos para el grado, intermediación y coeficiente de clustering a la hora de seleccionar los individuos que actuarán como *influencers* en las campañas de VM y el número de individuos semilla óptimo para maximizar las ganancias obtenidas durante dichas campañas.

Las *simheuristics* también se han aplicado con éxito en sectores ajenos al marketing. En [NT+17], por ejemplo, los autores proponen un algoritmo para optimizar el encaminamiento en una red de transporte de forma dinámica. El algoritmo recomienda a cada transportista seguir rutas dependientes del volumen de tráfico en cada instante y de las preferencias del usuario, de forma que se maximice el rendimiento global de la red de transporte respetando, en lo posible, las preferencias del usuario. Los autores aplican dicho algoritmo sobre un modelo ABM que simula el transporte de residuos sólidos urbanos en zonas urbanas buscando la optimización de su recogida y el transporte. En [He+19], por su parte, se propone un algoritmo heurístico para medir la influencia de la calidad de la comida y la localización de los restaurantes en un mercado de pedidos y entregas de comida en línea simulado usando ABMs y optimizar el beneficio de los clientes, de los restaurantes y de las plataformas de comida online de forma conjunta. Por último, en [NJJ19] se aplica el algoritmo de optimización basado en nubes de partículas [KE95] sobre un ABM que se utiliza para componer servicios de Cloud Computing usando parámetros de calidad de servicio. Este trabajo trata de resolver un problema complejo ya que la forma de elegir los servicios adecuados en una plataforma de Cloud Computing se realiza ajustando un alto número parámetros de calidad de servicio para realizar una asignación optimizada de los recursos.

### 3.3. Metaheurísticas para calibración y optimización de estrategias

Los problemas asociados a la calibración de ABMs (Sección 3.1) y a la optimización de estrategias usándolos como base de simulación (Sección 3.2) pueden abordarse utilizando herramientas como las metaheurísticas. Como comentamos en secciones anteriores, las metaheurísticas son un conjunto de técnicas de optimización no lineales aplicables en problemas complejos de búsqueda, optimización y aprendizaje. Las metaheurísticas han mostrado un excelente rendimiento sobre entornos de búsqueda de soluciones *NP-hard* [BFM97] para los que los algoritmos de búsqueda u optimización tradicionales se muestran incapaces de encontrar soluciones satisfactorias u óptimas.

El inicio de la utilización de las estrategias evolutivas en la solución de este tipo de problemas data del año 1960 cuando John Holland [Hol92] planteó la posibilidad de incorporar los mecanismos naturales de selección y supervivencia a la resolución de problemas de Inteligencia Artificial. La simulación de procesos de evolución natural de las especies da como resultado una técnica de optimización estocástica que posteriormente fue llamada algoritmos evolutivos, y que se enmarcó dentro de las técnicas no convencionales de optimización para problemas reales. A partir de la creación de los GAs aparecieron otras estrategias evolutivas alternativas como son: Programación Genética [Koz94], Programación Evolutiva [Fog99], y Estrategias de Evolución [BFM97].

Los algoritmos evolutivos constituyen una técnica paralela de búsqueda y optimización muy potente, inspirada en el principio Darwiniano de selección natural y reproducción genética. Se basa en el principio de selección de los individuos más aptos (elitismo), que tienen mayor recorrido y, por tanto, mayor probabilidad de reproducción. Los descendientes de estos individuos tienen, a su vez, una mayor posibilidad de transmitir sus códigos genéticos a las próximas generaciones. La aparición de computadores de altas prestaciones y bajo coste a mediados de los años 80 permitió la aplicación de los algoritmos evolutivos a la resolución de ciertos problemas de ingeniería que antes eran inabordables. Desde entonces el desarrollo de estas técnicas ha sido continuo, siendo aplicadas con éxito en campos tan diversos como el cálculo de estrategias de



mercado, reconocimiento de patrones, ingeniería aeroespacial, astronomía y astrofísica, química, programación y secuenciación de operaciones, programación de rutas entre otros problemas que involucran procesos de optimización [CWM12].

La sección actual describe las metaheurísticas y algoritmos de optimización avanzados utilizados en esta tesis doctoral para la calibración y optimización de ABMs. La sección se divide en cinco secciones, que abordan los problemas de optimización multiobjetivo (MOPs, por sus siglas en inglés) y los problemas de optimización multimodal (MMOPs, por sus siglas en inglés). Por un lado, las Secciones 3.3.1, 3.3.2 y 3.3.3 introducen los conceptos de la optimización multiobjetivo, las características principales de los algoritmos evolutivos multiobjetivo utilizados en esta tesis y los indicadores de calidad utilizados para medir y comparar sus rendimientos, respectivamente. Por otro, las Secciones 3.3.4 y 3.3.5 desarrollan el concepto de optimización multimodal y exponen las características de los algoritmos utilizados en la calibración de ABMs.

### 3.3.1. Optimización multiobjetivo

Los MOPs presentan un cierto número de funciones objetivo que deben ser optimizadas conjuntamente. Como en un problema de optimización monoobjetivo, también suele existir un número de restricciones que debe satisfacer cualquier solución factible. Dado un espacio  $X$  compuesto por vectores  $n$ -dimensionales de variables  $x = x_1, \dots, x_n$ , un MOP consiste en encontrar un vector  $x^*$  que minimice (o maximice) un conjunto de  $K$  funciones objetivo  $z(x) = f_1(x), \dots, f_K(x) \in Y$ :

$$\begin{aligned} \text{Maximizar} \mid \text{Minimizar} : z(x) &= (f_1(x), f_2(x), \dots, f_K(x))^T \\ \text{Sujeto a} : g_i(x) &\leq 0, 0 \leq i \leq N \\ h_j(x) &= 0, 0 \leq j \leq M. \end{aligned} \tag{3.1}$$

donde  $x = (x_1, x_2, \dots, x_n) \in X \subset R_n$  es un vector de decisión  $n$ -dimensional, y  $X$  es un espacio de decisión  $n$ -dimensional. Además,  $y = (y_1, y_2, \dots, y_m) \in Y \subset R_m$  es un vector objeto  $m$ -dimensional, e  $Y$  es el espacio objeto  $m$ -dimensional. La función objetivo  $F(x)$  define  $m$  funciones de mapeo del espacio de decisión al espacio objetivo.  $g_i(x) \leq 0$  ( $i = 1, 2, \dots, q$ ) son las

$q$  restricciones de desigualdad, y  $h_j(x) = 0$  ( $j = 1, 2, \dots, p$ ) son las  $p$  restricciones de igualdad. Sobre esta base, se dan las siguientes definiciones:

*Definition 3.1* (Solución factible). Para un  $x \in X$ , si  $x$  satisface las restricciones de la Ecuación 3.1, es decir,  $g_i(x) \leq 0$  ( $i = 1, 2, \dots, q$ ) y  $h_j(x) = 0$  ( $j = 1, 2, \dots, p$ ),  $x$  se llama solución factible.

*Definition 3.2* (Conjunto de soluciones factibles). El conjunto de todas las soluciones factibles en  $X$  se llama conjunto de soluciones factibles, denotado como  $X_f$ , y  $X_f \subseteq X$ .

*Definition 3.3* (Pareto dominancia). A partir de la Ecuación 3.1, dados  $x_A, x_B \in X_f$ ,  $x_A$  domina (lo que se denomina Pareto dominancia) a otra solución  $x_B$  o  $x_A > x_B$  si satisface las dos condiciones siguientes:

$$\forall i = 1, 2, \dots, m \exists f_i(x_A) \leq f_i(x_B) \wedge \exists j = 1, 2, \dots, m \exists f_j(x_A) < f_j(x_B) \quad (3.2)$$

*Definition 3.4* (Pareto-optimalidad). Una solución  $x^* \in X_f$  se llama solución pareto-óptima (no dominante) si  $\neg \exists x \in X_f : x > x^*$

*Definition 3.5* (Conjunto pareto-optimal). El conjunto de todas las soluciones pareto-óptimas se denomina conjunto pareto-óptimo.

$$P^* \triangleq \{x^* | \neg \exists x \in X_f : x > x^*\} \quad (3.3)$$

*Definition 3.6* (Frente pareto-óptimo). La superficie que se genera combinando todos los vectores de la función objetivo correspondientes a las soluciones óptimas de Pareto en el conjunto  $P^*$  se llama conjunto Pareto-óptimo ( $PF^*$ ):

$$PF^* \triangleq \{F(x^*) = (f_1(x^*), f_2(x^*), \dots, f_m(x^*))^T | x^* \in P^*\} \quad (3.4)$$

Pueden distinguirse dos tipos de problemas multiobjetivo principalmente:

1. **Problemas multiobjetivo lineales y no-lineales:** Cuando todas las funciones objetivo y las restricciones son lineales el problema multiobjetivo asociado se considera lineal. Por

el contrario, si alguna de las funciones objetivo o de las restricciones del problema no son lineales se considera al problema multiobjetivo como no lineal.

2. **Problemas multiobjetivo convexos y no convexos:** Un MOP es convexo si todas las funciones objetivo son convexas y la región factible del espacio de soluciones también es convexa. Otra forma de plantear la convexidad de un MOP es decir que todas las restricciones de no igualdad son no convexas y las restricciones de igualdad son lineales. De acuerdo con la definición anterior un problema multiobjetivo lineal es convexo. La convexidad es un factor importante a la hora de tratar un MOP.

### 3.3.1.1. Dominancia y Pareto-optimalidad

El concepto de dominancia proporciona una forma de comparar soluciones con múltiples objetivos. La mayor parte de los algoritmos de optimización multiobjetivo usan el concepto de dominancia para comparar soluciones en base a los valores de sus funciones objetivo. Así, en este tipo de algoritmos, la preferencia entre dos soluciones se especifica en función de la dominancia de una frente a la otra.

En la Ecuación 3.2 definimos el concepto de Pareto dominancia entre dos soluciones  $x$  e  $y$ . El operador de dominancia,  $>$ , trabaja a nivel de valores de función objetivo entre pares de soluciones. Cuando una solución  $x$  se dice dominada por otra  $y$  y, si y solo si se cumplen las siguientes condiciones:

1. La solución  $x$  no es peor que  $x$  en todos los objetivos;  $f_j(x) > f_j(y) \forall j = 1, 2, \dots, M$ .
2. La solución  $x$  es estrictamente mejor que  $y$  en al menos un objetivo.

Cuando no se cumple alguna de estas dos condiciones, la solución  $x$  no domina a la solución  $y$ . Si la solución  $x$  domina a la solución  $y$  podremos decir que:

1.  $y$  es dominada por  $x$ .
2.  $x$  es no dominada por  $y$ .
3.  $x$  es no inferior a  $y$ .

Para ilustrar mejor el concepto, supondremos un problema con dos funciones objetivo a minimizar:  $f_1$  y  $f_2$ . La optimización conjunta de  $f_1$  y  $f_2$  dificulta la tarea de encontrar una solución que sea la mejor con respecto a ambos objetivos. No obstante, utilizando la definición de dominancia podremos discernir qué solución es la mejor entre dos soluciones dadas en términos de  $f_1$  y  $f_2$ . Por ejemplo, si comparamos las soluciones  $S_1$  y  $S_2$  de la Figura 3.2, observamos que la solución  $S_1$  es mejor que la  $S_2$  en la función objetivo  $S_1$  y la solución 1 es también mejor que la  $S_2$  en el segundo objetivo. Esto supone que se cumplen las dos condiciones de dominancia, luego podremos afirmar que la solución  $S_1$  domina a la solución  $S_2$ .

De forma intuitiva, podemos afirmar que si una solución  $x$  domina a otra  $y$ , entonces  $x$  es mejor que  $y$  en el problema abordado.

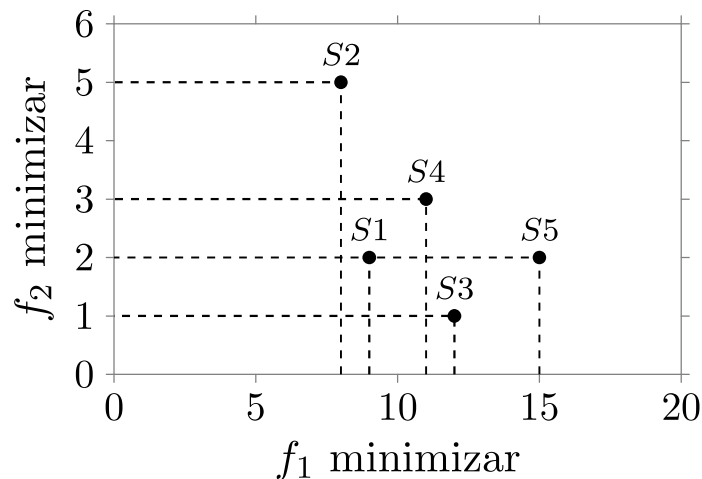


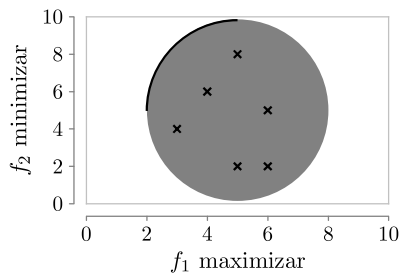
Figura 3.2: Ejemplo de relaciones de dominancia entre soluciones dadas dos funciones a minimizar  $f_1$  y  $f_2$  en un problema multiobjetivo

En la Figura 3.2 comparamos la solución  $S_3$  con la solución  $S_5$ , observamos que la solución  $S_5$  es mejor que la solución  $S_3$  en el objetivo 1 mientras que la solución  $S_5$  es peor que la solución 3 en el segundo objetivo. Vemos que la primera condición no se cumple para ambas soluciones, lo que simplemente nos dice que no podemos concluir que la solución  $S_5$  domine a la solución  $S_3$  ni tampoco que la  $S_3$  domine a la  $S_5$ . Cuando ocurre esto, las soluciones  $S_3$  y  $S_5$  son no dominadas entre sí. Teniendo en cuenta ambos objetivos, no podemos decir cuál de las dos soluciones es la mejor.

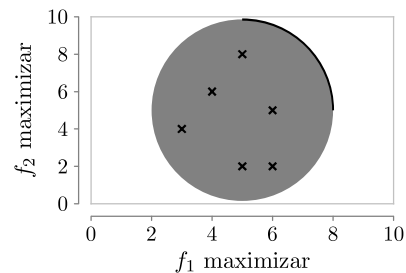
Para un conjunto de soluciones dado, podemos realizar todas las posibles comparaciones de

pares de soluciones y encontrar cuáles son dominadas y cuáles no dominadas. El procedimiento persigue encontrar un conjunto en el que todas las soluciones son no dominadas entre sí y dominan al resto de soluciones existentes, lo que significa que las soluciones de este conjunto son mejores comparadas con el resto de soluciones. Este conjunto se conoce como conjunto de soluciones no-dominadas o Frente de Pareto y constituye la solución final a un MOP.

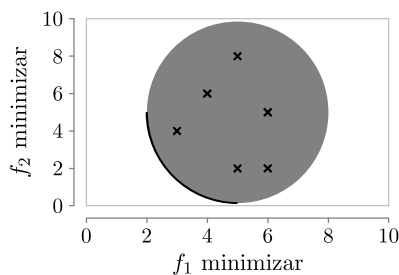
Dado un conjunto de soluciones  $P$ , el conjunto de soluciones no dominadas  $P'$  está formado por aquellas que no son dominadas por ningún miembro del conjunto  $P$ . Cuando el conjunto  $P$  es el espacio de búsqueda completo,  $P = S$ , el conjunto no dominado resultante se denomina conjunto Pareto-óptimo. En la Figura 3.3 podemos observar distintos ejemplos de problemas de optimización con dos objetivos en los que el conjunto de soluciones Pareto-óptimas aparecen resaltados con una línea negra.



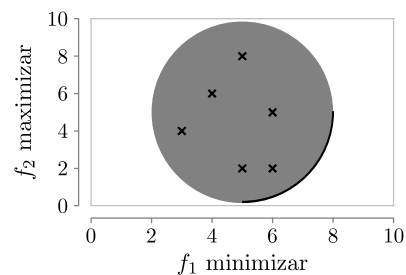
(a) Problema multiobjetivo donde se maximiza el objetivo  $f_1$  mientras se minimiza  $f_2$ .



(b) Problema multiobjetivo donde se maximizan los objetivos  $f_1$  y  $f_2$



(c) Problema multiobjetivo donde se maximizan los objetivos  $f_1$  y  $f_2$



(d) Problema multiobjetivo donde se minimiza el objetivo  $f_1$  mientras se maximiza  $f_2$

Figura 3.3: Tipos de frentes óptimos de Pareto para el mismo espacio de soluciones. La forma de cada frente es distinta dependiendo de si los objetivos del problema  $f_1$  y  $f_2$  son maximizados o minimizados.

### 3.3.2. Metaheurísticas multiobjetivo para optimización de ABMs

En esta sección se introducen las características generales de las metaheurísticas multiobjetivo utilizadas para la optimización de ABMs durante esta tesis doctoral. En la sección se abordan, por una parte, las adaptaciones multiobjetivo de algoritmos monoobjetivo para la optimización de ABMs y, por otra, las metaheurísticas intrínsecamente multiobjetivo aplicadas a la optimización de dichos problemas.

#### 3.3.2.1. NSGA-II

NSGA-II [Deb+02] es el algoritmo EMO más conocido. En este algoritmo, la población de descendientes  $Q_t$  se crea a partir de la población de padres  $P_t$  (también compuesta por  $N$  individuos) aplicando el mecanismo de selección considerado y los operadores genéticos. Después, las dos poblaciones se unen para formar una única población intermedia  $R_t$  con dos veces el tamaño de la población ( $2N$ ). A continuación, la población  $R_t$  se divide en varios frentes según sus grados de dominancia mediante un proceso de ordenación por no dominancia. Aunque este proceso de ordenación requiere un gran esfuerzo, se justifica por el hecho de permitir una verificación global de la dominancia dentro de la población de padres y descendientes.

Una vez finalizado el proceso de ordenación por no dominancia, se genera la nueva población a partir de las configuraciones de los frentes no dominados. Esta nueva población se construye primero con el mejor frente no dominado, continúa con las soluciones del segundo frente, el tercero, y así sucesivamente hasta que se ocupen las  $N$  posiciones disponibles. Cada miembro de cada grupo de dominancia se inserta en la nueva población hasta que no quede ningún lugar libre en todo el frente. Para el caso del primer frente que no pueda ser asignado completamente en la nueva población, se aplica un mecanismo de preservación de la diversidad. Se realiza un cálculo de la distancia de apilamiento para cada individuo de ese último frente considerado y los individuos más diversos son los que se incluyen en la nueva población.

La selección de los individuos que permanecen dentro de las sucesivas poblaciones de NSGA-II se realiza mediante un torneo por apilamiento, que considera que una solución tiene un rango

de no dominancia  $r_i$  y una distancia de apilamiento  $d_i$  asociados. La distancia de apilamiento  $d_i$  de una solución  $i$  es la medida del espacio de búsqueda alrededor de  $i$  que no está ocupado por otra solución en la población. Tenemos pues que la mejor configuración a través del operador de selección por torneo por apilamiento es aquella que tiene mejor rango de no-dominancia, es decir, mejores valores de  $r_i$  y  $d_i$ . Por ejemplo, si tuviésemos dos soluciones situadas en el mismo frente de Pareto, la ganadora del torneo sería aquella solución que tiene mejor (más alta) distancia de apilamiento. La distancia de apilamiento  $d_{I_j^m}$  para cada solución  $j$ , según un índice  $I$ , se calcula haciendo uso de la Ecuación 3.5:

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{max} - f_m^{min}} \quad (3.5)$$

donde  $f_m^{max}$  y  $f_m^{min}$  son el valor máximo y mínimo de la función objetivo  $m$ .  $f_m^{(I_{j+1}^m)}$  y  $f_m^{(I_{j-1}^m)}$  son las soluciones vecinas a la configuración  $j$  para cada una de las funciones objetivo  $m$ . Las distancias consideran todas las funciones objetivo y se asigna el valor de infinito a las soluciones extremas del frente de Pareto considerado. Por ser éstas las que cuentan con el mejor valor de una de las funciones objetivo del frente, la distancia resultante es la suma de las distancias en cada una de las direcciones de las funciones objetivo del problema.

NSGA-II usa los operadores genéticos básicos y promueve hacia el siguiente ciclo generacional aquellas configuraciones que ocupen los mejores frentes, y las más diversas, a través de las distancias de apilamiento.

### 3.3.2.2. MOEA/D

MOEA/D [ZL07] es un algoritmo EMO basado en descomponer explícitamente un MOP en  $M$  subproblemas de optimización escalar. El algoritmo resuelve estos subproblemas simultáneamente mediante la evolución de una población de soluciones. En cada generación, la población está compuesta por la mejor solución encontrada hasta el momento para cada subproblema. Las relaciones de vecindad entre estos subproblemas se definen en función de las distancias entre sus vectores de coeficientes de agregación. Cada subproblema se optimiza en MOEA/D utilizando

únicamente la información de sus subproblemas vecinos.

Existen varios enfoques para convertir el problema de aproximación a un frente de Pareto en una serie de problemas de optimización escalar. Utilizamos el enfoque de Tchebycheff ( $g^{te}$ ) para nuestros experimentos debido a los buenos resultados en términos de viabilidad y eficiencia obtenidos en [ZL07]. Matemáticamente, sea  $\lambda = (\lambda_1, \dots, \lambda_m)^W$  un vector de pesos y  $m$  el número de subproblemas, es decir,  $\lambda_i \geq 0$  para todo  $i = 1, \dots, m$  y  $\sum_{i=1}^m \lambda_i = 1$ . El enfoque de Tchebycheff considera un problema de optimización escalar de la forma:

$$\begin{aligned} \min g^{te}(x|\lambda, z^*) &= \min_{1 \leq i \leq m} \{\lambda_i |f_i(x) - z_i^*|\} \\ &\text{sujeto a } x \in \Omega, \end{aligned} \tag{3.6}$$

donde  $z^* = (z_1^*, \dots, z_m^*)^W$  es el punto de referencia. Este punto se inicializa como el valor más bajo de la función objetivo  $f_i$  encontrado en la población inicial. Para cada punto óptimo de Pareto  $x^*$  hay un vector de pesos  $\lambda$  tal que  $x^*$  es la solución óptima de la ecuación 3.6 si es una solución óptima de Pareto de la función objetivo. Por lo tanto, podemos obtener diferentes soluciones óptimas de Pareto alterando el vector de pesos.

### 3.3.3. Indicadores de calidad multiobjetivo

Consideramos dos indicadores de rendimiento multiobjetivo unarios y dos binarios [Zit+03; CLVV+07; Deb01] para evaluar el rendimiento de las metaheurísticas multiobjetivo utilizadas.

#### 3.3.3.1. Indicadores de calidad unarios

El primero de los indicadores unarios que vamos a utilizar en esta tesis doctoral es la cardinalidad de cada aproximación del conjunto de Pareto (es decir, el número de soluciones que lo componen). El segundo indicador unario es el ratio de hipervolumen ( $HVR$ ) [CLVV+07] que mide conjuntamente la distribución y la convergencia de una aproximación del frente de Pareto.



El  $HVR$  se define como sigue:

$$HVR = \frac{HV(P)}{HV(P^*)}, \quad (3.7)$$

donde  $HV(P)$  y  $HV(P^*)$  son el área (hipervolumen) de la aproximación del frente de Pareto obtenida y del frente de Pareto real, respectivamente. Cuando  $HVR$  es 1, la aproximación del frente de Pareto y el frente de Pareto real son iguales. Los valores de  $HVR$  inferiores a 1 indican que la aproximación del frente de Pareto generada no es tan bueno como el frente real.

Dado que trabajamos con problemas del mundo real, uno de los inconvenientes es la falta de información sobre los frentes de Pareto reales, lo que dificulta el cálculo del indicador. Para superar este obstáculo y poder calcular el indicador  $HVR$ , consideramos un frente de Pareto pseudo-óptimo. Un frente de Pareto pseudo-óptimo es una aproximación al verdadero frente de Pareto obtenida mediante la fusión de todas las aproximaciones del frente de Pareto generadas para cada instancia del problema por cada algoritmo en cada ejecución independiente.

### 3.3.3.2. Indicadores de calidad binarios

Los indicadores binarios miden el rendimiento de dos algoritmos EMO diferentes comparando las aproximaciones del frente de Pareto generadas por cada uno de ellos. Consideramos dos indicadores binarios principalmente: El indicador  $I_\epsilon$  y el indicador de cobertura del conjunto ( $C$ ). El indicador  $I_\epsilon$  [Zit+03] es un método de evaluación de la calidad asociado a la optimización multiobjetivo que evita las dificultades particulares de los métodos unarios y clásicos. Elegimos la variante multiplicativa del indicador  $I_\epsilon$ . Dadas dos aproximaciones del frente de Pareto,  $P$  y  $Q$ ,  $I_\epsilon(P, Q)$  se calcula como sigue:

$$I_\epsilon(P, Q) = \inf_{\epsilon \in \mathbb{R}} \forall z^2 \in Q, \exists z^1 \in P : z^1 \leq_\epsilon z^2, \quad (3.8)$$

donde  $z^1 \leq_\epsilon z^2$  si  $z_i^1 \leq_\epsilon z_i^2, \forall i \in 1, \dots, o$ , siendo  $o$  el número de objetivos. Suponiendo un problema de minimización,  $I_\epsilon(P, Q) < I_\epsilon(Q, P)$  indica que el conjunto  $P$  es mejor que el conjunto  $Q$  cuando el valor mínimo de  $\epsilon$  necesario para que el conjunto de aproximación  $P \in$

domine  $Q$  es menor que el valor de  $\epsilon$  necesario para que  $Q \in$  domine  $P$ .

El segundo indicador binario considerado es el indicador de cobertura del conjunto  $C$  [Zit+03], que se calcula como sigue:

$$C(P, Q) = \frac{|q \in Q; \exists p \in P : p \leq q|}{|Q|}, \quad (3.9)$$

donde  $p \leq q$  indica que la solución  $p$  de la aproximación del frente de Pareto  $P$  domina débilmente la solución  $q$  del frente de Pareto  $Q$ , en un problema de minimización. Por lo tanto,  $C(P, Q) = 1$  significa que todas las soluciones en  $Q$  están dominadas o son iguales a las soluciones en  $P$ .  $C(P, Q) = 0$  significa que ninguna solución en  $Q$  está cubierta por ninguna en  $P$ . Hay que considerar tanto  $C(P, Q)$  como  $C(Q, P)$  ya que  $C(P, Q)$  no es necesariamente igual a  $1 - C(Q, P)$ .

### 3.3.4. Optimización multimodal

Tal y como expusimos en la Sección 3.1, la validación de modelos computacionales es una tarea de vital importancia para que dichos modelos puedan generar salidas y recomendaciones fiables a los usuarios modeladores. Para la validación de ABMs es fundamental tanto la calibración de los parámetros desconocidos del modelo y su ajuste a valores y dinámicas reales, como tener una visión global de las distintas opciones o soluciones de calibración. Este último punto es indispensable para el modelador para analizar los distintos valores que puede tomar cada parámetro de calibración así como para reforzar su conocimiento del modelo, en forma de análisis de sensibilidad de los parámetros del mismo.

Los MMOPs son problemas de optimización caracterizados por presentar múltiples óptimos locales o globales (ver Figura 3.4), es decir, por tener varias soluciones localmente óptimas. En la resolución de estos problemas, por tanto, es necesario obtener el mayor número posible de esos óptimos locales o globales.

Los algoritmos evolutivos son una de las mejores herramientas disponibles para resolver MOPs.

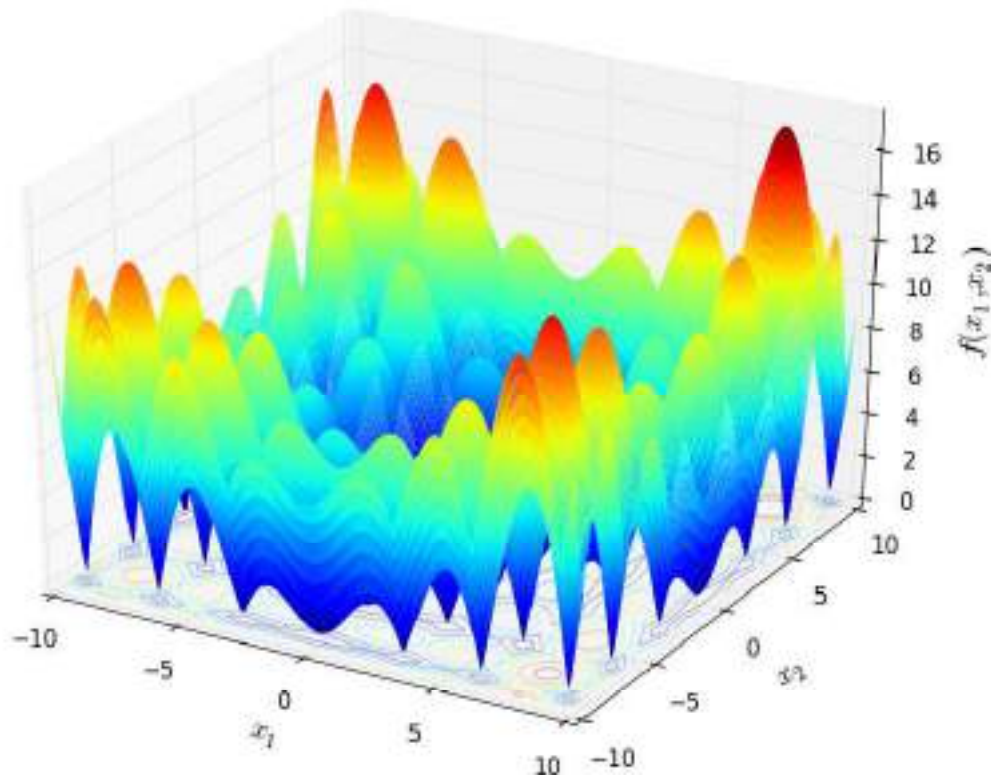


Figura 3.4: Ejemplo de un espacio de soluciones multimodal generado por la función Alpine 1, ampliamente utilizada para testar algoritmos de optimización. Las regiones con valores más bajos (mínimos) para Alpine 1 aparecen coloreadas en azul mientras que las regiones con valores más altos (máximos) aparecen en rojo.

Para ello, se ha propuesto una variante específica de algoritmos evolutivos denominada MMEAs. Estos algoritmos dividen la población en diferentes subpoblaciones (conocidas como nichos) siguiendo principios inspirados en la naturaleza para obtener múltiples soluciones en una sola ejecución [SK98]. El objetivo principal de estas técnicas es mantener la diversidad necesaria para lograr una búsqueda amplia en diferentes zonas prometedoras del espacio de soluciones de un problema, permitiendo que el método alcance varios óptimos de forma simultánea. Los MMEAs se han aplicado con éxito a diferentes problemas [Ibá+10; BDT12; BD13; Chi+17].

La existencia de un rango válido de ajustes de parámetros en la calibración de ABMs posiciona a los MMEAs como una técnica efectiva para este fin. Su aplicación a la calibración de ABMs podría proporcionar indicaciones adicionales sobre la validez del modelo a los modeladores (especialmente si los parámetros son diferentes entre sí). Además, permitirían analizar un conjunto de soluciones múltiples con el que descubrir propiedades ocultas relacionadas con el

diseño del modelo reforzando su validación.

En las siguientes secciones se describen tanto las técnicas clásicas de optimización evolutiva multimodal: *Fitness Sharing*, *Crowding* y *Clearing*, como MMEAs más avanzados como SHADE, L-SHADE, DE/NRAND/2, MOBiDE, NichePSO y PNA-NSGA-II. En capítulos posteriores (Capítulo 6) haremos uso de estas técnicas para diseñar métodos de calibración de ABMs.

### 3.3.5. Algoritmos evolutivos multimodales

#### 3.3.5.1. Algoritmo genético de *Fitness Sharing*

Introducido en [GR+87] y extendido en [Gol89], el algoritmo de compartición de *fitness* o *Fitness Sharing* fue uno de los primeros intentos de tratar directamente la localización y preservación de soluciones múltiples en GAs. *Fitness Sharing* restringe el crecimiento múltiple de un tipo de individuos haciendo que cada individuo de la población comparta su valor de *fitness* con los elementos cercanos de la población formando un nicho.

Cuando un determinado número de individuos ocupa un nicho, este método favorece que otros individuos busquen un nuevo nicho disponible en el espacio de búsqueda. En este sentido, el algoritmo debe encontrar un equilibrio entre el número de individuos que componen un nicho y la calidad del mismo, lo que lleva a un estado en el que el número de individuos que ocupan un nicho es proporcional al *fitness* o calidad del mismo. Este mecanismo se implementa degradando el ajuste de un individuo debido a la presencia de otros individuos en su vecindario, y la cantidad de reparto que aporta cada individuo a su vecino depende de la proximidad entre ambos.

El algoritmo de *Fitness Sharing* requiere un umbral de disimilitud  $\sigma_{share}$  que define la distancia máxima permitida para que un individuo comparta su *fitness* con otros miembros de la población (también llamado corte de distancia o radio del nicho) y un parámetro constante  $\alpha$  que regula la forma de la función de reparto. Normalmente,  $\alpha$  se fija en 1 y la función de reparto resultante se denomina función de reparto triangular [Gol89].

### 3.3.5.2. Algoritmo genético de *Crowding*

El GA de *Crowding* [DJ75] está motivado por la analogía con la competencia por los recursos limitados entre miembros similares de una población natural. Estos individuos tienden a ocupar los mismos nichos ambientales y deben competir por los mismos recursos limitados. Para determinar la similitud entre los individuos, el GA de *Crowding* utiliza una métrica de distancia. Las funciones de distancia más comunes son la distancia de *Hamming* entre cadenas binarias y la distancia Euclidiana. Cuando un nicho ha alcanzado su capacidad de carga, los miembros más débiles de ese nicho serán desplazados de la población por los más fuertes. Los miembros más antiguos del nicho acabarán siendo sustituidos por los más aptos de los miembros más jóvenes. Este método sigue la estructura de un GA estándar con la excepción de que sólo una fracción de la población se reproduce y muere en cada generación, mientras que un porcentaje de la población, especificado por la brecha generacional ( $G$ ), se elige mediante una selección proporcional a su valor de fitness para someterse a cruces y mutaciones. El resultado final es que en una población de tamaño fijo en equilibrio, los nuevos miembros de una especie concreta sustituyen a los miembros más antiguos de esa especie para mantener la diversidad preexistente de una población. Se toma una muestra aleatoria de  $CF$  individuos de la población, donde  $CF$  se denomina factor de aglomeración. De los individuos  $CF$ , se sustituye el más parecido al elemento que se inserta.

### 3.3.5.3. Algoritmo genético de *Clearing*

Desarrollado y presentado por Petrowski [Pet96; Pét97], el procedimiento de *Clearing* es una variante multimodal del GA clásico que mantiene al mejor individuo de la población de cada nicho y elimina a todos los demás individuos en lugar de repartir uniformemente los recursos disponibles entre los individuos de una subpoblación. En consecuencia, el método sigue una estrategia elitista.

*Clearing* se aplica después de evaluar el fitness de los individuos y antes de aplicar el operador de selección. La población se ordena de mejor a peor según los valores de fitness. A continuación,

todas las soluciones que tienen una medida de distancia crítica ( $\sigma_{clear}$ ) de las mejores soluciones  $k$  de la población se eliminan, lo que significa que sus valores de fitness se ponen a cero. El fitness de las mejores soluciones  $k$  se mantiene sin modificar. Este procedimiento continúa hasta que se consideran todas las soluciones. La deriva genética debida al ruido de selección se reduce significativamente en el caso del GA de *Clearing* y también el tamaño de la población requerido (es decir, es menor que *Fitness Sharing* cuando se utiliza con un operador de selección adecuado).

#### 3.3.5.4. SHADE y L-SHADE

El algoritmo de evolución diferencial adaptativa basada en históricos de éxito [TF13] (SHADE, por sus siglas en inglés) y su extensión L-SHADE [TF14] son variantes basadas en históricos del algoritmo de evolución diferencial [SP97] (DE, por sus siglas en inglés), un método de búsqueda estocástica que ha sido ampliamente utilizado en problemas prácticos de optimización debido a su buen rendimiento y simplicidad de implementación [DS11; DMS16].

DE genera nuevas soluciones combinando los individuos existentes con un *vector donante* que se crea siguiendo la siguiente ecuación:  $x_i(G + 1) = x_{r1}(G) + F(x_{r2}(G) - x_{r3}(G))$ , donde  $x_i$  es el vector donante generado y  $r1$ ,  $r2$  y  $r3$  son diferentes soluciones en la generación  $G$ . Para cada generación, se genera un vector donante  $x_i$  para cada individuo  $i$  y sus valores se combinan utilizando un parámetro  $CR$ . A continuación, para cada gen del individuo recién creado, el algoritmo toma el valor del vector donante con probabilidad  $CR$  o el valor original del individuo  $i$  en caso contrario. Si el valor de *fitness* del individuo resultante es mejor o igual que el del individuo  $i$ , éste sustituirá a  $i$  en la población.

SHADE, L-SHADE y otras variantes adaptativas de DE modifican los valores de los tres principales parámetros de control de una DE estándar (la tasa de cruce ( $CR$ ), la tasa de mutación ( $F$ ) y el tamaño de la población) durante el proceso de optimización. En el caso de SHADE, los valores *sucedidos* de  $CR$  y  $F$  se almacenan en una memoria histórica. Se considera que una combinación de parámetros es exitosa si la solución generada con ella mejora al individuo anterior [TF14]. L-SHADE amplía SHADE incorporando la reducción lineal del tamaño de la población (LPSR, por sus siglas en inglés), un sencillo método determinista de

redimensionamiento de la población que reduce continuamente el tamaño de la población de acuerdo con una función lineal durante una ejecución de SHADE. El LPSR reduce la población para que coincida con una función lineal donde el tamaño de la población en la primera generación es  $N_{init}$  y al final de la ejecución es  $N_{min}$ .

### 3.3.5.5. DE/NRAND/2

DE/nrand/2 es una nueva variante multimodal de DE basada en la estrategia de mutación DE/rand/2 [PSL06] propuesta por Epitropakis et al. en [EPV11]. La estrategia de mutación DE/nrand/2 es capaz de manejar eficientemente funciones multimodales incorporando información relativa a la vecindad de cada solución potencial para localizar y mantener con precisión muchos optimizadores globales sin necesidad de parámetros adicionales. Los esquemas algorítmicos sobre DE pueden clasificarse utilizando la notación DE/base/num/cross, donde el método de selección del padre que constituye el individuo base se indica con *base*. Por ejemplo, DE/rand/num/cross selecciona el progenitor del individuo base al azar, mientras que en DE/best/num/cross el progenitor del individuo base es el mejor individuo de la población. El número de diferencias entre individuos que se utilizan para perturbar al individuo base se indica mediante *num*. Por último, *cross* representa el tipo de cruce utilizado por la estrategia de mutación (es decir, *exp* para exponencial y *bin* para binomial).

DE/nrand/2 aprovecha la dinámica y la tendencia a la agrupación de las estrategias de mutación clásicas DE/rand/1 y DE/rand/2, e incorpora en sus esquemas de mutación información local de la población actual. Para localizar y mantener eficientemente los óptimos globales, el método adapta cada individuo aplicando como vector base su individuo vecino más cercano, en un intento de mantener el individuo en la proximidad de un óptimo y de explorar simultáneamente el espacio de búsqueda incorporando vectores aleatorios en las diferencias del esquema de mutación. El método utiliza un factor de mutación  $F$  para controlar la amplificación de la diferencia entre dos individuos y para evitar el riesgo de estancamiento del proceso de búsqueda.

### 3.3.5.6. MOBiDE

MOBiDE es un MMEA basado en DE e inspirado en el GA multipoblación biobjetivo (BMPGA, por sus siglas en inglés) [YKG10; YKG09] y NSGA-II. El método propone una formulación bi-objetivo de los MMOPs (proceso conocido como multiobjetivización [KWC01]) y utiliza DE junto con la ordenación no dominante y la ordenación basada en la medida de hipervolumen para detectar un conjunto de soluciones correspondientes a los óptimos globales y locales. A diferencia de BMPGA y NSGA-II, MOBiDE no requiere el gradiente real o estimado de la función multimodal para formar su segundo objetivo. En la formulación de MOBiDE, mientras que el primer objetivo sigue siendo la función multimodal bajo prueba, el segundo objetivo se elige como la distancia Euclidiana promediada de una solución respecto a todos los demás miembros de la población y se maximiza para evitar que toda la población converja a un único óptimo. Es necesario un balance adecuado entre los dos objetivos para encontrar el mayor número de óptimos globales. La situación más estable se consigue cuando diferentes miembros de la población ocupan todas las soluciones globales, situación a la que se espera que el algoritmo se acerque gradualmente. Este método incorpora un archivo externo para mantener un registro de las soluciones que tienen los mejores valores de aptitud actuales. El archivo desempeña un papel importante, ya que evita la generación de nuevas soluciones cerca de las soluciones ya almacenadas en él y reduce el número total de evaluaciones necesarias para detectar todos los picos globales con éxito.

### 3.3.5.7. NichePSO

El algoritmo de optimización de enjambres de partículas con anidamiento (NichePSO) [BEB02] es una extensión basada en anidamiento de la optimización de enjambre de partículas (PSO, por sus siglas en inglés) [KE95] para la optimización multimodal. Se ha demostrado que los algoritmos PSO resuelven eficazmente problemas de optimización unimodal, pero no están bien equipados para localizar soluciones óptimas en problemas multimodales. De este modo, NichePSO extiende al PSO unimodal para localizar de forma eficiente múltiples soluciones óptimas en MMOPs utilizando un enfoque de anidamiento (basado en subenjambres [LRK01]) para mantener la



diversidad en la población, y el algoritmo PSO de Convergencia Garantizada [BE02; VDB+07] (CGPSOA, por sus siglas en inglés) para optimizar los subenjambres de NichePSO. Así, el algoritmo NichePSO localiza múltiples soluciones óptimas para los MMOPs mediante el uso de múltiples subenjambres que crecen a partir de un enjambre de partículas inicial mediante el control del *fitness* de las partículas individuales y un CGPSOA. El método utiliza un parámetro de radio máximo permitido  $R_{max}$  para reducir las soluciones en subenjambres, y dos parámetros ( $\delta$  y  $\mu$ ) para crear y fusionar subenjambres. El parámetro  $\delta$  identifica y crea subenjambres uniendo una partícula y su vecino topológico más cercano cuando el *fitness* de la partícula muestra muy pocos cambios a lo largo de un pequeño número de iteraciones del algoritmo de aprendizaje, mientras que  $\mu$  fusiona subenjambres compuestos por partículas que son suficientemente similares entre sí.

#### 3.3.5.8. PNA-NSGA-II

El algoritmo NSGA-II asistido por anidamiento sin parámetros [BD13] (PNA-NSGA-II, por sus siglas en inglés) es una extensión del algoritmo multiobjetivo NSGA-II que combina el concepto de preservación de la dominancia y la diversidad de la optimización evolutiva multiobjetivo con el anidamiento del espacio variable para la optimización multimodal. Al igual que en MOBIDE, el problema original se convierte en un problema bi-objetivo en PNA-NSGA-II tomando el objetivo original como primer objetivo y diseñando un segundo objetivo adecuado. El problema bi-objetivo se resuelve utilizando NSGA-II con un criterio de dominancia modificado. También se añade una restricción adaptativa al problema bi-objetivo para evitar los óptimos locales. Además, no se introducen parámetros adicionales en el proceso, lo que hace que este enfoque sea fácil de utilizar.

El algoritmo NSGA-II viene con un operador de preservación de la diversidad que utiliza distancias de anidamiento calculadas en el espacio objetivo, lo que resulta ineficaz cuando los objetivos no están en conflicto. PNA-NSGA-II introduce un esquema de anidamiento en el espacio de decisión que utiliza los parámetros disponibles para ajustarse, mientras que la mayoría de los métodos de anidamiento introducen parámetros especiales cuyos valores pueden afectar

drásticamente al rendimiento del algoritmo. PNA-NSGA-II utiliza habitualmente la métrica de distancia *Manhattan* como medida de proximidad, ya que es preferible a las demás medidas en conjuntos de datos de alta dimensión.

## Parte II

# Propuesta



---

# Modelo CONSUMAT extendido

Sólo sé que no sé nada.

---

Sócrates

## 4.1. Introducción

Este capítulo se centra en la extensión propuesta para uno de los ABMs más conocidos en el área de modelado basado en agentes para implementar mercados de consumidores: *CONSUMAT* [JJ03]. Utilizamos el modelo *CONSUMAT* con el objetivo de simular los procesos de compra de los consumidores en un entorno de mercado competitivo. Las simulaciones ayudarán a entender cómo el comportamiento del consumidor se ve afectado por diferentes topologías de SNs y el éxito en la adquisición de los productos ofertados. Nuestra extensión del modelo *CONSUMAT* se centra en la incorporación de *awareness* al comportamiento del consumidor para hacerlo más cercano a las dinámicas de compra reales. De este modo, los individuos no dispondrán de un conocimiento completo de su entorno de compra sino que tendrán que adquirirlo gradualmente mediante un proceso de difusión WOM a través de la SN que los conecta. Consideraremos distintas topologías de SNs para representar las conexiones entre los agentes en el modelo *CONSUMAT* extendido y estudiar su influencia en la dinámica del sistema complejo. En concreto, utilizamos dos de los modelos de SN más conocidos: Redes de mundo pequeño [WS98] y redes libres de escala [BA99], que desarrollamos en la Sección 2.3.2 de Capítulo 2.

El uso del WOM como mecanismo de difusión proporciona un mayor realismo al modelo y permite obtener comportamientos emergentes más diversos que en el modelo *CONSUMAT*

original [Duz+07; RR11]. No obstante, para poder modelar estos procesos de difusión necesitamos un sistema capaz de simular las interacciones a nivel de red entre los clientes del mercado como los ABMs [RR11; MN14; Eps07]. Como vimos en la Sección 2.2, estas simulaciones permiten agregar interacciones a nivel individual a través de su SN, representativa de la SN real de los consumidores [NBW06].

El capítulo está estructurado como sigue. Comenzamos describiendo las características principales del modelo *CONSUMAT* y desarrollando las extensiones propuestas sobre el mismo en la Sección 4.2. A continuación, comparamos el comportamiento de los agentes en los modelos *CONSUMAT* original y extendido y analizamos las dinámicas de adquisición de productos sobre las redes libres de escala y de mundo pequeño en la Sección 4.3.

## 4.2. El modelo *CONSUMAT* extendido

En esta sección presentamos las principales características del modelo *CONSUMAT*, que utilizamos como base de nuestro modelo de comportamiento de consumidor, y las extensiones propuesta para el mismo. Los cambios propuestos para ampliar la funcionalidad del *CONSUMAT* original constan, por una parte, de la incorporación de un proceso de reconocimiento de marca (*awareness*) que limita el conocimiento que los agentes consumidores tienen de los productos ofertados en el mercado y, por otra parte, de un mecanismo de interacción entre agentes basado en la SN. De esta forma, los agentes están interconectados mediante una SN a través de la cual son capaces de difundir y recibir información mediante WOM y, por consiguiente, aumentar el *awareness* que tienen de los productos del mercado. Igualmente, incorporamos un mecanismo de olvido de productos. Estos mecanismos acercan a *CONSUMAT* a una forma de consumo más realista puesto que los agentes necesitan adquirir información sobre los productos del mercado antes de consumirlos.

En primer lugar, las Secciones 4.2.1 y 4.2.1.3 introducen las principales variables del mercado y las distintas heurísticas para la toma de decisiones de los consumidores en *CONSUMAT*, respectivamente. Seguidamente, la Sección 4.2.2 desarrolla la influencia de la SN en el consumo

de productos en el modelo ABM. Por último, la Sección 4.2.3 detalla los mecanismos de *awareness* y difusión de conocimiento mediante WOM añadidos a *CONSUMAT* que, como comentamos previamente, modelan el conocimiento de productos y la difusión de opiniones en los consumidores del mercado.

### 4.2.1. Consumidores en el modelo *CONSUMAT*

*CONSUMAT* es un ABM que modela mercados virtuales compuestos por una población de  $I$  agentes (consumidores) y  $J$  productos en competencia entre sí. Una simulación del modelo tiene  $T$  pasos y sigue una actualización síncrona [CR17]. Cada agente  $i \in \{1, 2, \dots, I\}$  tiene una probabilidad  $p_b \in [0, 1]$  de comprar un producto del mercado  $j \in \{1, 2, \dots, J\}$  en cada paso de simulación  $t \in \{1, 2, \dots, T\}$ . Así, la decisión de compra de un agente se produce usando la probabilidad de compra  $p_b$ . Cuando el valor aleatorio uniforme generado es inferior a  $p_b$  en el paso  $t$ , el agente  $i$  elige entre el conjunto de productos  $J$ . Una matriz  $X_i = x_i^1, x_i^2, \dots, x_i^T$  con  $x_i^t \in \{0, 1, \dots, J\}$  almacena los productos seleccionados por el agente  $i$  o establece el valor 0 cuando el agente no realiza compra alguna. El modelo *CONSUMAT* original asume que cada producto  $j$  tiene asociada una dimensión  $d_j \in [0, 1]$  que, a efectos prácticos es un compendio de las características (por ejemplo, el precio, la calidad o el diseño) de cada producto. Cada agente  $i$  tiene una preferencia personal  $\pi_i \in [0, 1]$  sobre sus necesidades en el mercado. La ecuación 4.1 calcula la utilidad  $\mu_{ij} \in [0, 1]$  del producto  $j$  para el agente consumidor  $i$  basándose en su preferencia  $\pi_i$  y en las características del producto  $d_j$ :

$$\mu_{ij} = 1 - |d_j - \pi_i|. \quad (4.1)$$

Las Secciones 4.2.1.1 y 4.2.1.2 describen las principales reglas de comportamiento de los agentes del modelo *CONSUMAT*:

#### 4.2.1.1. Utilidad de cada producto

La utilidad  $u_{ij}^t$  refleja la utilidad total esperada (tanto individual como social) de un agente  $i$  al consumir un producto  $j$  en el paso de simulación  $t$ :

$$u_{ij}^t = \beta\mu_{ij} + (1 - \beta)\gamma_{ij}^t, \quad (4.2)$$

donde  $\gamma_{ij}^t \in [0, 1]$  es la fracción de contactos directos del agente  $i$  en la SN que consumió el producto  $j$  en su última elección. La utilidad sesgada  $u_{ij}^t$  también está modulada por  $\beta \in [0, 1]$ , un parámetro de influencia de los pares sociales para todo el mercado artificial. Los valores altos de  $\beta$  significan que los consumidores están muy influenciados por sus contactos en la SN. Los valores bajos de  $\beta$  significan que las preferencias personales a la hora de adoptar un producto  $j$  son más importantes (por ejemplo, un mercado con consumidores más innovadores y poco convencionales).

La utilidad para el producto, basada únicamente en preferencias personales, es igual a  $1 - |\pi_i - d_j|$ . Como vemos en la ecuación 4.2, ambas componentes (la social y la personal) están ponderadas usando un valor  $\beta_i$ . Un valor alto de variable  $\beta_i$  significa una mayor apreciación de la preferencia personal a la hora de tomar decisiones (clientes que confían en sus elecciones) mientras que un valor bajo de  $\beta_i$  dará un mayor peso a las necesidades sociales del agente (consumidores dubitativos a la hora de tomar decisiones, que se dejan llevar por las decisiones de sus contactos). Así, se tienen en cuenta dos medidas para calcular la utilidad de la decisión del agente. La primera está relacionada con las preferencias de producto del agente y la segunda con la influencia social del mismo.

#### 4.2.1.2. Incertidumbre sobre las decisiones

La incertidumbre  $\phi_i^t$  de un agente  $i$  a la hora de consumir un producto  $j$  refleja su convicción de tomar la decisión correcta al elegir dicho producto. Cuántos más contactos tiene un consumidor comprando productos diferentes al suyo, menos convencido estará sobre la decisión de elegir el



producto más afín a sus preferencias personales:

$$\phi_i^t = \beta(1 - \gamma_{\neq x_i}^t), \quad (4.3)$$

donde  $\gamma_{\neq x_i}^t$  es la fracción de contactos del agente  $i$  que han comprado un producto diferente al comprado por el agente  $i$  en el paso  $t$  anterior. El parámetro social  $\beta$  modula el grado de incertidumbre de los agentes al tomar sus decisiones. Si el mercado tiene un alto componente social ( $\beta$ ), los agentes tendrán una mayor incertidumbre a la hora de comprar cuando sus respectivos contactos compraron un producto diferente al suyo.

#### 4.2.1.3. Heurísticas de decisión

En *CONSUMAT* los agentes toman decisiones en función de su nivel de satisfacción e incertidumbre. Los umbrales que desencadenan estos comportamientos son  $U$  y  $\Phi$ .  $U$  define el nivel mínimo de satisfacción de la utilidad de los agentes mientras que  $\Phi$  establece el nivel de tolerancia a la incertidumbre de los agentes. Estos dos umbrales especifican las condiciones del mercado artificial y definen el tipo de heurística junto con la utilidad sesgada  $u_{ij}^t$  y el valor de incertidumbre  $\phi_i^t$ .

Cuando un agente toma una decisión de compra, se adquiere un producto, el modelo actualiza los niveles de incertidumbre experimentados. De este modo, podemos distinguir entre los umbrales esperados y los experimentados teniendo en cuenta también las decisiones de su entorno de consumidores (esto es, su SN). Los primeros ( $u_{ij}^t$  y  $\phi_i^t$ ) modulan el estado de un agente al adquirir un producto que se está comprando en su entorno (estas expectativas varían según las decisiones del vecindario). Los segundos ( $U$  y  $\Phi$ ) permanecen estáticos para cada agente durante toda la simulación.

Como se hace en [JJ03], consideramos cuatro heurísticas de decisión diferentes para seleccionar un producto a adquirir del conjunto de productos que los agentes conocen (es decir, el subconjunto  $J_i^t \subseteq J$ ). Los agentes eligen dinámicamente la heurística a utilizar comparando sus variables internas con respecto a  $U$  y  $\Phi$ . Los siguientes cuatro tipos de procesos heurísticos o cognitivos

representan las acciones a tomar por los agentes y hacen uso de los umbrales anteriormente mencionados:

1. *Repetición*: El agente utiliza esta heurística cuando está satisfecho y seguro de su selección (es decir,  $u_{ij}^t \geq U$  y  $\phi_i^t \leq \Phi$ ). Se trata de un fenómeno de repetición de compra relacionado con la fidelidad y la satisfacción de un consumidor a una marca [EUG04]. Así, el agente  $i$  repite la compra del último producto adquirido:

$$x_i^t = x_i^{t-1}. \quad (4.4)$$

2. *Deliberación*: El agente consumidor no está satisfecho con su última compra pero está seguro (es decir,  $u_{ij}^t < U$  y  $\phi_i^t \leq \Phi$ ). Aquí, el agente  $i$  evalúa su utilidad  $u_{ij}^t$  para todos los productos del mercado que conoce y toma una decisión probabilística mediante una función logit<sup>1</sup>. Mediante esta función, el producto  $j$  tiene una probabilidad de ser elegido por el agente  $i$ :

$$prob_i^t(j) = \frac{e^{u_{ij}^t}}{\sum_{k=0}^{M_i^t} e^{u_{ik}^t}}, \forall j \in M_i^t. \quad (4.5)$$

3. *Imitación*: El agente está satisfecho pero no está seguro de su elección (es decir,  $u_{ij}^t \geq U$  y  $\phi_i^t > \Phi$ ). En consecuencia, el agente consumidor  $i$  evalúa los productos que conoce y la fracción de contactos directos que los han comprado. Se vuelve a utilizar una función logit para calcular las probabilidades de cada producto  $j$ . El producto con la mayor cuota entre los vecinos tiene la mayor probabilidad de ser elegido:

$$prob_i^t(j) = \frac{e^{2\gamma_{ij}^t}}{\sum_{k=0}^{M_i^t} e^{2\gamma_{ik}^t}}, \forall j \in M_i^t. \quad (4.6)$$

4. *Comparación social*: Un agente no está satisfecho ni seguro de su elección (es decir,  $u_{ij}^t < U$  y  $\phi_i^t > \Phi$ ), así que sólo evalúa los productos que compraron sus contactos directos en su última compra. Utilizamos una función logit similar a la de la ecuación 4.5 pero con un conjunto más reducido de productos  $J_i^t$  (es decir, un preproceso de cribado para filtrar

---

<sup>1</sup>La función logit es la inversa del “sigmoide”, o función “logística”. El modelo logit fue introducido por Joseph Berkson en 1944 y es una parte importante de la regresión logística.

alternativas):

$$prob_i^t(j) = \frac{e^{u_{ij}^t}}{\sum_{k=0}^{M_i^t} e^{u_{ik}^t}}, \forall j \in M_i^t. \quad (4.7)$$

### 4.2.2. Influencia de la SN en los agentes

En el modelo *CONSUMAT* las conexiones entre los clientes se representan utilizando SNs [WF94; Wat+07]. Por un lado, se representa matemáticamente la SN real mediante un grafo (dirigido o no dirigido) en el que los nodos representan a los individuos o agentes que forma la SN y las aristas a las conexiones entre ellos [BA99] y, por otro, hace uso de un mecanismo basado en patrones de conexión y creación de nodos para generar una topología de SN que imite la estructura de la SN real. Un aspecto importante en el estudio de las SNs es cómo los individuos interactúan entre sí para formar SNs amplias y conectadas.

Para nuestro modelo utilizaremos dos de los tipos más relevantes de SNs: Las redes libres de escala [BA99] y las redes de mundo pequeño [WS98]. Los mecanismos utilizados para la generación de estos dos tipos de SN fueron descritos en las Secciones 2.3.2.2 y 2.3.2.3, respectivamente.

### 4.2.3. *Awareness* y difusión de opiniones en la SN

Para obtener un modelo de consumo más realista incorporaremos un filtro adicional de conocimiento en los agentes. Con esta mejora, los agentes pasarán de tener un conocimiento completo del conjunto de productos disponibles (modelo *CONSUMAT* original [JJ03]) a tener ciertas limitaciones de conocimiento sobre los mismos. Esto se basa en la racionalidad limitada del comportamiento humano [PBJ91; GS02], ampliamente utilizada en marketing y ciencias sociales. Por lo tanto, nuestro modelo ampliado incorporará dos nuevas variables: La probabilidad de conocimiento, que permite a los agentes compartir parte de su conocimiento en su red de contactos, y la pérdida del conocimiento, que mide la probabilidad de que los agentes olviden parte de la información que tienen sobre los productos del mercado. El *awareness* de marca es la primera dimensión y el requisito previo del sistema de conocimiento de la marca en la mente del

consumidor, que ayuda a la identificación de la misma en diferentes condiciones [KS19] y filtra el conjunto final de opciones cuando el consumidor se enfrenta a una decisión sobre la marca que va a comprar. El *awareness* de marca puede definirse en varios niveles, como reconocimiento, notoriedad asistida, *top of mind* o conocimiento sugerido de la marca [Aak97].

En el modelo original, las decisiones de los agentes no estaban limitadas por su desconocimiento de todos los productos. Las características adicionales del nuevo modelo restringen sus decisiones entre todos los productos ofrecidos. Las decisiones de compra de los clientes también dependen del conocimiento, modificado a través de WOM con sus vecinos. La topología de la SN afectará a la mayor o menor difusión de su conocimiento del sistema complejo. La Figura 4.1 muestra los principales mecanismos del modelo *CONSUMAT* extendido.



Figura 4.1: Mecanismos involucrados en las dinámicas de compra de los consumidores en el modelo *CONSUMAT* extendido.

Utilizamos un mecanismo basado en probabilidades particulares de los agentes para modelar la ganancia y pérdida de *awareness*:

- Difusión de conocimiento:** Cada agente tiene una probabilidad de difusión  $p_\theta \in [0, 1]$  para hablar sobre cada producto del mercado que conoce. Para determinar si un consumidor habla de un producto, se lanza un número aleatorio en  $[0, 1]$ . Si este número no supera el valor  $p_\theta$ , el agente consumidor  $i$  difundirá su conocimiento del producto  $j$  a sus contactos

directos y éstos también conocerán este producto. Si un agente conoce un producto puede hablar de él con sus contactos directos en la SN, independientemente de su decisión de compra en el mismo paso temporal  $t$ .

- **Olvido de conocimiento:** Tras el proceso de WOM, todos los agentes podrían olvidar su conocimiento sobre algunos de los productos del mercado. Para todos los productos existentes, se establece un valor de olvido global  $\epsilon$  para todos los agentes y productos. Al igual que para el proceso de difusión, se genera un número aleatorio uniforme en  $[0, 1]$ . Si este valor no supera el umbral de olvido  $\epsilon$ , el agente consumidor  $i$  olvidará la información sobre el producto  $j$  y dejará de conocerlo. De lo contrario, el agente seguirá teniendo conocimiento sobre el producto y podrá adquirirlo en el siguiente paso  $t$ . Los valores altos de  $\epsilon$  harán que los agentes sean propensos a olvidar fácilmente la información que adquieren en la fase de difusión.

En el modelo *CONSUMAT* original los agentes reaccionaban a los estímulos en el momento de la compra y eran plenamente conscientes del estado de todos los productos en ese momento. Con la adición de este mecanismo de conocimiento, los agentes sólo pueden conocer una parte del conjunto de productos que se ofertan en el mercado y necesitan de la interacción con sus contactos para obtener información sobre el resto productos y para poder comprarlos. En la Figura 4.2, por ejemplo, se pueden observar las diferencias en el uso de heurísticas de decisión y en la adquisición de productos (tanto temporal como total) en los modelos *CONSUMAT* original y extendido sobre una topología de red aleatoria.

### 4.3. Comparación entre los modelos *CONSUMAT* base y extendido

Consideramos dos modelos de SNs (libre de escala y mundo pequeño) para la comparación de las dinámicas de los modelos *CONSUMAT* base y extendido. Los parámetros utilizados para la simulación de cada modelo son introducidos en la Sección 4.3.1. Por su parte, en las Secciones 4.3.2, 4.3.3 y 4.3.4 analizaremos las diferencias en la adquisición de productos y el

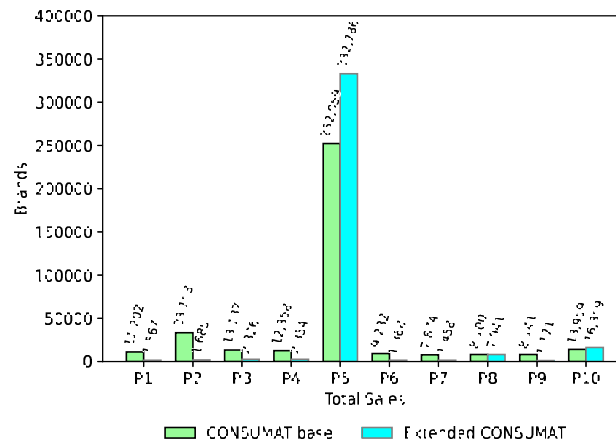
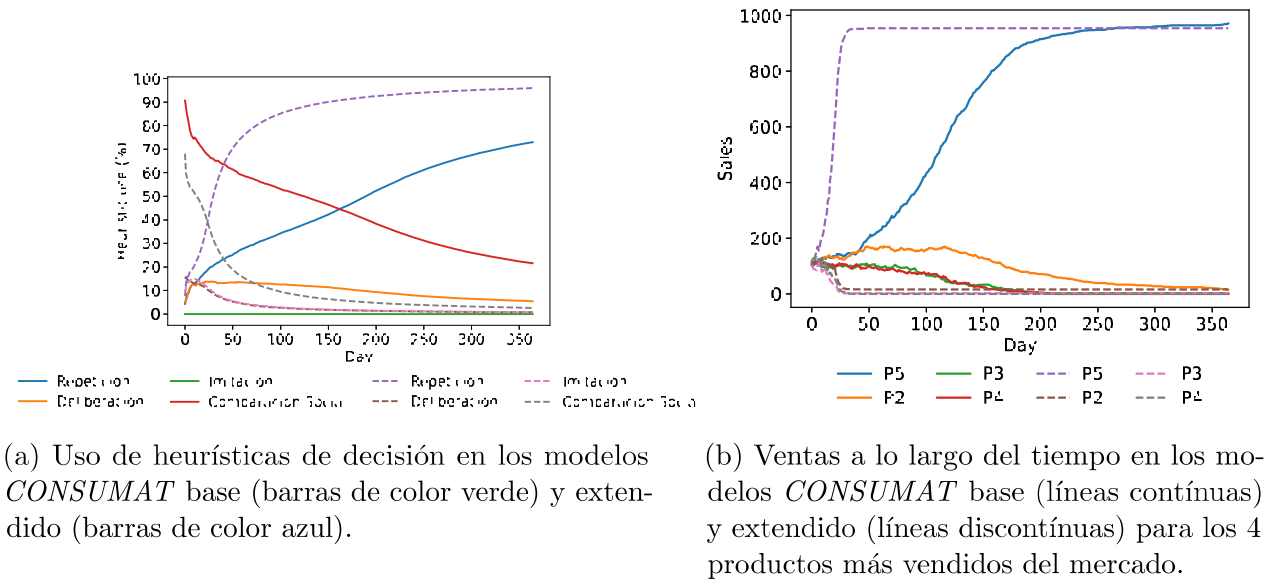


Figura 4.2: Uso de heurísticas de decisión y consumo de productos (tanto temporal como total) en los modelos *CONSUMAT* base y extendido en una topología de red aleatoria.

comportamiento de los agentes en los modelos original y extendido haciendo especial énfasis en describir cómo afecta el conocimiento limitado del mercado y la probabilidad de olvido a las dinámicas del modelo.

### 4.3.1. Configuración experimental

La Tabla 4.1 contiene los parámetros utilizados para simular el modelo original y ampliado de *CONSUMAT*.

Parámetro	Descripción	Valor
<b>Parámetros comunes</b>		
$ I $	Número de agentes	1000
$ B $	Número de marcas	8
$ T $	Pasos de simulación	365
$R$	Ejecuciones Monte Carlo	15
<b>Parámetros modelo <i>CONSUMAT</i> base</b>		
$U$	Utilidad mínima	0.4
$\Phi$	Incertidumbre mínima	0.2
$\beta$	Modulador consumo personal/social	0.6
<b>Parámetros modelo <i>CONSUMAT</i> extendido</b>		
$p_b$	Probabilidad de compra	1.0
$p_\theta$	Probabilidad de difusión	0.4
$\epsilon$	Probabilidad de olvido $b$	0.1

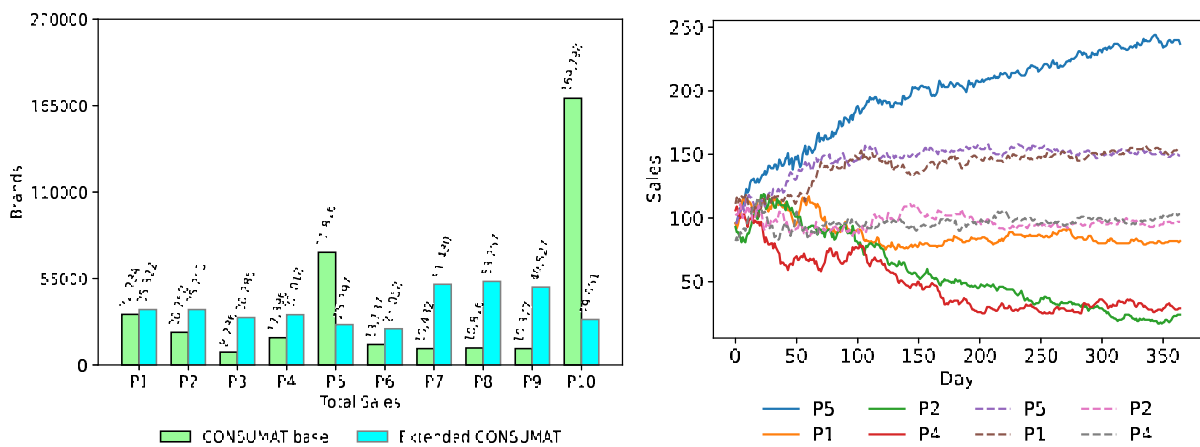
Tabla 4.1: Lista de parámetros para el modelo *CONSUMAT* base y extendido.

El parámetro  $p_b$  determina el número de agentes que actúan como consumidores (es decir, agentes que compran y difunden) en cada paso del modelo de simulación. Consideramos los mismos parámetros para cada tipo de SN utilizado en los experimentos. Las SN estarán formadas por un total de 1000 nodos, con un grado medio de 10. Para la red libre de escala, utilizamos  $m = 3$  como el número de nodos existentes que se conectan a un nuevo nodo durante el algoritmo de acoplamiento preferencial (para más información, ver Sección 2.3.2.2). Para la red de mundo pequeño, utilizamos  $p = 0.2$  como probabilidad de reconexión (ver Sección 2.3.2.3).

Por su parte,  $R$  determina el número de simulaciones Monte Carlo a realizar. Una simulación de Monte Carlo es una técnica matemática que se utiliza para estimar los posibles resultados de un suceso incierto [Moo97]. Cuando se utiliza un modelo probabilístico para simular una salida se obtienen resultados diferentes cada vez, tal y como ocurre en las simulaciones de ABMs. Por tanto, utilizaremos las simulaciones de Monte Carlo para repetir ejecuciones del modelo bajo distintas condiciones para así reducir los posibles sesgos que puedan producirse por aleatoriedad.

### 4.3.2. Impacto en redes libres de escala

La Figura 4.3 muestra la evolución de las ventas del modelo sobre redes libres de escala donde los agentes tienen un conocimiento completo del entorno. Los resultados mostrados se generan calculando la media de simulaciones de Monte Carlo independientes. Como puede observarse en la Figura 4.3, los clientes tienden a elegir un solo producto que acaba liderando el mercado. Cuando este conocimiento es limitado, el comportamiento del modelo cambia y la variabilidad del consumo de los productos aumenta (los clientes realizan compras más heterogéneas durante el período de simulación).



(a) Ventas totales para los 10 productos simulados en los modelos *CONSUMAT* base (barras de color verde) y extendido (barras de color azul).

(b) Ventas a lo largo del tiempo en los modelos *CONSUMAT* base (líneas continuas) y extendido (líneas discontinuas) para los 4 productos más vendidos del mercado.

Figura 4.3: Adquisición de productos (total y paso a paso) en los modelos *CONSUMAT* base y extendido en la red libre de escala.

Por el contrario, cuando el conocimiento de los productos es limitado en el entorno (uso de *awareness*), este intercambio de conocimientos y la compra se producen de forma gradual. Los consumidores con mayor grado (mayor número de contactos) tienen más influencia en el proceso de WOM ya que propagan sus elecciones a una parte de sus contactos. Este hecho provoca que el tiempo de difusión del producto sea más pausado y, por consiguiente, que disminuya el número de consumidores que acaban comprando un determinado producto.

La Figura 4.4 muestra la evolución en el uso de heurísticas de decisión en el modelo *CONSUMAT* (ver Sección 4.2.1.3). En el modelo base (con conocimiento pleno del entorno), los agentes



comienzan a utilizar cualquiera de las heurísticas de decisión que implican un cambio de producto (deliberación, imitación o comparación social). Cuando un producto comienza a destacar en el número de unidades vendidas, los consumidores empiezan a comprarlo para cubrir sus necesidades y, una vez cubiertas, repiten su última compra hasta el final del periodo de simulación. En nuestro modelo ampliado, la heurística de deliberación es más relevante. Los agentes no tienen incertidumbre sobre su última elección, pero aún no están satisfechos con ella. Por lo tanto, no hacen uso de las heurísticas de imitación y comparación. Como podemos observar en el gráfico situado a la derecha de la Figura 4.3, las características de la SN libre de escala y los procesos de WOM también contribuyen a que los productos elegidos por los agentes a través de la simulación sean muy heterogéneos.

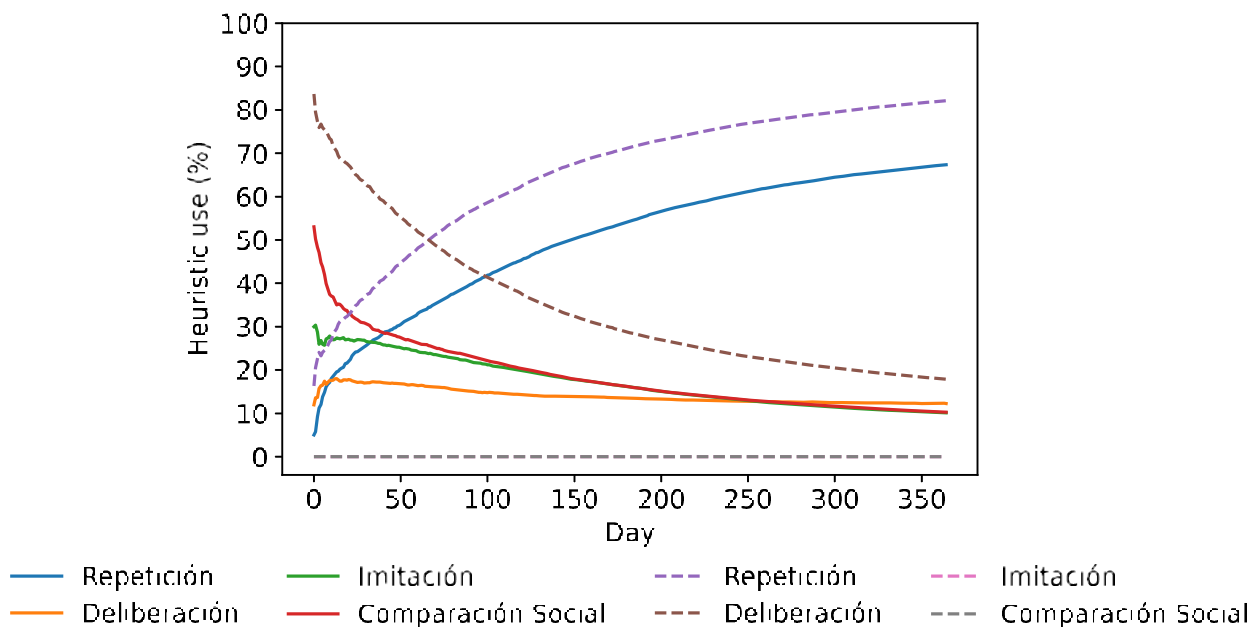
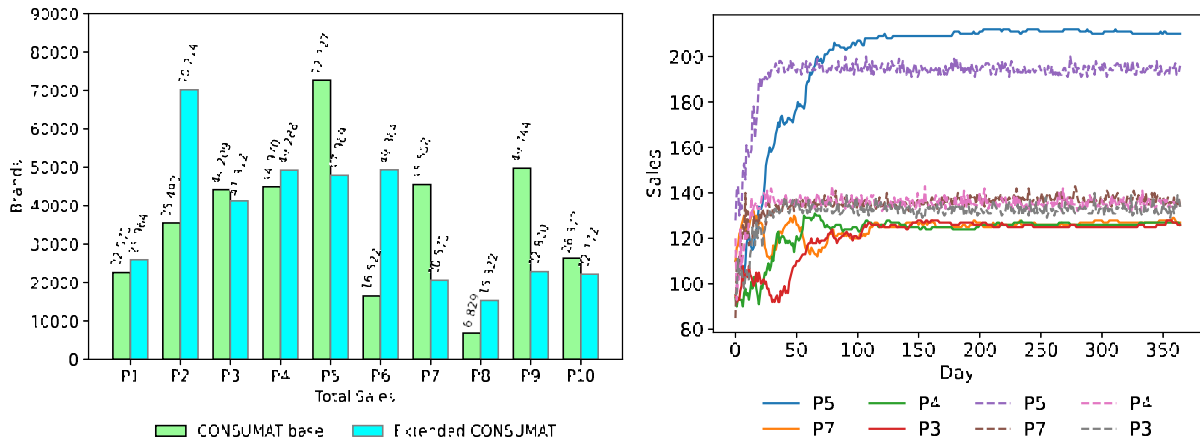


Figura 4.4: Uso de heurísticas de decisión en los modelos *CONSUMAT* base (líneas continuas) y extendido (líneas discontinuas) en la red libre de escala.

### 4.3.3. Impacto en redes de mundo pequeño

Como se puede ver en la Figura 4.5, el comportamiento de ambos modelos *CONSUMAT* es más similar que en las simulaciones realizadas sobre la red libre de escala. Los productos más difundidos en la SN son los más comprados en el mercado y su venta depende del tamaño de las comunidades existentes en la red. El proceso de difusión, sin embargo, es más lento debido a las

diferencias de distribución de grados entre ambos modelos de SN.



(a) Ventas a lo largo del tiempo en los modelos *CONSUMAT* base (líneas continuas) y extendido (líneas discontinuas) para los 4 productos más vendidos del mercado.

(b) Ventas totales para los 10 productos simulados en los modelos *CONSUMAT* base (barras de color verde) y extendido (barras de color azul).

Figura 4.5: Consumo de productos (total y paso a paso) en los modelos *CONSUMAT* base y extendido en la red de mundos pequeño.

En cuanto al uso de heurísticas de decisión, podemos ver en la Figura 4.6 como los agentes pasan de utilizar la imitación, la comparación social y la repetición en el modelo original a la deliberación y la repetición en el modelo extendido. Como habíamos visto en el apartado anterior, en el modelo base de *CONSUMAT*, los agentes hacen uso de la heurística que representa un cambio en su elección en las primeras iteraciones para cubrir sus umbrales de incertidumbre y utilidad. A partir de ese momento inicial, utilizan la repetición para reforzar el producto elegido. Limitando el uso del conocimiento por parte de los procesos de difusión, podemos obtener una elección de las heurísticas similar al caso de la red libre escala pero con un uso más pronunciado de la heurística de deliberación y repetición. Sin embargo, los procesos de WOM sobre los nuevos productos hacen que los clientes estén más insatisfechos y tengan que deliberar para elegir un producto mejor. Así, hasta el final de la simulación, los agentes deliberan y repiten su comportamiento.

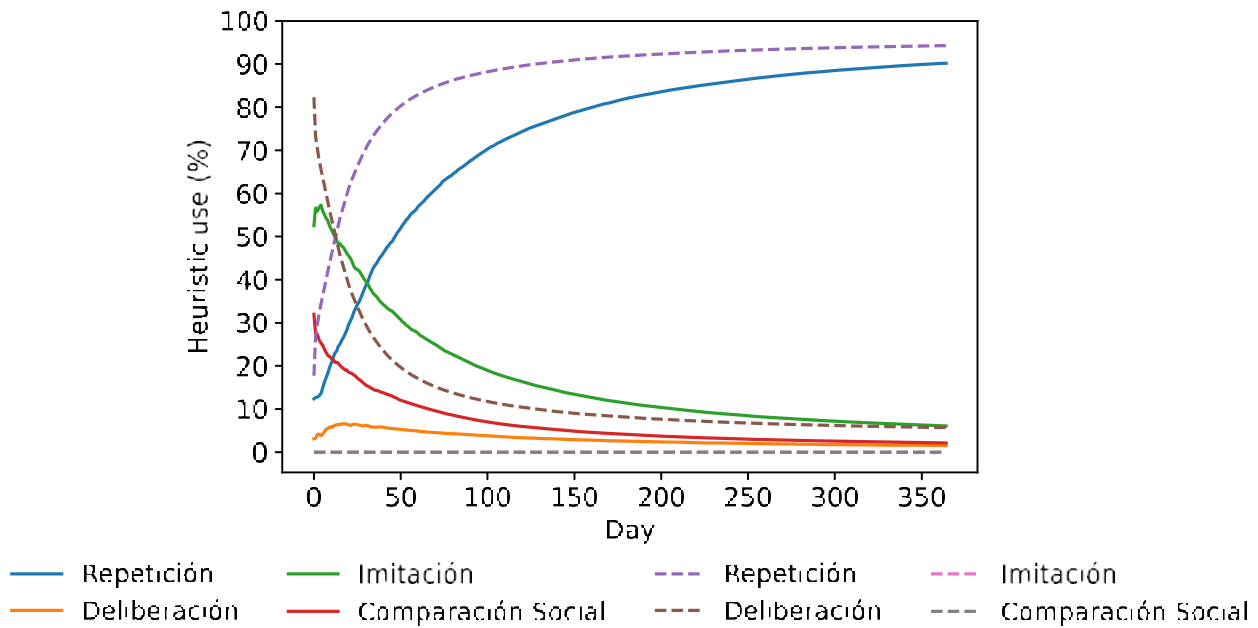


Figura 4.6: Uso de heurísticas de decisión en los modelos *CONSUMAT* base (líneas continuas) y extendido (líneas discontinuas) en la red de mundo pequeño.

#### 4.3.4. Discusión de los resultados

A la vista de los anteriores resultados podemos concluir que la limitación de *awareness* y el proceso de difusión para su obtención incorporados al modelo *CONSUMAT* proporciona un comportamiento más realista a los agentes consumidores. El patrón de consumo comienza a comportarse como un mercado real en el que los agentes tienen o no conocimiento de los productos que compran y son capaces de difundirlo a través de la SN subyacente. Además de la difusión de opiniones, el mecanismo encargado de modular el olvido de cierta información sobre los productos también influye de forma significativa en la toma de decisiones de los agentes. En el modelo *CONSUMAT* extendido, los procesos de compra se realizan de una forma más pausada debido a esta etapa adicional de ganancia/pérdida de conocimiento. Otro aspecto a tener en cuenta es que el uso de las heurísticas de decisión es más diverso en el modelo extendido y que se ven notablemente afectadas por las características de las diferentes topologías de SN utilizadas. En las redes libres de escala, por ejemplo, la difusión de la información sobre los productos se produce más rápidamente que en las SN de mundo pequeño, dando lugar a un mayor uso de la heurística de deliberación debido a la existencia de *hubs* en la red. Por su parte,

las heurísticas predominantes en la red de mundo pequeño son deliberación y repetición debido a la menor rapidez en la difusión de información en la red y la necesidad de los consumidores de repetir su compra ante la carencia de información de su entorno.

# Selección de *influencers* mono y multiobjetivo para campañas de marketing viral

Vivir no es sólo existir,  
sino existir y crear,  
saber gozar y sufrir  
y no dormir sin soñar.  
Descansar, es empezar a morir.

---

Gregorio Marañón

## 5.1. Introducción

Una vez ampliado y validado el modelo introducido en el capítulo anterior, procedemos a utilizarlo para un objetivo concreto: El diseño automático de campañas de VM mediante la identificación de consumidores influyentes, es decir, *influencers* o consumidores semilla.

La identificación y captación de *influencers* en SNs así como su posterior uso como difusores de productos ofrecen a las empresas y organizaciones es una importante baza para dirigir eficazmente sus estrategias de marketing. De este modo, se pueden realizar campañas publicitarias con un coste reducido que, en la mayoría de los casos, se limita al valor de los artículos o descuentos con los que obsequian a los *influencers*. Así, los individuos seleccionados difunden opiniones positivas a otros consumidores a través del WOM, proporcionando un punto de partida para las

estrategias de VM [LAH07]. La identificación efectiva de *influencers* depende de varios factores entre los que se encuentran las características de la SN subyacente y los procesos de difusión de WOM.

La búsqueda de individuos semilla en SNs fue presentada por primera vez como un problema de optimización por Domingos et al. [DR01]. Posteriormente, Kempe et al. mostraron que este problema es, en esencia, un problema de optimización *NP-hard* y propusieron un algoritmo voraz que requería un conocimiento global de la red para resolverlo [KKT03]. Otros autores abordaron los problemas de selección de *influencers* en VM desde enfoques descriptivos [RD02; SR03; LAH07].

Por su parte, autores como [SRW10; LC10] o [SR18] abordaron el problema utilizando metaheurísticas como los GAs y los algoritmos de enjambre de partículas, respectivamente. La selección de *influencers* en SNs también ha sido investigado usando enfoques y técnicas más actuales como el Deep Learning [Kei+19] o la detección de comunidades [BJP19] para explorar las estructuras subyacentes de las SN y utilizar métodos de difusión para maximizar la adopción de innovaciones a través de las mismas.

Este capítulo desarrolla dos propuestas de técnicas automáticas para abordar la selección de *influencers* para el diseño campañas de VM. En primer lugar, proponemos una técnica basada en GAs monoobjetivo que unifica el beneficio y coste de las campañas de VM en un único indicador de rendimiento de estas campañas. Una vez analizados los resultados de la propuesta monoobjetivo, desarrollamos una segunda propuesta basada en algoritmos EMO [Deb01; CLVV+07] que considera la optimización de los dos objetivos anteriores de forma independiente durante el proceso de búsqueda, en un enfoque multiobjetivo.

Utilizamos como base de nuestra experimentación el modelo *CONSUMAT* extendido que presentamos en el capítulo anterior. Este ABM se utiliza para evaluar el *fitness* de cada individuo (es decir, la eficacia de las campañas de VM) a través de los objetivos de coste de las campañas y adopción conseguida en las poblaciones de los algoritmos utilizados en las propuestas mono y multiobjetivo. Con la combinación de técnicas de optimización y un ABM sólido, podemos diseñar estrategias efectivas de VM. En la Figura 5.1 se muestra un esquema

de los módulos considerados en los métodos avanzados de VM para la selección de *influencers* en campañas de VM y su relación con el modelo *CONSUMAT* extendido.

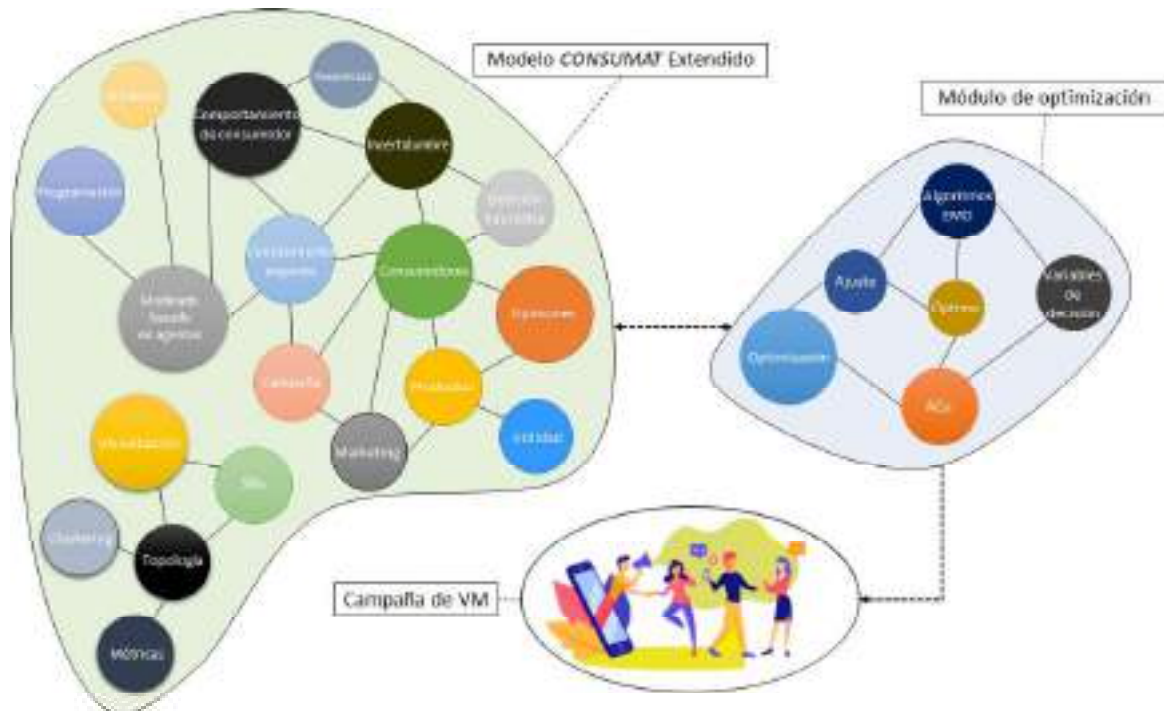


Figura 5.1: Metodología y módulos incluidos en los métodos avanzados de VM y relación con el modelo *CONSUMAT* extendido.

Las principales investigaciones y aplicaciones dedicadas a la identificación de los *influencers* más efectivos a los que dirigirse para lanzar campañas de VM tienen en cuenta un único objetivo: Maximizar los ingresos obtenidos de las ventas generadas en las campañas [SRW10; Sch+13]. En nuestra propuesta, se plantean dos enfoques alternativos. Por un lado, combinamos el beneficio y coste de las campañas de VM (dos objetivos en claro conflicto) en nuestro enfoque monoobjetivo. En la sección dedicada a la experimentación y análisis de resultados de esta propuesta comprobaremos que existen serias limitaciones a la hora de proponer soluciones realistas al problema usando este planteamiento. En consecuencia, proponemos un enfoque multiobjetivo para lidiar con la maximización de los ingresos de las campañas de marketing y la reducción de su coste. La aplicación de algoritmos EMO al problema permitirá trabajar con beneficio y coste al mismo tiempo que se localiza a los mejores individuos semilla utilizando métricas locales de SNs [NBW06].

Nuestra primera propuesta diseñará campañas de VM mediante el uso de GAs. Se compararán

los resultados de las estrategias de VM generadas y su impacto en función de las distintas topologías de redes y del número de consumidores semilla a seleccionar. Demostraremos que el *word-of-mouth* y las dinámicas de compra de los consumidores evolucionan más lentamente debido al filtro de conocimiento incorporado y que el GA, por su parte, es capaz de encontrar buenas soluciones para identificar a los *influencers* del mercado así como para adaptar su selección a distintas topologías de SN.

En nuestra segunda propuesta hacemos uso de dos de los algoritmos EMO más utilizados en el área de optimización multiobjetivo, NSGA-II [Deb+02] y MOEA/D [ZL07], para generar un conjunto de soluciones no dominadas que permitan a los responsables de marketing considerar múltiples opciones para focalizar sus campañas. Nuestra experimentación comprende diferentes escenarios de marketing utilizando cinco topologías de SN. Estas cinco SNs representan las conexiones entre los consumidores en el mercado y son: Una red real basada en las comunicaciones por correo electrónico entre académicos de Universidad [Gui+03], una red libre de escala [BA99], una red de mundo pequeño [WS98], una red aleatoria [ER+60] y una red de retículo regular. Además, y para validar mejor la escalabilidad de nuestro enfoque, utilizamos una SN generada por datos reales de 20,000 nodos, construida a partir de los usuarios de una aplicación *freemium* [CR17]. Comparamos la calidad de las soluciones no dominantes encontradas por los algoritmos EMO con una adaptación multiobjetivo del GA monoobjetivo desarrollado en nuestra primera propuesta y con un algoritmo voraz. El diseño de nuestra experimentación también nos permite analizar la influencia de cada topología de SNs en las soluciones y el rendimiento de los algoritmos. Por último, exploramos las políticas de marketing obtenidas a partir del conjunto de soluciones no dominadas generados. Los resultados muestran una clara superioridad de MOEA/D y NSGA-II sobre la adaptación multiobjetivo del GA monoobjetivo y el algoritmo voraz y una correlación entre el rendimiento de los algoritmos y las características de difusión de las SN sobre las que se aplican.

El capítulo está estructurado como sigue. Comenzamos revisando el estado del arte respecto al problema de la maximización de la influencia y VM en la Sección 5.2. A continuación, la Sección 5.3 desarrolla nuestra propuesta del método evolutivo monoobjetivo para la generación automática de campañas de VM. Los experimentos realizados sobre la versión extendida del



modelo y el análisis de los resultados quedan recogidos en la Sección 5.4. Finalmente, las Secciones 5.5 y 5.6 abarcan nuestra propuesta para el diseño de campañas de VM basada en algoritmos EMO y los experimentos realizados y su análisis, respectivamente. Tanto el modelo ABM como los algoritmos EMO utilizados en este capítulo fueron detallados en profundidad en las Secciones 4.2 y 3.3.2, respectivamente.

## 5.2. Estado del arte

Las SNs electrónicas, como Facebook o Instagram, hacen que las personas (consumidores potenciales) estén más conectadas que nunca. Con unas pocas acciones, los consumidores pueden comunicar al instante sus productos y las opiniones de las marcas. Los usuarios más influyentes de las SN aglutinan miles de amigos y un proceso de *word-of-mouth* puede crear una cascada de información positiva o negativa sobre una marca. Este proceso de *word-of-mouth* en las SNs es crucial para los vendedores y anunciantes [LAH07; Wat+07; HL17]. De hecho, las personas cada vez dan más valor a las recomendaciones de sus contactos que a las de los canales publicitarios tradicionales, como la televisión o la radio. El VM consiste en dirigirse a determinados consumidores para que promocionen el producto, provocando así una cascada que fomenta una adopción más rápida del producto [HL17]. La selección de estos individuos no es aleatoria, sino que tiene asociado un complejo proceso de optimización que implica el análisis de la SN de consumidores para favorecer una difusión positiva de la información. Diseñar las mejores estrategias de VM antes de ejecutarlas en el mercado real es posible mediante el uso de SN [Wat+07] y simulaciones que utilizan paradigmas como el modelado basado en agentes [Eps07]. El ABM puede describir y simular las interacciones a nivel de red entre los consumidores para reproducir los mecanismos del WOM que aportan realismo a los modelos de marketing y permiten a los modelizadores estudiar los comportamientos emergentes [JJ03; Eps07].

La búsqueda de individuos semilla en SN se presentó por primera vez como un problema de optimización denominado maximización de la influencia, *influence maximization* (IM) en [DR01].

Más tarde, Kempe et al. mostraron que la maximización de la influencia es esencialmente un problema de optimización *NP-hard* y propusieron un algoritmo voraz para resolverlo, aunque el algoritmo requería un conocimiento global de la red [KKT03]. Otros autores utilizaron un enfoque descriptivo para los problemas de VM [LAH07]. Mientras tanto, los investigadores más cercanos al área del marketing e investigación de mercados utilizaron ABMs para describir los procesos de adopción y caracterizar a los individuos que tienen un mayor efecto en la adopción global [Sch+13], mostrando que el papel de los *hubs* o *influencers* es importante para los procesos de adopción de nuevos productos. Algunos otros autores han mejorado la aproximación al problema de maximización de la influencia mediante el uso de metaheurísticas como los GAs [SRW10; LC10] o los algoritmos de inteligencia de enjambre [SR18]. Otros autores han abordado el problema de maximización de la influencia desde una perspectiva diferente. Sus enfoques consisten en el uso de Deep Learning [Kei+19], o la detección de comunidades [BJP19] para explorar las estructuras subyacentes de las SNs y luego utilizar métodos de difusión para maximizar la influencia a través de las comunidades o entre las redes interconectadas.

Entre las diferentes aproximaciones al problema de VM estudiadas por otros autores la más similar a la nuestra es la presentada en [SRW10], donde los autores propusieron políticas prescriptivas para el manejo de campañas de selección de *influencers* con un nivel de conocimiento de la SN asequible. Esta última propuesta utilizaba GAs para generar campañas VM utilizando medidas de análisis de SN locales y globales.

Aunque la optimización evolutiva multiobjetivo se ha utilizado ampliamente en el ámbito de las SNs, la mayoría de los esfuerzos relacionados con nuestro trabajo se llevaron a cabo para otros problemas relacionados, como la detección de comunidades. Autores como Gunasekara et al. consideraron algoritmos EMO para identificar a los actores clave en grandes SNs [GMM15]. En el momento en el que comenzamos a desarrollar nuestra propuesta no existía ninguna investigación basada en algoritmos EMO previa que tratase objetivos conflictivos a la hora de encontrar y captar a individuos semilla en campañas de VM. Normalmente, existe una simplificación del problema inherente de dos objetivos (maximizar los ingresos y minimizar los costes) en un único objetivo (véase [SRW10]). En nuestro caso, abordamos el problema desde una perspectiva multiobjetivo utilizando algoritmos EMO para tratar conjuntamente los dos objetivos clave de

las campañas. Merece la pena destacar que, en nuestra propuesta no tenemos en cuenta ninguna de las métricas global asociadas a las SNs (ver Sección 2.3.3.2), ya que asumimos como poco realista el manejar conocimiento global de la estructura de la SN cuando se lanzan las campaña de VM.

### 5.3. Diseño de un método automático para la definición de estrategias de VM basado en algoritmos genéticos

Utilizando el modelo *CONSUMAT* extendido (ver Sección 4.2) como motor de simulación, proponemos hacer uso de GAs [Gol89] para localizar y utilizar a los *influencers* como difusores para maximizar los beneficios de un producto o marca en campañas de VM. En general, los GAs son métodos de búsqueda potentes cuando se aplican sobre espacios de búsqueda complejos y extensos que han sido aplicados con éxito a diferentes problemas de marketing [Luo11], estudio de modelos de difusión [VKK04] y de maximización de influencia [SRW10].

El uso de GAs como base para la optimización de campañas de VM permite crear un entorno realista en el que simular la adquisición de productos por parte de los consumidores y optimizar sus ventas en diferentes escenarios. Nuestro enfoque se basa en la definición automática de una función heurística que combina diferentes métricas utilizadas para el análisis de SNs como el coeficiente de clustering, el grado y el número de vecinos para encontrar el conjunto óptimo de *influencers* que permita optimizar el beneficio de campañas de VM. Estas y otras métricas utilizadas en la teoría de redes están descritas en la Sección 2.3.3.

En las siguientes secciones describimos el diseño elegido para el GA. Comenzamos desarrollando las variables de decisión y las funciones objetivo utilizadas para evaluar las soluciones obtenidas en las Secciones 5.3.1 y 5.3.2, respectivamente. A continuación, describimos la inicialización y evaluación de los cromosomas, las estrategias de selección, los operadores de cruce y mutación en las Secciones 5.3.3, 5.3.4 y 5.3.5. Por último, la estrategia de reemplazo utilizada se desarrolla en la Sección 5.3.6.

### 5.3.1. Variables de decisión

En primer lugar, definimos una forma de ponderar lo “influyente” que es un agente utilizando una adaptación del enfoque definido por [SRW10]. De esta forma, medimos el grado de influencia de un agente combinando su grado  $d_i$ , el grado de sus vecinos directos  $d2S_i$  y su coeficiente de agrupamiento  $CC_i$ . Cada cromosoma del GA definirá una combinación de estas medidas para formar una solución al problema según la estructura de la SN subyacente. Nótese que las tres medidas seleccionadas son todas medidas locales y no globales (ver Sección 2.3.3.1). El motivo de usar estas métricas locales es que resultan más realistas para el diseño de campañas virales ya que pueden ser calculadas sin tener un conocimiento completo de la topología de la red social subyacente.

Las variables de decisión del proceso de optimización utilizadas por el GA son: i) los pesos de la combinación lineal de las tres medidas definidas anteriormente y ii) el número de *influencers* a seleccionar. La Figura 5.2 muestra la codificación de cada cromosoma de la población  $P$  utilizada por el GA:

$\omega_d$	$\omega_{2S}$	$\omega_{CC}$	$s$
------------	---------------	---------------	-----

Figura 5.2: Codificación de cromosomas en el GA.

donde  $\omega_d \in [0, 1]$  representa el peso aplicado a la ponderación del grado los nodos,  $\omega_{2S} \in [0, 1]$  representa el peso aplicado al número de clientes alcanzables a través de sus vecinos y  $\omega_{CC} \in [0, 1]$  representa el peso aplicado a su coeficiente de agrupación. También incorporamos una cuarta variable de decisión  $s \in \{1, 2, \dots, S_{max}\}$  que define el número de individuos semilla a los que seleccionar en las campañas de VM.  $S_{max}$  es el número máximo de *influencers* a seleccionar del total de agentes consumidores  $I$ .

Realizamos una combinación ponderada de las medidas anteriores usando la ecuación 5.1 para obtener un valor real  $q(i)$  para cada nodo  $i$  que indica el grado de influencia de cada agente dentro de la SN (los valores más altos significan mayor influencia):

$$q(i) = \omega_d \frac{\text{degree}(i)}{N-1} + \omega_{2s} \frac{\text{2-steps}(i)}{N-1} + \omega_{cc}(1 - cc(i)), \quad (5.1)$$

donde  $i$  es el índice del agente consumidor que está siendo evaluado como posible *influencer* para su posterior selección. El proceso para identificar el conjunto de influyentes ( $S$ ) consiste en evaluar primero todos los agentes calculando sus valores  $q(i)$ . Posteriormente, los ordenamos de forma descendente y elegimos los primeros  $s$  agentes para que actúen como difusores en la campaña de VM.

### 5.3.2. Funciones objetivo

Utilizamos el modelo *CONSUMAT* extendido para simular el mercado y evaluar la eficacia de las campañas de VM codificadas en los individuos después de dar a los  $s$  *influencers* seleccionados muestras gratuitas del producto  $k \in J$  (es decir, el producto cuyas ventas se pretenden maximizar en la campaña de VM).

La eficacia de una campaña se mide en base a dos objetivos: i) la maximización del número de ventas del producto  $k$  (es decir, maximización de los ingresos), y ii) la minimización los costes de la campaña. Obsérvese que los individuos semilla del conjunto  $S$  no pueden comprar productos durante la simulación ya que obtienen muestras gratuitas del producto  $k$  cuando ejercen sus acciones de compra. El segundo objetivo es el coste de la campaña asociado al coste de las muestras gratuitas del producto proporcionadas a los *influencers* seleccionados a lo largo del tiempo.

El *fitness* de una solución se obtiene simulando el modelo seleccionando a  $s$  agentes (valor de parámetro codificado en la solución candidata que está siendo evaluada) como *influencers* usando la combinación de pesos de  $\omega_a$ ,  $\omega_{2S}$  y  $\omega_{CC}$  (ver ecuación 5.1), valores también codificados en el cromosoma, y midiendo el beneficio (número de ventas de productos) y coste de la campaña de VM asociada a la estrategia diseñada. El objetivo de las campañas de VM es difundir rápidamente un producto a través de la SN del modelo a través de los *influencers* para obtener el mayor beneficio con el menor coste posible. Además, no solo es interesante maximizar el volumen de ventas sino también que dichas ventas se produzcan cuánto antes, lo que presenta varias ventajas como por ejemplo la reducción de los costes de almacenamiento de los productos. De este modo, utilizamos el Net Present Value (NPV) para calcular el coste

y beneficio de las campañas de VM. El NPV es una medida de los ingresos de una empresa cuando los consumidores compran el producto en el momento en que éste se ha comercializado y no varios meses después de su comercialización. Matemáticamente, se define como  $\sum_{t=1}^{\infty} |A^t| p \lambda^t$  donde  $|A^t|$  es el número de adoptadores o compradores del producto  $k$  en el momento  $t$ ,  $p$  es el beneficio que obtiene la empresa por la adquisición de un producto por parte de un consumidor, y  $\lambda$  es el factor de descuento. Este factor de descuento  $\lambda$  significa que es preferible una compra más temprana debido al efecto acumulativo de varios factores como los costes de oportunidad y la posible necesidad de fijar precios más bajos con el tiempo para ser más competitivos.

El beneficio de una campaña de VM es el NPV obtenido acumulando los ingresos de las compras del producto  $k$  en cada paso de tiempo  $t$ . El número de compras obtenidas del conjunto de adoptadores  $|A_j^t|$  se multiplica por el factor de descuento  $\lambda$  relacionado con el momento de la adquisición ya que fijamos  $p = 1$  para cada producto vendido. La Ecuación 5.2 describe el cálculo del NPV:

$$f^0(x) = NPV(A_k) = \sum_{t=1}^T |A_k^t| \lambda^t. \quad (5.2)$$

Por su parte, el coste de una campaña de VM está ligado a la inversión realizada al seleccionar a los  $s$  influencers del conjunto de consumidores  $I$  con muestras gratuitas del producto  $k$  cuando deciden comprar según la probabilidad  $p_b$ . De esta forma, el coste de una campaña es el de pagar a los individuos semilla seleccionados para que difundan un determinado producto. En este caso la inversión realizada por la empresa se corresponde la entrega gratuita de una unidad de producto a cada influencer seleccionado.

Obsérvese que el coste de una campaña entra en conflicto con su beneficio, ya que al aumentar el número de influencers ( $s$ ) se reduce el número de posibles consumidores que adoptan el producto ( $A_k^t$ ) y se aumenta el coste de la misma. Definimos  $S_{x^t=k}$  como el conjunto de influencers a los que se regala el producto en el paso de tiempo  $t$ . Los costes de la campaña se calculan multiplicando el número de muestras proporcionadas por un coste de fabricación de  $\frac{1}{10}$  y el

factor de descuento descrito en la Ecuación 5.3:

$$f^1(x) = Cost(S) = \frac{1}{10} \sum_{t=1}^T |S_{x^t=k}| \lambda^t. \quad (5.3)$$

### 5.3.3. Inicialización y evaluación de los cromosomas

La primera generación de la población del GA se inicializa aleatoriamente y se evalúa simulando el modelo *CONSUMAT* extendido utilizando  $s$  individuos semilla seleccionados mediante la combinación lineal de los valores de cada cromosoma definida en la Ecuación 5.1. El valor definitivo de *fitness* para cada cromosoma será el NPV (Ecuación 5.2) obtenido tras realizar la campaña de VM, consecuencia de las adquisiciones resultantes de la misma, sustrayendo los costes de la campaña que dependen del número de individuos semilla que hayan sido contratados en dicha campaña (Ecuación 5.3). La Ecuación 5.4 muestra el cálculo del *fitness* de las soluciones de nuestro método automático para la definición de estrategias de VM basado en GA:

$$Fitness = NPV - Coste(s). \quad (5.4)$$

### 5.3.4. Mecanismo de selección de los padres

La estrategia de selección del GA sigue un enfoque estacionario [Sys91] que utiliza una selección por torneo  $k$  binario con  $k = 2$ . Este tipo de selección escoge al azar dos individuos distintos de la población y compara sus valores objetivo para quedarse con el mejor. Repetimos esta operación dos veces hasta seleccionar dos individuos en cada generación y les aplicamos los operadores genéticos descritos en la Sección 5.3.5.

### 5.3.5. Operadores de cruce y mutación

El GA utiliza un operador de cruce BLX- $\alpha$  [ES93] para cruzar la descendencia con una probabilidad  $p_c \in [0, 1]$ . Dados dos padres  $C^1 = (C_1^1, \dots, C_n^1)$  y  $C^2 = (C_1^2, \dots, C_n^2)$ , BLX- $\alpha$

genera dos descendientes  $D^1 = (d_1^1, \dots, d_i^1, \dots, d_n^1)$  y  $D^2 = (d_1^2, \dots, d_i^2, \dots, d_n^2)$ , donde cada gen  $d_i$  se genera aleatoriamente en el rango  $[C_i^{min} - \alpha I, C_i^{max} + \alpha I]$ , con  $C_i^{max} = \max\{C_i^1, C_i^2\}$ ,  $C_i^{min} = \min\{C_i^1, C_i^2\}$ , e  $I = C_i^{max} - C_i^{min}$ .

Por su parte, el operador de mutación se aplica a cada individuo con una probabilidad  $p_m \in [0, 1]$  para reiniciar uno de sus genes al azar dentro del intervalo de valores específico de cada gen (mutación aleatoria). Si es elegido el primer, segundo o tercer gen, el operador de mutación genera un valor real dentro del rango definido, y un valor entero en el caso de mutar el cuarto gen.

### 5.3.6. Estrategia de reemplazo

Una vez obtenida la población de hijos  $P'$ , los dos hijos que la componen competirán con los individuos de la población actual para reemplazarlos en su población  $P$ . La estrategia de reemplazo se basará en una comparación entre dos hijos y los dos peores padres para obtener un lugar en  $P$ . Si el *fitness* de los hijos en  $P'$  es mejor que la de sus padres, entonces los reemplazarán en  $P$  y pasarán a la siguiente generación del GA.

En nuestro algoritmo estacionario se generan un 20% de nuevos individuos en cada generación que compiten con los peores individuos de la población para sobrevivir (o mantenerse) como soluciones en la próxima generación del algoritmo. El algoritmo sigue un enfoque intrínsecamente elitista, por lo que los mejores individuos (aquellos con mejor ajuste) se mantienen durante su ejecución.

## 5.4. Experimentos y análisis de resultados sobre la propuesta monoobjetivo

En esta experimentación estudiamos las diferencias de comportamiento entre el modelo *CONSUMAT* original y el extendido para comprender cómo se produce la difusión del *awareness*



en el modelo y los cambios que implica en las dinámicas de decisión y compra de los agentes. Además, analizamos la influencia de la topología de la SNs subyacente en los procesos de WOM y el impacto y efectividad de las estrategias de VM basadas en GA en la difusión de productos y su adquisición en el modelo de consumo.

### 5.4.1. Configuración experimental

La tabla 5.1 muestra los parámetros usados en el GA que encuentra las campañas de VM óptimas en las distintas SNs utilizadas. Los valores de parámetros utilizados para simular el modelo *CONSUMAT* así como sus descripciones pueden consultarse en la Sección 4.3.1.

Parámetro	Descripción	Valor
$R$	Número de ejecuciones	20
$P$	Población del algoritmo	100
$p_c$	Probabilidad de cruce	0.7
$p_m$	Probabilidad de mutación	0.1

Tabla 5.1: Lista de parámetros para el GA monoobjetivo.

### 5.4.2. Experimentos del método automático de diseño de campañas de VM en el modelo *CONSUMAT* extendido

En esta sección analizamos los resultados de la aplicación de nuestro método de diseño de estrategias de VM basado en GAs sobre el modelo *CONSUMAT* extendido sobre varias topologías de SN.

La Tabla 5.2 muestra los resultados obtenidos por el GA en las redes libre de escala y de mundo pequeño. La información mostrada en esa tabla recoge la composición de la mejor estrategia de marketing obtenida en cada caso, reflejada en forma de los pesos asociados a cada medida de análisis de SNs en la función de selección de individuos semilla. A la vista de las combinaciones de los parámetros  $\omega_d$ ,  $\omega_{2S}$ ,  $\omega_{CC}$  y  $s$ , no siempre los agentes con mayor grado son los más beneficiosos para identificar a los difusores de productos en las campaña de VM. Dependiendo de la topología de la SN y de los parámetros que modelan el comportamiento del

consumidor, hay casos en los que la estrategia se enfoca en dirigirse a aquellos agentes con un alto coeficiente de clustering ya que este tipo de *influencers* pueden acceder a un mayor número de consumidores a través de las conexiones de sus vecinos cercanos.

SN	Variables decisión mejor solución				<i>Fitness</i>	
	$\omega_d$	$\omega_{2S}$	$\omega_{CC}$	$s$	<i>Fitness</i>	Ventas
Libre de escala	0.5292	0.6703	0.6168	82	2678.651	621
Mundo pequeño	0.355	0.629	0.122	99	718.4988	948

Tabla 5.2: Resultados obtenidos por el GA para las redes libre de escala y de mundo pequeño.

En los resultados contenidos en la Tabla 5.2 podemos comprobar cómo en la red libre de escala las soluciones recomendadas por el GA proponen seleccionar como individuos semilla atendiendo primero al valor de grado en dos pasos, seguido del coeficiente de clustering (aunque ambos pesos están prácticamente al mismo nivel) y, por último del grado. En la red de mundo pequeño, por el contrario, la medida más importante es el grado en dos pasos, seguido del grado y del coeficiente de clustering. Como se puede observar en el gráfico situado a la izquierda de la Figura 5.3, en la mayoría de los casos, los productos comprados por los *hubs* de la red fueron los que lideraron el mercado en detrimento de los que fueron comprados por consumidores con menor grado en la red. En contraposición a la red libre de escala, los resultados sobre la red de mundo pequeño (segunda fila en la Tabla 5.2 e imagen derecha de la Figura 5.3), muestran estrategias que necesitan de un mayor número de *influencers* para generar campañas de VM efectivas: 99 individuos semilla contra 82 y resultando en un beneficio menor.

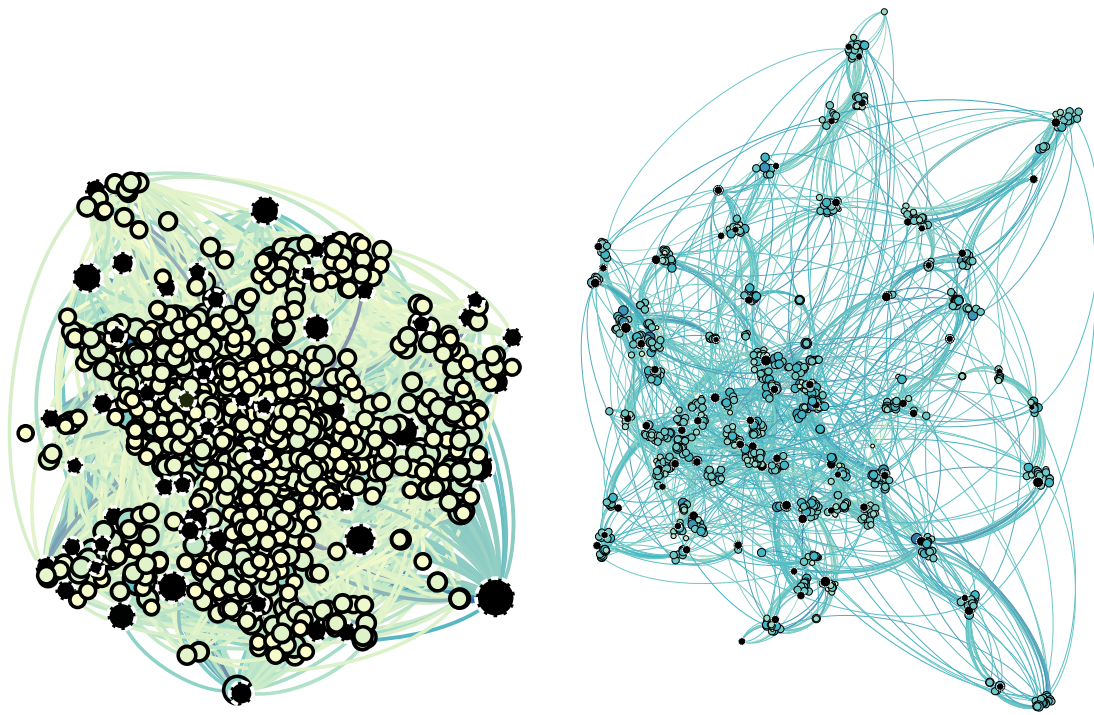


Figura 5.3: Resultado de la estrategia de VM propuesta por el GA en redes libre de escala (izquierda) y de mundo pequeño (derecha). Los *influencers* seleccionados aparecen delimitados con círculos negros con borde punteado blanco. Cada color representa un producto consumido.

A la vista de los resultados podemos concluir que el uso del GA monoobjetivo para la optimización de campañas de VM es efectivo ya que permite explorar diferentes estrategias en las que se aprovechan tanto la estructura como las características de las SNs y su influencia en la transmisión de información y obtener resultados beneficiosos.

## 5.5. Propuesta de un método automático de selección de *influencers* para campañas de VM basado en algoritmos evolutivos multiobjetivo

### 5.5.1. Diseño de los métodos de selección de *influencers* empleados

Aunque la optimización evolutiva multiobjetivo se ha utilizado ampliamente en el ámbito de las SNs, la mayoría de los esfuerzos se han enfocado en problemas relacionados, como la detección

de comunidades. Autores como Gunasekara et al. consideraron algoritmos EMO para identificar a los actores clave en grandes SN [GMM15]. Otros autores como [WMA13] estudiaron la mejora en la predicción de las dinámicas de difusión en SNs incorporando información acerca de la estructura de la red subyacente de forma que futuros productos involucrarían subconjuntos de la misma red. Por su parte, autores como [TRJ13] propusieron un enfoque descriptivo para identificar las condiciones sistemáticas que son estables a través de las dinámicas difusión y, por tanto, “transferibles” al lanzamiento de nuevos productos dentro de una red determinada utilizando datos de aplicaciones de Facebook.

En el momento en el que comenzamos a desarrollar nuestra propuesta no existía ninguna investigación basada en algoritmos EMOs previa que tratase objetivos conflictivos a la hora de identificar *influencers* en campañas de VM. Normalmente, existe una simplificación del problema inherente de dos objetivos (maximizar los ingresos y minimizar los costes) en un único objetivo (véase [SRW10]). En nuestro caso, abordamos el problema desde una perspectiva multiobjetivo utilizando algoritmos EMOs para tratar conjuntamente los dos objetivos clave de las campañas.

A continuación se introducen las decisiones de diseño comunes a los algoritmos EMO utilizados en nuestro método propuesto para el diseño de estrategias de VM. Las adaptaciones multiobjetivo del GA estacionario monoobjetivo (ver Sección 5.3) y del algoritmo voraz que utilizamos como métodos base contra los que comparar los resultados obtenidos por el resto de algoritmos EMO serán detallados en las Secciones 5.5.2 y 5.5.3. Las características generales de NSGA-II y MOEA/D fueron detalladas con anterioridad en las Secciones 3.3.2.1 y 3.3.2.2, respectivamente.

- **Representación:** Cada individuo de la población tiene cuatro genes que corresponden a las cuatro variables de decisión definidas en la Sección 5.3.1. Los tres primeros genes siguen una codificación real y representan los pesos de las medidas de análisis de redes sociales en la función heurística de selección de individuos semilla  $(\omega_d, \omega_{2s}, \omega_{cc})$ . El cuarto gen es un valor entero que indica el número de individuos semilla  $s$  a los que dirigir la campaña. La población inicial se genera al azar.
- **Selección:** MOEA/D y la adaptación multimodal del GA monoobjetivo siguen un enfoque de estado estacionario. MOEA/D utiliza un método de selección utilizando la función de

agregación de Tchebycheff y el vector de pesos introducidos en la Sección 3.3.2.2 para formar una población élite temporal de la que se escogen al azar dos soluciones a partir de las cuales se genera una nueva solución utilizando los operadores genéticos deseados. Por su parte, el GA multiobjetivo utiliza un método de selección de torneo binario en el que se escogen dos individuos de la población al azar y se comparan sus valores objetivo para quedarse con el mejor. Este procedimiento se replica dos veces hasta que se seleccionan dos individuos en cada generación y se les aplican los operadores genéticos. NSGA-II, por el contrario, sigue un enfoque generacional y utiliza un método de selección de individuos basado en no dominancia y distancias de apilamiento (ver Sección 3.3.2.1).

- **Operador de cruce:** Los tres algoritmos EMO utilizan un operador de cruce BLX- $\alpha$  para cruzar la descendencia con una probabilidad  $p_c \in [0, 1]$ . El funcionamiento de este operador puede consultarse en la Sección 5.3.5.
- **Operador de mutación** Se aplica a cada individuo con una probabilidad  $p_m \in [0, 1]$  para reajustar uno de sus genes al azar dentro del intervalo del gen específico (mutación aleatoria). Este operador de mutación es un reinicio con código real para los tres primeros genes y un reinicio con código entero para el cuarto gen.

Las variables de decisión consideradas en nuestra propuesta multiobjetivo son las mismas que utilizamos en la Sección 5.3.1. Al igual que en nuestra propuesta monoobjetivo, la selección de los  $s$  individuos que actuarán como *influencers* en las campañas de VM generadas por nuestro método hace uso de la Ecuación 5.1. Sin embargo, la evaluación de las soluciones considerará las Ecuaciones 5.2 y 5.3 de forma independiente.

### 5.5.2. Adaptación multiobjetivo de un algoritmo genético estacionario monoobjetivo

Implementamos un enfoque de objetivo único para resolver el problema de selección de *influencers* en VM multiobjetivo combinando linealmente los dos objetivos,  $f^0(x)$  y  $f^1(x)$  introducidos en la Sección 5.3.2, en un valor de *fitness* de objetivo único. Utilizamos el parámetro  $\rho \in [0, 1]$  para

ponderar ambos objetivos del problema por  $f(x) = \rho f^0(x) + (1 - \rho)f^1(x)$ .

Establecemos el siguiente procedimiento para poder generar una aproximación del frente de Pareto agregado del GA multiobjetivo con la que realizar una comparación justa con NSGA-II y MOEA/D:

1. Usamos 11 valores para el parámetro de combinación lineal  $\rho$ , generados entre 0 a 1 con un paso de 0.1 para poder cubrir un rango de soluciones diverso. Así, en las primeras ejecuciones del GA multiobjetivo ( $\rho = 0$ ) para medir la calidad de las estrategias de VM generadas se considera únicamente su coste ( $f^0$ ). Posteriormente, con cada nuevo valor de  $\rho$ , el algoritmo pondera menos el coste de la campaña para centrarse en el beneficio de la misma ( $f^1$ ), pasando por un estado de equidad entre coste y beneficio cuando  $\rho = 0.5$ .
2. Una vez ejecutado el algoritmo con distintos pesos de  $\rho$  para  $f^0$  y  $f^1$ , generamos el frente de Pareto agregado del GA multiobjetivo usando el conjunto no dominado de soluciones encontradas por el algoritmo en el paso anterior.

### 5.5.3. Adaptación multiobjetivo de un algoritmo voraz

El último algoritmo multiobjetivo utilizado en nuestra experimentación es un algoritmo voraz (*greedy*, en inglés). Para este enfoque *greedy* definimos cuatro combinaciones de pesos para las medidas de SNs *degree*, *2-steps* y *cc*. Las combinaciones de pesos utilizadas para la ordenación y selección de *influencers* en las campañas de VM optimizadas son las siguientes:

- [1, 0, 0]: Ordenación y selección en base a sus valores de medida de grado.
- [0, 1, 0]: Ordenación y selección en base a sus valores de medida de 2 pasos.
- [0, 0, 1]: Ordenación y selección en base a sus valores de coeficiente de agrupación.
- [0.33, 0.33, 0.33]: Ordenación y selección ponderando por igual su grado, 2 pasos y coeficiente de agrupación.

El algoritmo genera conjuntos de  $s$  *influencers* (desde 1 hasta un máximo  $S_{max}$ ), para cada una de las últimas combinaciones de pesos. Tras la generación de todas las soluciones, se construye

un frente de Pareto agregado con aquellas soluciones no dominadas de entre las generadas por el algoritmo para cada conjunto de pesos y número de influyentes objetivo  $s$ .

## 5.6. Experimentos y análisis de resultados sobre la propuesta multiobjetivo

En esta sección abordamos la fase experimental de nuestra propuesta multiobjetivo para el la selección de *influencers* en campañas de VM. Comenzamos desgranando el diseño experimental en la Sección 5.6.1. Después, introducimos la adaptación multiobjetivo del GA propuesto en la Sección 5.3 y de un algoritmo voraz en las Secciones 5.5.2 y 5.5.3. Por último, analizamos los resultados obtenidos en las Secciones 5.6.2 y 5.6.3.

### 5.6.1. Configuración experimental

Establecemos un factor de descuento del 10% ( $\lambda = 0.9$ ) como el utilizado en [SRW10] para las funciones objetivo (ver Ecuaciones 5.2 y 5.3). Todos los algoritmos EMO se ejecutan 30 veces con semillas aleatorias independientes. El tamaño de la población  $P$  es de 200 individuos. Las poblaciones de los algoritmos evolucionan durante 100 generaciones. Las probabilidades de cruce y mutación,  $p_c$  y  $p_m$ , son iguales a 0.6 y 0.2, respectivamente, y usamos  $\alpha = 0.5$  para el operador de cruce BLX- $\alpha$ . Se utiliza un vecindario de 40 individuos para MOEA/D debido al elevado tamaño de la población ( $P = 200$ ). El número máximo de agentes para seleccionar  $S$  se define como  $0.1 \cdot I$ . Por consiguiente, el número máximo de agentes que actúan como *influencers* en una simulación será, como máximo, un 40% del total de agentes consumidores  $I$ .

Los algoritmos EMO (esto es, NSGA-II, MOEA/D y las adaptaciones multiobjetivo del GA monoobjetivo y del algoritmo *greedy*) utilizan el modelo *CONSUMAT* extendido propuesto en el Capítulo 4 para evaluar los ingresos y los costes de las campañas de VM a optimizar. Definimos un total de  $J = 10$  productos disponibles en el mercado, incluyendo el producto  $k$  usado en la campaña VM. Las simulaciones se ejecutan durante  $T = 365$  pasos (donde cada paso representa

un día) y el número total de agentes  $I$  se establece en función del número de nodos de cada una de las cinco SNs utilizadas en la experimentación.

La Tabla 5.3 muestra las características de las cinco topologías de SNs y de la red construida a partir de datos de una aplicación *freemium*. La primera topología de SN es una red de correo electrónico real (*email*, en inglés) de la Universidad Rovira i Virgili en España [Gui+03]. El resto de las SNs se generaron utilizando características similares a las de la red de *emails* (un grado medio de 10 y una densidad de 0.01). Para la red libre de escala se utilizó el algoritmo BA [BA99] con  $m = 2$ . Para la red aleatoria, conectamos cada nodo al azar con otros nodos de la red manteniendo un grado medio de 10 ( $p = 0.01$ ). Para la red de mundo pequeño, utilizamos 0.2 como probabilidad de reconexión en el algoritmo de generación de Watts-Strogatz [WS98]. Utilizamos cuatro enlaces por nodo (esto es, vecindario *von Neumann*) con condiciones de contorno periódicas para la red de malla regular.

SN	$I$	Enlaces	Long. camino medio	Grado medio	Coef. clust.	Densidad
<i>Email</i>	1,133	5,451	3.60	9.622	0.254	0.009
Libre de escala	1,000	5,007	3.04	10.014	0.056	0.01
Aleatoria	1,000	4,980	3.27	9.852	0.008	0.01
Mundo pequeño	1,000	5,000	4.43	10	0.495	0.01
Regular	1,024	2,048	16.02	4	0	0.04

Tabla 5.3: Principales características de las topologías de SNs utilizadas.

El parámetro de influencia social  $\beta$  para la simulación del ABM se fija en 0.5. Aplicamos un valor de 0.5 a los umbrales  $U$  y  $\Phi$  que definen la incertidumbre y la utilidad mínima de los agentes en el mercado. Los valores de la probabilidad de compra y de difusión de los agentes son  $p_b = 0.8$  y  $p_\theta = 0.5$ , respectivamente. La desactivación del conocimiento  $\delta$  tiene un valor de 0.2. El conocimiento de productos en los agentes  $\theta_{ij}$  se inicializa aleatoriamente al principio de la simulación. Por último, los valores de los objetivos se promedian para 30 ejecuciones independientes de Monte Carlo de la simulación del ABM desarrollada para la evaluación de cada individuo.

Consideramos dos indicadores de rendimiento multiobjetivo unarios: cardinalidad y ratio de



hivervolumen ( $HVR$ ) [CLVV+07] (ver descripciones en Sección 3.3.3.1) y dos binarios: el indicador  $I_\epsilon$  y el indicador de cobertura del conjunto ( $C$ ) (ver descripciones en Sección 3.3.3.2) para evaluar el rendimiento de los algoritmos.

### 5.6.2. Análisis de resultados

En esta sección mostramos los resultados obtenidos por NSGA-II, MOEA/D, la adaptación multiobjetivo del GA monoobjetivo y el algoritmo *greedy* para las cinco topologías de SNs consideradas en nuestra experimentación: *email*, libre de escala, aleatoria, mundo pequeño y regular. En primer lugar, discutimos los valores obtenidos por los algoritmos EMO considerados para cada uno de los indicadores de rendimiento multiobjetivo seleccionados (esto es,  $HVR$ , cobertura del conjunto  $C$  e indicador  $I_\epsilon$ ). A continuación, mostramos cómo cada algoritmo cubre el espacio de búsqueda y su contribución al mejor conjunto de soluciones encontradas para cada escenario, analizando sus aproximaciones al frente de Pareto. Discutiremos los resultados a partir de aquellas topologías de red con mayor velocidad de difusión.

Por un lado, la Tabla 5.4 muestra los valores de los indicadores de rendimiento  $HVR$  y cardinalidad. Por otro lado, la Tabla 5.5 muestra los valores del indicador  $C$ . Por último, los valores para el indicador de rendimiento  $I_\epsilon$  se muestran en la Tabla 5.6. En todas las tablas, mostramos los valores de la media y la desviación estándar (entre paréntesis) para las 30 ejecuciones de los algoritmos. Los mejores valores para cada indicador se muestran en negrita.

SN	HVR				Cardinalidad			
	MOEA/D	NSGA-II	GA	Greedy	MOEA/D	NSGA-II	GA	Greedy
<i>Email</i>	<b>0.96</b> (0.017)	0.947 (0.007)	0.913 (0.01)	0.864 (0.008)	41.33 (3.32)	<b>65.47</b> (4.88)	10 (0.83)	32 (3.1)
Libre de escala	<b>0.957</b> (0.017)	0.938 (0.007)	0.928 (0.01)	0.862 (0.003)	35 (3.05)	<b>59.86</b> (6.21)	9.03 (1.03)	32 (2.54)
Aleatoria	<b>0.943</b> <b>0.023</b>	0.94 (0.009)	0.937 (0.01)	0.857 (0.02)	41 (3.85)	<b>65.5</b> (6.25)	9.56 (0.86)	29 (5.8)
Mundo pequeño	0.932 (0.028)	<b>0.934</b> (0.01)	0.878 (0.01)	0.789 (0.023)	39 (3.76)	<b>63.6</b> (5.35)	8.73 (1.01)	21 (2.7)
Regular	0.865 (0.045)	<b>0.897</b> (0.022)	0.687 (0.037)	0.653 (0.034)	28 (3.68)	<b>35</b> (3.74)	6.1 (0.8)	15 (4.21)

Tabla 5.4: Valores medios y desviación estándar de los indicadores HVR y cardinalidad (los mejores resultados se muestran en negrita).

SN	$C(\text{MOEA/D, NSGA-II}) / C(\text{NSGA-II, MOEA/D})$	$C(\text{MOEA/D, GA}) / C(\text{GA, MOEA/D})$	$C(\text{MOEA/D, Greedy}) / C(\text{Greedy, MOEA/D})$	$C(\text{NSGA-II, GA}) / C(\text{GA, NSGA-II})$	$C(\text{NSGA-II, Greedy}) / C(\text{Greedy, NSGA-II})$	$C(\text{GA, Greedy}) / C(\text{Greedy, GA})$
<i>Email</i>	<b>0.461</b> (0.019) / 0.358 (0.0246)	<b>0.604</b> (0.0152) / 0.068 (0.05)	<b>1.0</b> (0.0) / 0.0 (0.0)	<b>0.526</b> (0.022) / 0.082 (0.008)	<b>1.0</b> (0.0) / 0.0 (0.0)	<b>1.0</b> (0.0) / 0.0 (0.0)
Libre de escala	<b>0.4803</b> (0.0218) / 0.333 (0.0212)	<b>0.439</b> (0.0252) / 0.09 (0.075)	<b>1.0</b> (0.0) / 0.0 (0.0)	<b>0.421</b> (0.0142) / 0.145 (0.0076)	<b>1.0</b> (0.0) / 0.0 (0.0)	<b>1.0</b> (0.0) / 0.0 (0.0)
Aleatoria	<b>0.4529</b> (0.026) / 0.4072 (0.0277)	<b>0.448</b> (0.0121) / 0.139 (0.065)	<b>1.0</b> (0.0) / 0.0 (0.0)	<b>0.385</b> (0.0201) / 0.14 (0.093)	<b>1.0</b> (0.0) / 0.0 (0.0)	<b>1.0</b> (0.0) / 0.0 (0.0)
Mundo pequeño	0.3261 (0.0238) / <b>0.5537</b> (0.0263)	<b>0.3</b> (0.0151) / 0.249 (0.0136)	<b>1.0</b> (0.0) / 0.0 (0.0)	<b>0.414</b> (0.0116) / 0.175 (0.059)	<b>1.0</b> (0.0) / 0.0 (0.0)	<b>1.0</b> (0.0) / 0.0 (0.0)
Regular	0.2289 (0.0164) / <b>0.715</b> (0.0178)	<b>0.647</b> (0.0235) / 0.113 (0.0132)	<b>1.0</b> (0.0) / 0.0 (0.0)	<b>0.733</b> (0.0164) / 0.045 (0.013)	<b>1.0</b> (0.0) / 0.0 (0.0)	<b>1.0</b> (0.0) / 0.0 (0.0)

Tabla 5.5: Valores medios y desviación estándar del indicador de cobertura  $C$  (los mejores resultados se muestran en negrita).

SN	$I_c(\text{MOEA/D, NSGA-II}) / I_c(\text{NSGA-II, MOEA/D})$	$I_c(\text{MOEA/D, GA}) / I_c(\text{GA, MOEA/D})$	$I_c(\text{MOEA/D, Greedy}) / I_c(\text{Greedy, MOEA/D})$	$I_c(\text{NSGA-II, GA}) / I_c(\text{GA, NSGA-II})$	$I_c(\text{NSGA-II, Greedy}) / I_c(\text{Greedy, NSGA-II})$	$I_c(\text{GA, Greedy}) / I_c(\text{Greedy, GA})$
<i>Email</i>	<b>1.023</b> (0.009) / 1.04 (0.018)	<b>1.014</b> (0.007) / 1.092 (0.023)	<b>0.928</b> (0.001) / 1.22 (0.048)	<b>1.016</b> (0.009) / 1.093 (0.023)	<b>0.953</b> (0.002) / 1.21 (0.02)	<b>0.988</b> (0.002) / 1.12 (0.037)
Libre de escala	<b>1.022</b> (0.006) / 1.042 (0.017)	<b>1.019</b> (0.009) / 1.086 (0.023)	<b>0.91</b> (0.001) / 1.33 (0.028)	<b>1.025</b> (0.009) / 1.09 (0.022)	<b>0.952</b> (0.025) / 1.171 (0.035)	<b>0.973</b> (0.032) / 1.22 (0.04)
Aleatoria	<b>1.03</b> (0.018) / 1.034 (0.018)	<b>1.024</b> (0.013) / 1.108 (0.031)	<b>0.921</b> (0.004) / 1.22 (0.03)	<b>1.023</b> (0.008) / 1.103 (0.024)	<b>0.946</b> (0.002) / 1.19 (0.018)	<b>0.983</b> (0.005) / 1.12 (0.02)
Mundo pequeño	1.04 (0.019) / <b>1.031</b> (0.02)	<b>1.028</b> (0.014) / 1.094 (0.031)	<b>0.91</b> (0.001) / 1.325 (0.032)	<b>1.022</b> (0.008) / 1.09 (0.022)	<b>0.938</b> (0.003) / 1.268 (0.003)	<b>0.964</b> (0.012) / 1.25 (0.012)
Regular	1.083 (0.026) / <b>1.03</b> (0.027)	<b>1.023</b> (0.024) / 1.205 (0.059)	<b>0.94</b> (0.002) / 1.31 (0.03)	<b>1.01</b> (0.011) / 1.205 (0.045)	<b>0.928</b> (0.001) / 1.703 (0.002)	<b>0.993</b> (0.002) / 1.19 (0.023)

Tabla 5.6: Valores de la media y la desviación estándar de  $I_c$  multiplicativo (los mejores resultados se muestran en negrita).

En lo que respecta al indicador  $HVR$ , MOEA/D obtiene los valores más altos en tres de las cinco topologías de SNs. Por el contrario, NSGA-II muestra los mejores resultados en las otras dos topologías. El GA multiobjetivo muestra resultados pobres en comparación con NSGA-II y MOEA/D. Por último, el algoritmo *greedy* es el peor algoritmo en todos los escenarios y está muy lejos de los resultados del resto de algoritmos EMO.

Los resultados muestran que las diferencias entre MOEA/D, NSGA-II y el GA multiobjetivo son claras en función de las topologías de red con diferentes velocidades de difusión. MOEA/D rinde entre un 1.5% y un 2% mejor que los otros dos algoritmos en las topologías de red con mayor velocidad de difusión de la información (*email* y libre de escala). Mientras tanto, el GA multiobjetivo obtiene los peores valores para esas redes. En la red aleatoria, donde la difusión de la información es más lenta que en la red libre de escala, MOEA/D sigue siendo el mejor algoritmo, pero con un margen muy estrecho respecto a NSGA-II (en torno al 0.03%) y al GA (0.8%). Finalmente, NSGA-II obtiene los mejores valores de  $HVR$  para las redes de mundo pequeño y regular, las que presentan una menor velocidad de difusión.

NSGA-II es el algoritmo EMO que aporta el mayor número de soluciones no dominadas a las aproximaciones de los frentes de Pareto (es decir, indicador de cardinalidad). MOEA/D es el segundo en valores de cardinalidad, aportando alrededor de la mitad de las soluciones obtenidas por NSGA-II salvo en el escenario de la red regular (en este caso, se obtienen valores de cardinalidad similares). El enfoque *greedy* obtiene un buen número de soluciones no dominadas en sus aproximaciones al frente de Pareto, pero se aleja de los valores de cardinalidad de NSGA-II. El GA multiobjetivo tiene el valor de cardinalidad más bajo, ya que obtiene menos de diez soluciones no dominadas en cada escenario.

Los valores del indicador  $C$  de la Tabla 5.5 reflejan las mismas conclusiones observadas en la Tabla 5.4. MOEA/D obtiene los mejores valores de  $C$  en las redes *email*, libre de escala y aleatoria. Por el contrario, las soluciones de MOEA/D son cubiertas principalmente por las soluciones de NSGA-II para las redes de mundo pequeño y regular, los dos escenarios con menor velocidad de difusión de información. El GA multiobjetivo muestra un valor de cobertura bajo cuando se compara con los métodos multiobjetivo puros. Sólo cuando se considera la red SW,

el GA multiobjetivo cubre alrededor del 20% de las soluciones no dominadas encontradas por MOEA/D y NSGA-II. El algoritmo *greedy* es el de peor rendimiento en todos los escenarios ya que sus soluciones están cubiertas por las soluciones de los demás algoritmos EMO en todos los escenarios.

Los valores de  $I_\epsilon$  se muestran en la Tabla 5.6. De nuevo, los algoritmos siguen un comportamiento similar al mostrado con los indicadores de las Tablas 5.4 y 5.5. Las aproximaciones del frente de Pareto de MOEA/D necesitan multiplicarse por valores inferiores (siendo 1.03 el más alto) para alcanzar el conjunto de soluciones de NSGA-II para los tres primeros escenarios (*email*, libre de escala y aleatoria). Por otro lado, el NSGA-II necesita un valor mayor de  $\epsilon$  para aproximarse a los frentes de Pareto del MOEA/D. En el caso de las redes de mundo pequeño y regular observamos el comportamiento contrario, dado que NSGA-II presenta valores de  $I_\epsilon$  más bajos. El GA multiobjetivo presenta valores de  $I_\epsilon$  elevados, pero está lejos del algoritmo *greedy*, que vuelve a ser el de peor rendimiento.

Ahora centramos nuestro análisis en las superficies de éxito (*attainment surfaces*) de las aproximaciones del frente de Pareto obtenidas por los algoritmos. Las Figuras 5.4, 5.5 y 5.6 muestran las aproximaciones agregadas del frente de Pareto para tres de las cinco topologías de SNs (es decir, *email*, libre de escala y regular). En las superficies de éxito, cada algoritmo se representa con un color y una forma en las figuras: Cuadrados verdes para las soluciones de MOEA/D, triángulos azules para las soluciones de NSGA-II, cruces rojas para las soluciones del GA multiobjetivo y diamantes naranjas para las soluciones del algoritmo *greedy*. Las soluciones del frente de Pareto pseudo-óptimo, construidas a partir de todas las ejecuciones, se representan con círculos negros. Obsérvese que en el análisis de los indicadores anteriores se utilizaron las aproximaciones independientes del frente de Pareto obtenidas por los algoritmos en las 30 ejecuciones. Sin embargo, estas superficies de éxito muestran los frentes de Pareto agregados de los diferentes algoritmos en las 30 ejecuciones para una mejor visualización.

MOEA/D proporcionó las mejores soluciones para la red *email* (ver Figura 5.4). Sólo una solución de NSGA-II domina a las de MOEA/D en la parte superior del frente de Pareto pseudo-óptimo. La aproximación del frente de Pareto agregada del GA multiobjetivo sólo aporta dos soluciones

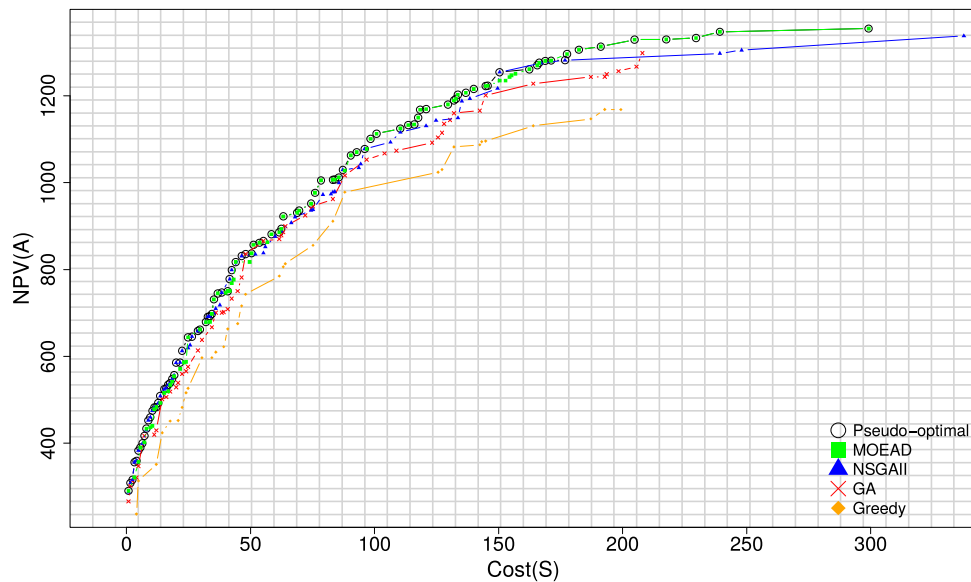


Figura 5.4: Aproximaciones del frente de Pareto obtenidas para la red de *emails*. En el gráfico vemos cómo el frente de Pareto agregado obtenido por MOEA/D domina claramente a los obtenidos por el resto de algoritmos EMO.

al frente de Pareto pseudo-óptimo (ambas en su parte inferior). El resto de soluciones obtenidas por el GA multiobjetivo están dominadas por las soluciones de MOEA/D y NSGA-II. Todas las soluciones del algoritmo *greedy* están dominadas por los algoritmos MOEA/D, NSGA-II y GA multiobjetivo.

MOEA/D vuelve a ser el mejor algoritmo para la red libre de escala (Figura 5.5). 41 de las 49 soluciones del frente de Pareto pseudo-óptimo provienen de las ejecuciones del MOEA/D. Sólo unas pocas soluciones obtenidas por el NSGA-II dominan a las del MOEA/D y son aquellas con ingresos en  $[1, 500]$  y costes en  $[1, 20]$ , situadas en la parte inferior izquierda del frente de Pareto. La mayoría de las soluciones del GA multiobjetivo están dominadas por NSGA-II y MOEA/D. Vemos claramente que MOEA/D consigue mejores resultados y proporciona un mayor número de soluciones al frente de Pareto pseudo-óptimo. El rendimiento de MOEA/D en estos dos primeros escenarios muestra su potencial cuando se tienen topologías de red con una distribución de grados que sigue la ley de potencias, es decir, que tienen *hubs* que aumentan la velocidad de difusión de la información.

La Figura 5.6 muestra las superficies de éxito de las campañas VM cuando se utiliza la red regular, la de menor velocidad de difusión de la información. En este escenario, podemos ver

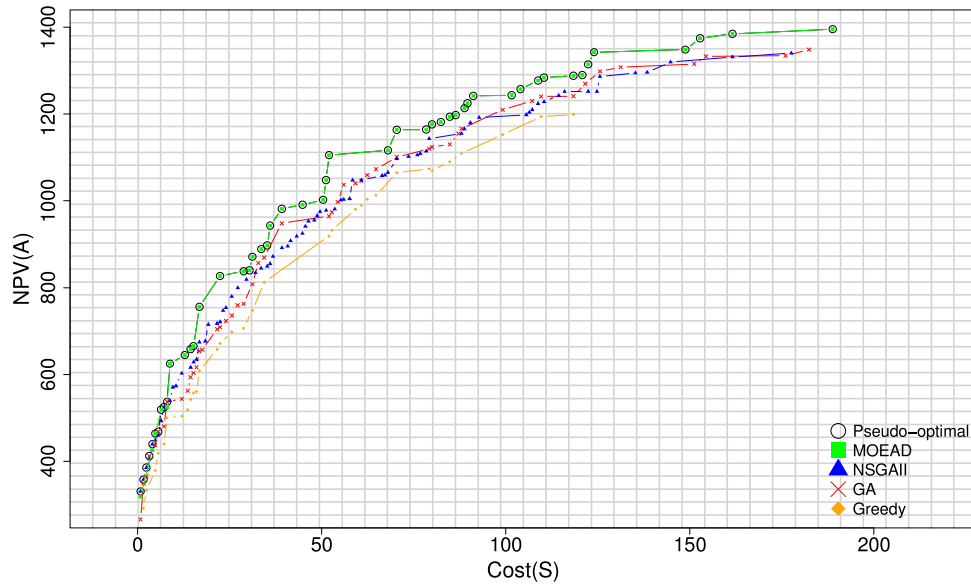


Figura 5.5: Aproximaciones del frente de Pareto obtenidas para la red SF. De nuevo, el frente de Pareto agregado obtenido por MOEA/D domina a los obtenidos por el resto de algoritmos EMO.

claramente que NSGA-II supera a MOEA/D generando un número significativamente mayor de soluciones de frentes de Pareto pseudo-óptimos. Los frentes obtenidos por los dos algoritmos EMO dominan ampliamente la aproximación del frente de Pareto de la adaptación multiobjetivo del GA monoobjetivo. Sólo dos soluciones del GA multiobjetivo forman el frente de Pareto pseudo-óptimo en su parte central. El algoritmo *greedy* no tiene soluciones en el frente de Pareto pseudo-óptimo. Su aproximación del frente de Pareto está totalmente dominado por los obtenidos por los otros algoritmos.

Por último, ampliamos el conjunto de SNs de la experimentación añadiendo un caso de estudio real. La red utilizada en este caso de estudio está generada a partir de los datos de una aplicación *freemium* y presenta una distribución de grado fuertemente bimodal [CR17]. Esto significa que hay un grupo numeroso de usuarios con una o dos conexiones y otro grupo, también numeroso, con un elevado número de conexiones (alrededor de 100). La SN está formada por 20,000 nodos y 481,907 enlaces, la longitud del camino medio es 3.02 y su grado medio, coeficiente de clustering y densidad son 48.19, 0.006, 0.002, respectivamente. Este caso de estudio nos permite demostrar que los algoritmos EMO pueden escalarse a grandes SNs reales y comprobar si se comportan de forma similar que con las cinco SNs anteriormente analizadas.

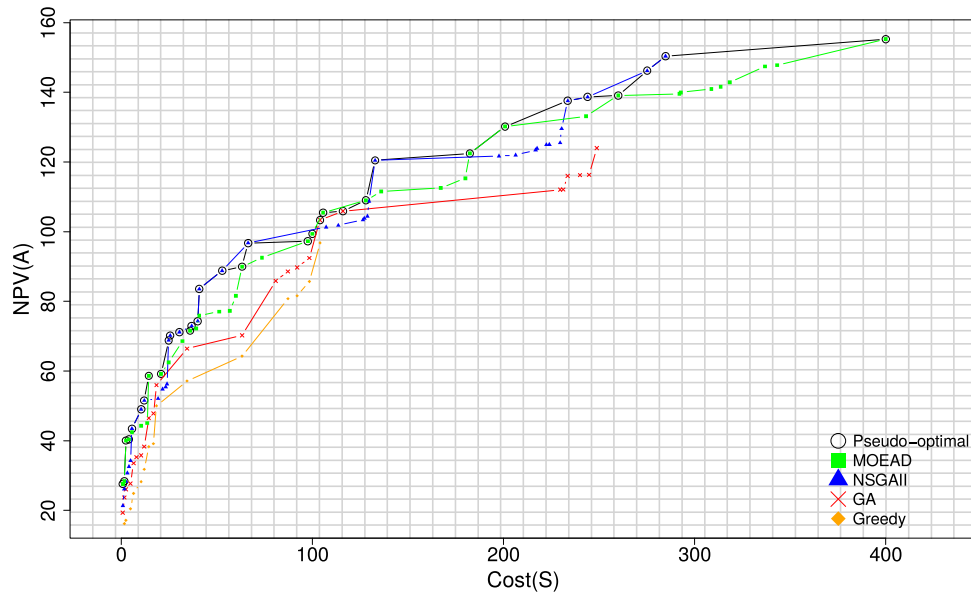


Figura 5.6: Aproximaciones del frente de Pareto obtenidas para la red regular. En este caso, vemos cómo es el frente de Pareto agregado obtenido por NSGA-II el que domina, en la mayoría de los casos, a los obtenidos por MOEA/D y las adaptaciones multiobjetivo del GA y del algoritmo *greedy*.

Utilizamos los mismos indicadores para medir el rendimiento de los algoritmos (Tabla 5.7) y mostramos los frentes de Pareto con el conjunto de soluciones no dominadas encontradas por los algoritmos EMO de MOEA/D, NSGA-II y GA multiobjetivo (Figura 5.7). En este caso no hacemos uso del algoritmo *greedy* ya que lo descartamos a la vista del bajo rendimiento que mostró en la Sección 5.6.2

		MOEA/D	NSGA-II	GA
HVR		<b>0.999 (0.0008)</b>	0.997 (0.0002)	0.94 (0.0031)
Cardinality		22.6 (3.51)	<b>34.2 (5.18)</b>	12.5 (1.2)
$I_c$	MOEA/D	-	<b>1.002 (0.002)</b>	<b>1.005 (0.001)</b>
	NSGA-II	1.008 (0.001)	-	<b>1.003 (0.002)</b>
	GA	43.08 (0)	42.7 (0.02)	-
$C$	MOEA/D	-	<b>0.742 (0.121)</b>	<b>1.0 (0)</b>
	NSGA-II	0 (0.001)	-	<b>1.0 (0.001)</b>
	GA	0 (0)	0.05 (0.001)	-

Tabla 5.7: Valores medios y desviación estándar de los indicadores de rendimiento multiobjetivo para el caso real.

Teniendo en cuenta los resultados de la Tabla 5.7 podemos observar la superioridad de MOEA/D sobre NSGA-II y la adaptación multiobjetivo del GA monoobjetivo. MOEA/D obtiene los mejores valores para todos los indicadores de rendimiento multiobjetivo, excepto para la cardinalidad,

donde NSGA-II es el algoritmo con mejor rendimiento. En esta red, MOEA/D tiene valores de  $I_\epsilon$  y  $C$  muy superiores a los de NSGA-II y el GA multiobjetivo. Vemos, por ejemplo, que MOEA/D cubre alrededor del 75% de las soluciones obtenidas por NSGA-II. La Figura 5.7 muestra los frentes de Pareto de los algoritmos EMO donde vemos que MOEA/D aporta el mayor número de soluciones al frente de Pareto pseudo-óptimo. Sólo tres soluciones del NSGA-II pertenecen al frente de Pareto pseudo-óptimo, pero estas soluciones no dominan ninguna de las soluciones del frente de Pareto de MOEA/D. El GA multiobjetivo muestra resultados bajos y su rendimiento está muy lejos del MOEA/D y del NSGA-II.

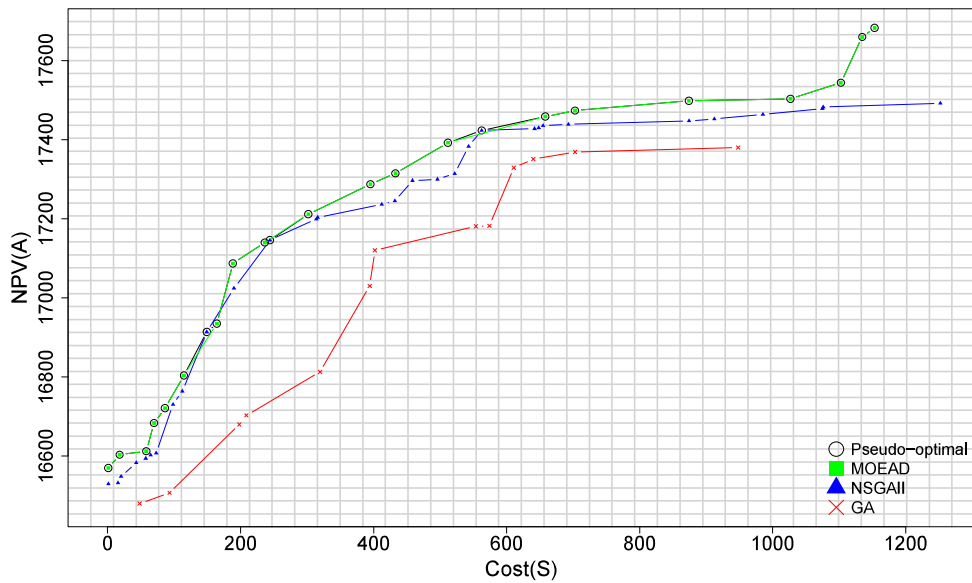


Figura 5.7: Aproximaciones del frente de Pareto obtenidas para la red APP. Como ocurría en el caso de los resultados obtenidos para las redes libre de escala y de mundos pequeños, el frente de Pareto agregado obtenido por MOEA/D domina a los frentes obtenidos por el resto de algoritmos EMO en la mayor parte de los casos.

### 5.6.3. Análisis de las soluciones obtenidas

Concluimos este capítulo analizando y validando las soluciones a las campañas de VM obtenidas por los algoritmos EMO desde la perspectiva del experto en marketing. Para ello, evaluaremos la calidad de las campañas de marketing diseñadas por los tres algoritmos evolutivos con mejores resultados. Para facilitar este análisis, sólo consideramos los escenarios sobre las redes de *email* y regular puesto que representan las topologías de SN más diferentes entre sí.

La Tabla 5.8 muestra la composición de tres soluciones diferentes seleccionadas de la aproximación



del frente de Pareto generada por cada algoritmo en cada escenario. De estas soluciones, dos presentan el mejor valor para cada objetivo individual (ingresos y coste de la campaña de VM). La tercera se corresponde con un trade-off entre estos dos objetivos. La solución trade-off (en la que se busca un balance entre los dos objetivos optimizados) se elige como sigue: Para cada solución del conjunto aproximado de Pareto obtenido, lanzamos 1,000 valores aleatorios para un peso  $\rho \in [0, 1]$  y calculamos el valor de la combinación ponderada de los dos objetivos (ingresos y costes de la campaña) como  $f(x) = \rho \cdot NPV(A) - (1 - \rho) \cdot Cost(S)$  para cada peso. Después, sumamos los 1,000 valores obtenidos para la función  $f_\rho(x)$  obteniendo el valor agregado  $f(x)$  y seleccionamos la solución con el mayor valor agregado  $f(x)$ . En lo que respecta a los valores de los genotipos de los individuos, los tres primeros valores corresponden a los pesos  $\omega$  asociados a las medidas de análisis de la SN local mientras que el cuarto es el número de individuos semilla seleccionados en la campaña,  $s$ .

SN	MOEA/D			NSGA-II			GA			Greedy		
	$W_d$	$W_{2s}$	$W_{cc}$	$F_S$	$W_d$	$W_{2s}$	$W_{cc}$	$F_S$	$W_d$	$W_{2s}$	$W_{cc}$	$F_S$
<i>email</i>	Highest revenue solution			Highest revenue solution			Highest revenue solution			Highest revenue solution		
	$(f^0(x)=1398.17, f^1(x)=191.25)$			$(f^0(x)=1350.2, f^1(x)=174.38)$			$(f^0(x)=1354.2, f^1(x)=182.25)$			$(f^0(x)=1180, f^1(x)=201.3)$		
	0.58	0.9	0.23	170	0.48	0.98	0.29	155	0.34	0.78	0.22	162
	Best trade-off solution			Best trade-off solution			Best trade-off solution			Best trade-off solution		
	$(f^0(x)=972.12, f^1(x)=76.3)$			$(f^0(x)=958.4, f^1(x)=79.88)$			$(f^0(x)=950.8, f^1(x)=84.6)$			$(f^0(x)=912.7, f^1(x)=92.2)$		
	0.7	0.84	0.27	69	0.64	0.96	0.32	71	0.58	0.75	0.27	75
Regular	Lowest cost solution			Lowest cost solution			Lowest cost solution			Lowest cost solution		
	$(f^0(x)=204.92, f^1(x)=15.75)$			$(f^0(x)=238.8, f^1(x)=22.5)$			$(f^0(x)=70.03, f^1(x)=14.625)$			$(f^0(x)=52.12, f^1(x)=26.8)$		
	0.91	0.87	0.4	14	0.87	0.85	0.49	20	0.86	0.72	0.37	13
	Highest revenue solution			Highest revenue solution			Highest revenue solution			Highest revenue solution		
	$(f^0(x)=155.4, f^1(x)=398.25)$			$(f^0(x)=148.1, f^1(x)=282.38)$			$(f^0(x)=123.43, f^1(x)=148.5)$			$(f^0(x)=120.5, f^1(x)=152.3)$		
	0.31	0.52	0.47	354	0.72	0.85	0.68	251	0.5	0.64	0.53	132
Best trade-off solution			Best trade-off solution			Best trade-off solution			Best trade-off solution			
$(f^0(x)=112.4, f^1(x)=132.4)$			$(f^0(x)=122.38, f^1(x)=125.03)$			$(f^0(x)=108.6, f^1(x)=111.58)$			$(f^0(x)=89.31, f^1(x)=120.45)$			
0.34	0.56	0.32	117	0.47	0.63	0.025	111	0.74	0.58	0.49	99	
Lowest cost solution			Lowest cost solution			Lowest cost solution			Lowest cost solution			
$(f^0(x)=27.13, f^1(x)=1.13)$			$(f^0(x)=22.24, f^1(x)=1.13)$			$(f^0(x)=15.41, f^1(x)=1.13)$			$(f^0(x)=10.58, f^1(x)=1.13)$			
0.27	0.62	0.27	1	0.41	0.26	0.23	1	0.82	0.47	0.53	1	

Tabla 5.8: Valores del espacio de decisión obtenidos por las soluciones de MOEA/D, NSGA-II y GA en las SN *email* y regular.

Podemos encontrar diferencias interesantes en la composición de las soluciones en la Tabla 5.8. En primer lugar, podemos observar que los algoritmos devuelven soluciones con diferentes pesos para las métricas de la SN al definir la estrategia de selección de los *influencers* (véase la composición de los individuos con mayores ingresos encontrados por MOEA/D y NSGA-II). Estos pesos se adaptan a las características de la topología de la SN para conseguir una mayor difusión en la campaña. Por ejemplo, en la red *email*, vemos que los pesos de las medidas relacionadas con el grado de los nodos tienen un valor más alto para utilizar los *hubs* de la red como individuos semilla. También podemos observar que hay diferencias entre los individuos en función de los objetivos. Como era de esperar, las campañas representadas en las soluciones tienen un mayor o menor número de *influencers* en función de los ingresos y el coste de la campaña, así como en función de la topología de la red (por ejemplo, se requiere un número significativamente mayor para obtener buenos ingresos en la red regular debido a su menor velocidad de difusión de la información).

También podemos comprobar como en la red *email* las soluciones tienen valores de pesos más altos, tanto para las soluciones de altos ingresos como para las de bajo coste. Observamos dos diferencias principales al comparar las soluciones de mayores ingresos y las de menores costes para la red *email*. La primera diferencia está relacionada con el número de consumidores seleccionados como *influencers*, que es mayor para las soluciones con mayores ingresos, ya que así se permite difundir rápidamente el producto a través de la SN. La segunda es que para la solución con mayores ingresos el grado de los contactos alcanzables a través de dos pasos ( $\omega_{2s}$ ) se pondera después del grado ( $\omega_d$ ) y del coeficiente de clustering. Esto es una consecuencia directa de la distribución de grados siguiendo la ley de potencia de esta topología de red: Al utilizar un bajo número de individuos semilla con valores altos de grado y de  $\omega_{2s}$ , la campaña VM se dirige a aquellos consumidores que son *hubs* en la SN (es decir, aquellos que pueden difundir rápidamente información positiva sobre el producto).

La composición de las soluciones para la red regular presenta menos variabilidad que la de la red *email*. Las soluciones para regular muestran valores similares de  $\omega_d$ ,  $\omega_{2s}$  y  $\omega_{cc}$ , y valores altos de  $s$ , tanto para las soluciones de mayores ingresos como para las de menores costes. En este caso, la principal diferencia entre estas soluciones reside en el número de consumidores objetivo  $s$  y su

distribución en la SN. Las características de la red regular hacen que los algoritmos encuentren la mejor distribución de los *influencers* a través de la topología de la red, seleccionando también un número elevado de semillas para obtener soluciones con altos ingresos. Por supuesto, hay un bajo número de individuos semilla cuando se tiene una solución de campaña de bajo coste. De hecho, es el menor número posible (esto es, 1) como consecuencia de la lenta difusión de información asociada a esta topología de la red.

# Optimización multimodal para calibración de modelos basados en agentes

Estudia el pasado si quieres pronosticar el futuro.

---

Confucio

## 6.1. Introducción

Este capítulo abarca la propuesta de un método para la calibración de ABMs avanzado que permite obtener soluciones óptimas y diversas basado en MMEAs [Das+11; Pre15; Won15] para la calibración de ABMs. A diferencia del Capítulo 5, en el que propusimos un método avanzado para la generación de campañas de VM en ABMs basado en algoritmos EMO, en el presente capítulo hacemos uso de MMEAs para calibrar o ajustar los parámetros de dichos ABMs. Por tanto, cambiamos el foco en el uso de las técnicas de optimización y del problema, pasando de aplicarlas para explotar un modelo ABM previamente calibrado y validado para aplicarlas para mejorar la calibración de los modelos.

Como ya desarrollamos en la Sección 3.1 del Capítulo 3, la modelización basada en agentes suele implicar una etapa de calibración en la que es necesario estimar un conjunto de parámetros. Este proceso de calibración puede realizarse de forma automática mediante algoritmos de optimización que buscan una configuración óptima de los parámetros para obtener ajustes de calidad del

modelo. La calibración automática considera una medida de error para comparar los datos del mundo real con la salida del modelo simulado y ajusta un conjunto de parámetros del modelo, seleccionado previamente, para que coincida con los datos [Oli03; Sar10]. El proceso de calibración automática de ABMs suele hacer uso de un método de optimización que modifica los parámetros del modelo de forma sistemática minimizando la medida de error.

Un modelo correctamente calibrado puede ser utilizado como apoyo a la toma de decisiones para evaluar políticas en escenarios hipotéticos y comprender cómo funciona un determinado sistema [CR17]. No obstante, la construcción de ABMs es una tarea compleja que implica establecer los valores de un gran número de parámetros que suelen ser desconocidos. Además, una vez ajustados estos parámetros, los modeladores están obligados a estudiarlos cuidadosamente y a analizar las salidas del modelo para garantizar una buena correspondencia con el funcionamiento del sistema real [Sal+10]. Los modelos de simulación no lineales no pueden entenderse sin explorar su comportamiento bajo diferentes ajustes de parámetros [Lee+15] por lo que es necesario abordar una serie de problemas durante la validación y calibración de ABMs [Muñ+15; Fag+19]. Uno de los problemas más críticos implica la obtención de la mejor configuración de parámetros de entre las varias soluciones sub-óptimas disponibles en el espacio de búsqueda del problema, que suele ser multimodal [GR+87], ya que generalmente exige un proceso computacionalmente costoso que hace difícil encontrar una solución óptima en un período de tiempo factible. Otro problema importante, esta vez desde el punto de vista de la optimización, es la exploración de un espacio de búsqueda multimodal que implica la presencia de distintas soluciones subóptimas de igual calidad y que dificulta encontrar una solución única para los parámetros. Esto también se conoce como “identificabilidad del sistema” [BBC08]. Cuando el problema de optimización multimodal se aborda con algoritmos de optimización tradicionales, el procedimiento de búsqueda podría estancarse en óptimos locales sin devolver el óptimo global o, en su caso, el conjunto de óptimos globales.

Los problemas anteriormente mencionados hacen que el uso de métodos de optimización multimodal sea interesante para abordar la calibración de ABMs ya que pueden evitar el estancamiento en la búsqueda y proporcionar un conjunto de soluciones óptimas, tanto en términos de diversidad como de calidad de ajuste de igual preferencia para los modeladores,

todo ello en una sola ejecución y en un tiempo razonable. El uso de estos algoritmos permite a los modeladores la validación de los modelos ABM a partir de varias alternativas de calibración, proporcionando además una información más rica para analizar la dinámica del modelo estudiado. Además, la obtención de varios conjuntos de parámetros con un rendimiento óptimo garantiza una mayor robustez del modelo y, en consecuencia, ayuda tanto a validar como a analizar mejor la sensibilidad de los parámetros calibrados [Chi+17]. No obstante, los modeladores deben utilizar los métodos automáticos de calibración con cuidado y de forma iterativa y controlada para filtrar manualmente las distintas alternativas obtenidas. En caso contrario, los valores de los parámetros calibrados se verán forzados a coincidir con el comportamiento histórico, con el consiguiente riesgo de tratar el modelo como una caja negra (*black-box* en inglés) [Sal+10].

Nuestra propuesta se centra en el diseño de un método avanzado de calibración de ABMs basado en MMEAs y considera el uso y posterior comparación de nueve MMEAs competitivos, tanto clásicos como recientes, para tal fin. Durante el desarrollo de este capítulo analizaremos el rendimiento de una batería de MMEAs en 12 instancias de un ABM para marketing (es decir, en 12 mercados virtuales diferentes) en los que calibramos entre 24 y 129 parámetros para generar dos resultados principales: El reconocimiento o *awareness* histórico de la marca y el volumen del WOM. De este modo, podremos comprobar el comportamiento de la metodología propuesta a través del uso de los distintos algoritmos al calibrar los ABM con un número creciente de parámetros.

De la amplia familia de MMEAs consideramos tres métodos clásicos de anidamiento multimodal basados en GAs como Sharing [GR+87], Crowding [DJ75] y Clearing [Pet96], cuatro extensiones multimodales de DE como SHADE [TF13], L-SHADE [TF14], DE/NRAND/2 [EPV11], y MOBiDE [BDT12], un algoritmo de optimización de enjambre de partículas con anidamiento multimodal (NichePSO) [BEB02], y una extensión asistida por nichos de NSGA-II para la optimización multimodal llamada PNA-NSGA-II [BD13]. Mediremos el rendimiento de los algoritmos anteriores comparando la media, la desviación estándar y el valor mínimo de *fitness* de sus soluciones. Además, utilizaremos pruebas estadísticas para clasificar los algoritmos en función de su rendimiento, determinando así cuales son los más adecuados para nuestra metodología, y técnicas de visualización para analizar el conjunto de soluciones diferentes

devueltas por los de mejor rendimiento, explotando en consecuencia su capacidad para la validación de ABMs.

El capítulo está estructurado como sigue. Comenzamos revisando el estado del arte en optimización multimodal y calibración de ABMs en la Sección 6.2. La Sección 6.3 detalla las características del ABM cuyos parámetros pretendemos calibrar. La introducción a la optimización multimodal y las principales características y configuraciones de los nueve algoritmos multimodales utilizados para diseñar nuestra propuesta quedaron descritos y detallados en las Secciones 3.3.4 y 3.3.5. En la Sección 6.4 describimos el marco de trabajo propuesto para la calibración de ABMs utilizando MMEAs. Por su parte, la Sección 6.5 se centra en la presentación de los experimentos llevados a cabo para validar nuestra propuesta. Finalmente, el análisis de los resultados obtenidos queda desarrollado en la Sección 6.6.

## 6.2. Estado del arte

La calibración representa una fase fundamental durante la validación de cualquier modelo computacional [Oli03; Chi+17] y determina si puede o no ser utilizado con posterioridad. Aunque la calibración pueda realizarse manualmente de forma similar a un análisis de sensibilidad global [BVL16b] (simulando repetidamente el modelo y ajustando sus parámetros en función de los resultados observados), este enfoque resulta prohibitivo para muchos modelos realistas debido a la gran cantidad de parámetros que los caracterizan. La calibración automática aparece como un enfoque eficaz para abordar la calibración de modelos computacionalmente costosos [Oli03; Chi+17].

Las técnicas de calibración automática se han aplicado para diseñar modelos computacionales no lineales de diferentes áreas como el marketing y el estudio de comportamiento del consumidor [Chi+17; MCC19; Moy+21a], la economía [Fab13; Tad16], la simulación de tráfico [NM12; PK12], o la salud [Ser+16; LaT+19]. Por un lado, algunos autores han considerado el uso de métodos exactos basados en simplex [KR03] o en búsqueda de descenso de gradiente [TKG14], métodos estocásticos como la entropía cruzada [NM12] o la regresión lineal [CBM20] para la



calibración de ABMs. Sin embargo, estos enfoques están limitados a calibrar no más de 20 parámetros, lo que los hace inútiles cuando se consideran modelos con una mayor dimensión. Por el contrario, las metaheurísticas aparecen como métodos adecuados para calibrar problemas de alta dimensionalidad debido a su capacidad para explorar rangos amplios de parámetros y para considerar interacciones potencialmente no lineales entre ellos [SRW10; SR14].

Explorar el espacio de búsqueda de un problema de optimización complejo suele causar dificultades a cualquier algoritmo de optimización a la hora de encontrar el conjunto de soluciones óptimas locales y/o globales. Esto ha sido otro punto a favor para la aplicación de las metaheurísticas a la calibración de los ABMs dada su capacidad para explorar espacios multidimensionales y complejos en tiempo razonable y con soluciones de alta calidad. En consecuencia, existen varios ejemplos de calibración de ABMs usando metaheurísticas sobre modelos diseñados para las ciencias sociales y biológicas [CH05; Fab13; WEM16; Moy+17; Liu+17; MCC19]. Igualmente, existen numerosas contribuciones en la literatura que consideran la aplicación de metaheurísticas para la calibración de modelos y la estimación de parámetros, como los GAs [Dai+09; Bul+09], diferentes versiones de DE [ZC15; LaT+19] y otros tipos de algoritmos evolutivos y bioinspirados [ZCG14; Moy+21a]. Otros autores también han abordado la calibración de ABMs desde una perspectiva multiobjetivo [MD13; MCC21] donde son varios los indicadores clave de rendimiento (KPIs, por sus siglas en inglés) que se ajustan durante el proceso de calibración.

La selección de la metaheurística es decisiva para encontrar soluciones de calidad ya que muestran diferentes capacidades para explorar y explotar el espacio de búsqueda de un problema. Teniendo en cuenta esto y la naturaleza intrínsecamente multimodal de la calibración de ABMs, los algoritmos de optimización multimodal [SD06; Das+11; TI20] pueden considerarse una técnica eficaz para obtener soluciones diversas y de alta calidad en problemas grandes y complejos ya que son capaces de devolver diferentes configuraciones óptimas de parámetros con ajustes similares. De esta forma, los modeladores tienen a su disposición una herramienta que les permite seleccionar las soluciones que mejor se adaptan a sus necesidades o aumentar las alternativas que tienen para apoyar sus decisiones, disponiendo de información adicional para analizar la sensibilidad y la robustez de sus modelos [Chi+17].

Desde Goldberg y Richardson [GR+87], muchos investigadores han propuesto métodos basados en la introducción de técnicas de preservación de la diversidad en los algoritmos evolutivos [EPV11; BDT12; TF13; TF14]. Así, la familia de MMEAs ha ido creciendo y ha sido ampliamente utilizada en un gran número de problemas de optimización multimodal.

## 6.3. Descripción del ABM de inversión publicitaria en un mercado competitivo

### 6.3.1. Estructura del modelo

En este caso, vamos a utilizar un ABM distinto al de los Capítulos 4 y 5. El modelo ABM utilizado en nuestra propuesta fue introducido en [Moy+17] y simula un número determinado de semanas ( $T$ ) de un mercado que comprende un conjunto de marcas competidoras  $B$ . Utilizando un *time step* semanal, el modelo simula el comportamiento de los agentes  $I$  y sus reacciones a la exposición a las influencias sociales a través de una SN en un proceso de WOM y a las influencias externas procedentes de los medios publicitarios  $C$ . El modelo tiene dos KPIs: El *awareness* histórico de la marca y el número de interacciones WOM entre los consumidores, que denominamos volumen WOM. Estos KPIs se seleccionan por su importante papel en la dinámica del mercado [MS00; LMP13]. Presentamos un esquema general del modelo en la Figura 6.1:



Figura 6.1: Esquema general del modelo ABM tomado de [MCC19]. Los agentes expuestos a la publicidad de los medios publicitarios pueden conocer la marca que se promociona y hablar de ella con sus contactos directos (aumentando el volumen de WOM).

En resumen, el ABM intenta reproducir el impacto de las interacciones sociales en un mercado concreto y permitir a las marcas influir en dichas interacciones a través de la publicidad. A continuación, detallamos el conjunto de los distintos parámetros que definen las dinámicas del modelo.

### 6.3.2. Estados de los agentes y toma de decisiones

Los valores de *awareness* de los agentes se modelan mediante una variable de estado  $a_i^b \in [0, 1]$ .  $a_i^b(t) = 1$  representa que el agente  $i \in \{1, \dots, I\}$  conoce la marca  $b \in \{1, \dots, B\}$  en el instante  $t$ ; mientras que  $a_i^b(t) = 0$  representa que el agente no conoce la marca  $b$  en ese momento. Esta variable de estado se inicializa utilizando un conjunto de parámetros de *awareness* inicial ( $a^b(0) \in [0, 1]$ ), que representa el *awareness* global de la población al principio de la simulación, cumpliendo  $a^b(0) = \frac{1}{I} \sum_{i=1}^I a_i^b(0)$ . Por tanto, el porcentaje de agentes con *awareness* de las diferentes marcas al inicio de la simulación depende del valor del parámetro de *awareness* inicial de cada marca.

Los valores de *awareness* de los agentes pueden cambiar en cada paso de la simulación. Por un lado, los agentes pueden adquirir *awareness* de una marca debido a la publicidad o a la

interacción con otros agentes a través de un proceso de difusión de WOM. Por otro lado, el *awareness* de una marca puede perderse con el tiempo debido a un proceso de desactivación si no se refuerza con nuevos estímulos (similar al utilizado en el modelo *CONSUMAT* extendido descrito en la Sección 4.2). Estos efectos de pérdida/ganancia se modelan con parámetros adicionales, descritos en el siguiente párrafo.

Los efectos o dinámicas de ganancia y desactivación de *awareness* se regulan con dos parámetros. El parámetro que regula la velocidad a la que se pierde el *awareness* con el tiempo se llama probabilidad de desactivación del *awareness* ( $d \in [0, 1]$ ). Este proceso se modela comprobando cada marca  $b$  de que conoce el agente  $i$  ( $a_i^b(t) = 1$ ) al comienzo de cada paso  $t$  y desactivándola con una probabilidad  $d$ , estableciendo  $a_i^b(t) = 0$ . Si la desactivación surte efecto y el agente olvida la marca en cuestión, el agente podría volver a adquirir *awareness* debido a la difusión de WOM y/o a la acción de los medios publicitarios durante los siguientes pasos de la simulación.

Además del *awareness* de la marca, cada agente almacena el número de conversaciones producidas durante su proceso de difusión. Esta información se utiliza para calcular el volumen de WOM generado por cada marca para toda la población ( $\omega^b(t)$ ). Cada vez que un agente inicia un proceso de difusión y habla con sus contactos directos en la SN, la variable  $\omega_i^b(t)$  se actualiza incrementándola con el número total de conversaciones, que corresponde al número de contactos directos del agente.

### 6.3.3. Red social subyacente e interacciones WOM

Los agentes están conectados a través de una SN [WS98; BA99] donde cada nodo es un agente y los enlaces representan las conexiones de los agentes con sus contactos directos. Modelamos esta SN utilizando una red artificial libre de escala [BA99] con una distribución de grados que sigue la ley de potencias. Esto significa que unos pocos nodos tienen un número significativamente grande de conexiones (*hubs* de la SN) y la mayoría de los nodos tienen un número muy bajo de conexiones. Nuestra red libre de escala se genera a través del algoritmo BA [BA99] que utiliza un parámetro  $m$  para regular la tasa de crecimiento de la red y durante la ejecución del algoritmo que generará su densidad final.

Los agentes del modelo pueden difundir sus valores de *awareness* durante la simulación a través de la SN. Las interacciones entre los consumidores se producen durante todos los pasos de la simulación y facilitan el proceso de difusión de información [Rog10]. Cada agente  $i$  tiene una probabilidad de hablar  $p(t)_i^b \in [0, 1]$  para difundir su *awareness* en el instante  $t$  para cada marca  $b$ . Esta probabilidad especifica si el agente  $i$  habla con todos sus contactos directos en la SN en ese momento. El efecto de contagio se modela mediante el parámetro de impacto del *awareness* de WOM ( $\alpha^{WOM} \in [0, 1]$ ), que representa la probabilidad de que un agente vecino  $j$  (contacto directo de  $i$ ) obtenga reconocimiento de la marca o *awareness* tras mantener una conversación sobre ella con el agente  $i$ .

#### 6.3.4. Descripción de los medios publicitarios

Modelamos las influencias externas (esto es, las relaciones entre las marcas y los consumidores, como la publicidad) utilizando un enfoque similar al aplicado en la SN. Las influencias externas se parametrizan para definir las diferencias entre los medios publicitarios (es decir, prensa, radio y televisión). Los medios publicitarios en  $C$  influyen en los agentes en función del potencial del medio para llegar a la población y de la cantidad de inversión de cada marca. El porcentaje máximo de población al que puede llegar un medio publicitarios (punto de contacto o *touchpoint* en inglés) está limitado por la naturaleza del propio medio. Por ejemplo, el porcentaje máximo de población al que puede llegar un anuncio de televisión está limitado por el porcentaje máximo de población que ve la televisión. Modelamos estas diferentes propiedades con un parámetro de alcance ( $r_c \in [0, 1], \forall c \in \{1, \dots, C\}$ ), que limita el número máximo de agentes a los que un medio  $c$  puede llegar en un *time step*. Este proceso de alcance a los agentes o consumidores virtuales se realiza de manera estocástica dentro del modelo atendiendo a los parámetros propios de cada medio.

Las campañas publicitarias realizadas por los medios publicitarios se modelan mediante puntos de rating bruto (GRPs, por sus siglas en inglés). En publicidad [Far+10], un GRP es una medida de la magnitud de las impresiones programadas para medios publicitarios. En concreto, utilizamos la convención de que un GRP significa llegar al 1% de la población objetivo. La

variable  $\chi_c^b(t)$  modela las unidades de inversión en GRP para el *touchpoint*  $c$  por marca  $b$  y *time step*  $t$ . Cada medio tiene diferentes costes para las unidades de GRPs invertidas y las marcas tienen que elegir cuidadosamente su inversión, ya que aumentar el *awareness* de la población o el número de conversaciones utilizando los medios publicitarios implica un coste publicitario para la marca. Es por ello que las marcas tienen que definir su mix de marketing entre los medios existentes a partir de su presupuesto publicitario. De manera similar a la forma en la que modelamos las interacciones sociales, todos los medios  $c \in C$  consideran un parámetro de impacto de sensibilización ( $\alpha_c \in [0, 1], \forall c \in \{1, \dots, C\}$ ) que modela la probabilidad de que el agente conozca la marca después de un impacto del medio.

El efecto de la publicidad transmitida por los medios publicitarios puede producir un efecto viral<sup>1</sup> o viral en el agente alcanzado, el cual potencia el número de conversaciones sobre la marca anunciada, aumentando la probabilidad de hablar de los agentes alcanzados ( $p_i^b$ ). Este efecto se modela mediante una variable denominada incremento de *virialidad* ( $\tau_c$ ) definida para cada medio  $c$ . El incremento producido en la probabilidad de hablar del agente se calcula como un porcentaje de incremento sobre la probabilidad inicial de hablar del agente  $i$  ( $p_i^b(0)$ ) y queda representado por la variable  $\sigma_c$ . Sin embargo, si la virialidad generado no se refuerza, su efecto podría decaer con el tiempo al olvidarse las interacciones anteriores. Utilizamos una variable  $d\tau_c$ , llamada desactivación de la virialidad, para reducir los incrementos anteriores en la probabilidad de hablar  $\sigma_c$ . La Ecuación 6.1 define el proceso de actualización de la probabilidad de hablar del agente  $i$  para la marca  $b$  debido a los efectos de incremento y decaimiento de la virialidad del medio  $c$ .

$$p_i^b(t+1) = p_i^b(t) + p_i^b(0) \cdot \tau_c - \sigma_{i_c}^b(t) \cdot d\tau_c, \quad (6.1)$$

donde  $\sigma_{i_c}^b(t) = \sum_{s=1}^t (p_i^b(s) - p_i^b(0) \cdot \tau_c)$ .

---

<sup>1</sup>El marketing viral trata aprovechar el boca a boca a favor de una marca para incrementar su reconocimiento o *awareness* de marca. Lo consigue creando expectación alrededor de algo lo suficientemente interesante, novedoso o útil como para que la gente hable de ello y comparta su experiencia.

## 6.4. Propuesta de un método de calibración de ABMs basado en algoritmos evolutivos multimodales

En esta sección describimos nuestro marco de trabajo global para la calibración de ABMs utilizando MMEAs. En la Figura 6.2 se muestra un diagrama que ilustra los componentes de la propuesta. La sección se estructura como sigue. Comenzamos presentando cómo se manejan los objetivos del problema y los parámetros considerados durante la calibración en la Sección 6.4.1. A continuación, en la Sección 6.4.2 se explica el papel de los MMEAs, que son un componente central del marco, en la calibración de los modelos. Por último, en la Sección 6.4.3 se explica la importancia del análisis estadístico y la visualización para la validación de los modelos calibrados.

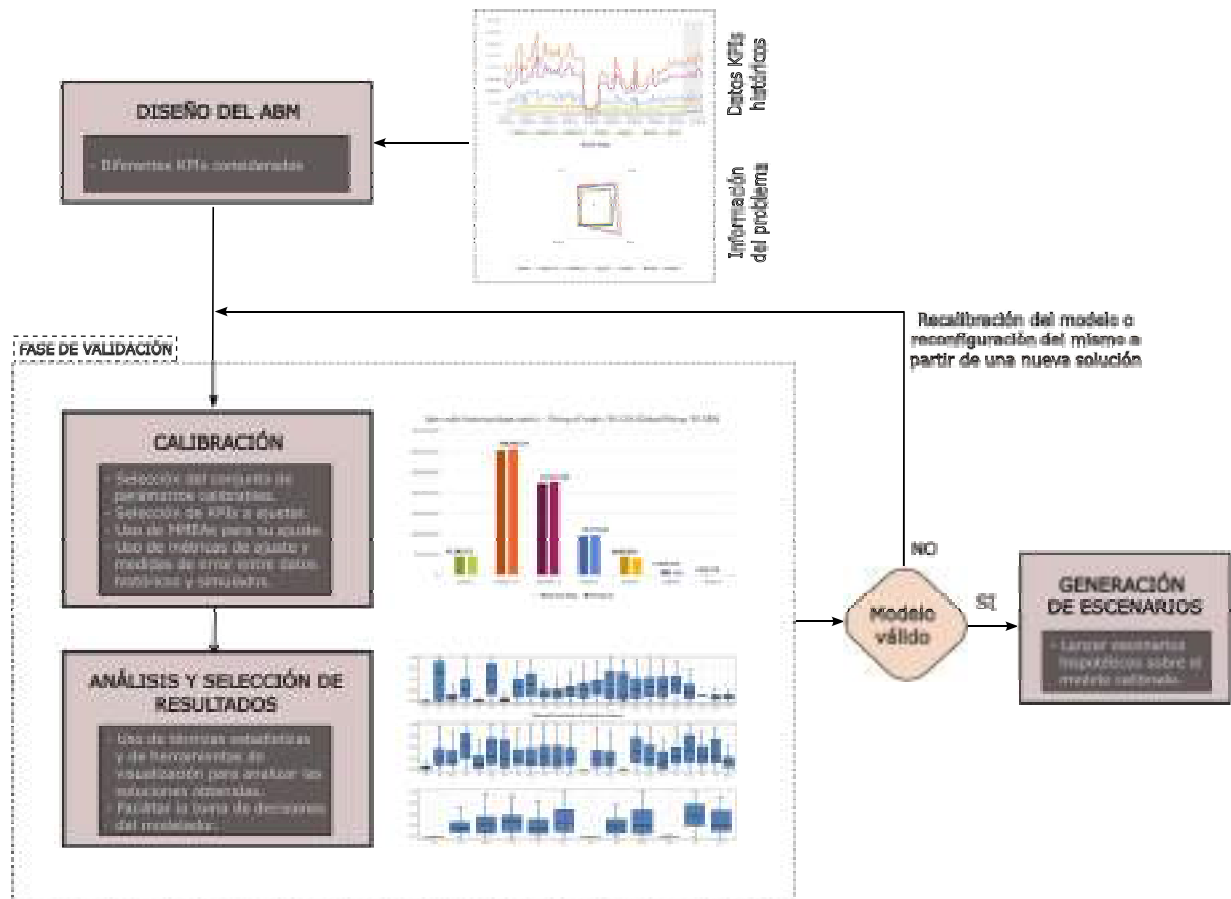


Figura 6.2: Diagrama que ilustra los componentes y el flujo de trabajo de nuestro marco global para la calibración de ABMs utilizando MMEAs.

### 6.4.1. Parámetros y objetivos de calibración

En un proceso de calibración automático, los valores de los parámetros del ABM se ajustan para hacer coincidir sus salidas con los datos reales (KPIs históricos) del escenario modelado. Cada configuración de parámetros  $X = (x_1, \dots, x_n)$  se define como un vector de  $n$  variables de decisión. El modelador debe ser cuidadoso a la hora de seleccionar los parámetros del modelo que se desean calibrar, ya que la dificultad de validar las configuraciones calibradas aumenta con el número de parámetros calibrados. Por una parte, se recomienda considerar aquellos parámetros con mayor incertidumbre y los más difíciles de definir de acuerdo con la información disponible. Por otra, también deben tenerse en cuenta los parámetros sensibles, es decir, aquellos en los que ligeras modificaciones sobre sus valores producen cambios significativos en las respuestas y resultados del modelo. Nuestro enfoque considera la calibración de parámetros utilizando valores tanto enteros como reales.

En nuestro enfoque evaluamos la calidad de una determinada configuración de parámetros del modelo con respecto a dos objetivos de calibración que se corresponden con los KPIs históricos considerados y que pueden estar o no en conflicto. Evaluamos la calidad del modelo con respecto al ajuste de su resultado a los datos reales e históricos proporcionados para los KPIs definidos para una marca objetivo: i) *awareness* y ii) volumen de WOM. Merece la pena mencionar que en nuestro enfoque, medimos el ajuste de una determinada solución combinando los valores de los objetivos considerados. Otros enfoques abordan el ajuste de cada objetivo por separado [MCC19]. Para evitar un ajuste excesivo en los parámetros calibrados resultantes, los datos históricos pueden dividirse en datos de entrenamiento y de *hold-out*<sup>2</sup>. Se recomienda considerar los conjuntos de *hold-out* cuando se dispone de datos reales al calibrar para poder testar si el modelo es lo suficientemente general y evitar el sobreajuste<sup>3</sup> [CR17]. Estos conjuntos son útiles para comprobar los resultados del proceso de calibración y mejorar la confianza que se tiene en modelo. No obstante, dado que las implementaciones de ABMs suelen utilizarse como modelos explicativos más que como predicciones [Has+13], el uso de conjuntos de *hold-out*

---

<sup>2</sup>Período durante el cual se conservan los datos para probar un modelo

<sup>3</sup><https://aws.amazon.com/es/what-is/overfitting/>



puede ser innecesario en algunos casos. El ajuste sobre datos históricos de entrenamiento se computa utilizando una medida de desviación o desajuste ( $\epsilon$ ). Esta medida de desviación es una distancia entre el dato real histórico y el resultado del modelo simulado que guía el algoritmo de optimización y puede calcularse utilizando cualquiera de las medidas de desviación estándar (RMSE, MAPE o MARE [HK06], por ejemplo). Es responsabilidad del modelador escoger aquella medida más adecuada para ajustar los objetivos considerados.

El objetivo del algoritmo de optimización es minimizar  $F(X) = (f_1(X), \dots, f_M(X))$ , donde  $M$  representa el número de objetivos. Cada función de *fitness*  $f_j(X)$  calcula el error asociado al objetivo  $j$ , y se define mediante la Ecuación 6.2:

$$\begin{aligned} \min F(X) &= \min(f_1(X), \dots, f_j(X), \dots, f_M(X)), \\ &\text{donde } f_j(X) = \epsilon^j \cdot (\sigma^j(X), \tilde{\sigma}^j), \end{aligned} \quad (6.2)$$

donde  $\tilde{\sigma}^j$  representa los valores reales del objetivo para la salida  $j$ -ésima y  $\sigma^j(X)$  representa los valores simulados del modelo utilizando la configuración de parámetros  $X$ . Obsérvese que  $\epsilon^j$  es independiente para cada objetivo  $j \in [1, M]$  y pueden utilizarse diferentes medidas de desviación. Nuestra propuesta y definición algorítmica no impone restricciones en este sentido.

Para la implementación y posterior experimentación utilizaremos en este problema el error medio porcentual absoluto (MAPE, por sus siglas en inglés). Las ecuaciones 6.3 y 6.4 definen las funciones de MAPE,  $f_1$  y  $f_2$ , que comparan los datos históricos con respecto a las salidas de la simulación durante cada evaluación de una solución candidata.

$$f_1 = \frac{100}{T \cdot B} \sum_{b=1}^B \sum_{t=1}^T \left| \frac{a^b(t) - \tilde{a}^b(t)}{\tilde{a}^b(t)} \right|, \quad (6.3)$$

$$f_2 = \frac{100}{T \cdot B} \sum_{b=1}^B \sum_{t=1}^T \left| \frac{\omega^b(t) - \tilde{\omega}^b(t)}{\tilde{\omega}^b(t)} \right|, \quad (6.4)$$

donde  $\tilde{a}$  y  $\tilde{\omega}$  representan los valores objetivo históricos de *awareness* y volumen de WOM de toda la población y se utilizan para medir el impacto de cada medio en sus consumidores a través de la recopilación de todas las interacciones WOM y el reconocimiento total de las campañas de

la marca para cada semana.

Los valores simulados se generan ejecutando varias simulaciones de Monte Carlo del ABM considerando la configuración de parámetros codificada en la solución evaluada y calculando la media de esas ejecuciones independientes.  $f_1$  y  $f_2$  se combinan en una función objetivo final  $f$  como se muestra en la Ecuación 6.5:

$$f = \frac{1}{R} \sum_{i=1}^R \beta f_1^i + (1 - \beta) f_2^i, \quad (6.5)$$

donde  $R$  es el número de ejecuciones de Monte Carlo y  $\beta \in [0, 1]$  es el parámetro utilizado para la combinación ponderada de  $f_1$  y  $f_2$ .

#### 6.4.2. Algoritmo de calibración basado en MMEAs

El MMEA es un componente central de nuestro marco de calibración. Los algoritmos multimodales son metaheurísticas basadas en poblaciones que representan las soluciones del problema como individuos dentro de una población. Estas técnicas pueden proporcionar diferentes configuraciones del modelo (es decir, conjuntos de valores de los parámetros), diversas (valores de parámetros suficientemente distintos entre soluciones) y con un valor óptimo de *fitness* en una sola ejecución. En nuestra propuesta para la calibración de ABMs, cada individuo de la población tiene  $n$  genes que se corresponden con las  $n$  variables de decisión que representan cada configuración del modelo, siendo estos genes valores codificados reales o enteros. Las distintas configuraciones de parámetros obtenidas por los MMEAs se devuelven ordenadas según su valor de ajuste al final del proceso de calibración para que el modelador pueda analizarlas.

Pese a que cualquier MMEA puede llevar a cabo un proceso de calibración, deben tenerse en cuenta las características del modelo que se está calibrando a la hora de seleccionarlo. Así, si el número de variables de decisión es bajo (menor de 50) se podría considerar cualquier MMEA básico como los GAs de *Fitness Sharing* [GR+87; Gol89], Crowding [DJ75], Clearing [Pet96; Pét97] o un algoritmo basado en enjambres de partículas como NichePSO [BEB02]. Cuando

el número de variables de decisión es alto, se hace necesario el uso de técnicas más avanzadas de optimización multimodal que sean capaces de explorar el espacio de soluciones del modelo a calibrar y de explotarlo (encontrando las configuraciones de parámetros de mejor ajuste y diversidad) satisfactoriamente y en un tiempo razonable. Entre estos métodos avanzados se encuentran SHADE [TF13], L-SHADE [TF14], DE/NRAND/2 [PSL06] o MOBiDE [YKG09]. Sin embargo, esta decisión debe ser tomada por el modelador, ya que depende del espacio de búsqueda específico del problema de calibración definido. Los MMEAs anteriormente mencionados fueron descritos en profundidad en la Sección 3.3.5.

### 6.4.3. Análisis estadístico y visual de resultados de calibración

El uso eficaz de la simulación de los ABMs para representar sistemas complejos depende en gran medida de la transparencia de los modelos subyacentes. Los modeladores de ABMs y los usuarios interesados en el uso de esta tecnología deben entender cómo recrea el modelo un comportamiento determinado, ya que la simulación de ABMs se utiliza con frecuencia para definir y prever escenarios hipotéticos [JO06; Voi+16]. Esto puede lograrse desde una perspectiva de caja blanca (*white-box* en inglés) [SWM17], en la que tanto los modeladores como las partes interesadas pueden hacer uso de herramientas de visualización para aumentar la explicabilidad del modelo.

Mejorar la comprensión de los modelos basados en la AI es uno de los objetivos del área de la AI explicable [SWM17]. Se alienta a modeladores e investigadores a abrir modelos *black-box* para que su comportamiento pueda entenderse de forma sencilla y para que sus resultados puedan explicarse de una forma más simple y clara. La AI explicable también potencia las soluciones proporcionadas por los modelos *white-box*, ya que el aumento de la transparencia de las soluciones proporcionadas debería aumentar la confianza en el comportamiento y en el rendimiento de estas soluciones. Esto pone de relieve el papel de los métodos de visualización para la calibración de modelos, ya que son herramientas poderosas que aumentan la comprensión del modelador sobre el modelo calibrado y sus ajustes de parámetros [Chi+17]. El uso de la visualización aumenta la transparencia de los indicadores de calidad (centrados principalmente

en el ajuste del modelo a los datos reales) para la validación del modelo calibrado [Lee+15; Chi+17]. Así, mostrar visualmente las relaciones subyacentes entre una configuración de entrada y su correspondiente salida del modelo se convierte en un componente crítico del proceso de validación.

La validación de las soluciones obtenidas utilizando MMEAs requiere del uso de técnicas de visualización capaces de medir e identificar cómo se comporta cada parámetro calibrado dentro del conjunto óptimo de soluciones devuelto por el método de optimización. Adicionalmente, se requiere acompañar a los métodos de visualización con medidas estadísticas que permitan medir de forma cuantitativa la variabilidad de cada configuración. Esto refuerza la confianza del modelador en los resultados, su interpretación y su posterior uso. Entre las técnicas de visualización utilizadas para analizar las soluciones obtenidas por los MMEAs se encuentran los *heatmaps* y los *boxplots*. Esta última técnica permite realizar un análisis tanto cuantitativo como cualitativo de los datos como veremos posteriormente. Cuando se considera el ajuste de dos o más objetivos conflictivos, se requieren métodos de visualización específicos para la validación del modelo [MCC19]. La Figura 6.3 muestra un ejemplo de aplicación de análisis visual sobre soluciones obtenidas por un MMEA aplicado a la calibración de modelos ecológicos poblacionales [Chi+17]. El método construye un histograma usando la frecuencia de aparición de valores para cada parámetro del modelo considerando todas las soluciones de calibración del conjunto devuelto por el MMEA.

## 6.5. Diseño experimental

### 6.5.1. Definición de escenarios

La experimentación considera 12 instancias diferentes de un ABM que modela un escenario de marketing bancario real. La dimensionalidad de cada instancia depende directamente del número de medios  $|C|$  en el mercado virtual. Es importante mencionar que el número de medios en un mercado específico (independientemente de las instancias que utilicemos en nuestra configuración

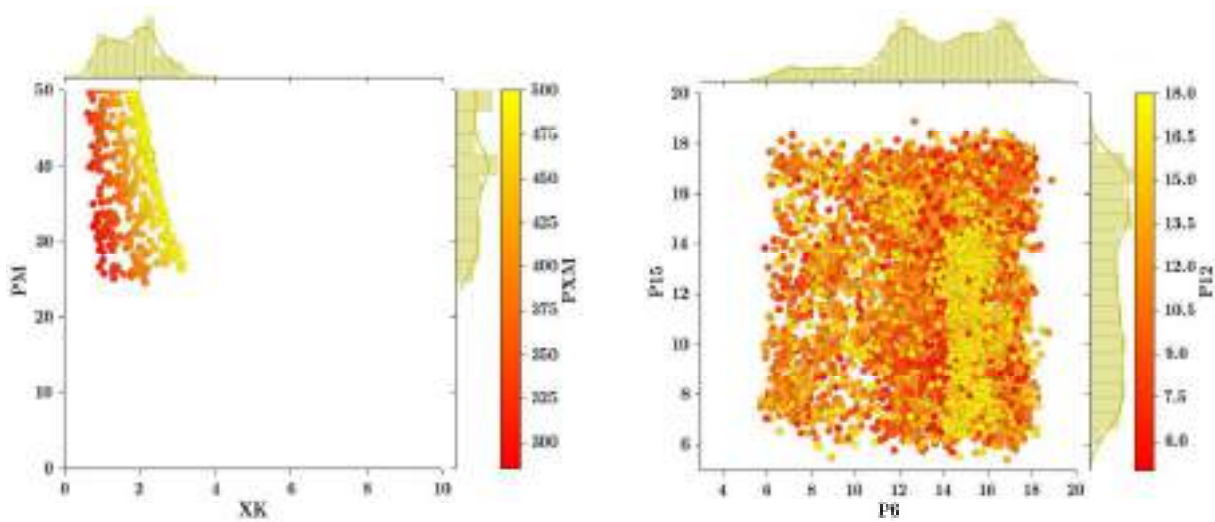


Figura 6.3: Ejemplo de método de análisis visual sobre soluciones obtenidas por un MMEA durante la calibración de dos modelos tomado de [Chi+17]. A la izquierda de la figura se presenta un análisis de sensibilidad realizado sobre dos parámetros,  $p_M$  y  $X_k$ , con respecto a un tercer parámetro  $P_{X_m}$  en un modelo ecológico poblacional. A la derecha, se realiza el mismo análisis de sensibilidad sobre los parámetros  $P_{15}$  y  $P_6$  respecto a  $P_{12}$  en un modelo de dinámica de sistemas.

experimental) es variable y depende del mercado. Dado que el método de calibración propuesto debe ser genérico para ser útil en cualquier mercado,  $|C|$  no puede ser un número concreto, sino una variable que define el modelador en cada mercado.

El conjunto final de parámetros que se seleccionan para la calibración viene determinado por el tamaño de la instancia del modelo: Tres parámetros para cada *touchpoint* más tres parámetros sociales fijos. Así, el número de parámetros que se calibran se calcula como  $(|C| + 1) \cdot 3$ . En resumen, estos parámetros son los siguientes:

- **Parámetros de los medios publicitarios:** Para cada *touchpoint* definido  $c \in C$ , calibramos su impacto en el *awareness* ( $\alpha_c$ ), el incremento de la viralidad ( $\tau_c$ ) y su desactivación ( $d\tau_c$ ).
- **Parámetros de la SN:** Calibramos la probabilidad inicial de hablar ( $p^b(0)$ ), el impacto del *awareness* social ( $\alpha^{WOM}$ ) y su desactivación ( $d$ ).

Disponemos de una instancia inicial *baseline*, denominada P1(24), correspondiente a un mercado real con  $|C| = 7$  medios, a partir de la cual se generan sintéticamente el resto de instancias. El conjunto completo de parámetros del modelo correspondiente a esta instancia de referencia así

como sus valores se recoge en la Tabla 6.1. Cada variación de la instancia introduce medios publicitarios adicionales que se generan a partir de los iniciales modificando sus valores de inversión de forma aleatoria. Los valores objetivo de  $\tilde{a}$  como los de  $\tilde{\omega}$  proceden de agencias de medios externas contratadas por una entidad bancaria colaboradora.

Nombre	Descripción	Valor
$ I $	Número de agentes	1,000
$ B $	Número de marcas	8
$ C $	Número de medios publicitarios	7
$ T $	Número de pasos	52
$m$	Parámetros para el generador de SNs	4
$a^b(0)$	<i>Awareness</i> inicial para las marcas $b$	$a^1(0) = 0.71, a^2(0) = 0.76, a^3(0) = 0.59, a^4(0) = 0.26,$ $a^5(0) = 0.08, a^6(0) = 0.43, a^7(0) = 0.4, a^8(0) = 0.34$
$r_c$	Alcance de cada <i>touchpoint</i> $c$	$r_1 = 0.93, r_2 = 0.58, r_3 = 0.55, r_4 = 0.04,$ $r_5 = 0.43, r_6 = 0.38, r_7 = 0.69$

Tabla 6.1: Lista y configuración de parámetros fijos de la instancia base de nuestro modelo de mercado (instancia P1(24)).

Además, cada modelo incluye una perturbación aleatoria de los valores históricos objetivo para ambos KPIs, el *awareness* y el volumen de WOM. Cada una de las nuevas instancias generadas aumenta la dimensionalidad de la anterior, incluyendo nuevas variables de decisión para permitir una comparación más completa de los diferentes MMEAs, como se ve en la Tabla 6.2. Obsérvese que cada instancia está etiquetada con su número de variables de decisión: P1(24), P2(39), P3(45), P4(54), P5(60), P6(69), P7(75), P8(84), P9(90), P10(99), P11(114) y P12(129).

Instancia	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Medios ( $ C $ )	7	12	14	17	19	22	24	27	29	32	37	42
Variables de decisión	24	39	45	54	60	69	75	84	90	99	114	129

Tabla 6.2: Nombre, número de medios publicitarios y variables de decisión asociados a cada instancia del problema.

Las variaciones sobre los medios publicitarios existentes  $C$  consisten en aumentar o reducir la inversión original de cada marca para cada uno de los instantes de tiempo, multiplicando su valor por un factor determinado. Consideramos factores de reducción para los valores originales de 15 %, 30 %, 45 % y 60 %, y factores de aumento de 100 %, 200 %, 300 % y 400 %. Estas últimas decisiones se toman al azar y no se modifican en cada paso. En cambio, las modificaciones de los valores objetivo de *awareness* y volumen de WOM se aplican sumando o restando una cantidad determinada a cada uno de sus pasos temporales. De este modo, cada modificación de

los valores objetivo de *awareness* añade o resta un 2%, 5%, 8% o 10% a los valores objetivo de la marca. Los valores de *awareness* resultantes se truncan entre el 1% y el 100% para evitar valores objetivo poco realistas. Con respecto al volumen de WOM objetivo, cada modificación creciente o decreciente es de 1,000, 2,000, 4,000 o 6,000 conversaciones, manteniendo los valores de conversación siempre positivos (esto es,  $> 0$ ). La Tabla 6.3 muestra un ejemplo del proceso llevado a cabo para generar las instancias.

P2(40)									
Generación de nuevos medios									
Medio original	Nuevo medio	<i>b1</i>	<i>b2</i>	<i>b3</i>	<i>b4</i>	<i>b5</i>	<i>b6</i>	<i>b7</i>	<i>b8</i>
3	8	+100 %	-15 %	-15 %	-45 %	+200	-60 %	+300 %	-30 %
5	9	+100 %	-30 %	+400 %	+200 %	+300 %	-30 %	-30 %	-15 %
5	10	-15 %	-15 %	+300 %	+300 %	+200 %	-30 %	+300 %	-30 %
3	11	-60 %	+100 %	+300 %	-45 %	+200 %	+400 %	-45 %	-15 %
6	12	+200 %	-15 %	-60 %	-30 %	-30 %	+400 %	-60 %	200 %
Modificación del <i>awareness</i> objetivo									
	<i>b1</i>	<i>b2</i>	<i>b3</i>	<i>b4</i>	<i>b5</i>	<i>b6</i>	<i>b7</i>	<i>b8</i>	
	-10 %	-8 %	+10 %	+10 %	-8 %	-5 %	+5 %	+5 %	
Modificación del volumen de WOM (Conversaciones) objetivo									
	<i>b1</i>	<i>b2</i>	<i>b3</i>	<i>b4</i>	<i>b5</i>	<i>b6</i>	<i>b7</i>	<i>b8</i>	
	-6,000	-6,000	-4,000	+4,000	+6,000	-2,000	+4,000	-1,000	

Tabla 6.3: Resumen de las transformaciones de parámetros realizadas en la generación de P2(40) a partir de la instancia P1(24). Las modificaciones de inversión se muestran como un porcentaje de incremento/decremento del medio original al nuevo. También se incluye la modificación de los valores históricos objetivo (*awareness* y volumen de WOM).

### 6.5.2. Parámetros de calibración

La Tabla 6.4 muestra el conjunto de parámetros seleccionados para ser calibrados automática usando nuestra propuesta basada en MMEAs. Los parámetros considerados son aquellos que controlan las dinámicas pérdida/ganancia del *awareness* de los agentes y el volumen de conversaciones que mantiene dicho agente, ya que son los más inciertos y los más complejos de estimar por el modelador utilizando los datos disponibles [MCC19]. El resto de instancias comparten esta configuración inicial, junto con el correspondiente valor del parámetro de alcance  $r_c$  para cualquier nuevo *touchpoint* que tome el valor del canal original empleado en su generación. Es decir, si se genera un nuevo medio masivo  $c_{12}$  utilizando el medio original  $c_3$ , éste comparte el valor del parámetro de alcance (es decir,  $r_{12} = r_3$ ).

Nombre	Descripción	Valor
$d$	Probabilidad de desactivación del <i>awareness</i>	-
$p_i^b(0)$	Probabilidad de hablar inicial (igual para cada marca $b$ )	-
$\alpha^{WOM}$	Impacto de las interacciones sociales en el <i>awareness</i>	-
$\alpha_c$	Impacto de los medios en el <i>awareness</i> $c$	-
$\tau_c$	Incremento de viralidad de los medios $c$	-
$d\tau_c$	Desactivación de viralidad por medio $c$	-

Tabla 6.4: Lista de variables de decisión asociadas a la instancia P1(24).

El proceso de calibración asigna cada uno de los parámetros seleccionados del modelo a una única variable de decisión, limitando el rango de valores posibles de los parámetros a un valor real en  $[0, 1]$ . La Tabla 6.5 muestra un ejemplo de solución de calibración para la instancia P2(39).

$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$
4.420592e-05	0.9275268	0.91726940	0.358706832	1.991031e-04	0.188117322	7.712790e-03
$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$	$\tau_6$	$\tau_7$
0.919016368	0.85484480	0.445861067	0.037229668	0.42503753	0.7193559683	0.046021892
$d\tau_1$	$d\tau_2$	$d\tau_3$	$d\tau_4$	$d\tau_5$	$d\tau_6$	$d\tau_7$
0.013241515	0.19740914	0.4053775552	0.008095596	0.670678402	0.803071854	0.463061986
$p^b$	$\alpha^{WOM}$	$d$	$\alpha_8$	$\tau_8$	$d\tau_8$	
0.1287573	0.02508721	0.06805826	0.780366135	0.134720552	0.6739496654	
$\alpha_9$	$\tau_9$	$d\tau_9$	$\alpha_{10}$	$\tau_{10}$	$d\tau_{10}$	
0.8341758	0.004248453	0.737444678	0.6065266	0.80506945	0.226626727	
$\alpha_{11}$	$\tau_{11}$	$d\tau_{11}$	$\alpha_{12}$	$\tau_{12}$	$d\tau_{12}$	<i>Fitness</i>
0.790572098	0.128404770	0.739637188	0.0015857279	0.7423187907	0.831388290	38.15518

Tabla 6.5: Ejemplo de una solución de calibración para la instancia P2(39) donde las variables de decisión coinciden con los parámetros codificados reales del modelo (que se limitan a  $[0, 1]$ ) y el valor final representa el *fitness* de la solución.

### 6.5.3. Configuración experimental

Realizamos 20 ejecuciones independientes de cada MMEA utilizando diferentes semillas aleatorias. Para adaptar los algoritmos al tamaño del problema y a la dimensionalidad de su espacio de búsqueda, definimos un tamaño de población variable  $P = 10 \cdot D$ , donde  $D$  es el número de variables que se calibran. Además, establecemos como criterio de parada del algoritmo un número de evaluaciones proporcional a la dimensión del problema multiplicando  $P$  por 50 para cada instancia del problema. El ajuste de los individuos se calcula como el valor medio de  $f$



para  $R = 15$  simulaciones de Monte Carlo del ABM, lo cual demostró ser suficiente en una experimentación previa. Se considera un peso  $\beta = 0.5$  en la función objetivo.

Todos los MMEAs seleccionados se implementan en Java utilizando el framework ECJ [Luk98]. Las poblaciones iniciales de los algoritmos se inicializan aleatoriamente. La configuración de los hiperparámetros para los MMEAs se definió mediante una extensa experimentación preliminar en la que cada algoritmo se ejecutó con un conjunto diferente de valores. Probamos los algoritmos empezando con las configuraciones que sus autores recomiendan en la literatura y luego ampliamos estos valores utilizando rangos mínimos y máximos. Finalmente, seleccionamos de todo el rango de valores las configuraciones de hiperparámetros que mejor se comportaron, obteniendo una referencia para cada algoritmo con los siguientes valores finales:

- **Métodos tradicionales:** Fijamos el radio de distancia  $\sigma_{share}$  a 0.033 tanto para Sharing como para Crowding. La brecha generacional ( $G$ ) y el factor de aglomeración ( $CF$ ) se fijan en 3 para el método Crowding. Por último, fijamos una capacidad de nicho  $k = 3$  para Clearing. Los tres métodos utilizan los mismos operadores de selección, cruce y mutación: Un método de selección de torneo ternario, un operador de cruce BLX- $\alpha$  con  $\alpha = 0.6$  y probabilidad de cruce  $p_c = 0.3$  y un operador de mutación aleatorio aplicado a cada individuo con una probabilidad  $p_m = 0.25$ .
- **Métodos basados en DE:** Al tratarse de métodos adaptativos (sin parámetros), no es necesario establecer ningún parámetro para los mecanismos de anidamiento. El valor de la probabilidad de cruce se establece en  $CR = 0.9$  para todos los métodos, mientras que la probabilidad de mutación se establece en  $F = 0.5$  para SHADE, L-SHADE y DE/nrand/2, y en  $F = 0.8$  para MOBiDE. Tanto para SHADE como para L-SHADE, fijamos el tamaño de la memoria histórica a la dimensionalidad del modelo que se está calibrando,  $D$ . El  $N_{init}$  y el  $N_{min}$  para el método L-SHADE se establecen en  $P$  y 4, respectivamente.
- **Niche-PSO:** Fijamos el radio máximo permitido en  $R_{max} = 0.1$  y los valores de  $\delta$  y  $\mu$  en  $-10^{-4}$  y  $-10^{-3}$ , respectivamente.
- **PNA-NSGA-II:** Como método adaptativo, no es necesario establecer los parámetros del mecanismo de anidamiento. Definimos el número de objetivos en los que hay que

descomponer el objetivo original a 2 para utilizar el método NSGA-II. A continuación, utilizamos la misma configuración de parámetros que en [BD13], utilizando también un cruce binario simulado (SBX) [DA+95] para el espacio de búsqueda continuo.

## 6.6. Análisis de resultados

En esta sección vamos a proceder con el análisis de los resultados obtenidos. En primer lugar, definimos los criterios de evaluación y el filtrado de soluciones utilizados durante el análisis de los experimentos. A continuación, analizamos y comparamos los rendimientos de los diferentes MMEAs, mostrando las diferentes soluciones obtenidas por el MMEA de mejor rendimiento en el espacio de decisión y, finalmente, analizamos cómo cambian las distribuciones de ajuste para algunas instancias del problema.

Consideramos tres medidas diferentes para evaluar y clasificar las mejores soluciones de calibración en los diferentes escenarios del modelo: Eficacia (el ajuste de calibración definido en la ecuación 6.5), eficacia basada en múltiples soluciones (capacidad de encontrar múltiples óptimos) y diversidad en el conjunto final de soluciones. Esta evaluación del rendimiento comprueba si el algoritmo de calibración explora adecuadamente el espacio de búsqueda del problema cuando los óptimos pertenecen a regiones diferentes y si es capaz de explotar eficientemente aquellas donde hay soluciones óptimas similares. Comparamos el rendimiento de calibración de los diferentes MMEAs clasificándolos en términos de rendimiento de *fitness* (es decir, ajuste del modelo) considerando los valores medios de *fitness* del conjunto de soluciones devueltas por los MMEAs. Además, realizamos pruebas estadísticas sobre los diferentes resultados y utilizamos herramientas de visualización como los mapas de calor para visualizar el espacio de búsqueda de los parámetros y situar en ellos el conjunto de soluciones devueltas por los MMEA con objeto de ilustrar las capacidades de validación del marco general de calibración multimodal de ABMs propuesto.

### 6.6.1. Comparación de resultados

La Tabla 6.6 muestra la media, la desviación estándar y los valores mínimos de *fitness* obtenidos por los MMEAs en las 12 instancias. La última columna de la Tabla 6.6 muestra los ajustes mínimos y máximos medios obtenidos para cada instancia. Complementamos esta última información con un test estadístico que considera el ranking de los algoritmos y aplica varios procedimientos post-hoc para resaltar las diferencias significativas en su rendimiento al final de la sección. Realizamos un test no paramétrico de Friedman, un test de Bonferroni-Dunn y un test de Holm.

	SHADE	L-SHADE	NichePSO	PNA-NSGA-II	MOBiDE	Crowding GA	Sharing GA	DE/nrand/2	Clearing GA	Fitness Value [min., max.]
P1	$\bar{x}$ (Std) <i>best</i> 27.14 (1.53) <b>25.27</b>	29.07 (1.11) 26.66 39.13 (0.36) 38.67	28.27 (1.09) 27.41 43.95 (2.43) 39.11	29.52 (2.38) 25.37 38.66 (0.27) 38.21	28.82 (2.08) 26.51 39.97 (0.86) 38.87	32.58 (0.39) 31.61 50.82 (0.82) 49.12	33.53 (0.57) 32.24 52.58 (1.24) 49.60	33.53 (0.59) 32.12 52.37 (1.68) 47.39	33.65 (0.41) 32.99 53.05 (0.59) 51.98	[27.14, 33.65] [38.43, 53.05] [25.69, 30.3]
P2	$\bar{x}$ (Std) <i>best</i> 26.78 (2.83) <b>22.92</b>	25.69 (0.47) 24.87 34.78 (1.64) 32.19	28.76 (1.5) 24.65 34.2 (2.8) 31.63	26.74 (0.86) 25.12 35.78 (2.38) 31.21	27.49 (1.02) 25.13 32.70 (0.47) 31.99	28.86 (0.28) 28.28 38.39 (0.25) 37.61	30.11 (0.5) 28.48 40.28 (0.43) 39.52	30.75 (0.82) 29.24 42.5 (0.51) 41.55	30.3 (0.3) 29.61 41.79 (0.58) 40.68	[25.69, 30.3] [31.21, 42.5] [28.02, 37.21] [40.44, 55.4]
P3	$\bar{x}$ (Std) <i>best</i> 28.02 (1.34) <b>25.69</b>	29.67 (0.40) 28.50 41.01 (0.09) 40.84	29.96 (2.5) 26.07 40.87 (1.54) 39.71	29.55 (1.49) 27.07 41.91 (0.30) 41.35	28.73 (1.43) 27.31 43.88 (1.75) 40.69	33.26 (0.18) 32.69 45.48 (0.34) 44.93	35.6 (0.33) 34.96 51.98 (0.71) 50.6	37.16 (0.54) 36.19 54.59 (0.70) 52.67	37.21 (0.51) 35.96 55.4 (0.82) 53.62	[31.21, 42.5] [28.02, 37.21] [40.44, 55.4]
P4	$\bar{x}$ (Std) <i>best</i> 35.63 (0.92) <b>33.82</b>	38.16 (1.23) 35.54 32.79 (1.40) 31.39	<b>35.25</b> (1.72) 34.06 32.2 (1.59) 29.47	36.78 (1.94) 34.31 34.37 (1.25) 32.32	36.77 (1.28) 34.92 37.48 (0.64) 34.89	42 (0.22) 41.64 48.20 (0.38) 46.91	45.04 (0.28) 44.45 53.27 (0.47) 52	45.93 (0.43) 45.19 54.35 (0.58) 52.68	46.59 (0.42) 46.02 55.86 (0.74) 54.52	[35.25, 46.59] [29.42, 55.86]
P5	$\bar{x}$ (Std) <i>best</i> 25.53 (0.85) <b>24.34</b>	29.30 (0.62) 27.66 44.99 (0.82) 43.39	28.04 (1.73) 25.18 47.85 (2.11) 43.98	29.54 (2.17) 25.87 47.30 (0.56) 46.31	30.85 (0.62) 29.92 50.25 (3.20) 44.79	40.32 (0.36) 39.55 53.91 (0.27) 53.45	46.76 (0.66) 45.27 67.62 (1.32) 64.15	48.37 (0.63) 46.44 70.16 (1.21) 68.52	49.49 (0.45) 48.74 73.46 (1.47) 70.41	[25.53, 49.49] [43.97, 73.46]
P6	$\bar{x}$ (Std) <i>best</i> 27.69 (1.25) <b>25.36</b>	30.22 (0.57) 29.09 <b>27.38</b> (0.22) 27.07	28.43 (0.89) 26.31 32.18 (1.74) 27.93	32.5 (1.54) 28.69 29.64 (0.46) 29	32.87 (1.36) 29.5 32.59 (1.52) 29.89	42.64 (0.35) 41.85 36.93 (0.45) 35.97	47.04 (0.43) 46.26 40.73 (0.36) 39.98	47.64 (0.68) 46.05 41.37 (0.44) 40.56	48.57 (0.32) 47.85 42.06 (0.60) 40.05	[27.69, 48.57] [27.38, 42.06]
P7	$\bar{x}$ (Std) <i>best</i> 28.78 (3.70) <b>26.36</b>									

Tabla 6.6: Promedio, desviación estándar y mejores valores de ajuste encontrados por los MMEAs para cada instancia del problema (los mejores resultados aparecen resaltados en negrita).

Los valores de la Tabla 6.6 muestran a SHADE como el algoritmo de mejor rendimiento, con un rango medio de 1.33. SHADE obtiene los valores de *fitness* medios más bajos (más cercanos a los datos históricos) en 9 de las 12 instancias del modelo, además de obtener el valor de *fitness* mínimo para todas las instancias. L-SHADE es el segundo algoritmo con mejor rendimiento, con un rango medio de 3, consiguiendo el mejor valor medio de *fitness* en las instancias del modelo P3 y P12. Con una mínima diferencia respecto a L-SHADE, NichePSO es el tercer mejor MMEA, obteniendo un rango medio de 3.08. NichePSO alcanza el mejor valor medio de *fitness* en la instancia del modelo P7 y el segundo mejor *fitness* medio en 5 de las 12 instancias del modelo, siendo también el segundo algoritmo mejor clasificado en términos de *fitness* mínimo (7 de las 12 instancias). PNA-NSGA-II ocupa la cuarta posición en términos de *fitness* medio con un rango medio de 3.58, mientras que MOBIDE obtiene la quinta posición de los cinco MMEAs de mejor rendimiento con un rango medio de 4. Por el contrario, Crowding, Sharing, DE/nrand/2 y Clearing, con rangos medios de 6, 7.17, 8 y 8.83, respectivamente, muestran el peor rendimiento, entre 1.1 % y 1.6 % peor que el mejor algoritmo.

Aunque se encuentra a cierta distancia de los cinco algoritmos con mejores resultados, el GA Crowding es el MMEA clásico que mejor se comporta, obteniendo mejores resultados que un método más actual como DE/nrand/2. Los resultados muestran que, aunque la dimensión del problema aumenta con cada instancia del modelo, SHADE sigue proporcionando el mejor rendimiento promedio. Esto permite demostrar que SHADE es independiente de la dimensionalidad del problema de calibración y sugiere que sería un método adecuado para la calibración de instancias ABM de alta dimensión.

Sobre la base del análisis anterior, podemos dividir claramente los MMEAs en dos grupos en función de su rendimiento. Por un lado tenemos los tres MMEAs clásicos (Crowding, Clearing y Sharing) así como DE/nrand/2. Por otro lado, tenemos a SHADE, L-SHADE, MOBIDE, NichePSO y PNA-NSGA-II como los MMEAs de mejor rendimiento en nuestro problema.

Además, extraemos dos conclusiones adicionales interesantes de los valores de la Tabla 6.6. En primer lugar, observamos que las diferencias de ajuste entre los peores y los mejores algoritmos aumentan gradualmente con las dimensiones del modelo. Así, podemos ver en la última columna

de la Tabla 6.6 que estas diferencias son menores de P1 a P7 (alrededor del 20%) que de P8 a P12, donde el rendimiento del peor algoritmo es en torno a un 47% peor que el valor del mejor algoritmo. Por tanto, podemos afirmar que los mejores algoritmos son más adecuados para instancias ABM de alta dimensionalidad.

La segunda conclusión proviene del análisis de las diferencias entre la media, la desviación estándar y los mejores ajustes encontrados para cada instancia del problema. Los resultados obtenidos por cada algoritmo cambian para cada instancia. Así, aunque se utiliza el mismo mercado virtual (P1) para generar las instancias del problema, observamos que tanto la dimensión del espacio de búsqueda como la dinámica del mercado cambian de una instancia a otra, lo que da lugar a comportamientos diferentes de las instancias del ABM (de hecho, comprenden mercados virtuales con características y complejidades diferentes). Con objeto de hacernos una idea de la forma de las funciones de *fitness* en cada instancia del problema, generamos y evaluamos 200,000 soluciones aleatorias para construir los histogramas que se muestran en la Figura 6.4. La Figura 6.5 demuestra cómo los rangos de *fitness* de las soluciones obtenidas por SHADE en el proceso de calibración también cambian de una instancia del problema a otra. Las dos figuras demuestran la diversidad de comportamiento de las instancias y el potencial de los algoritmos de mejor rendimiento para seguir encontrando configuraciones de parámetros prometedoras mientras aumentan la dimensión y la complejidad de la dinámica del mercado.

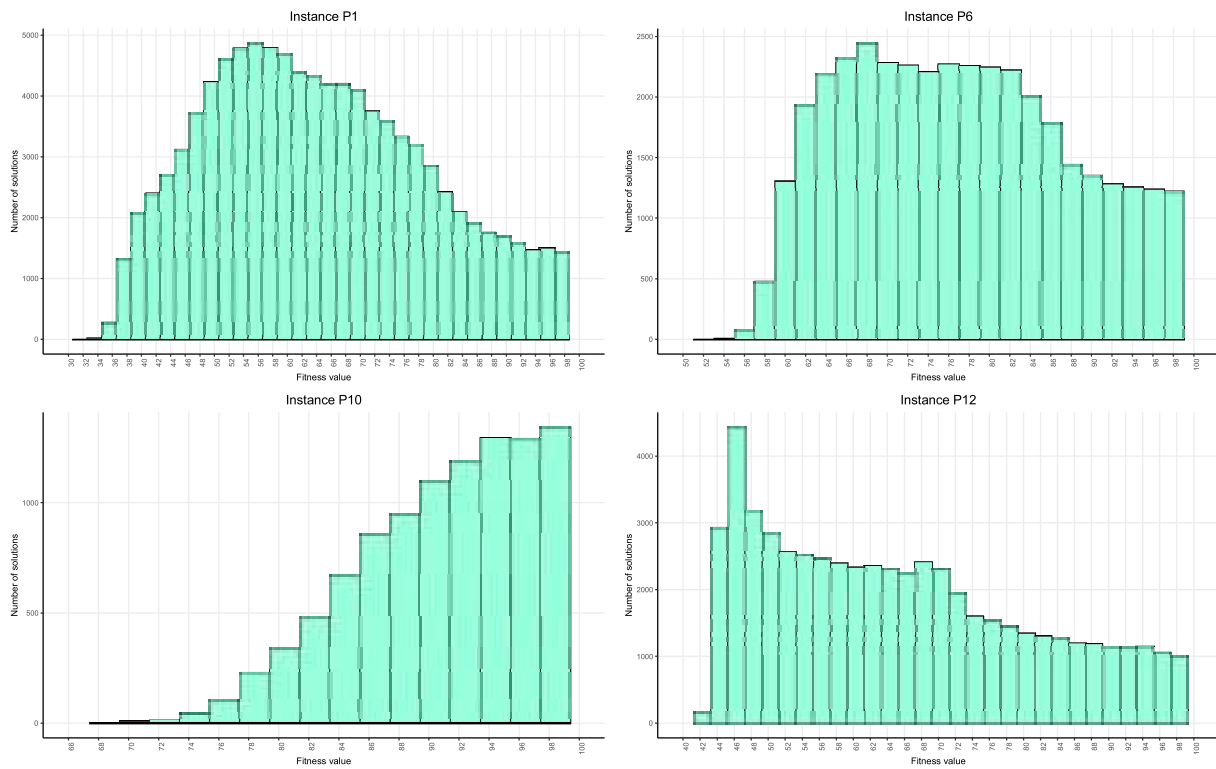


Figura 6.4: Distribución de *fitness* para 200,000 soluciones aleatorias para las instancias P1, P6, P10 y P12. Los histogramas ilustran las diferencias en el espacio objetivo para las instancias consideradas.

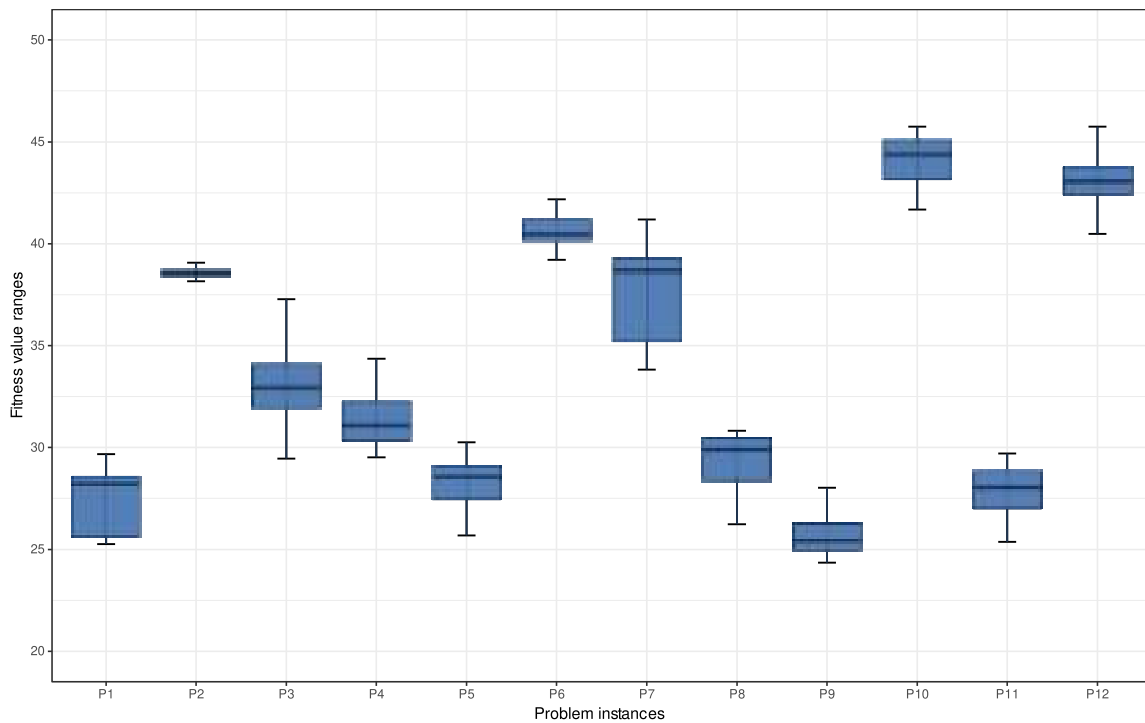


Figura 6.5: Rangos de *fitness* obtenidos por SHADE en las 12 instancias del problema.

Por último, la Tabla 6.7 presenta la clasificación media y los test de valores  $p$ -resultantes de los tests de Bonferroni-Dunn y Holm utilizando SHADE como método de control. Con respecto al test de Friedman, el resultado de su aplicación es  $\chi_F^2 = 85.62$  y su correspondiente valor  $p$  es  $3.59 \cdot 10^{-5}$ , que es inferior al nivel de significación deseado ( $\alpha = 0.01$ ), con la conclusión de que existen diferencias significativas entre el rendimiento de los algoritmos.

MMEA	Ranking	Bonferroni $p$	Holm $p$
SHADE	1.33	—	—
L-SHADE	3	0.92	0.2
NichePSO	3.08	0.79	0.2
PNA-NSGA-II	3.58	0.27	0.1
MOBiDE	4	0.09	0.05
Crowding GA	6	$< 10^{-5}$	$< 10^{-5}$
Sharing GA	7.17	$< 10^{-7}$	$< 10^{-7}$
DE/nrand/2	8	$< 10^{-9}$	$< 10^{-9}$
Clearing GA	8.83	$< 10^{-11}$	$< 10^{-11}$

Tabla 6.7: Clasificación media de los MMEAs y los  $p$ -valores resultantes utilizando SHADE como método de control.

Teniendo en cuenta las posiciones del ranking en la Tabla 6.7, podemos comprobar como el algoritmo SHADE supera a todos los demás métodos en cuanto al valor de *fitness* mínimo obtenido. Además, consigue un comportamiento más robusto en 9 de las 12 instancias en los valores medios de *fitness*.

### 6.6.2. Análisis de los mejores resultados de calibración

A continuación mostramos cómo se comporta el algoritmo al explorar el espacio de parámetros de algunas de las 12 instancias del modelo calibradas en nuestra propuesta. Utilizamos los mapas de calor obtenidos a partir de los pares de variables del modelo calibrado como herramienta para explicar mejor el rendimiento de SHADE y para compararlo con el tercer MMEA de mejor rendimiento (NichePSO) y con uno de los de peor rendimiento (Crowding GA). Aunque tanto L-SHADE como NichePSO obtuvieron una posición similar en el ranking mostrado en la Tabla 6.7, elegimos NichePSO para la comparación visual ya que L-SHADE es una extensión del mejor método (SHADE) y ofreció soluciones similares. Concluimos la sección analizando



la configuración de parámetros que SHADE obtiene para algunas instancias del problema (es decir, los parámetros calibrados para algunos de los ABM de diferentes instancias).

Las Figuras 6.6 y 6.7 exponen el conjunto de soluciones obtenidas por SHADE, NichePSO y Crowding en la calibración de las instancias P1 y P6. Tanto las Figuras 6.6 como 6.7 son proyecciones sobre dos de las variables del conjunto de parámetros calibrado (con un mínimo de 24 parámetros en la instancia P1). Los gráficos de la primera fila de la Figura 6.6 representan los valores de los parámetros impacto del *awareness* (eje X) y probabilidad de desactivación del *awareness* (eje Y) para SHADE y NichePSO, mientras que el gráfico de la segunda fila representa los mismos valores para Crowding GA. La Figura 6.7 representa el impacto del *awareness* en los medios publicitarios 7 y 15 (eje X e Y, respectivamente). En ambos casos, los círculos negros representan los modelos mejor calibrados en términos de ajuste con los datos históricos. Por último, la Figura 6.8 presenta un zoom de las soluciones obtenidas por SHADE, el MMEA de mejor rendimiento, utilizando la misma combinación de parámetros que en las Figuras 6.6 y 6.7 para apreciar mejor su diversidad en el espacio de búsqueda del problema.

Las distribuciones de los parámetros que se muestran en las Figuras 6.6, 6.7 y 6.9 son el resultado de agrupar las mejores soluciones de las 20 ejecuciones independientes para cada instancia del problema.

A pesar de que el conjunto de soluciones obtenidas por SHADE no parece cubrir el espacio de búsqueda como lo hace el NichePSO, es capaz de explorar las regiones más prometedoras (ver Figura 6.8), encontrando las mejores configuraciones de parámetros aunque el espacio de búsqueda tenga una alta complejidad a menor escala. Además, SHADE sigue mostrando una buena exploración y explotación del espacio de búsqueda en instancias de alta dimensionalidad como puede observarse, por ejemplo, en la instancia P6 de la Figura 6.7. Por su parte, NichePSO comienza a reducir su rendimiento, siendo incapaz de encontrar soluciones óptimas o localizar soluciones en las mejores regiones del espacio de búsqueda.

El Crowding GA muestra rendimiento muy pobre tanto en las instancias P1 como P6. Aunque parece ofrecer soluciones más diversas en ambas instancias, es incapaz de encontrar soluciones locales o globales de alta calidad, o de centrarse en una región cercana del espacio de búsqueda y

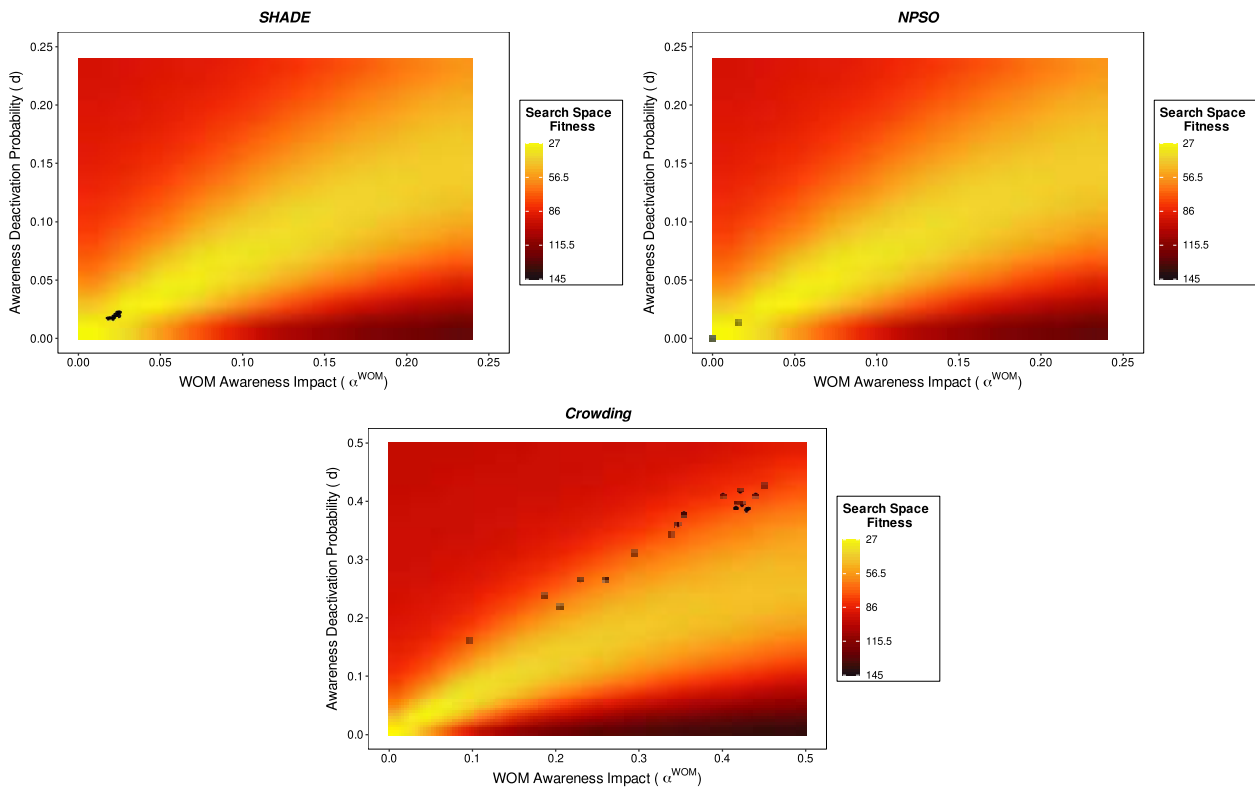


Figura 6.6: Representación de los conjuntos de soluciones de calibración obtenidos por SHADE, NichePSO y Crowding GA en los gráficos 2D del espacio de búsqueda multimodal de la instancia P1 correspondiente a los parámetros  $d$  y  $\alpha^{WOM}$ . Las soluciones encontradas por SHADE están todas en la región de menor valor de *fitness* (mejor calidad), mientras que las soluciones encontradas por NichePSO y Crowding GA sólo logran situar unas pocas (o ninguna) soluciones en la región de mejor rendimiento, respectivamente.

buscar allí soluciones óptimas. Las Figuras 6.6 y 6.7 muestran cómo la exploración del espacio de búsqueda realizada por Crowding se aleja de las regiones donde se encuentran las configuraciones de parámetros más prometedoras.

Aunque SHADE tiene un mejor balance de dispersión en comparación con NichePSO y Crowding GA, no cubre todo el espacio de búsqueda óptimo disponible para las combinaciones de parámetros analizadas en las Figuras 6.6 y 6.7. Una razón para este comportamiento podría ser la existencia de algunos factores desconocidos que limitan el rango de buenas soluciones. Sin embargo, es importante tener en cuenta que estamos mostrando sólo las relaciones de dos parámetros escogidos de entre un amplio conjunto de parámetros de calibración y que las relaciones multidimensionales pueden ser mucho más complejas y mostrar sesgos particulares para diferentes pares de parámetros. Por tanto, no podemos confirmar que la cobertura SHADE del espacio de búsqueda óptimo se comporte de forma similar para todas las combinaciones

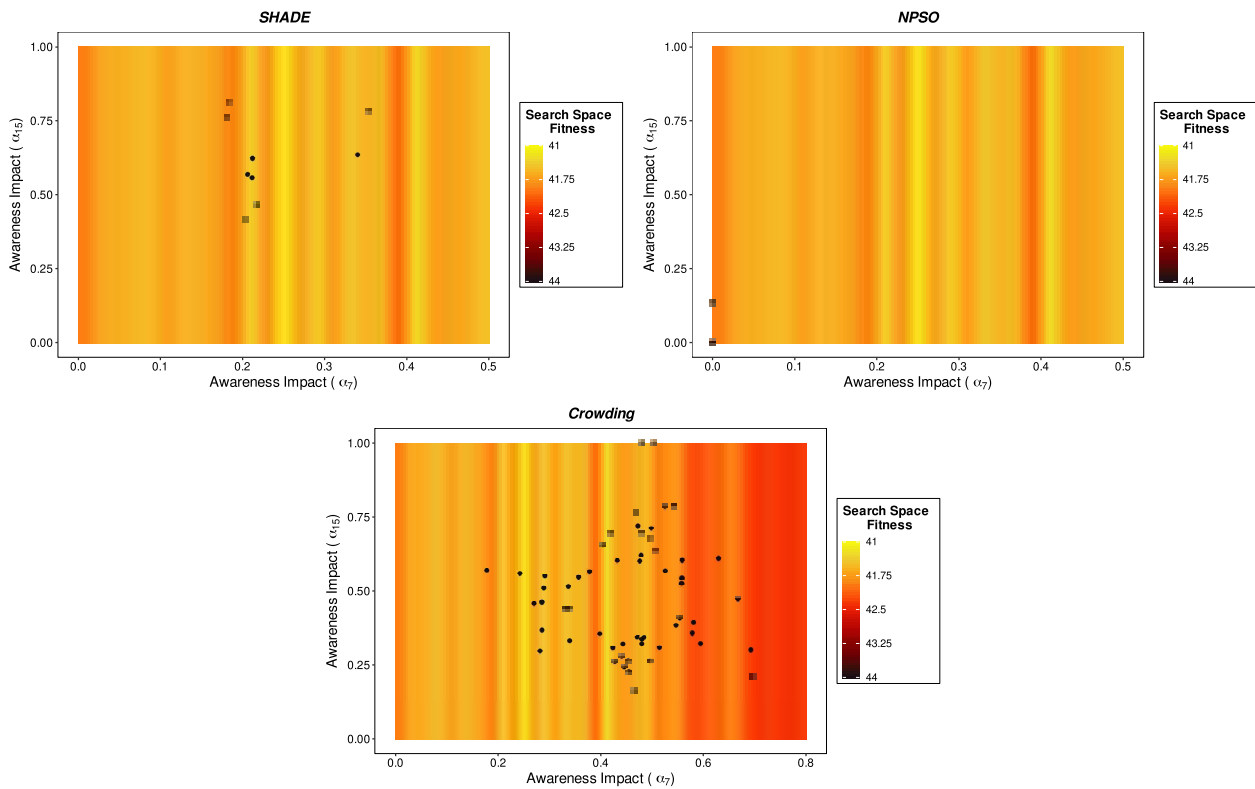


Figura 6.7: Representación de los conjuntos de soluciones de calibración obtenidos por SHADE, NichePSO y Crowding GA en los gráficos 2D del espacio de búsqueda multimodal de la instancia P6 ABM correspondiente a los parámetros  $\alpha_{15}$  y  $\alpha_7$ . Las soluciones encontradas por SHADE se encuentran todas en las regiones de mejor *fitness*, mientras que las soluciones de NichePSO están fuera de ellas. Las soluciones de Crowding GA cubren una mayor parte del espacio de búsqueda, pero no se encuentran en las mejores regiones.

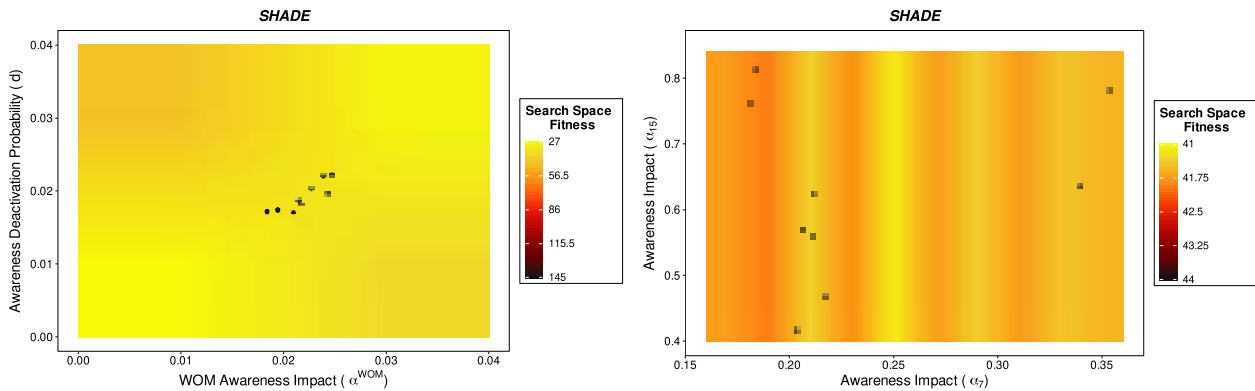


Figura 6.8: Detalle de los conjuntos de soluciones de calibración SHADE en los gráficos 2D del espacio de búsqueda multimodal de las instancias P1 y P6 mostradas en las Figuras 6.6 y 6.7.

de parámetros de todas las instancias del problema consideradas en nuestra propuesta. La Figura 6.9 nos permite observar las distribuciones de los valores de cada parámetro de calibración encontrados por el mejor MMEA encuentra para diferentes instancias del problema.

La Figura 6.9 nos permite comprender cómo cambian las combinaciones de parámetros a medida que aumenta el tamaño de las instancias. Como se puede observar, desde la instancia base (esquina superior izquierda) hasta la instancia de dimensionalidad más alta (esquina inferior derecha), los rangos de valores para parámetros como  $\alpha^{WOM}$ ,  $d$ ,  $\alpha_c$  o  $\tau_c$  cambian para cada *touchpoint*. Por un lado, este fenómeno demuestra que el algoritmo es capaz de mantener un buen rendimiento a pesar del aumento de la complejidad y el tamaño del espacio de búsqueda. Por otro, también ilustra que los cambios en la estructura del modelo afectan directamente a las relaciones entre los parámetros que se calibran, haciendo que tanto los parámetros como los algoritmos implicados en el proceso de calibración dependan del contexto del problema [Car21].

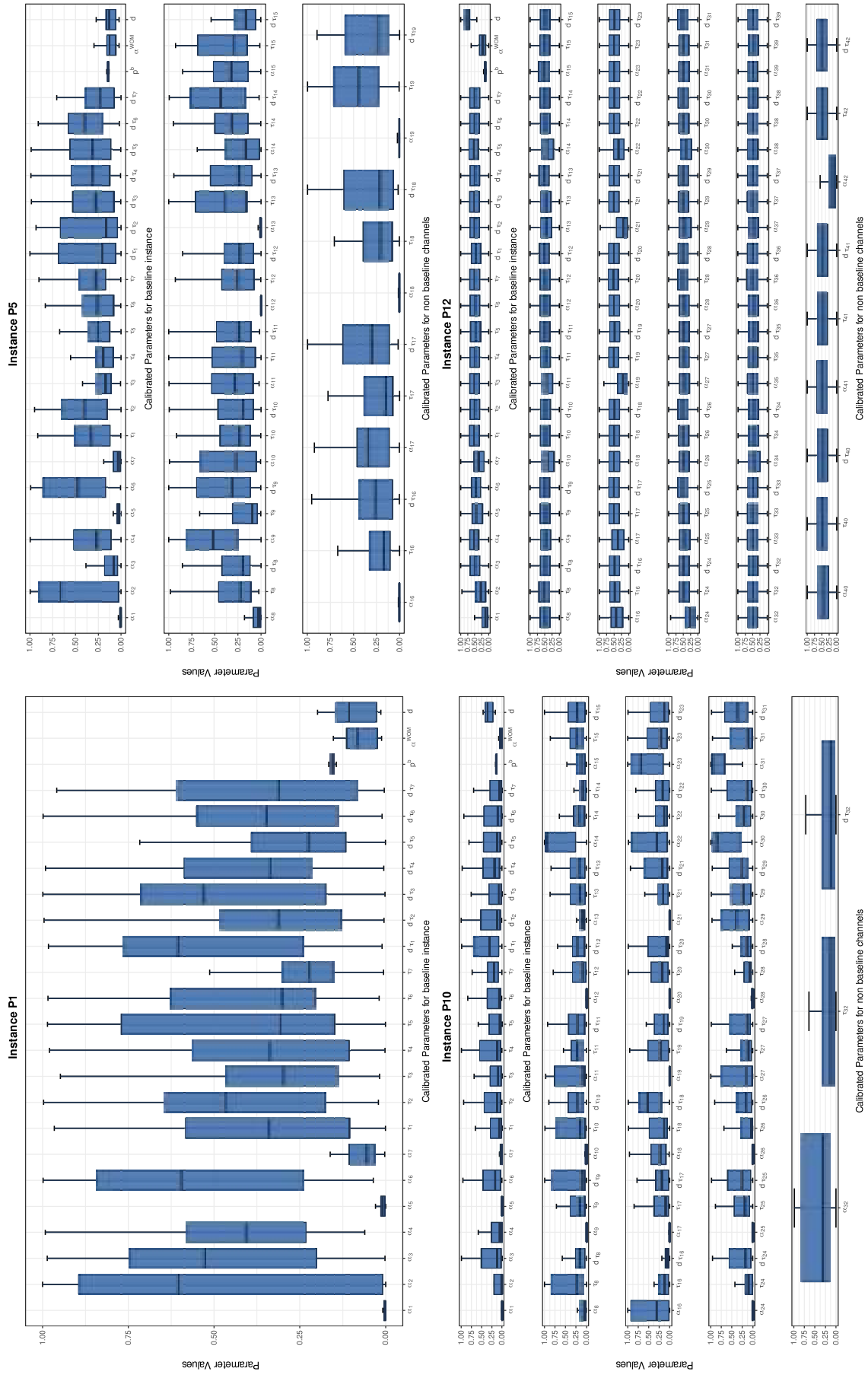


Figura 6.9: Mejores combinaciones de parámetros devueltas por SHADE en las instancias P1, P5, P10, and P12.



# MultiCalib4DEB: Optimización multimodal para calibración en modelos ecológicos poblacionales

La vida es una serie de colisiones con el futuro; no es una suma de lo que hemos sido, sino de lo que anhelamos ser.

---

José Ortega y Gasset

## 7.1. Introducción

El objetivo de este capítulo es introducir MultiCalib4DEB, un paquete software de calibración basada en MMEAs [Das+11], que se integra directamente con la herramienta *DEBtool* [Koo21; Mar+12], un marco de calibración cómodo y accesible y una amplia biblioteca de funciones y ejemplos extensamente utilizado para la simulación y calibración de modelos ecológicos poblacionales. Más concretamente, modelos basados en la teoría del balance energético dinámico (DEB, por sus siglas en inglés), que representan una metodología de simulación *individual-based* parecida a la utilizada en los ABMs. De este modo, esta nueva propuesta supone una extensión del desarrollo realizado en el capítulo anterior a un nuevo dominio de aplicación.

La teoría DEB [Koo+08; Koo09] se ha convertido en un enfoque de gran aceptación para describir adecuadamente la bioenergética del individuo en modelos ecológicos basados en poblaciones a lo largo de su ciclo vital que, además, ha sido ampliamente utilizada para representar el

metabolismo de diferentes especies [Sar+13; Sma+17]. No obstante, la necesidad de calibrar un gran conjunto de parámetros utilizando muchos conjuntos de datos simultáneamente es un obstáculo para el uso de los modelos DEB y uno de los aspectos más desafiantes de su modelización. Además, los parámetros deben calibrarse para miles de especies, por lo que el problema se ha convertido en una tarea fundamental [Lik+14].

La creación del proyecto Add-my-Pet<sup>1</sup> [Ano21] y la introducción de la toolbox *DEBtool* [Koo21; Mar+12] se han convertido en dos de las razones del éxito de la teoría DEB, ya que facilitan la simulación de modelos DEB, el manejo de sus parámetros, el estudio de nuevas especies y su calibración. Además, tanto Add-my-Pet como *DEBtool* se actualizan constantemente para mejorar la experiencia de modelado [Lik+11; Lik+14; Mar+18; Mar+19]. Por este motivo, varios autores han utilizado la toolbox como base para la calibración de los parámetros de las especies con las que trabajan [Sar+13], han propuesto varios métodos para mejorar tanto los resultados de calibración de *DEBtool* [MCM19] como su validación [Acc+20], y han contribuido al proyecto Add-my-Pet. Actualmente hay unas 3,361 especies disponibles en la base de datos del proyecto.

El módulo de calibración existente en *DEBtool* utiliza un método simplex para la calibración de los parámetros de los modelos. Concretamente, el algoritmo de Nelder-Mead [NM65]. Aunque algunos autores han propuesto procedimientos para llevar a cabo el proceso de calibración de una manera más eficiente [MCM19], el principal problema de estos métodos es que no son capaces de hacer frente a los inconvenientes del espacio de búsqueda multimodal de la calibración de los parámetros de la teoría DEB, como la existencia de diferentes conjuntos de valores de los parámetros calibrados que con frecuencia producen ajustes de similar calidad [Chi+17]. Los resultados que devuelven los métodos de calibración tradicionales se reducen a una única solución, limitando la exploración del espacio de búsqueda multimodal y las alternativas y conocimientos ofrecidos al modelador. Como hemos visto en el capítulo anterior, existen enfoques de optimización más sofisticados en el campo de la investigación operativa y la AI que abordan mejor estos problemas de optimización multimodal.

La naturaleza multimodal del problema de calibración y la existencia de interacciones no

---

<sup>1</sup>[http://www.bio.vu.nl/thb/deb/deblab/add\\_my\\_pet/](http://www.bio.vu.nl/thb/deb/deblab/add_my_pet/)



lineales entre el amplio conjunto de parámetros a calibrar suelen hacer de los algoritmos de optimización aproximados, como los algoritmos evolutivos [BFM97], el mejor enfoque para abordar los diferentes problemas de optimización multimodal [Chi+17; Fil+20]. En concreto, los MMEAs han sido reconocidos como un potente método para obtener soluciones diversas y de alta calidad en problemas grandes y complejos, mejorando la validación de los resultados en un tiempo razonable. Como hemos visto en el Capítulo 6, los MMEAs son capaces de devolver diferentes configuraciones óptimas de parámetros con ajustes similares que son igualmente preferibles entre sí, proporcionando información adicional para el análisis de sensibilidad y sobre la robustez de los modelos [Chi+17]. Gracias al reciente descubrimiento de los límites del espacio de parámetros de los modelos DEB [Lik+14] y a la aplicación de filtros, es posible mejorar la calibración de los modelos DEB mediante MMEAs. Estos límites representan los valores dentro de los cuales los parámetros de calibración pueden fluctuar, evitando resultados inviables.

Para sacar el máximo partido a la multimodalidad del espacio de soluciones de la calibración de los modelos basados en la teoría DEB hemos desarrollado *MultiCalib4DEB*, un paquete software para *MATLAB* (v.9.2 (R2017a)) que está directamente integrado en la toolbox *DEBtool* y que utiliza algoritmos de optimización evolutiva multimodal para encontrar múltiples soluciones de calibración con un rendimiento optimal desde un punto de vista local y global, y con una composición diversa para los modelos DEB. *MultiCalib4DEB* añade potentes mecanismos de calibración, análisis estadísticos y métodos de visualización a *DEBtool*, al mismo tiempo que proporciona una amplia gama de resultados, diferentes alternativas de calibración y herramientas específicas para reforzar su módulo de calibración y ayudar a los usuarios de *DEBtool* a evaluar el rendimiento de los resultados obtenidos. *MultiCalib4DEB* v.1.0 está disponible en GitHub<sup>2</sup> bajo la licencia GNU.

Nuestro paquete software aprovecha el potencial de SHADE [TF13] y su extensión L-SHADE [TF14] para la calibración de modelos basado en la teoría DEB. Tanto SHADE como L-SHADE se han aplicado con éxito en áreas como la economía [JGD21], la energía [BSA17; Gao+21; Tra+21], las matemáticas [CA+21], y la farmacia [KSC20]. *MultiCalib4DEB* ofrece una gama más amplia de resultados, diferentes alternativas de soluciones de calibración y herramientas específicas para

---

<sup>2</sup><https://github.com/JuanfranRobles/MultiCalib4DEB>

evaluar la incertidumbre de los parámetros y la validación. *MultiCalib4DEB* puede aplicarse a las bibliotecas de especies existentes en *Add-my-Pet* y *DEBtool* para calibrar los parámetros del modelo, evaluar su solidez y realizar análisis de sensibilidad visual en una sola ejecución del algoritmo. Además de mejorar el ajuste de los modelos, una de las principales ventajas de *MultiCalib4DEB* es que devuelve un conjunto óptimo y diverso de soluciones de calibración igualmente deseables que pueden ser utilizadas por el modelizador para realizar mejores análisis de sensibilidad sobre sus resultados o para lanzar nuevas calibraciones.

Comenzamos describiendo las principales funcionalidades de *MultiCalib4DEB*, el proceso de calibración de parámetros y los MMEAs integrados en la herramienta en la Sección 7.2. Finalizamos evaluando y analizando el rendimiento del módulo de calibración de *MultiCalib4DEB* en la Sección 7.3.

## 7.2. Descripción de la herramienta

### 7.2.1. Funcionalidades principales e integración con *DEBtool*

*MultiCalib4DEB* está desarrollado para calibrar eficientemente los parámetros de los modelos DEB al mismo tiempo que devuelve un conjunto óptimo de soluciones diversas utilizando MMEAs [Pre15; Won15]. Nuestra toolbox se ha integrado en *DEBtool* para mejorar su rendimiento cuando se calibran modelos con una gran cantidad de parámetros y para ayudar a los modeladores cuando analizan los resultados de sus calibraciones. Para ello se ha incluido en *MultiCalib4DEB* un conjunto de funcionalidades estadísticas y de visualización.

*MultiCalib4DEB* sigue el mismo esquema que en *DEBtool* para utilizar, ejecutar, guardar, visualizar y trabajar fácilmente con los resultados de calibración de los modelos DEB. Por tanto, es sencillo empezar a trabajar con *MultiCalib4DEB* para alguien que esté familiarizado con el módulo de calibración de *DEBtool*. *MultiCalib4DEB* está totalmente integrado en *DEBtool*.

Nombre del módulo	Descripción
<code>charts</code>	Múltiples opciones de gráficos para la validación de los resultados de calibración de <i>MultiCalib4DEB</i> , como mapas de calor y gráficos de dispersión.
<code>configuration</code>	Archivo que controla las opciones de calibración
<code>examples</code>	Ejemplos para la calibración de diferentes especies utilizando los algoritmos y opciones de calibración de <i>MultiCalib4DEB</i> . También contiene ejemplos para los módulos de visualización e informes estadísticos.
<code>functions</code>	Funciones auxiliares para los algoritmos de calibración, gráficos y estadísticas.
<code>methods</code>	Código de los algoritmos SHADE, L-SHADE y de búsqueda local que se utilizan para la calibración en <i>MultiCalib4DEB</i> .
<code>results</code>	Funciones que generan los gráficos a partir de los resultados de la calibración.
<code>statistics</code>	Funciones que generan los informes estadísticos a partir de los resultados de la calibración.
<code>utils</code>	Utilidades para guardar y generar informes de soluciones.

Tabla 7.1: Descripción de los módulos de *MultiCalib4DEB*

La toolbox también es flexible. Los usuarios pueden editar sus ajustes de calibración, desde el tiempo total de calibración hasta la solución inicial y los rangos de los parámetros a calibrar. *MultiCalib4DEB* proporciona una amplia variedad de salidas, diferentes soluciones de calibración y herramientas específicas para evaluar la incertidumbre de los parámetros y su validación, que permiten a los usuarios interactuar con el proceso de calibración a lo largo de su ejecución y también cuando termina.

### 7.2.2. Calibración de parámetros con *MultiCalib4DEB*

*MultiCalib4DEB* ha sido diseñado para imitar el proceso de calibración de parámetros de *DEBtool* para facilitar su uso a los usuarios de *DEBtool*. Así, es posible iniciar un proceso de calibración seleccionando las especies a calibrar, cargando los datos de las especies y los filtros y métodos de la función de ajuste de *DEBtool*, y configurando algunas opciones de calibración. Los usuarios pueden lanzar procesos de calibración con *MultiCalib4DEB* estableciendo la opción `mmea` en las opciones de estimación de *DEBtool* con `estim_options('method', 'mmea');`. Una

vez finalizado el proceso de calibración, el usuario puede realizar análisis de sensibilidad tanto estadísticos como visuales para extraer información de los resultados de la calibración. La lista completa de módulos que integran *MultiCalib4DEB* puede consultarse en la Tabla 7.1.

*MultiCalib4DEB* devuelve diferentes soluciones que son igualmente preferibles entre sí. Estas soluciones pueden guardarse en un archivo para utilizarlas en análisis posteriores o para iniciar un nuevo proceso de calibración para explorar un área más específica del espacio de búsqueda del problema. En este sentido, *MultiCalib4DEB* no realiza un proceso de calibración cerrado, sino que invita a un proceso de calibración iterativo en el que cualquiera de las soluciones devueltas por el método multimodal puede volver a utilizarse para lanzar nuevos procesos de calibración (por ejemplo, un usuario puede fijar algunos valores de los parámetros o modificar sus rangos mínimos y máximos). La Figura 7.1 muestra un diagrama que resume el proceso de calibración multimodal de un modelo DEB utilizando la caja de herramientas *MultiCalib4DEB*.

Aunque un proceso de calibración puede iniciarse con una configuración por defecto, el usuario puede personalizar esta configuración con las opciones de la Tabla 7.2. El usuario puede seleccionar entre dos criterios de parada de la calibración: i) un número máximo de evaluaciones (utilizando el parámetro `max_fun_evals`) y ii) un tiempo máximo (usando el parámetro `max_calibration_time`). El valor mínimo recomendado para el parámetro `max_fun_evals` es de 10,000. Una pauta para definir el valor de `max_fun_evals` es establecer 1,000 evaluaciones por cada parámetro de calibración. Por lo tanto, si una especie tiene 15 parámetros para calibrar, el número máximo de evaluaciones debe establecerse, al menos, en 15,000 evaluaciones. Por otra parte, el tiempo mínimo de calibración recomendado es de 1 hora, mientras que un criterio adecuado para definir el tiempo de calibración puede ser de 10 minutos por cada parámetro de calibración. Por defecto, el proceso de calibración se detiene cuando se alcanza uno de los criterios anteriormente mencionados, incluso si no se alcanza la convergencia. El manual de usuario de *MultiCalib4DEB*<sup>3</sup> contiene una explicación más detallada de los criterios de calibración, el tiempo máximo y otras opciones de calibración. Los Ejemplos 1 y 2 muestran cómo lanzar diferentes procesos de calibración para especies como *Clarias Gariepinus* utilizando diferentes opciones de calibración de *MultiCalib4DEB*.

---

<sup>3</sup><https://github.com/JuanfranRobles/MultiCalib4DEB>

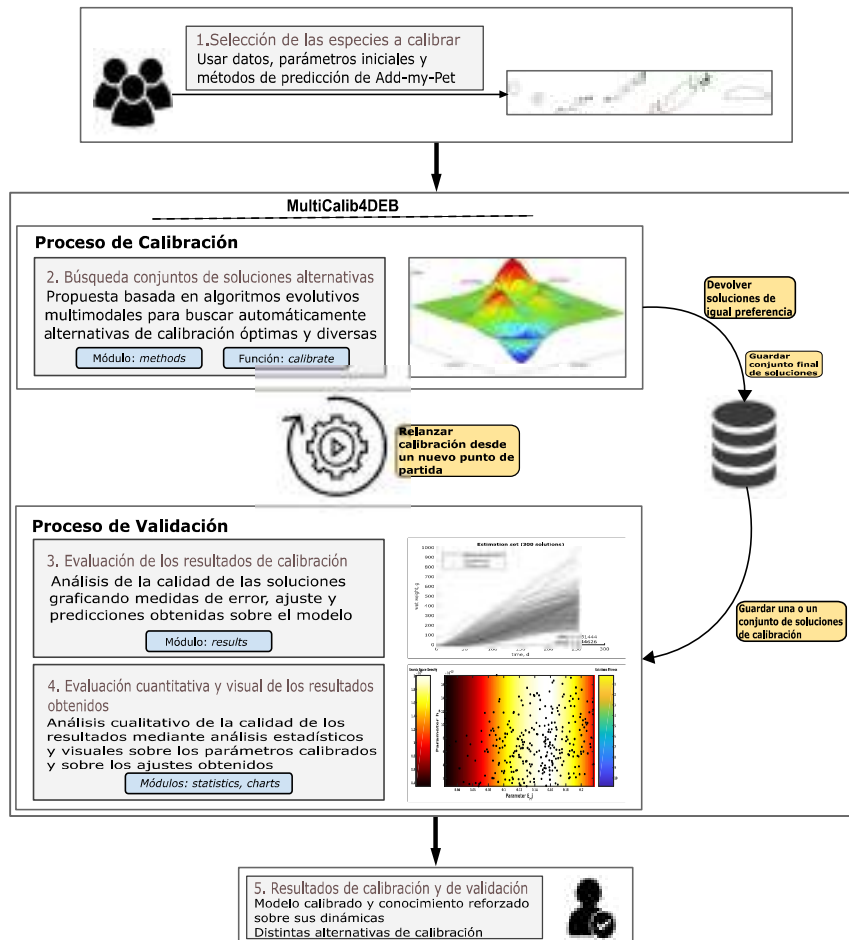


Figura 7.1: Diagrama que ilustra el funcionamiento de *MultiCalib4DEB* y sus etapas durante un proceso de calibración de un modelo DEB.

Nombre	Descripción	Valores
<code>method</code>	Define el MMEA usado para calibrar.	mm1: SHADE. mm2: L-SHADE.
<code>num.results</code> <sup>1</sup>	Define el número de individuos de las poblaciones SHADE y L-SHADE que se devolverán tras el proceso de calibración.	[50, 500]
<code>gen.factor</code> <sup>2</sup>	Es un valor porcentual utilizado para construir los rangos a partir de los cuales generar el primer conjunto de soluciones.	(0.0, 1.0)
<code>bounds.from.ind</code>	Es un parámetro que controla de dónde se toman los parámetros para inicializar la población inicial de individuos.	0: A partir de los valores de los pseudodatos. 1: A partir de valores iniciales.
<code>add.initial</code>	Controla si los parámetros de las especies (aquellos parámetros de las especies existentes en el proyecto Add-my-Pet) se introducen en la primera población del MMEA.	0: No. 1: Sí.
<code>refine.best</code>	Controla si la mejor solución se refina utilizando un procedimiento de búsqueda local después del proceso de calibración.	0: No. 1: Sí.
<code>refine.prob</code>	Es un valor probabilístico que controla sobre qué soluciones aplicar el proceso de refinamiento después de la calibración con MMEAs <sup>3</sup> .	[0.0, 1.0]
<code>max.fun.evals</code>	Define el número máximo de evaluaciones para el proceso de calibración.	[10, 000, 100, 000]
<code>max.calibration.time</code>	Define el tiempo máximo de calibración en minutos.	[60, 10,080 (una semana)]
<code>num.runs</code>	Define el número de ejecuciones independientes a realizar durante la calibración. Cada ejecución tiene una semilla independiente.	[1, 15]
<code>verbose</code>	Controla si se imprime la información sobre los mejores parámetros encontrados a través del proceso de calibración.	0: No. 1: Sí.
<code>verbose.options</code> <sup>4</sup>	Define el número de soluciones de calibración que se imprimirán. Este parámetro sólo se utiliza cuando la opción <code>verbose</code> está activada.	[10, <i>num.results</i> ]
<code>ranges</code>	Permite definir un conjunto de rangos para todos o un subconjunto de los parámetros a calibrar. Este parámetro recibe un parámetro-rango o parámetro-porcentaje para definir los valores mínimos y máximos para cada parámetro a calibrar.	[ <i>min-param</i> , <i>max-param</i> ] <sup>5</sup> o valor % <sup>6</sup> .
<code>results.output</code>	Imprime los resultados después del proceso de calibración.	Basic: No mostrar los resultados por pantalla sino un resumen de los mejores resultados en texto Best: Muestra la predicción del mejor resultado usando el formato <i>DEBtool</i> Set: Muestra la predicción del conjunto de soluciones en formato <i>DEBtool</i> .
<code>results.filename</code>	Define un nombre para el fichero de resultados.	Si el parámetro no es definido, el nombre del fichero es generado por defecto.

<sup>1</sup> Cuanto mayor sea el valor de este parámetro, más lento será el proceso de calibración.

<sup>2</sup> Un valor de 0.9 significa que, para un valor inicial de los parámetros de 1.0, el rango de generación es  $[(1 - 0.9) \cdot 1.0, 1.0 \cdot (1 + 0.9)]$ . Así, los valores generados se seleccionarán aleatoriamente del rango [0.1, 1.9].

<sup>3</sup> Un valor de 0 significa que no se aplica ningún refinamiento, mientras que un valor de 1 significa que se aplica un refinamiento sobre todas las soluciones. Si el valor de este parámetro es 0.05, el proceso de refinamiento se aplica sobre el 5% de las soluciones.

<sup>4</sup> Es preferible establecer un valor bajo para este parámetro para mostrar mejor los mejores parámetros que los algoritmos de calibración encuentran a través del proceso de calibración.

<sup>5</sup> Cuando se define un rango [*min.*, *max.*] para un parámetro, los valores generados a través del proceso de calibración no pueden superar los valores establecidos como mínimo y máximo en el rango. Cuando un valor calibrado está por debajo del mínimo su valor se convierte en el valor mínimo establecido en el rango. Cuando el valor está por encima del máximo entonces cambia al valor máximo definido en el rango del parámetro.

<sup>6</sup> Cuando se define un porcentaje para un parámetro, los rangos mínimo y máximo para estos parámetros se calculan como [*min.* = *parameter-value* · (1.0 - *valor* %), *max.* = *parameter-value* · (1.0 + *valor* %)] donde *%value* es el factor que se aplica al valor del parámetro original para generar los rangos de parámetros utilizados para inicializar la primera población de individuos.

Tabla 7.2: Descripción de las opciones de calibración de *MultiCalib4DEB*

**Ejemplo 1:** Calibra los parámetros para la especie *Clarias gariepinus* con SHADE usando los parámetros iniciales en la inicialización del algoritmo, ejecutando una única ejecución del mismo, sin rangos para los parámetros libres que se calibran, y 10,000 evaluaciones como criterio de parada. La información de la calibración no se muestra por pantalla.

```
close all;
global pets

pets = {'Clarias_gariepinus'}; % Especie cuyos parámetros calibramos
check_my_pet(pets); % Revisión de la consistencia de la especie
% Opciones de calibración por defecto para DEBtool
% (filtros, función de ajuste, ...)
estim_options('default');

calibration_options('default'); % Definición de las opciones de calibración
calibration_options('method', 'mm1'); % Calibramos con SHADE
% Paramos la calibración con 10.000 evaluaciones
calibration_options('max_fun_evals', 10000);
% Tomamos los parámetros de calibración del fichero pseudo-data de DEBtool
calibration_options('bounds_from_ind', 1);
% Añadimos los valores iniciales a la inicialización de la población del \acs{MMEA}
calibration_options('add_initial', 1);
% Desactivamos la información del proceso de calibración
calibration_options('verbose', 0);
[best, info, out, best_fvalues] = calibrate; % Lanzamos la calibración
```

**Ejemplo 2:** Calibra los parámetros con L-SHADE utilizando los valores iniciales de los parámetros del fichero ‘pars\_init’ de la especie<sup>a</sup>. A continuación, se genera la población inicial a partir de los valores de las especies aplicando un factor de generación del 20% sobre los valores base de los parámetros, no se añaden los valores iniciales de los parámetros a la inicialización del algoritmo, y se ejecutan cinco ejecuciones diferentes del algoritmo. Se refina la mejor solución al final del proceso de calibración. El criterio de parada de la calibración es alcanzar 30 minutos de ejecución. Se activa la opción que muestra la información de la calibración por pantalla, imprimiéndose los diez mejores valores de la función objetivo.

```
close all;
global pets

pets = {'Clarias_gariepinus'}; % Especie cuyos parámetros calibramos
check_my_pet(pets); % Revisión de la consistencia de la especie

% Opciones de calibración por defecto para DEBtool
% (filtros, función de ajuste, ...)
estim_options('default');

calibration_options('default'); % Opciones de calibración por defecto
calibration_options('method', 'mm2'); % Calibración con L-SHADE
% Parar tras 30 minutos de calibración
calibration_options('max_calibration_time', 30);
calibration_options('bounds_from_ind', 0); % Parámetros de especie (DEBtool)
calibration_options('gen_factor', 0.2); % Límites para rangos de calibración
% No introducir parámetros iniciales en la inicialización de L-SHADE
calibration_options('add_initial', 0);
% Correr 5 rondas de calibración con semillas aleatorias independientes
calibration_options('num_runs', 5);
% Refinar mejor solución con búsqueda local
calibration_options('refine_best', 1);
calibration_options('verbose', 1); % Activar información durante calibración
calibration_options('verbose_options', 10); % Mostrar 10 mejores soluciones
[best, info, out, best_fvalues] = calibrate; % Lanzar calibración
```

<sup>a</sup>[http://www.bio.vu.nl/thb/deb/deblab/add\\_my\\_pet/entries\\_web/Clarias\\_gariepinus/Clarias\\_gariepinus\\_res.html](http://www.bio.vu.nl/thb/deb/deblab/add_my_pet/entries_web/Clarias_gariepinus/Clarias_gariepinus_res.html)



### 7.2.3. Módulo de calibración multimodal

#### 7.2.3.1. Algoritmos multimodales

Como hemos indicado anteriormente, se han incluido dos MMEAs en la caja de herramientas de calibración *MultiCalib4DEB*: La evolución diferencial adaptativa basada en históricos de éxito (SHADE) [TF13] y su extensión L-SHADE [TF14]. Tanto el algoritmo SHADE como L-SHADE son extensiones multimodales del conocido algoritmo de DE [SP97] (más detalles en la Sección 3.3.5.4). Además de su excelente rendimiento de optimización, SHADE y L-SHADE son MMEAs adaptativos. Esto evita la necesidad de afinar cualquier parámetro y hace posible su uso sin ningún entrenamiento previo, una característica muy deseable para su inclusión en un paquete software que será empleado por usuario sin experiencia en el campo de la computación evolutiva.

Los pocos parámetros que SHADE y L-SHADE necesitan para su ejecución se fijan en *MultiCalib4DEB* siguiendo las recomendaciones de su autor [TF13; TF14]. Establecemos una probabilidad de cruce de  $CR = 0.9$  mientras que la probabilidad de mutación es de  $F = 0.5$ . Tanto para SHADE como para L-SHADE fijamos el tamaño mínimo de la memoria histórica ( $P$ ) en 100. El tamaño máximo de este parámetro lo define el usuario (para más información sobre el parámetro `num_results`, consultar la Tabla 7.2). El  $N_{init}$  y el  $N_{min}$  para el método L-SHADE se establecen en el valor de  $P$  y 5, respectivamente. A pesar de que L-SHADE reduce el número de soluciones a través de su ejecución, mantenemos el tamaño del conjunto de resultados finales a  $P$ . Por lo tanto,  $P$  es el único parámetro que no es fijo en *MultiCalib4DEB* y puede ser establecido por el usuario para controlar el número de resultados de calibración que la herramienta devuelve después del proceso de calibración.

#### 7.2.3.2. Refinamiento local de las soluciones de calibración

Para mejorar la explotación del espacio de búsqueda del problema, *MultiCalib4DEB* permite aplicar un proceso de refinamiento sobre todas, las mejores, o un conjunto aleatorio de soluciones

contenidas en la población devueltas por SHADE y L-SHADE. El usuario elige si desea aplicar el proceso de refinamiento después de la calibración con el MMEA escogida y cómo hacerlo. Para ello, se utilizan los parámetros `refine_best` y `refine_prob` mostrados en la Tabla 7.2. El parámetro `refine_best` es la opción que activa/desactiva el proceso de refinamiento sobre la mejor solución encontrada tras el proceso de calibración mientras que el parámetro `refine_prob` controla sobre cuántas soluciones aplicar el proceso de refinamiento. Por defecto, el parámetro `refine_best` está activado y el proceso de refinamiento se aplica sobre la mejor solución (es decir, aquella con menor valor de la función de ajuste). La aplicación de un proceso de refinamiento depende del nivel deseado de exploración del espacio de búsqueda del problema. Consideramos que la aplicación del refinamiento es una opción adecuada para buscar un óptimo global en el conjunto de soluciones finales. Además, esta opción no afecta significativamente al tiempo de ejecución de *MultiCalib4DEB*, ya que se limita al número máximo de evaluaciones o al tiempo total de calibración definido por el usuario.

La aplicación de un proceso de refinamiento es una novedad respecto a métodos como el propuesto por Filgueira et al. [Fil+20] ya que, además de explorar el espacio de soluciones en busca de diferentes óptimos, también se explota la vecindad de estos óptimos durante la búsqueda. El proceso de refinamiento utiliza un método simplex [NM65] basado en Nelder-Mead para explorar la vecindad de un óptimo local para minimizar el valor de su función de ajuste. El método Nelder-Mead puede utilizar diferentes criterios de parada, como la tolerancia simplex (que controla la precisión de la solución en cada iteración simplex) o un criterio de convergencia (que detiene el proceso de optimización si no se alcanza un umbral mínimo de convergencia). Siguiendo el enfoque de *DEBtool*, se establece un número fijo de evaluaciones de la función objetivo para buscar la convergencia. A continuación, se utilizan ejecuciones de un máximo de 500 pasos para el método, utilizando técnicas de continuación numérica [AG90] para reiniciar el proceso de optimización si se alcanza la convergencia.

Nombre	Descripción
<code>set_size</code>	Número de soluciones devueltas tras el proceso de calibración.
<code>solutions_set</code>	Conjunto de soluciones devueltas tras el proceso de calibración. Contiene <code>set_size</code> soluciones, cada una de ellas con un conjunto diferente de parámetros de calibración.
<code>fun_values</code>	Valores de la función de ajuste para cada solución en <code>solutions_set</code> .
<code>par_names</code>	Lista de parámetros que se seleccionan para la calibración. Es un campo de información que se utiliza en la ejecución de los módulos <code>statistics</code> , <code>results</code> y <code>charts</code> .
<code>results</code>	Campo struct de <i>MATLAB</i> que contiene tanto información general sobre las especies cuyos parámetros se calibran como información específica sobre cada solución en <code>solutions_set</code> . Todas las soluciones de calibración aparecen en este campo como subcampos con el nombre “solution_ + solución_número” e incluyen los archivos “par” y “metaPar” con los parámetros de las especies en formato <i>DEBtool</i> . A continuación, la información general se almacena debajo de la lista de soluciones y contiene los ficheros “data”, “auxData”, “txtPar”, “metaData”, “txtData”, y “weights” también en formato <i>DEBtool</i> . La información de este campo facilita el trabajo con uno o un conjunto de resultados después del proceso de calibración. Además, las soluciones en el campo <code>results</code> se puede utilizar para generar informes más tarde utilizando el módulo <code>results</code> de <i>MultiCalib4DEB</i> .

Tabla 7.3: Descripción de la información proporcionada tras el proceso de calibración en *MultiCalib4DEB*.

#### 7.2.4. Análisis y visualización de resultados

Una vez finalizado el proceso de calibración, los resultados se almacenan en un objeto *MATLAB* denominado `solutions_set`. Un usuario puede recuperar una solución correcta o un conjunto de soluciones de este objeto para volver a lanzar iterativamente un proceso de calibración con diferentes opciones de calibración si lo desea. Los campos `solutions_set` se describen en la Tabla 7.3.

A partir de la información de salida de la Tabla 7.3 el usuario también puede realizar análisis estadísticos y visuales utilizando las funciones de los módulos `statistics` y `charts`. La lista completa de métodos estadísticos y de visualización disponibles en los módulos anteriormente mencionados se enumeran a continuación:

- **Funciones estadísticas sobre los valores de la función de ajuste:** Cardinalidad, media, mínimo, máximo, desviación estándar y distancia media entre el conjunto de la función de ajuste.
- **Funciones estadísticas sobre los valores de los parámetros:** Media, desviación estándar, dispersión, mínimo, máximo, curtosis, asimetría, coeficiente bimodal y la distancia media entre los valores finales de los parámetros y sus rangos mínimo y máximo.

- **Gráficos de visualización:** Mapa de calor de la densidad (puede complementarse con un gráfico de dispersión que incluya los valores de la función de ajuste), gráfico de dispersión (simple, ponderado por el valor de la función de ajuste, o que muestre la densidad de las soluciones en el espacio de búsqueda) y gráfico de error (valores de calibración y ajuste de la función de ajuste sobre ellos).

*MultiCalib4DEB* también permite visualizar y analizar los resultados de la calibración con el módulo `charts`. Este módulo incluye dos métodos:

1. `plot_chart`: Método para generar diferentes gráficos (como gráficos de dispersión y mapas de calor) para pares de parámetros de calibración.
2. `plot_results`: Método para visualizar gráficos con los valores reales y estimados de los resultados de la calibración. Este método es capaz de trazar un resultado de predicción aislado para una sola solución de calibración o un informe completo que incluya los valores de predicción para todo el conjunto de resultados de calibración.

Enumeramos las opciones que están disponibles tanto para el método `plot_chart` como `plot_results` en la Tabla 7.4. Con el método `plot_chart`, el usuario puede explorar las relaciones entre un par de parámetros de calibración en el espacio de búsqueda. Los diferentes gráficos que ofrece este método pueden ser útiles para comprender mejor cómo se relacionan los parámetros y el espacio de búsqueda que exploran. Las opciones del método `plot_results` pueden utilizarse para representar los valores de la función de ajuste obtenidos o los errores.

Las Figuras 7.2 y 7.3 muestran un ejemplo de gráficos que pueden generarse con los métodos `plot_chart` y `plot_results`, respectivamente. En los gráficos representados en la Figura 7.2, se comparan los pares de parámetros  $zoom(z)$ , que se define como la longitud volumétrica máxima de un organismo en centímetros, y  $f_tW$ , que representa una respuesta funcional escalada para el peso húmedo en la pubertad de una especie. Por su parte, en la Figura 7.3, se muestra el conjunto de soluciones obtenido para los dos procesos de calibración de diez minutos y una hora. Los círculos azules contenidos en la Figura 7.3 representan los valores reales observados,

Método	Opciones	Descripción
plot_chart	density_hm	Genera un mapa de calor con la distribución de los valores de la función de ajuste de los resultados de <i>MultiCalib4DEB</i> en un espacio de búsqueda bidimensional. El mapa de calor final se genera para un par de parámetros de calibración.
	density_hm_scatter	Genera un mapa de calor junto con un gráfico de dispersión en el que se representan los valores del par de parámetros seleccionados.
	scatter	Genera un gráfico de dispersión con los valores de un par de parámetros a partir de los resultados de calibración devueltos por <i>MultiCalib4DEB</i> .
	weighted_scatter	Genera un gráfico de dispersión en el que los valores de los parámetros se ponderan utilizando el valor de su función de ajuste.
	density_scatter	Genera un gráfico de dispersión en el que cada punto se colorea con la densidad espacial de los puntos cercanos. La función utiliza la función de suavizado del núcleo <sup>4</sup> para calcular el valor de la densidad de cada punto.
	prediction	Genera un gráfico de predicción, el gráfico por defecto que <i>DEBtool</i> devuelve después de un proceso de calibración.
plot_results	Basic	Genera los resultados de ajuste
	Best	Genera un gráfico de ajuste a partir de los parámetros de la mejor solución.
	Set	Genera un gráfico con los ajustes de todo el conjunto de resultados junto con la media de MRE y SMSE.
	Complete	Genera los gráficos <b>Basic</b> , <b>Best</b> , y <b>Set options</b> .

Tabla 7.4: Opciones para trazar resultados y gráficos de la salida de *MultiCalib4DEB*.

mientras que las líneas grises y la línea negra muestran los valores de predicción para cada solución en el conjunto de soluciones finales y el menor (mejor) valor de la función de ajuste, respectivamente. En la calibración más larga, la estimación de todas las soluciones se solapa, como puede verse en el zoom realizado sobre el gráfico.

Los Ejemplos 3 y 4 muestran dos llamadas a funciones de código para algunos de los métodos de la Tabla 7.4. Los resultados obtenidos en los Ejemplos 3 y 4 se ilustran en las Figuras 7.4 y 7.5, respectivamente.

### Ejemplo 3: Lanzamiento del método `plot_chart` en *MultiCalib4DEB*:

```
global pets

% Especie a calibrar
pets = {'Clarias_gariepinus'};

% Probar consistencia de la especie
check_my_pet(pets);

% Cargar datos de la especie
[data, auxData, metaData, txtData, weights] = mydata_pets;

% Carga de resultados
load('solutionSet_Clarias_gariepinus_20-Apr-2021_20:42:00.mat');

% Generación de gráficos
plot_chart(solutions_set, 'density_hm', {'kap'; 'E_G'}, true, 20);
```

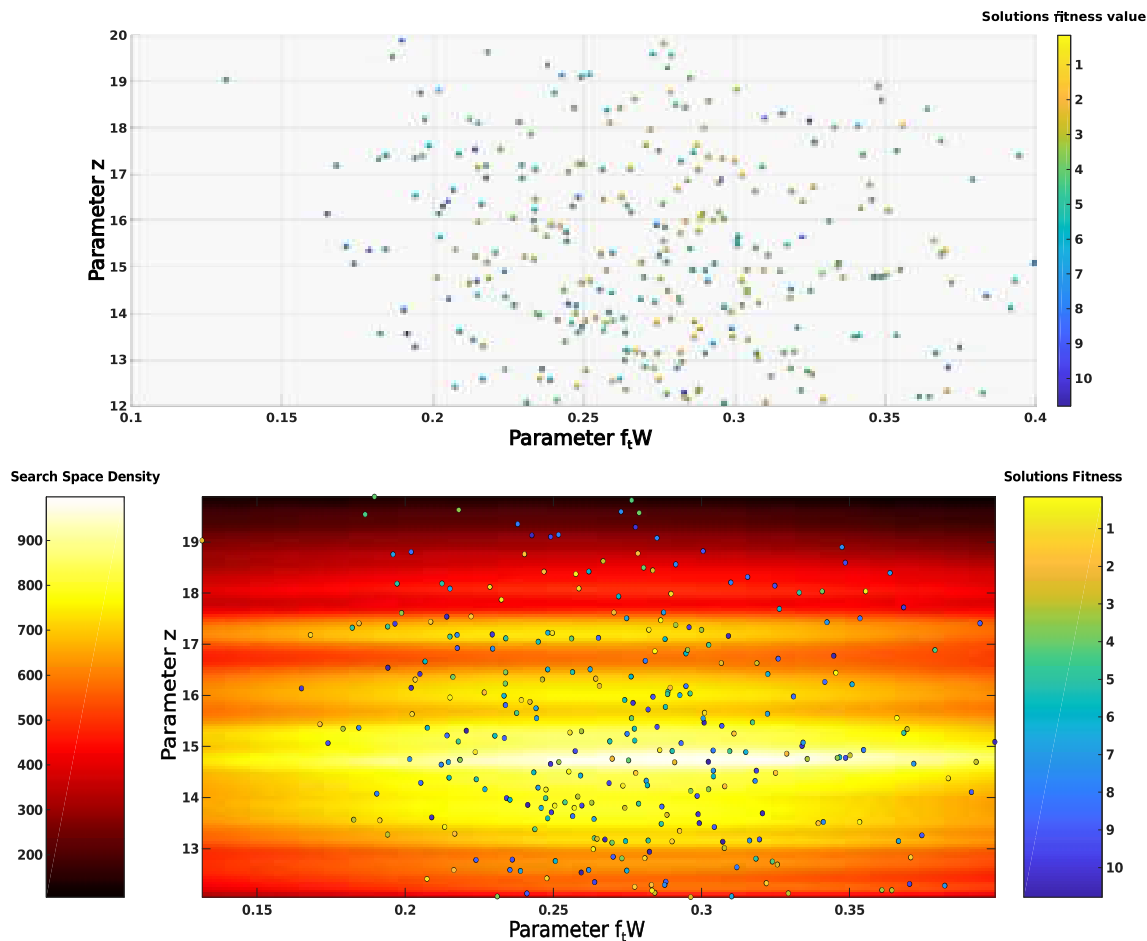


Figura 7.2: Ejemplo de los gráficos disponibles en el módulo `charts`. La figura de la parte superior muestra la salida del método `textttscatter` mientras que la figura de la parte inferior muestra la salida del método `density_hm_scatter`. Los pares de parámetros comparados en los gráficos son el factor de zoom ( $z$ ) y  $f_iW$ .  $z$  se define como la longitud volumétrica máxima de un organismo en centímetros mientras que  $f_iW$  es una respuesta funcional escalada para el peso húmedo en la pubertad de una especie.

#### Ejemplo 4: Uso del comando `plot_results` en *MultiCalib4DEB*:

```
global pets

pets = {'Clarias_gariepinus'}; % Especie a calibrar
check_my_pet(pets); % Check species consistence

% Leer datos de la especie
[data, auxData, metaData, txtData, weights] = mydata_pets;
% Cargamos los resultados
load('solutionSet_Clarias_gariepinus_20-Apr-2021_20:42:00.mat')

% Generamos los gráficos
plot_results(solutions_set, solutions_set.results.txtPar, ...,
            solutions_set.results.data, ...,
            solutions_set.results.auxData, metaData, ...,
            solutions_set.results.txtData, weights, 'Set');
```

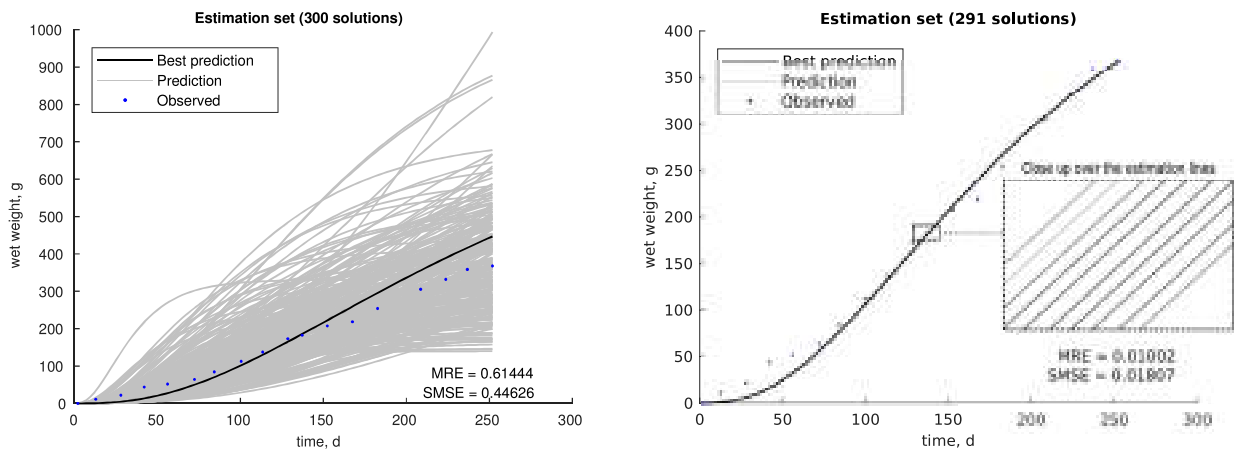


Figura 7.3: Ejemplo ilustrativo de dos procesos de calibración de diez minutos (izquierda) y una hora (derecha) sobre la especie *Clarias gariepinus*. El eje X contiene el tiempo medido en días mientras que el eje Y contiene los valores del “peso húmedo” (en gramos).

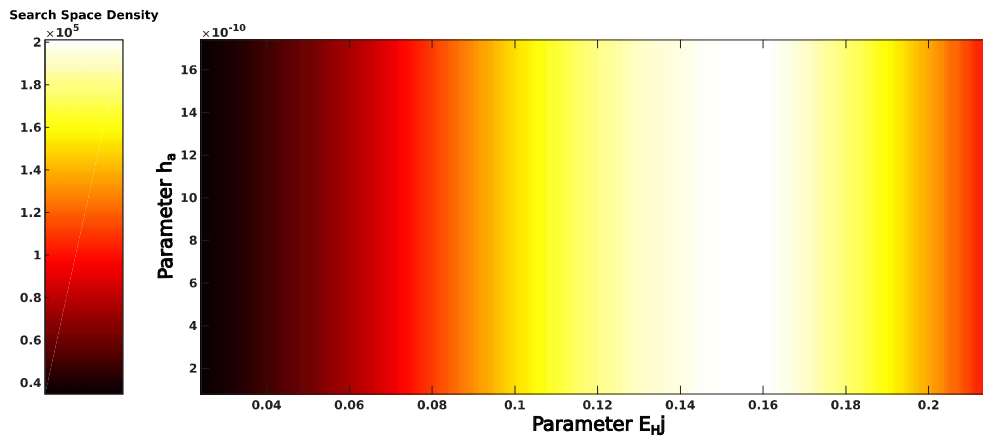


Figura 7.4: Ejemplo de la salida del módulo charts de *MultiCalib4DEB* usando como datos de entrada los resultados del Ejemplo 3.

### 7.3. Evaluación del rendimiento del módulo de calibración

*MultiCalib4DEB* devuelve diferentes resultados a lo largo del proceso de calibración. La Figura 7.3 muestra dos ejemplos de resultados de calibración de dos procesos de calibración ejecutados durante 10 minutos (panel izquierdo) y 1 hora (panel derecho) respectivamente al calibrar el modelo base de la especie *Clarias gariepinus*. La Figura 7.3 compara el “peso húmedo” (medido en gramos), que mide la masa de una especie, con el tiempo (medido en días), respectivamente. Como puede observarse, la calidad de las soluciones y las medidas de error son considerablemente

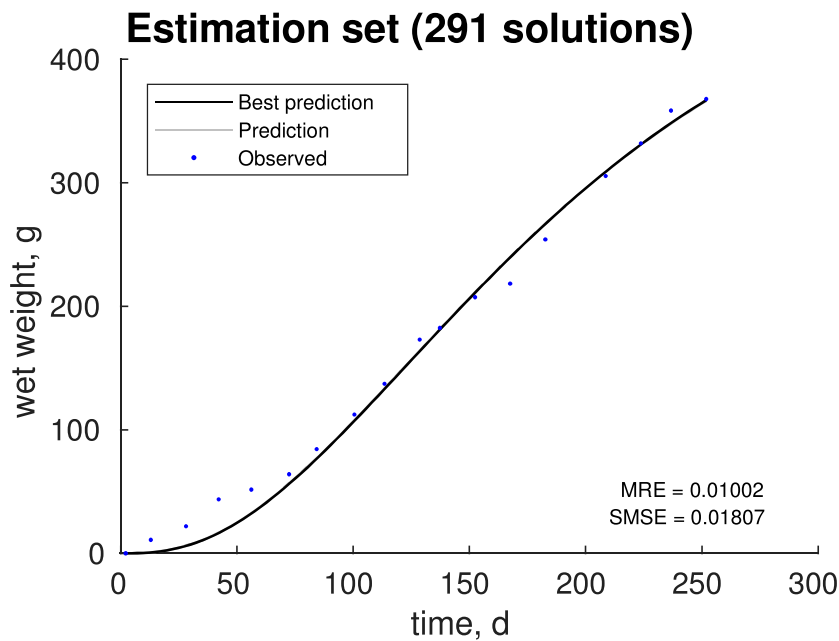


Figura 7.5: Ejemplo de la salida del módulo `plots` de *MultiCalib4DEB* usando los resultados Ejemplo 4.

mejores en el proceso de calibración de una hora en comparación con el de diez minutos porque, en este caso, el algoritmo no tiene tiempo para explorar el espacio de búsqueda del problema y converger. Dentro del conjunto final de soluciones del proceso de calibración de una hora, hay una que tiene el menor (mejor) valor de la función objetivo. Las soluciones restantes tienen valores de la función objetivo más altos que la mejor solución, pero tienen predicciones precisas y son igualmente deseables. Mientras que los métodos numéricos como Nelder-Mead se limitan a proporcionar una única solución, los MMEAs generan un conjunto de soluciones que podrían permitir a los usuarios explorar los parámetros de sus modelos DEB y su comportamiento.

Se lleva a cabo otro ejercicio numérico para comparar el rendimiento de Nelder-Mead y SHADE con refinamiento local mediante la calibración de 13 especies diferentes del repositorio de especies de laboratorio del proyecto Add-my-Pet<sup>5</sup>. La coincidencia entre los datos históricos y las predicciones se cuantifica mediante la bondad del ajuste utilizando el valor de la función de ajuste, un error MAPE que se denomina error medio relativo (MRE, por sus siglas en inglés) en la literatura de la teoría DEB, y el error medio cuadrático simétrico (SMSE, por sus siglas en inglés) [Mar+19]. El MRE evalúa las diferencias entre los datos y las predicciones de forma

<sup>5</sup>[http://www.bio.vu.nl/thb/deb/deblab/add\\_my\\_pet/species\\_list](http://www.bio.vu.nl/thb/deb/deblab/add_my_pet/species_list)



aditiva, mientras que el SMSE lo hace de forma multiplicativa. El MRE y el SMSE se calculan tal y como se definen en las Ecuaciones 7.1 y 7.2:

$$MRE = \frac{1}{n'} \sum_{i=1}^n \sum_{j=1}^{n_i} \frac{w_{ij} |p_{ij} - d_{ij}|}{w_i |d_i|}, \quad (7.1)$$

$$\text{donde } w_i = \sum_{j=1}^{n_i} w_{ij} > 0 \text{ y } d_i = \frac{1}{n_i} \sum_{j=1}^{n_i} d_{ij}$$

$$SMSE = \frac{1}{n'} \sum_{i=1}^n \sum_{j=1}^{n_i} \frac{w_{ij} (p_{ij} - d_{ij})^2}{w_i (p_i^2 + d_i^2)}, \quad (7.2)$$

$$\text{donde } w_i = \sum_{j=1}^{n_i} w_{ij} > 0 \text{ y } p_i = \frac{1}{n_i} \sum_{j=1}^{n_i} p_{ij}$$

donde  $i$  se refiere al conjunto de datos de la especie y  $j$  a un punto dado en  $i$ .  $d_{ij}$  representa los datos,  $p_{ij}$  la predicción del modelo y  $w_{ij}$  el coeficiente de peso asociado. Los coeficientes de peso cuantifican la confianza del usuario en los conjuntos de datos, así como para el dato específico. Se establecen automáticamente a  $w_{ij} = \frac{1}{n_i}$  hasta que el usuario los sobrescribe.  $n'$  es el número de conjuntos de datos con  $w_i > 0$ . MRE toma valores de 0 a infinito, mientras que SMSE toma valores de 0 a 1. En ambos casos, 0 significa que las predicciones coinciden exactamente con los datos históricos.

Para realizar una comparación justa de ambos algoritmos de calibración, el diseño experimental considera 20,000 evaluaciones de la función de ajuste como criterio de parada (15,000 para SHADE y ejecuciones de un máximo de 500 pasos para el proceso de refinamiento utilizando Nelder-Mead). Cuando SHADE termina su búsqueda, el proceso de refinamiento se aplica sobre la mejor solución encontrada. El refinamiento se ejecuta hasta que se alcanza el número máximo de evaluaciones o la función de ajuste deja de mejorar. El método Nelder-Mead simple sigue el mismo criterio de parada que el utilizado en SHADE (es decir, 20,000 evaluaciones). El método Nelder-Mead se ejecuta de forma iterativa con ejecuciones de un máximo de 500 pasos (es decir, 500 evaluaciones de la función de ajuste) siguiendo las recomendaciones del procedimiento de estimación. Así, cuando el método alcanza los 500 pasos, la búsqueda continúa a partir del (mejor) resultado obtenido anteriormente. Cuando se alcanzan las 20,000 evaluaciones o

no hay mejora, el método se detiene. Las opciones de calibración se establecen por defecto para los motores de calibración *DEBtool* y *MultiCalib4DEB*. En consecuencia, el número de soluciones a mantener durante la calibración con SHADE ( $P$ ) se establece en 200. Así, se devuelve un conjunto de 200 soluciones óptimas después del proceso de calibración para cada especie calibrada. Tanto SHADE como Nelder-Mead comienzan la calibración con los parámetros iniciales de las especies definidos en la base de datos del proyecto Add-my-Pet.

Como se puede observar en la Tabla 7.5, los valores iniciales de la función de ajuste suelen estar cerca de los valores mínimos de la función de ajuste devueltos por los algoritmos, ya que a veces la calibración comienza en la vecindad de la mejor solución. La mejora entre Nelder-Mead y SHADE se calcula como el incremento porcentual de los valores mínimos (mejores) de la función de ajuste de SHADE frente a los de Nelder-Mead. Un valor de cero significa que SHADE no mejora los resultados de Nelder-Mead, mientras que un valor mayor que cero significa que SHADE supera a Nelder-Mead (por ejemplo, un valor de 0.05 significa una mejora del 5% de SHADE sobre Nelder-Mead). La lista completa de las especies que seleccionamos se muestra en la Tabla 7.5 incluyendo su modelo tipificado<sup>6</sup>, el número de parámetros a calibrar y el valor inicial de la función de ajuste.

La Tabla 7.5 muestra que SHADE es capaz de calibrar con éxito especies con diferentes complejidades. SHADE supera a Nelder-Mead según el valor final de la función de ajuste en 9 de las 13 especies, mientras que Nelder-Mead alcanza el mismo valor final de la función de ajuste que SHADE en las otras cuatro especies. El algoritmo SHADE es capaz de reducir el valor de la función de ajuste de las especies calibradas entre un uno (por ejemplo, para las especies *Dipodomys herrmanni*, *D. merriami* y *Lepus timidus*) y un cuatro por ciento (por ejemplo, para las especies *Dipodomys deserti* y *Magallana gigas*) con respecto a los resultados de Nelder-Mead. Además, el algoritmo SHADE devuelve 200 soluciones óptimas y diversas para cada especie en lugar de la única solución que devuelve Nelder-Mead. Por otro lado, los valores medios de la función de ajuste para el conjunto de 200 soluciones de cada especie se acercan a los valores mínimos de la función de ajuste, lo que demuestra que cualquiera de las soluciones

---

<sup>6</sup>Un modelo tipificado es la clase de las especies que la teoría DEB permite para la construcción de modelos DEB. Cada modelo tipificado corresponde a una especie o grupo taxonómico similar. Consultar [http://www.debtheory.org/wiki/index.php?title=Typified\\_models](http://www.debtheory.org/wiki/index.php?title=Typified_models) para más información sobre los modelos tipificados.

obtenidas es óptima y puede considerarse como una solución de calibración factible. SHADE también consigue los mejores valores de SMSE y MRE en siete especies. Además, SHADE supera a Nelder-Mead al abordar la especie más compleja en términos de número de parámetros (*Magallana gigas*, 18 parámetros a calibrar) según el valor de la función de ajuste final, SMSE y MRE.

Especies (Modelo tipificado)	Params. a calibrar	Nelder-Mead (1 solución)			SHADE (conj. 200 soluciones)				
		Ajuste inicial	Mejor ajuste	SMSE	MRE	Mejor ajuste	Ajuste medio del conjunto	SMSE	MRE
<i>Cyclops vicinus</i> (abp)	9	6.3819	0.4848	0.1714	0.198	0.4848 (0.0)	0.485	<b>0.1713</b>	<b>0.1974</b>
<i>Lobatus gigas</i> (abj)	9	0.0623	0.0623	0.0381	0.0365	0.0623 (0.0)	0.0623	<b>0.0378</b>	<b>0.0364</b>
<i>Dipodomys deserti</i> (stx)	10	0.2326	0.2326	0.021	0.0189	<b>0.2234</b> (0.04)	0.2266	<b>0.0185</b>	<b>0.0169</b>
<i>Dipodomys herrmanni</i> (stx)	10	0.3289	0.3289	<b>0.042</b>	0.024	<b>0.3237</b> (0.016)	0.3241	0.0439	<b>0.0228</b>
<i>Dipodomys merriami</i> (stx)	10	0.3167	0.3135	0.018	0.017	<b>0.3102</b> (0.011)	0.3113	<b>0.0165</b>	<b>0.0154</b>
<i>Lepus timidus</i> (stx)	10	0.3068	0.3066	0.013	0.013	<b>0.3028</b> (0.0124)	0.303	<b>0.0075</b>	<b>0.0073</b>
<i>Clarias gariepinus</i> (abj)	11	0.1531	0.15293	0.019	0.013	<b>0.1516</b> (0.009)	0.1525	<b>0.0181</b>	<b>0.011</b>
<i>Heterobranchius longifilius</i> (abj)	11	0.12205	0.1183	0.0161	0.0131	0.1183 (0.0)	0.1404	0.0161	0.0131
<i>Porcellio scaber</i> (std)	12	0.1887	0.0988	0.0472	0.0522	0.0988 (0.0)	0.1294	0.0472	0.0522
<i>Pleurobrachia bachei</i> (abj)	12	1.3614	1.3503	0.24	<b>0.182</b>	<b>1.3174</b> (0.0244)	1.3182	<b>0.2364</b>	0.1838
<i>Homo sapiens</i> (stx)	12	0.28	0.28	0.057	0.051	<b>0.2794</b> (0.0022)	0.2794	<b>0.0547</b>	<b>0.0495</b>
<i>Asterias rubens</i> (abj)	15	0.98	0.8443	0.1431	0.1432	<b>0.8442</b> (0.0002)	0.8442	<b>0.143</b>	<b>0.1431</b>
<i>Magallana gigas</i> (asj)	18	16.7519	15.527	0.463	0.405	<b>14.8837</b> (0.0414)	15.0611	<b>0.4553</b>	<b>0.3965</b>

Tabla 7.5: Comparación entre los resultados de calibración de Nelder-Mead y SHADE para diferentes especies. La tabla contiene el número de parámetros a calibrar y la función de ajuste inicial para cada especie. También se incluyen los valores de la función de ajuste mínima (mejor), SMSE y MRE tanto para el algoritmo Nelder-Mead como para el SHADE. El promedio de la función de ajuste de las 200 soluciones que devuelve SHADE se muestra a la derecha de los mejores valores de la función de ajuste. Los mejores valores de la función de ajuste, el SMSE y el MRE aparecen en negrita.

## Parte III

### Observaciones finales



---

# Conclusiones y trabajos futuros

Los hombres geniales empiezan grandes obras, los hombres trabajadores las terminan.

---

Leonardo da Vinci

## 8.1. Conclusiones

La principal motivación de esta tesis doctoral es proponer técnicas de diseño de modelos de simulación social haciendo uso de algoritmos multiobjetivo y multimodales. En este sentido, se han combinado diferentes técnicas de ciencia de datos e AI como las metaheurísticas y los ABMs para abordar y resolver problemas con un alto nivel de complejidad. La presente tesis reúne una serie de contribuciones con carácter multidisciplinar dado que son transversales a la Ciencia de Datos e AI, el marketing y la biología.

La primera contribución abarca la extensión del modelo *CONSUMAT*, un ABM consolidado en el campo de la simulación social, al que incorporamos un proceso de *awareness* (que limita el conocimiento que los agentes consumidores tienen de los productos ofertados en el mercado) y un mecanismo de difusión de innovaciones/reconocimiento de marca entre agentes a través de la SN que los interconecta basado en WOM. De este modo, los agentes del modelo *CONSUMAT* extendido no disponen de un conocimiento completo de todos los productos y marcas ofertados en el mercado, sino que lo adquieren gradualmente a través de la comunicación con el resto de consumidores dentro de la SN.

Una vez desarrollado, testado y validado, se procedió a la explotación del mismo. Para ello, creamos un marco para la generación de campañas basadas en VM usando el modelo extendido como base y GAs para seleccionar una cierta cantidad de individuos influyentes, utilizando para ello distintas métricas locales de SNs. Los *influencers* actúan como difusores de productos en el mercado simulado y su correcta identificación y selección cumplen el objetivo definido en la explotación del modelo; maximizar la venta de los productos de una determinada marca y su reconocimiento en el mercado.

La tercera contribución de la tesis pretendió ir un paso más allá en la explotación del modelo desarrollado en la primera contribución. Para ello, decidimos hacer uso de algoritmos EMO para mejorar la calidad de los *influencers* seleccionados como difusores de productos en las campañas de VM considerando dos objetivos en conflicto: El coste de la contratación de dichos *influencers* y el beneficio generado por las campañas de VM realizadas. Para abordar este problema hicimos uso de los dos algoritmos EMO más utilizados en el área de la optimización multimodal: NSGA-II y MOEA/D. Ambos algoritmos fueron enfrentados a una adaptación multiobjetivo de un GA estacionario y a un enfoque greedy. Para medir la calidad de los resultados de cada algoritmo y su capacidad para afrontar diferentes configuraciones del modelo, se realizaron pruebas sobre seis topologías de SN: Una red aleatoria, una red regular, una red de mundo pequeño, una red libre de escala, una red real basada en intercambios de *emails* y, por último, una red bimodal obtenida de una aplicación *Freemium* de un juego online. Asimismo, para demostrar la adaptabilidad de los algoritmos a ABMs aumentamos la complejidad y tamaño de cada SN, pasando de los 1,000 nodos y 5,000 enlaces de la red regular a los 20,000 nodos y 481,000 enlaces de la red bimodal. El resultado de esta contribución mostró un claro dominio de las soluciones devueltas por *NSGA-II* y *MOEA/D* sobre los dos algoritmos restantes y un mejor desempeño de *MOEA/D* frente a *NSGA-II* en las diferentes redes y configuraciones utilizadas sobre el modelo *CONSUMAT* extendido.

En una cuarta y quinta contribución nos enfocamos en el uso de MMEAs para la calibración de ABMs y en modelos poblacionales basados en la teoría DEB dentro del área de la ecología, respectivamente. En primer lugar, propusimos un marco general multimodal de calibración de ABMs para el cual consideramos una amplia variedad de MMEAs. Los resultados fueron claros,



*SHADE*, *L-SHADE* y *Niche-PSO* fueron los algoritmos de mejor desempeño. A lo largo de la experimentación pudimos comprobar el magnífico rendimiento de *SHADE*, siendo significativamente mejor que el resto de algoritmos en la mayoría de las instancias de experimentación. El estudio realizado confirmó la utilidad de los MMEAs para dar a los modeladores un conjunto de soluciones diversas con un ajuste de calidad frente a una única solución global al problema. Estas soluciones resultaban de igual preferencia para el modelador dado que representaban soluciones de calibración factibles. Esto permitió, por un lado, tomar mejores decisiones y, por otro, tener una mejor visión del entorno de soluciones del modelo a calibrar.

Finalmente, aprovechamos los conocimientos adquiridos a partir de la calibración multimodal de ABMs con MMEAs y los extendimos a la calibración de modelos basados en DEB, una metodología que modela individuos en poblaciones ecológicas. Para ello, desarrollamos el paquete software MultiCalib4DEB que integra, por una parte, los beneficios de la optimización multimodal para obtener soluciones de calibración óptimas y que diversas y, por otra, una serie de herramientas de análisis y visualización de los resultados de calibración que permiten a los usuarios estudiarlos de forma efectiva. La herramienta desarrollada se integró dentro de la herramienta *DEBtool*, un software ya existente para la simulación de dinámicas de balance energético y otros procesos biológicos asociados, de gran importancia en biología. MultiCalib4DEB fue utilizado para la calibración de 13 especies diferentes y con complejidades distintas, obteniendo un gran desempeño en todas ellas y superando al algoritmo utilizado para la calibración de modelos DEB en el estado del arte.

En base a los resultados obtenidos en cada una de las contribuciones mencionadas anteriormente, podemos concluir que los resultados del trabajo realizado cumplen con los objetivos definidos en la presente tesis doctoral. Además, las contribuciones realizadas han supuesto una aportación notable al estado del arte de distintas disciplinas como la AI, el marketing y la biología.

## 8.2. Trabajos futuros

En lo referente a trabajos futuros, se plantea el escalado de los modelos desarrollados durante esta tesis doctoral. Comprender y, sobre todo, ser capaces de representar y simular los comportamientos emergentes en poblaciones del orden de millones de individuos es uno de los retos más desafiantes, ya que requiere la ejecución de simulaciones masivas sobre ABMs. Estos procesos de simulación requerirán de computación intensiva y del uso de tecnologías y herramientas de Big Data para poder procesar la ingente cantidad de información para definir los modelos y las ejecución de sus dinámicas.

Ser capaces de extender los modelos de una granularidad mucho mayor, además de acercar los ABMs a los modelos de consumo reales, ayudará a entender mejor cómo surgen y evolucionan las interacciones sociales en el mercado, cómo se producen las decisiones de los consumidores y, ante todo, reducirá los sesgos producidos por las limitaciones que implica no poder hacer un representación 1:1 del consumidor real y del agente virtual. Así, es imprescindible el escalado de ABM para hacer frente a la dimensión de sistemas complejos para la obtención de teorías válidas derivadas de la simulación. Afortunadamente, la potencia computacional ya disponible y el desarrollo de protocolos ABMs en entornos de alto rendimiento pueden facilitar la labor [Kan15; Zha+16; SBR18].

Con respecto al aprovechamiento de los ABMs para la selección de *influencers* en campañas de VM, un trabajo futuro podría consistir en utilizar enfoques distintos, como el *seeding* secuencial [Jan+17], que se podrían aplicar y comparar con nuestro enfoque. Además, se podrían incorporar y combinar las medidas para clasificar los nodos de la SN utilizadas en nuestro enfoque con otras medidas globales para tratar de mejorar la selección de *influencers* y los resultados obtenidos por los algoritmos EMO.

En lo referente al uso de MMEAs para la calibración de ABMs, dado que seleccionar las mejores configuraciones de parámetros en problemas de alta dimensionalidad puede resultar una tarea compleja para un modelador, podría ser interesante utilizar algunas herramientas cuantitativas y cualitativas para ayudar a los modeladores a la hora de seleccionar las configuraciones de

parámetros del conjunto final de soluciones. Alternativamente, se podría explorar el uso de algoritmos coevolutivos para resolver problemas de optimización real de alta dimensión en este campo [LY12; MLY14; Kaz+19]. Otros trabajos futuros asociados a esta contribución podría enfocarse en explorar de forma exhaustiva el espacio de parámetros calibrados para detectar posibles cambios en la dinámica de los ABMs con el fin de probar los beneficios de usar MMEAs y mejorar la validación sus resultados [Fag+19].

### 8.3. Publicaciones

Por último, esta sección presenta las publicaciones científicas realizadas en el curso de esta tesis doctoral.

#### Artículos publicados en revistas indexadas en el JCR-SCI (2):

1. Robles, J. F., Chica, M., & Cordon, O. (2020). **Evolutionary Multiobjective Optimization to Target Social Network Influentials in Viral Marketing**. *Expert systems with applications*, 147, 113183. DOI: 10.1016/j.eswa.2020.113183 (JCR 2020; factor de impacto: 6.954; Cat.: OPERATIONS RESEARCH & MANAGEMENT SCIENCE; Pos.: 23/139; Q1). Relacionado con el Capítulo 5.
2. Robles, J. F., Bermejo, E., Chica, M., & Cordon, O. (2021). **Multimodal Evolutionary Algorithms for Easing the Complexity of Agent-Based Model Calibration**. *Journal of Artificial Societies and Social Simulation*. DOI: 10.18564/jasss.4606 (JCR 2021; Factor de impacto: 3.506; Cat.: SOCIAL SCIENCES, INTERDISCIPLINARY; Pos.: 20/111; Q1). Relacionado con el Capítulo 6.

#### Artículos publicados en revistas no indexadas en el JCR-SCI (1):

1. Robles, J. F., Chica, M., Filgueira, R., Agüera, A., & Damas, S. (2023). **MultiCalib4DEB: A Toolbox Bringing Multimodal Optimisation for DEB Parameters Calibration**. arXiv: 2301.07548 [cs.NE]. DOI: 10.48550/arXiv.2301.07548. Relacionado con el Capítulo 7.

**Artículos publicados en congresos internacionales (2):**

1. Robles, J. F., Chica, M., & Cordon, Ó. (2016, Julio). **Incorporating Awareness and Genetic-based Viral Marketing Strategies to a Consumer Behavior Model**. In 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 2016, pp. 5178-5185, doi: 10.1109/CEC.2016.7748346. Relacionado con los Capítulos 4 y 5.
2. Robles, J. F., Chica, M., Filgueira, R., Agüera, A., Damas, S. (2021, Mayo). **MultiCa-lib4DEB: A Toolbox Bringing Multimodal Optimisation for DEB Parameters Calibration**. In Seventh International Symposium and Thematic School on DEB Theory for metabolic organization (DEB2021). Relacionado con el Capítulo 7.

**Artículos publicados en congresos nacionales (1):**

1. Robles, J. F., Chica, M., y Cordon, Ó. (2016, Septiembre). **Algoritmos Genéticos para Estrategias de Marketing en un Modelo de Comportamiento de Consumo**. En Actas del XI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2016), pp. 69-80, Ediciones Universidad de Salamanca. Relacionado con los Capítulos 4 y 5.

## 8.4. Agradecimientos

Esta tesis doctoral ha sido apoyada por el Ministerio de Ciencia, Innovación y Universidades, la Junta de Andalucía, la Universidad de Granada, la Agencia Estatal de Investigación (AEI) y los Fondos Europeos de Desarrollo Regional (FEDER) con las ayudas EXASOCO (PGC2018-101216-B-I00), AIMAR (A-TIC-284-UGR18) y SIMARK (PY18-4475).

# Bibliografía

- [Aak97] Jennifer L. Aaker. «Dimensions of Brand Personality». En: *Journal of Marketing Research* 34:3 (1997), págs. 347-356. eprint: <https://doi.org/10.1177/002224379703400304>.
- [Acc+20] Chiara Accolla et al. «DEB-tox and Data Gaps: Consequences for individual-level outputs». En: *Ecological Modelling* 431 (2020).
- [AG90] Eugene L. Allgower y Kurt Georg. *Numerical Continuation Methods: An Introduction*. Berlin, Heidelberg: Springer-Verlag, 1990.
- [An12] Li An. «Modeling human decisions in coupled human and natural systems: Review of agent-based models». En: *Ecological Modelling* 229 (2012). Modeling Human Decisions, págs. 25-36.
- [Ano21] Anonymous. *Add-my-Pet*. [https://www.bio.vu.nl/thb/deb/deblab/add\\_my\\_pet/](https://www.bio.vu.nl/thb/deb/deblab/add_my_pet/). [Online; accessed 10-June-2021]. 2021.
- [Art09] W Brian Arthur. «Complexity and the economy». En: *Handbook of Research on Complexity*. Edward Elgar Publishing, 2009.
- [BA99] Albert-László Barabási y Réka Albert. «Emergence of scaling in random networks». En: *Science* 286:5439 (1999), págs. 509-512.
- [Bae07] Walter RJ Baets. *Complexity, learning and organizations: A quantum interpretation of business*. Routledge, 2007.
- [BBC08] Julio Banga y Eva Balsa-Canto. «Parameter estimation and optimal experimental design». En: *Essays in Biochemistry* 45 (sep. de 2008), págs. 195-210.
- [BCZ10] Jonathan D Bohlmann, Roger J Calantone y Meng Zhao. «The effects of market network heterogeneity on innovation diffusion: An agent-based modeling approach». En: *Journal of Product Innovation Management* 27:5 (2010), págs. 741-760.

- [BD13] Sunith Bandaru y Kalyanmoy Deb. «A parameterless-niching-assisted bi-objective approach to multimodal optimization». En: *2013 IEEE Congress on Evolutionary Computation*. 2013, págs. 95-102.
- [BDT12] Aniruddha Basak, Swagatam Das y Kay Chen Tan. «Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection». En: *IEEE Transactions on Evolutionary Computation* 17:5 (2012), págs. 666-685.
- [BE02] F. van den Bergh y A.P. Engelbrecht. «A new locally convergent particle swarm optimiser». En: *IEEE International Conference on Systems, Man and Cybernetics*. Vol. 3. 2002, 6 pp. vol.3-.
- [BEB02] Riaan Brits, Andries P Engelbrecht y F Van den Bergh. «A niching particle swarm optimizer». En: 2 (2002), págs. 692-696.
- [BFM97] Thomas Back, David B. Fogel y Zbigniew Michalewicz. *Handbook of Evolutionary Computation*. 1.<sup>a</sup> ed. GBR: IOP Publishing Ltd., 1997.
- [BJP19] Suman Banerjee, Mamata Jenamani y Dilip Kumar Pratihar. «ComBIM: A community-based solution approach for the Budgeted Influence Maximization Problem». En: *Expert Systems with Applications* 125 (2019), págs. 1-13.
- [BL13] Jaume Bacardit y Xavier Llorà. «Large-scale data mining using genetics-based machine learning». En: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3:1 (2013), págs. 37-61.
- [Blo15] Per Block. «Reciprocity, transitivity, and the mysterious three-cycle». En: *Social Networks* 40 (2015), págs. 163-173.
- [Boe+15] Riccardo Boero et al. *Agent-based models of the economy: from theories to applications*. Springer, 2015.
- [Bon02] Eric Bonabeau. «Agent-Based Modeling: Methods And Techniques for Simulating Human Systems». En: *Proceedings of the National Academy of Sciences of the United States of America* 99 Suppl 3 (jun. de 2002), págs. 7280-7.

- [Bon87] Phillip Bonacich. «Power and Centrality: A Family of Measures». En: *American Journal of Sociology* 92:5 (1987), págs. 1170-1182.
- [BS15] Federico Bianchi y Flaminio Squazzoni. «Agent-based models in sociology». En: *Wiley Interdisciplinary Reviews: Computational Statistics* 7:4 (2015), págs. 284-306.
- [BSA17] Partha P Biswas, Ponnuthurai N Suganthan y Gehan AJ Amaratunga. «Optimal placement of wind turbines in a windfarm using L-SHADE algorithm». En: *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2017, págs. 83-88.
- [Bul+09] Tom Bulatewicz et al. «Calibration of a crop model to irrigated water use using a genetic algorithm». En: *Hydrology and Earth System Sciences* 13:8 (2009), págs. 1467-1483.
- [BV03] Lisa Brouwers y Harko Verhagen. «Applying the consumat model to flood management policies». En: (ene. de 2003).
- [BVL16a] Guus ten Broeke, George van Voorn y Arend Ligtenberg. «Which Sensitivity Analysis Method Should I Use for My Agent-Based Model?» En: *Journal of Artificial Societies and Social Simulation* 19:1 (2016), pág. 5.
- [BVL16b] Guus ten Broeke, George van Voorn y Arend Ligtenberg. «Which Sensitivity Analysis Method Should I Use for My Agent-Based Model?» En: *Journal of Artificial Societies & Social Simulation* 19:1 (2016), pág. 5.
- [Byr02] David Byrne. *Complexity theory and the social sciences: An introduction*. Routledge, 2002.
- [CA+21] K.B. Cantún-Avila et al. «Optimizing functionals using Differential Evolution». En: *Engineering Applications of Artificial Intelligence* 97 (2021), pág. 104086.
- [Cai08] Peter M Cain. «Limitations of conventional marketing mix modelling». En: *Admap Magazine, (493)* (2008), págs. 48-51.
- [Cao99] Junwei Cao. «Evaluation of advertising effectiveness using agent-based modeling and simulation». En: *Proceedings of 2nd UK Workshop of SIG on Multi-Agent Systems. Bristol, UK, December 6*. Vol. 7. 1999.

- [Car21] Ernesto Carrella. «No Free Lunch when Estimating Simulation Parameters». En: *Journal of Artificial Societies and Social Simulation* 24:2 (2021), pág. 7.
- [CBM20] Ernesto Carrella, Richard Bailey y Jens Koed Madsen. «Calibrating Agent-Based Models with Linear Regressions». En: *Journal of Artificial Societies and Social Simulation* 23:1 (2020), pág. 7.
- [CH05] Benoît Calvez y Guillaume Hutzler. «Automatic tuning of agent-based models using genetic algorithms». En: *International Workshop on Multi-Agent Systems and Agent-Based Simulation*. Springer. 2005, págs. 41-57.
- [Cha+85] V Chankong et al. «Multiple criteria optimization; a state of the art review». En: *Decision making with multiple objectives* (1985), págs. 36-90.
- [CHBG21] Manuel Chica, Juan Hernández y Jacques Bulchand-Gidumal. «A collective risk dilemma for tourism restrictions under the COVID-19 context». En: *Scientific Reports* 11 (mar. de 2021).
- [Chi+16] Manuel Chica et al. «Identimod: Modeling and managing brand value using soft computing». En: *Decision Support Systems* 89 (2016), págs. 41-55.
- [Chi+17] M. Chica et al. «Multimodal optimization: an effective framework for model calibration». En: *Information Sciences* 375 (2017), págs. 79-97.
- [Chi+20] Manuel Chica et al. «Why simheuristics? Benefits, limitations, and best practices when combining metaheuristics with simulation». En: *SOFT-Statistics and Operations Research Transactions* 44:2 (2020), págs. 311-334.
- [CI06] Gilbert A Churchill y Dawn Iacobucci. *Marketing research: methodological foundations*. Vol. 199. 1. Dryden Press New York, 2006.
- [CLVV+07] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen et al. *Evolutionary algorithms for solving multi-objective problems*. Vol. 5. Springer, 2007.
- [Coa+19] Graham Coates et al. «Agent-based modeling and simulation to assess flood preparedness and recovery of manufacturing small and medium-sized enterprises». En: *Engineering Applications of Artificial Intelligence* 78 (2019), págs. 195-217.



- [CR17] Manuel Chica y William Rand. «Building agent-based decision support systems for word-of-mouth programs. A freemium application». En: *Journal of Marketing Research* 54 (2017), págs. 752-767.
- [CSM10] Wai Kin Victor Chan, Young-Jun Son y Charles M. Macal. «Agent-based simulation tutorial - simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation». En: *Proceedings of the 2010 Winter Simulation Conference*. 2010, págs. 135-150.
- [CWM12] Raymond Chiong, Thomas Weise y Zbigniew Michalewicz. *Variants of Evolutionary Algorithms for Real-World Applications*. Springer Publishing Company, Incorporated, 2012.
- [DA+95] Kalyanmoy Deb, Ram Bhushan Agrawal et al. «Simulated binary crossover for continuous search space». En: *Complex systems* 9:2 (1995), págs. 115-148.
- [Dai+09] Chunni Dai et al. «Parameter optimization for growth model of greenhouse crop using genetic algorithms». En: *Applied Soft Computing* 9:1 (2009), págs. 13-19.
- [Das+11] Swagatam Das et al. «Real-parameter evolutionary multimodal optimization — A survey of the state-of-the-art». En: *Swarm and Evolutionary Computation* 1 (jun. de 2011), págs. 71-88.
- [Deb01] K. Deb. *Multi-objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.
- [Deb+02] K. Deb et al. «A fast and elitist multiobjective genetic algorithm: NSGA-II». En: *IEEE Transactions on Evolutionary Computation* 6:2 (2002), págs. 182-197.
- [Del+10] Sebastiano A Delre et al. «Will it spread or not? The effects of social influences and network topology on innovation diffusion». En: *Journal of Product Innovation Management* 27:2 (2010), págs. 267-282.
- [Dha+20] Sandeep Dhakal et al. «Climate change induced migration and the evolution of cooperation». En: *Applied Mathematics and Computation* 377:C (2020).
- [DJ75] Kenneth Alan De Jong. «Analysis of the behavior of a class of genetic adaptive systems». Tesis doct. University of Michigan, 1975.

- [DMS16] Swagatam Das, Sankha Subhra Mullick y Ponnuthurai N Suganthan. «Recent advances in differential evolution—an updated survey». En: *Swarm and Evolutionary Computation* 27 (2016), págs. 1-30.
- [DR01] Pedro Domingos y Matt Richardson. «Mining the network value of customers». En: *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2001, págs. 57-66.
- [DS11] Swagatam Das y Ponnuthurai Nagarathnam Suganthan. «Differential Evolution: A Survey of the State-of-the-Art». En: *IEEE Transactions on Evolutionary Computation* 15:1 (2011), págs. 4-31.
- [Du+17] Erhu Du et al. «Exploring the role of social media and individual behaviors in flood evacuation processes: An agent-based modeling approach». En: *Water Resources Research* 53:11 (2017), págs. 9164-9180.
- [Duz+07] Dejan Duzevik et al. «The effects of word-of-mouth: An agent-based simulation of interpersonal influence in social networks». En: *2007 Word of Mouth Research Symposium*. 2007.
- [Eps07] Joshua M Epstein. «Generative social science: studies in agent-based computational modeling». En: *Journal of Ecological Anthropology* 11 (2007), pág. 76.
- [Eps99] Joshua M Epstein. «Agent-based computational models and generative social science». En: *Complexity* 4:5 (1999), págs. 41-60.
- [EPV11] Michael G. Epitropakis, Vassilis P. Plagianakos y Michael N. Vrahatis. «Finding multiple global optima exploiting differential evolution's niching capability». En: *2011 IEEE Symposium on Differential Evolution (SDE)*. 2011, págs. 1-8.
- [ER+60] Paul Erdős, Alfréd Rényi et al. «On the evolution of random graphs». En: *Publ. Math. Inst. Hung. Acad. Sci* 5:1 (1960), págs. 17-60.
- [ES+12] Abdulrahman M El-Sayed et al. «Social network analysis and agent-based modeling in social epidemiology». En: *Epidemiologic Perspectives & Innovations* 9:1 (2012), págs. 1-9.

- [ES93] Larry J Eshelman y J David Schaffer. «Real-coded genetic algorithms and interval-schemata». En: *Foundations of genetic algorithms*. Vol. 2. Elsevier, 1993, págs. 187-202.
- [EUG04] Andrew SC Ehrenberg, Mark D Uncles y Gerald J Goodhardt. «Understanding brand performance measures: using Dirichlet benchmarks». En: *Journal of Business Research* 57:12 (2004), págs. 1307-1325.
- [Fab13] Annalisa Fabretti. «On the problem of calibrating an agent based model for financial markets». En: *Journal of Economic Interaction and Coordination* 8:2 (2013), págs. 277-293.
- [Fag+19] Giorgio Fagiolo et al. «Validation of Agent-Based Models in Economics and Finance». En: *Computer Simulation Validation: Fundamental Concepts, Methodological Frameworks, and Philosophical Perspectives*. Ed. por Claus Beisbart y Nicole J. Saam. Cham: Springer International Publishing, 2019, págs. 763-787.
- [Far+10] Paul W Farris et al. *Marketing metrics: The definitive guide to measuring marketing performance*. 2.<sup>a</sup> ed. Pearson Education, 2010.
- [Far+16] Saber Farhadi et al. «An Agent-based-Nash Modeling Framework for Sustainable Groundwater Management: A Case Study». En: *Agricultural Water Management* 177 (ago. de 2016).
- [FF09] J Doyne Farmer y Duncan Foley. «The economy needs agent-based modelling». En: *Nature* 460:7256 (2009), págs. 685-686.
- [Fil+20] Ramón Filgueira et al. «Embracing multimodal optimization to enhance Dynamic Energy Budget parameterization». En: *Ecological Modelling* 431 (2020), págs. 109-139.
- [FMWFM11] Enrique Frias-Martinez, Graham Williamson y Vanessa Frías-Martínez. «An Agent-Based Model of Epidemic Spread Using Human Mobility and Social Network Information». En: oct. de 2011, págs. 57-64.
- [Fog99] Lawrence J Fogel. *Intelligence through simulated evolution: Forty years of evolutionary programming*. John Wiley & Sons, Inc., 1999.

- [Fre+11] Karen Freberg et al. «Who are the social media influencers? A study of public perceptions of personality». En: *Fuel and Energy Abstracts* 37 (mar. de 2011), págs. 90-92.
- [FRM79a] Linton C Freeman, Douglas Roeder y Robert R Mulholland. «Centrality in social networks: ii. experimental results». En: *Social Networks* 2:2 (1979), págs. 119-141.
- [FRM79b] Linton C Freeman, Douglas Roeder y Robert R Mulholland. «Centrality in social networks: II. Experimental results». En: *Social networks* 2:2 (1979), págs. 119-141.
- [Gao+21] Shangce Gao et al. «A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models». En: *Energy Conversion and Management* 230 (2021), págs. 113-784.
- [Gat93] Hubert Gatignon. «Chapter 15 Marketing-mix models». En: *Marketing*. Vol. 5. Handbooks in Operations Research and Management Science. Elsevier, 1993, págs. 697-732.
- [GMM15] R Chulaka Gunasekara, Kishan Mehrotra y Chilukuri K Mohan. «Multi-objective optimization to identify key players in large social networks». En: *Social Network Analysis and Mining* 5:1 (2015), págs. 1-20.
- [Gol89] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing, 1989.
- [GP14] Peijun Guo y Witold Pedrycz. *Human-centric decision-making models for social sciences*. Springer, 2014.
- [GR+87] David E Goldberg, Jon Richardson et al. «Genetic algorithms with sharing for multimodal function optimization». En: (1987), págs. 41-49.
- [GS02] Gerd Gigerenzer y Reinhard Selten. *Bounded rationality: The adaptive toolbox*. MIT press, 2002.
- [GT00] Nigel Gilbert y Pietro Terna. «How to build and use agent-based models in social science». En: *Mind & Society* 1:1 (2000), págs. 57-72.

- [GT05] Nigel Gilbert y Klaus Troitzsch. *Simulation for the social scientist*. McGraw-Hill Education (UK), 2005.
- [GT15] Anindya Ghose y Vilma Todri. «Towards a Digital Attribution Model: Measuring the Impact of Display Advertising on Online Consumer Behavior». En: *SSRN Electronic Journal* (ene. de 2015).
- [Gui+03] Roger Guimera et al. «Self-similar community structure in a network of human interactions». En: *Physical Review E* 68:6 (2003), págs. 65-103.
- [HAN13] Farhad Hosseinali, Ali A Alesheikh y Farshad Nourian. «Agent-based modeling of urban land-use development, case study: Simulating future scenarios of Qazvin city». En: *Cities* 31 (2013), págs. 105-113.
- [Has+13] Samer Hassan et al. «Asking the Oracle: Introducing Forecasting Principles into Agent-Based Modelling». En: *Journal of Artificial Societies and Social Simulation* 16:3 (2013), pág. 13.
- [Hay15] Philip Haynes. *Managing complexity in the public services*. Routledge, 2015.
- [He+19] Zhou He et al. «Evolutionary food quality and location strategies for restaurants in competitive online-to-offline food ordering and delivery markets: An agent-based approach». En: *International Journal of Production Economics* 215 (2019), págs. 61-72.
- [HG09] Lynne Hamill y Geoffrey Gilbert. «Social circles: A simple structure for agent-based social network models». En: *Journal of Artificial Societies and Social Simulation* 12:2 (2009).
- [HH95] Per Hage y Frank Harary. «Eccentricity and centrality in networks». En: *Social Networks* 17:1 (1995), págs. 57-63.
- [HK06] Rob J. Hyndman y Anne B. Koehler. «Another look at measures of forecast accuracy». En: *International Journal of Forecasting* 22:4 (2006), págs. 679-688.
- [HL17] Michael Haenlein y Barak Libai. «Seeding, Referral, and Recommendation: Creating Profitable Word-of-Mouth Programs». En: *California Management Review* 59:2 (2017), págs. 68-91.

- [HL71] Paul W Holland y Samuel Leinhardt. «Transitivity in structural models of small groups». En: *Comparative group studies* 2:2 (1971), págs. 107-124.
- [Hoe+20] Nicolas Hoertel et al. «A stochastic agent-based model of the SARS-CoV-2 epidemic in France». En: *Nature Medicine* 26 (sep. de 2020), págs. 1-5.
- [Hol92] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [HS08] Florian Hauser y Klaus Schredelseker. *Complexity and Artificial Markets*. Springer, 2008.
- [Ibá+10] Oscar Ibáñez et al. «Multimodal genetic algorithms for craniofacial superimposition». En: *Nature-inspired informatics for intelligent applications and knowledge discovery: implications in business, science, and engineering*. IGI Global, 2010, págs. 119-143.
- [Jan+17] Jarosław Jankowski et al. «Balancing Speed and Coverage by Sequential Seeding in Complex Networks». En: *Scientific Reports* 7:1 (2017), pág. 891.
- [Jan+93] Svante Janson et al. «The birth of the giant component». En: *Random Structures & Algorithms* 4:3 (1993), págs. 233-358.
- [JGD21] R. K. Jana, Indranil Ghosh y Debojyoti Das. «A differential evolution-based regression framework for forecasting Bitcoin price». En: *Annals of Operations Research* 306:1 (2021), págs. 295-320.
- [JJ03] Marco A Janssen y Wander Jager. «Simulating market dynamics: Interactions between consumer psychology and social networks». En: *Artificial Life* 9:4 (2003), págs. 343-356.
- [JJ12] Wander Jager y Marco A. Janssen. «An updated conceptual framework for integrated modeling of human decision making: The Consumat II». En: 2012.
- [JJV01] Wander Jager, Marco A Janssen y Charles Viek. «Experimentation with household dynamics: the consumat approach». En: *International Journal of Sustainable Development* 4:1 (2001), págs. 90-100.

- [JO06] Marco Janssen y Elinor Ostrom. «Empirically Based, Agent-based models». En: *Ecology and Society* 11 (dic. de 2006).
- [Kan14] A Kangur. «Simulating the transition to electric cars using the consumat agent rationale». Tesis doct. University of Groningen, 2014.
- [Kan15] Vijayarathy Kannan. «A Distributed Approach to EpiFast using Apache Spark». Tesis doct. Virginia Tech, 2015.
- [Kaz+19] Borhan Kazimipour et al. «Bandit-based cooperative coevolution for tackling contribution imbalance in large-scale optimization problems». En: *Applied Soft Computing* 76 (2019), págs. 265-281.
- [KE95] J. Kennedy y R. Eberhart. «Particle swarm optimization». En: *Proceedings of ICNN'95 - International Conference on Neural Networks*. Vol. 4. 1995, págs. 1942-1948.
- [Kei+19] Mohammad Mehdi Keikha et al. «Influence Maximization across Heterogeneous Interconnected Networks based on Deep Learning». En: *Expert Systems with Applications* 140 (ago. de 2019), pág. 112905.
- [Kie+12] Elmar Kiesling et al. «Agent-based simulation of innovation diffusion: a review». En: *Central European Journal of Operations Research* 20:2 (2012), págs. 183-230.
- [KKT03] David Kempe, Jon Kleinberg y Éva Tardos. «Maximizing the Spread of Influence through a Social Network». En: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '03. Washington, D.C.: Association for Computing Machinery, 2003, 137–146.
- [Koo00] Sebastiaan ALM Kooijman. *Dynamic Energy and Mass Budgets in Biological Systems*. 2.<sup>a</sup> ed. Cambridge University Press, 2000.
- [Koo+08] Sebastiaan ALM Kooijman et al. «From food-dependent statistics to metabolic parameters, a practical guide to the use of dynamic energy budget theory». En: *Biological Reviews* 83:4 (2008), págs. 533-552.
- [Koo09] Sebastiaan ALM Kooijman. *Dynamic Energy Budget Theory for Metabolic Organisation*. 3.<sup>a</sup> ed. Cambridge University Press, 2009.

- [Koo21] Sebastiaan ALM Kooijman. *Add-my-Pet project*. [https://add-my-pet.github.io/DEBtool\\_M/docs/](https://add-my-pet.github.io/DEBtool_M/docs/). [Online; accessed 10-June-2021]. 2021.
- [Koz94] John R. Koza. «Genetic programming as a means for programming computers by natural selection». En: *Statistics and Computing* 4 (1994), págs. 87-112.
- [KR03] Kyu-Ok Kim y L. R. Rilett. «Simplex-Based Calibration of Traffic Microsimulation Models with Intelligent Transportation Systems Data». En: *Transportation Research Record* 1855:1 (2003), págs. 80-89.
- [KS19] Kevin Lane Keller y Vanitha Swaminathan. «Strategic brand management: building, measuring, and managing brand equity». En: (2019).
- [KSC20] Manjit Kaur, Dilbag Singh y Vijay Chahar. «Drug Synergy Prediction Using Dynamic Mutation Based Differential Evolution». En: *Current Pharmaceutical Design* 27:8 (nov. de 2020), págs. 1103-1111.
- [Kum14] Vineet Kumar. «Making "freemium" work». En: *Harvard business review* 92:5 (2014), págs. 27-29.
- [KWC01] Joshua D Knowles, Richard A Watson y David W Corne. «Reducing local optima in single-objective problems by multi-objectivization». En: (2001), págs. 269-283.
- [LAH07] Jure Leskovec, Lada A. Adamic y Bernardo A. Huberman. «The Dynamics of Viral Marketing». En: *ACM Trans. Web* 1:1 (2007), 5-es.
- [LaT+19] Antonio LaTorre et al. «Model calibration using a parallel differential evolution algorithm in computational neuroscience: simulation of stretch induced nerve deficit». En: *Journal of Computational Science* (2019), pág. 101053.
- [Lav05] Michael Laver. «Policy and the dynamics of political competition». En: *American Political Science Review* 99:2 (2005), págs. 263-281.
- [Laz+09] David Lazer et al. «Computational social science». En: *Science* 323:5915 (2009), págs. 721-723.



- [LC10] Mayank Lahiri y Manuel Cebrian. «The Genetic Algorithm as a General Diffusion Model for Social Networks». En: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. 2010, págs. 494-499.
- [LD08] Mikola Lysenko y Roshan M D'Souza. «A framework for megascale agent based model simulations on graphics processing units». En: *Journal of Artificial Societies and Social Simulation* 11:4 (2008), pág. 10.
- [Lee+15] Ju-Sung Lee et al. «The complexities of agent-based modeling output analysis». En: *Journal of Artificial Societies and Social Simulation* 18:4 (2015), pág. 4.
- [Lib+10] B. Libai et al. «Customer-to-customer interactions: Broadening the scope of word of mouth research». English. En: *Journal of Service Research* 13:3 (ago. de 2010), págs. 267-282.
- [Lik+11] Konstadia Lika et al. «The “covariation method” for estimating the parameters of the standard Dynamic Energy Budget model I: Philosophy and approach». En: *Journal of Sea Research* 66:4 (2011), págs. 270-277.
- [Lik+14] Konstadia Lika et al. «The bijection from data to parameter space with the standard DEB model quantifies the supply–demand spectrum». En: *Journal of Theoretical Biology* 354 (2014), págs. 35-47.
- [Lin92] Long-Ji Lin. «Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching». En: *Mach. Learn.* 8:3–4 (1992), 293–321.
- [Liu+17] Zhengchun Liu et al. «A simulation and optimization based method for calibrating agent-based emergency department models under data scarcity». En: *Computers & Industrial Engineering* 103 (2017), págs. 300-309.
- [LJS16] Runjiao Liu, Difei Jiang y Lei Shi. «Agent-based simulation of alternative classroom evacuation scenarios». En: *Frontiers of Architectural Research* 5:1 (2016), págs. 111-125.
- [LMP13] Barak Libai, Eitan Muller y Renana Peres. «Decomposing the value of word-of-mouth seeding programs: Acceleration versus expansion». En: *Journal of Marketing Research* 50:2 (2013), págs. 161-176.

- [LRK01] Morten Løvbjerg, Thomas Kiel Rasmussen y Thiemo Krink. «Hybrid Particle Swarm Optimiser with Breeding and Subpopulations». En: *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*. GECCO'01. San Francisco, California: Morgan Kaufmann Publishers Inc., 2001, 469–476.
- [Luk98] Sean Luke. *ECJ Evolutionary Computation Library*. Publicly available at <http://cs.gmu.edu/> 1998.
- [Luo11] Lan Luo. «Product line design for consumer durables: an integrated marketing and engineering approach». En: *Journal of Marketing Research* 48:1 (2011), págs. 128-139.
- [LY12] Xiaodong Li y Xin Yao. «Cooperatively Coevolving Particle Swarms for Large Scale Optimization». En: *IEEE Transactions on Evolutionary Computation* 16:2 (2012), págs. 210-224.
- [Mac05] Charles M Macal. «Model verification and validation». En: *Workshop on "Threat Anticipation: Social Science Methods and Models*. 2005.
- [Mar+12] Benjamin T Martin et al. «Dynamic Energy Budget theory meets individual-based modelling: a generic and accessible implementation». En: *Methods in Ecology and Evolution* 3:2 (2012), págs. 445-449.
- [Mar+18] Gonçalo M Marques et al. «The AmP project: comparing species on the basis of dynamic energy budget parameters». En: *PLoS computational biology* 14:5 (2018), e1006100.
- [Mar+19] Gonçalo M. Marques et al. «Fitting multiple models to multiple data sets». En: *Journal of Sea Research* 143 (2019). Ecosystem based management and the biosphere: a new phase in DEB research, págs. 48-56.
- [MBA14] Cristian Morosan, John T Bowen y Morgan Atwood. «The evolution of marketing research». En: *International Journal of Contemporary Hospitality Management* (2014).

- [MCC19] Ignacio Moya, Manuel Chica y Óscar Cordón. «A multicriteria integral framework for agent-based model calibration using evolutionary multiobjective optimization and network-based visualization». En: *Decision Support Systems* 124 (2019), pág. 113111.
- [MCC21] Ignacio Moya, Manuel Chica y Oscar Cordon. «Evolutionary multiobjective optimization for automatic agent-based model calibration: A comparative study». En: *IEEE Access* 9 (2021), págs. 55284-55299.
- [MCM19] JV Morais, AL Custódio y GM Marques. «Calibration of parameters in Dynamic Energy Budget models using Direct-Search methods». En: *Journal of Mathematical Biology* 78:5 (2019), págs. 1439-1458.
- [MD13] Daniele Muraro y Rui Dilão. «A parallel multi-objective optimization algorithm for the calibration of mathematical models». En: *Swarm and Evolutionary Computation* 8 (2013), págs. 13-25.
- [Mil67] Stanley Milgram. «The small world problem». En: *Psychology today* 2:1 (1967), págs. 60-67.
- [MK03] Eve Mitleton-Kelly. *Complex systems and evolutionary perspectives on organisations: the application of complexity theory to organisations*. Elsevier Science Ltd, 2003.
- [MLY14] Yi Mei, Xiaodong Li y Xin Yao. «Cooperative Coevolution With Route Distance Grouping for Large-Scale Capacitated Arc Routing Problems». En: *IEEE Transactions on Evolutionary Computation* 18:3 (2014), págs. 435-449.
- [MN14] Charles Macal y Michael North. «Introductory tutorial: Agent-based modeling and simulation». En: *Proceedings of the winter simulation conference 2014*. IEEE, 2014, págs. 6-20.
- [Mom22] Ida Momennejad. «Collective minds: social network topology shapes collective cognition». En: *Philosophical Transactions of the Royal Society B* 377:1843 (2022).

- [Moo97] Christopher Z Mooney. *Monte Carlo simulation*. 116. SAGE Publications Inc., 1997.
- [Moy+17] Ignacio Moya et al. «An agent-based model for understanding the influence of the 11-M terrorist attacks on the 2004 Spanish elections». En: *Knowledge-Based Systems* 123 (2017), págs. 200-216.
- [Moy+21a] Ignacio Moya et al. «Coral reefs optimization algorithms for agent-based model calibration». En: *Engineering Applications of Artificial Intelligence* 100 (2021), págs. 104-170.
- [Moy+21b] Ignacio Moya et al. «Simulating the influence of terror management strategies on the voter ideological distance using agent-based modeling». En: *Telematics and Informatics* 63 (2021), págs. 101-656.
- [MPM18] Magnus Moglia, Aneta Podkalicka y James McGregor. «An agent-based model of residential energy efficiency adoption». En: *Journal of Artificial Societies and Social Simulation* 21:3 (2018).
- [MS00] Emma K Macdonald y Byron M Sharp. «Brand Awareness Effects on Consumer Decision Making for a Common, Repeat Purchase Product: A Replication». En: *Journal of Business Research* 48:1 (2000), págs. 5-15.
- [MT10] Melanie Mitchell y Zoltan Toroczkai. «Complexity: A Guided Tour». En: *Physics Today* 63 (feb. de 2010).
- [Muñ+15] Mario A Muñoz et al. «Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges». En: *Information Sciences* 317 (2015), págs. 224-245.
- [MW02] Michael W Macy y Robert Willer. «From factors to actors: Computational sociology and agent-based modeling». En: *Annual review of sociology* (2002), págs. 143-166.
- [NBW06] Mark Ed Newman, Albert-László Ed Barabási y Duncan J Watts. *The structure and dynamics of networks*. Princeton university press, 2006.

- [NC16] Akira Namatame y Shu-Heng Chen. *Agent-based modeling and network dynamics*. Oxford University Press, 2016.
- [NJN19] Afshin Naseri y Nima Jafari Navimipour. «A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm». En: *Journal of Ambient Intelligence and Humanized Computing* 10:5 (2019), págs. 1851-1864.
- [NM12] D. Ngoduy y M.J. Maher. «Calibration of second order traffic models using continuous cross entropy method». En: *Transportation Research Part C: Emerging Technologies* 24 (2012), págs. 102-121.
- [NM65] John A Nelder y Roger Mead. «A simplex method for function minimization». En: *The Computer Journal* 7:4 (1965), págs. 308-313.
- [NMM06] Giuseppe Narzisi, Venkatesh Mysore y Bud Mishra. «Multi-objective evolutionary optimization of agent-based models: An application to emergency response planning.» En: *Computational Intelligence 2006* (2006), págs. 224-230.
- [NRW05] Prasad A Naik, Kalyan Raman y Russell S Winer. «Planning marketing-mix strategies in the presence of interaction effects». En: *Marketing Science* 24:1 (2005), págs. 25-34.
- [NT+17] Khanh Nguyen-Trong et al. «Optimization of municipal solid waste transportation by integrating GIS analysis, equation-based, and agent-based model». En: *Waste management* 59 (2017), págs. 14-22.
- [NY14] Ashkan Negahban y Levent Yilmaz. «Agent-based simulation applications in marketing research: an integrated review». En: *Journal of Simulation* 8:2 (2014), págs. 129-142.
- [OAS10] Tore Opsahl, Filip Agneessens y John Skvoretz. «Node Centrality in Weighted Networks: Generalizing Degree and Shortest Paths». En: *Social Networks - SOC NETWORKS* 32 (jul. de 2010), págs. 245-251.

- [OEBY20] Sherin M Omran, Wessam H El-Behaidy y Aliaa AA Youssif. «Decomposition based multi-objectives evolutionary algorithms challenges and circumvention». En: *Science and Information Conference*. Springer. 2020, págs. 82-93.
- [Oli03] Rogelio Oliva. «Model calibration as a testing strategy for system dynamics models». En: *European Journal of Operational Research* 151:3 (2003), págs. 552-568.
- [Ove+19] Gijs Overgoor et al. «Letting the computers take over: Using AI to solve marketing problems». En: *California Management Review* 61:4 (2019), págs. 156-185.
- [PBJ91] J Payne, JR Bettman y EJ Johnson. «Consumer decision making». En: *Handbook of consumer behaviour* (1991), págs. 50-84.
- [Pet96] A. Petrowski. «A clearing procedure as a niching method for genetic algorithms». En: *Proceedings of IEEE International Conference on Evolutionary Computation*. 1996, págs. 798-803.
- [Pét97] Alan Pétrowski. «An efficient hierarchical clustering technique for speciation». En: *Evolution. Technical report, Institute National des Telecommunications, Evry, France* (1997).
- [PK12] A Poole y Apostolos Kotsialos. «METANET model validation using a genetic algorithm». En: *IFAC Proceedings Volumes* 45:24 (2012), págs. 7-12.
- [Pre15] Mike Preuss. *Multimodal optimization by means of evolutionary algorithms*. Springer, 2015.
- [PSL06] Kenneth Price, Rainer M Storn y Jouni A Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.
- [RD02] Matthew Richardson y Pedro Domingos. «Mining Knowledge-Sharing Sites for Viral Marketing». En: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '02. Edmonton, Alberta, Canada: Association for Computing Machinery, 2002, 61–70.
- [Rog10] Everett M Rogers. *Diffusion of innovations*. Simon y Schuster, 2010.

- [RR11] William Rand y Roland T Rust. «Agent-based modeling in marketing: Guidelines for rigor». En: *International Journal of Research in Marketing* 28:3 (2011), págs. 181-193.
- [RT04] Alex Rogers y Peter von Tessin. «Multi-Objective Calibration For Agent-Based Models». En: *Agent-Based Simulation 5 (30/04/04)*. Event Dates: May 2004. 2004.
- [Sal+10] Mohamed Saleh et al. «A comprehensive analytical approach for policy analysis of system dynamics models». En: *European Journal of Operational Research* 203:3 (2010), págs. 673-683.
- [Sar10] Robert G Sargent. «Verification and validation of simulation models». En: *Proceedings of the 2010 winter simulation conference*. IEEE. 2010, págs. 166-183.
- [Sar+13] G Sarà et al. «Parameterisation of bivalve functional traits for mechanistic eco-physiological dynamic energy budget (DEB) models». En: *Marine Ecology Progress Series* 480 (2013), págs. 99-117.
- [Saw05] R. Keith Sawyer. *Social Emergence: Societies As Complex Systems*. Cambridge University Press, 2005.
- [SBR18] M. Ehsan Shafiee, Zachary Barker y Amin Rasekh. «Enhancing water system models by integrating big data». En: *Sustainable Cities and Society* 37 (2018), págs. 485-491.
- [Sch+13] Christian Schlereth et al. «Optimal Product-Sampling Strategies in Social Networks: How Many and Whom to Target?» En: *International Journal of Electronic Commerce* 18:1 (2013), págs. 45-72.
- [SD06] Gulshan Singh y Kalyanmoy Deb. «Comparison of Multi-Modal Optimization Algorithms Based on Evolutionary Algorithms». En: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. GECCO '06. Seattle, Washington, USA: Association for Computing Machinery, 2006, 1305–1312.
- [Ser+16] Leidy Y. Serna et al. «Optimization techniques in respiratory control system models». En: *Applied Soft Computing* 48 (2016), págs. 431-443.

- [Sil+20] Petrônio CL Silva et al. «COVID-ABS: An agent-based model of COVID-19 epidemic to simulate health and economic effects of social distancing interventions». En: *Chaos, Solitons & Fractals* 139 (2020), pág. 110088.
- [SK98] Bruno Sareni y Laurent Krähenbühl. «Fitness sharing and niching methods revisited». En: *Evolutionary Computation, IEEE Transactions on* 2 (oct. de 1998), págs. 97-106.
- [Sma+17] Isabel M Smallegange et al. «Mechanistic description of population dynamics using dynamic energy budget theory incorporated into integral projection models». En: *Methods in Ecology and Evolution* 8:2 (2017), págs. 146-154.
- [SP97] Rainer Storn y Kenneth Price. «Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces». En: *Journal of Global Optimization* 11 (ene. de 1997), págs. 341-359.
- [SR03] Mani R. Subramani y Balaji Rajagopalan. «Knowledge-Sharing and Influence in Online Social Networks via Viral Marketing». En: *Commun. ACM* 46:12 (2003), 300–307.
- [SR14] Forrest Stonedahl y William Rand. «When does simulated data match real data?». En: *Advances in Computational Social Science*. Springer, 2014, págs. 297-313.
- [ŞR18] Aybike ŞİMŞEK y KARA Resul. «Using swarm intelligence algorithms to detect influential individuals for influence maximization in social networks». En: *Expert Systems with Applications* 114 (2018), págs. 224-236.
- [SRW10] Forrest Stonedahl, William Rand y Uri Wilensky. «Evolving Viral Marketing Strategies». En: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*. GECCO '10. Portland, Oregon, USA: Association for Computing Machinery, 2010, 1195–1202.
- [Str01] Steven Strogatz. «Strogatz, S.H.: Exploring Complex Networks. Nature 410, 268». En: *Nature* 410 (abr. de 2001), págs. 268-76.
- [SV18] Shweta Sharma y H. Verma. «Social Media Marketing: Evolution and Change». En: feb. de 2018, págs. 19-36.



- [SW10] Forrest Stonedahl y Uri Wilensky. «Evolutionary Robustness Checking in the Artificial Anasazi Model». En: *AAAI Fall Symposium: Complex Adaptive Systems*. 2010.
- [SWM17] Wojciech Samek, Thomas Wiegand y Klaus-Robert Müller. «Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models». En: *arXiv preprint arXiv:1708.08296* (2017).
- [Sys91] Gilbert Syswerda. «A study of reproduction in generational and steady-state genetic algorithms». En: *Foundations of genetic algorithms*. Vol. 1. Elsevier, 1991, págs. 94-101.
- [SZM13] Arnaud Sallaberry, Faraz Zaidi y Guy Melançon. «Model for generating artificial social networks having community structures with small-world and scale-free properties». En: *Social Network Analysis and Mining* 3:3 (2013), págs. 597-609.
- [Tad16] Emmanuel M. Tadjouddine. «Calibration based on entropy minimization for a class of asset pricing models». En: *Applied Soft Computing* 42 (2016), págs. 431-438.
- [Tal09] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.
- [TBP08] Michael Trusov, Randolph Bucklin y Koen Pauwels. «Effects of Word-of-Mouth Versus Traditional Marketing: Findings from an Internet Social Networking Site». En: *Journal of Marketing* 73 (abr. de 2008).
- [Tei+17] Andreia Sofia Teixeira et al. «Using Spark and GraphX to Parallelize Large-Scale Simulations of Bacterial Populations over Host Contact Networks». En: *International Conference on Algorithms and Architectures for Parallel Processing*. Springer. 2017, págs. 591-600.
- [TF13] Ryoji Tanabe y Alex Fukunaga. «Success-history based parameter adaptation for Differential Evolution». En: *2013 IEEE Congress on Evolutionary Computation*. 2013, págs. 71-78.

- [TF14] Ryoji Tanabe y Alex S. Fukunaga. «Improving the search performance of SHADE using linear population size reduction». En: *2014 IEEE Congress on Evolutionary Computation (CEC)*. 2014, págs. 1658-1665.
- [TI20] Ryoji Tanabe e Hisao Ishibuchi. «A Review of Evolutionary Multimodal Multi-objective Optimization». En: *IEEE Transactions on Evolutionary Computation* 24:1 (2020), págs. 193-200.
- [TKG14] Jan C. Thiele, Winfried Kurth y Volker Grimm. «Facilitating Parameter Estimation and Sensitivity Analysis of Agent-Based Models: A Cookbook Using NetLogo and 'R'». En: *Journal of Artificial Societies and Social Simulation* 17:3 (2014), pág. 11.
- [Tra+21] Viet-Phu Tran et al. «Success-history based adaptive differential evolution method for optimizing fuel loading pattern of VVER-1000 reactor». En: *Nuclear Engineering and Design* 377 (2021), pág. 111125.
- [TRJ13] Michael Trusov, William Rand y Yogesh V Joshi. «Improving prelaunch diffusion forecasts: Using synthetic networks as simulated priors». En: *Journal of Marketing Research* 50:6 (2013), págs. 675-690.
- [Tru13] Richard J Trudeau. *Introduction to graph theory*. Courier Corporation, 2013.
- [TSN20] Shih-Hsien Tseng y Tien Son Nguyen. «Agent-Based Modeling of Rumor Propagation Using Expected Integrated Mean Squared Error Optimal Design». En: *Applied System Innovation* 3:4 (2020), pág. 48.
- [VB10] Alexey Voinov y Francois Bousquet. «Modelling with stakeholders». En: *Environmental Modelling & Software* 25:11 (2010). Thematic Issue - Modelling with Stakeholders, págs. 1268-1281.
- [VDB+07] Frans Van Den Bergh et al. «An analysis of particle swarm optimizers». Tesis doct. University of Pretoria, 2007.
- [VKK04] Rajkumar Venkatesan, Trichy V Krishnan y Vineet Kumar. «Evolutionary estimation of macro-level diffusion models using genetic algorithms: An alternative to nonlinear least squares». En: *Marketing Science* 23:3 (2004), págs. 451-464.

- [VO+19] PR Van Oel et al. «Using data on social influence and collective action for parameterizing a geographically-explicit agent-based model for the diffusion of soil conservation efforts». En: *Environmental Modeling & Assessment* 24:1 (2019), págs. 1-19.
- [Voi+16] Alexey Voinov et al. «Modelling with stakeholders – Next generation». En: *Environmental Modelling & Software* 77 (2016), págs. 196-220.
- [VT10] Kaan Varnali y Aysegul Toker. «Mobile marketing research: The-state-of-the-art». En: *International Journal of Information Management* 30 (abr. de 2010), págs. 144-151.
- [Wal18] M. Mitchell Waldrop. «Free agents». En: *Science* 360:6385 (2018), págs. 144-147.
- [Wat+07] Duncan J. Watts et al. «Influentials, Networks, and Public Opinion Formation». En: *Journal of Consumer Research* 34:4 (2007), págs. 441-458.
- [WEM16] Jonathan Ward, Andrew Evans y Nicolas Malleson. «Dynamic calibration of agent-based models using data assimilation». En: *Royal Society Open Science* 3 (abr. de 2016), págs. 150-703.
- [WF94] Stanley Wasserman y Katherine Faust. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 1994.
- [Wil+20] Meike Will et al. «Combining social network analysis and agent-based modelling to explore dynamics of human interaction: A review». En: *Socio-Environmental Systems Modelling* 2 (feb. de 2020).
- [WMA13] Lilian Weng, Filippo Menczer y Yong-Yeol Ahn. «Virality Prediction and Community Structure in Social Networks». En: *Scientific reports* 3 (ago. de 2013), pág. 2522.
- [Won15] Ka chun Wong. «Evolutionary Multimodal Optimization: A Short Survey». En: *ArXiv* abs/1508.00457 (2015).
- [WR15] Uri Wilensky y William Rand. *An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo*. Mit Press, 2015.

- [WS98] Duncan J Watts y Steven H Strogatz. «Collective dynamics of 'small-world' networks». En: *Nature* 393:6684 (1998), págs. 440-442.
- [XH16] Yu Xiao y Jingti Han. «Forecasting new product diffusion with agent-based models». En: *Technological Forecasting and Social Change* 105 (2016), págs. 167-178.
- [XWX11] Haoxiang Xia, Huili Wang y Zhaoguo Xuan. «Opinion dynamics: A multidisciplinary review and perspective on future research». En: *International Journal of Knowledge and Systems Science (IJKSS)* 2:4 (2011), págs. 72-91.
- [YKG09] Jie Yao, Nawwaf Kharmay y Peter Grogono. «Bi-objective multipopulation genetic algorithm for multimodal function optimization». En: *IEEE Transactions on Evolutionary Computation* 14:1 (2009), págs. 80-102.
- [YKG10] Jie Yao, Nawwaf Kharmay y Peter Grogono. «Bi-Objective Multipopulation Genetic Algorithm for Multimodal Function Optimization». En: *IEEE Transactions on Evolutionary Computation* 14:1 (2010), págs. 80-102.
- [ZC15] Jinghui Zhong y Wentong Cai. «Differential evolution with sensitivity analysis and the Powell's method for crowd model calibration». En: *Journal of Computational Science* 9 (2015). Computational Science at the Gates of Nature, págs. 26-32.
- [ZCG14] Elmer César Trejo Zúñiga, Irineo Lorenzo López Cruz y Agustín Ruíz García. «Parameter estimation for crop growth model using evolutionary and bio-inspired algorithms». En: *Applied Soft Computing* 23 (2014), págs. 474-482.
- [ZGG11] Ting Zhang, Sonja Gensler y Rosanna Garcia. «A study of the diffusion of alternative fuel vehicles: An agent-based modeling approach». En: *Journal of Product Innovation Management* 28:2 (2011), págs. 152-168.
- [Zha+15] J. Zhang et al. «Leveraging social influence to address overweight and obesity using agent-based models: The role of adolescent social networks». En: *Social Science & Medicine* 125 (2015). Special Issue: Social Networks, Health and Mental Health, págs. 203-213.

- [Zha+16] Qiang Zhang et al. «Agent based urban growth modeling framework on apache spark». En: *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*. 2016, págs. 50-59.
- [Zit+03] E. Zitzler et al. «Performance assessment of multiobjective optimizers: an analysis and review». En: *IEEE Transactions on Evolutionary Computation* 7:2 (2003), págs. 117-132.
- [ZL07] Qingfu Zhang y Hui Li. «MOEA/D: A multiobjective evolutionary algorithm based on decomposition». En: *IEEE Transactions on Evolutionary Computation* 11:6 (2007), págs. 712-731.
- [ZSM12] Faraz Zaidi, Arnaud Sallaberry y Guy Melançon. «Generating artificial social networks with small world and scale free properties». Tesis doct. INRIA; LIRMM, 2012.
- [ZWD09] Brent Zenobia, Charles Weber y Tugrul Daim. «Artificial markets: A review and assessment of a new venue for innovation research». En: *Technovation* 29:5 (2009), págs. 338-350.