Un Marco de Trabajo para la Optimización Evolutiva de la Evacuación de Multitudes

Ángel Fernández-Civila[1], Carlos Cotta[1,2][0000−0001−8478−7549], and Antonio J. Fernández-Leiva[1,2][0000−0002−5330−5217]

[1] Dept. Lenguajes y Ciencias de la Computación,
Universidad de Málaga, 29071 Málaga, España
angelfdezcivila@uma.es,{ccottap,afdez}@lcc.uma.es
[2] ITIS Software, Universidad de Málaga, 29071 Málaga, España

Resumen La evacuación de recintos es un problema de extraordinaria importancia y complejidad. En este artículo presentamos un marco de trabajo integrado que combina heurísticas de simulación (simheurísticas) con un entorno de análisis y visualización. El modelado del comportamiento de la muchedumbre se realiza mediante autómatas celulares probabilísticos basados en agentes e incorpora tanto la toma de decisiones racionales como la interacción social. El funcionamiento del marco de trabajo se ilustra con un caso de estudio en el que se optimiza la localización de puertas de emergencia, empleando algoritmos evolutivos.

Keywords: Evacuación de multitudes · Simheurísticas · Autómatas celulares · Algoritmos evolutivos.

1. Introducción

Es evidente que gran parte de nuestras vidas se desarrollan en espacios interiores (escuelas, entornos laborales, comercios, locales de ocio, etc.), tal como varios estudios evidencian [8,16], y como nuestra experiencia personal probablemente confirme. Esto subraya lo importante que es poder evacuar de manera segura y rápida en caso de una emergencia en estos lugares. Naturalmente, alcanzar este objetivo plantea una problemática con múltiples facetas en la que no solo se involucran varios factores, sino que también proporciona diversas visiones para abordarla [12].

Claramente, alcanzar una evacuación efectiva y segura es un empeño que está trufado de decisiones de diseño sujetas a optimización. Sin embargo, dicha optimización es de enorme complejidad. Una de las razones fundamentales es la dependencia que cualquier estrategia de evacuación va a tener del comportamiento de una multitud en una situación de emergencia. Modelar dicho comportamiento puede llegar a ser extraordinariamente complejo y es, en cualquier caso, un objetivo que no puede alcanzarse mediante una fórmula matemática cerrada. Por el contrario, es necesario recurrir a modelos de simulación. El empleo de estos en entornos de optimización conlleva toda una tipología propia de consideraciones que deberán tenerse en cuenta durante el proceso, lo que ha llevado

a acuñar el término *simheurísticas* para designar a este enfoque de optimización [15]. En este contexto, dichas consideraciones no se limitan a aspectos del problema que deban ser incorporados en el proceso de optimización (estocasticidad, ruido, complejidad computacional, etc.), sino que también incorporan el propio análisis y visualización de estrategias y entornos, de manera que el experto humano pueda formar parte activa del proceso y sea capaz de interpretar y adaptar los resultados. En este sentido, en este trabajo se presenta una herramienta holística que aúna un marco metaheurístico de optimización, con un modelado del comportamiento de multitudes y un sistema de visualización integrado en el cual toma una importancia muy significativa el involucrar al humano en el análisis de los resultados (i.e., el llamado *human-in-the-loop*).

2. Preliminares

Como se indicaba en la Sección 1, por más que sea posible modelar matemáticamente el comportamiento de una multitud, no es posible alcanzar una expresión matemática cerrada que capture la configuración final de esta. Es posible, sin embargo, confeccionar conjuntos de ecuaciones o reglas que definan la evolución del estado de la multitud durante la evacuación.

Los detalles de estos modelos matemáticos dependerán de diversos factores; por ejemplo, la granularidad de la simulación ([4,20]). Así, podemos tener [17]: *modelos macroscópicos*, en los que la multitud se considera un fluido en un medio continuo (e.g., [1,11]) o *modelos microscópicos*, en los que cada miembro de la multitud (usaremos el término *peatón*) es un agente cuyo comportamiento está definido por ciertas reglas y por la interacción con otros peatones.

Los modelos microscópicos son especialmente interesantes en contextos donde la heterogeneidad de los individuos y las interacciones detalladas entre ellos son determinantes. Así, mientras que los modelos macroscópicos tratan a la multitud como una masa homogénea similar a un fluido basado en promedios locales de densidad y velocidad, los enfoques microscópicos permiten integrar diferencias individuales en parámetros como edad, género, velocidad deseada y capacidades de decisión racional o instintiva [14]. Este nivel de detalle es esencial en espacios confinados o de geometría compleja, donde las interacciones físicas y psicológicas locales generan fenómenos de autoorganización, la formación espontánea de carriles o situaciones de pánico [4]. Por tanto, su uso es preferible cuando se requiere una alta fidelidad y realismo para predecir riesgos específicos y comportamientos espontáneos que los modelos macroscópicos no pueden capturar con precisión.

Concentrándonos, pues, en este tipo de modelos, se pueden considerar dos grandes categorías dentro de los mismos:

(a) *Modelos de fuerza social*, en los que la multitud es una colección de partículas individuales en un espacio continuo cuya dinámica está controlada por su interacción entre sí y con el entorno (e.g., [3,18]).
(b) *Modelos de autómata celular (CA)*, en los que los individuos viven en un espacio discreto sujeto a reglas de evolución local (e.g., [22,23]).

Cada uno de estos enfoques ofrece un equilibrio diferente entre dos objetivos contrapuestos: simplicidad y realismo (véase [4,9,17] para más detalles). Los modelos de autómatas celulares resultan atractivos por su comprensibilidad y su capacidad para capturar comportamientos emergentes. Precisamente en esta línea, hemos definido anteriormente un modelo probabilístico de autómatas celulares basados en agentes que integra una estrategia de escape racional y un comportamiento de aversión a las multitudes, el cual puede utilizarse dentro de un marco de optimización basado en simulación [5,6].

3. Modelo de simulación

El modelo de simulación empleado para capturar la geometría del recinto, así como el comportamiento de las personas que se hallan en su interior y cuya evacuación se pretende, se basa en un autómata celular probabilista [19].

Un autómata celular es un sistema dinámico discreto que evoluciona sobre una cuadrícula, donde cada celda puede adoptar un estado según reglas específicas que determinan el comportamiento del autómata. En nuestro caso, partimos de un entorno original $\mathcal{E}$, que consiste básicamente en una representación de la planta del recinto o espacio considerado. Sobre dicho diseño de planta se traza una rejilla cuadrada con un tamaño de celda específico c, lo que proporciona una discretización del entorno y define la matriz de estados sobre la que el autómata funcionará. Como resultado de esta discretización, algunas celdas estarán `bloqueadas` (ocupadas por un obstáculo físico, una pared, un mueble, etc.) y permanecerán así durante toda la simulación. Otras celdas serán, sin embargo, `navegables`, y son las que serán susceptibles de ser ocupadas por una persona. Por lo tanto, estas últimas podrán estar generalmente en uno de dos estados: `libre` u `ocupada`. Adicionalmente a lo anterior, algunas celdas estarán etiquetadas como `salida` y representan los puntos a través de los cuales las personas pueden salir del entorno. Dicho de otra forma, alcanzar una de estas celdas por parte de una persona equivale a que ha sido satisfactoriamente evacuada.

Inicialmente, se sitúa un número n configurable de personas repartidas por el entorno. Cada una de estas personas ocupará una celda libre y podrá desplazarse a otras celdas libres adyacentes (en nuestro caso, se considera la vecindad de Moore, es decir, las ocho celdas que rodean la casilla actual) en cada iteración del autómata celular. Más concretamente, se realiza un barrido aleatorio por la multitud [21] (para evitar situaciones patológicas y ciclos en la simulación), y cada una de las personas determina un movimiento en dicho orden. La lógica del movimiento es como sigue [2]:

- Existe un campo estático $\mathcal{A}$ definido sobre todas las celdas de la rejilla. Dicho campo captura la distancia de cada casilla a la salida más cercana. Dicha distancia se calcula mediante el algoritmo de Dijkstra antes de la ejecución del autómata, y su inversa (normalizada al intervalo $[0, 1]$) define la atracción $\mathcal{A}_{ij}$ de cada celda. Véase la Fig. 1.
- Existe asimismo un campo dinámico R cuya naturaleza es, en este caso, repulsiva. Dicho campo está también normalizado, en este caso al intervalo

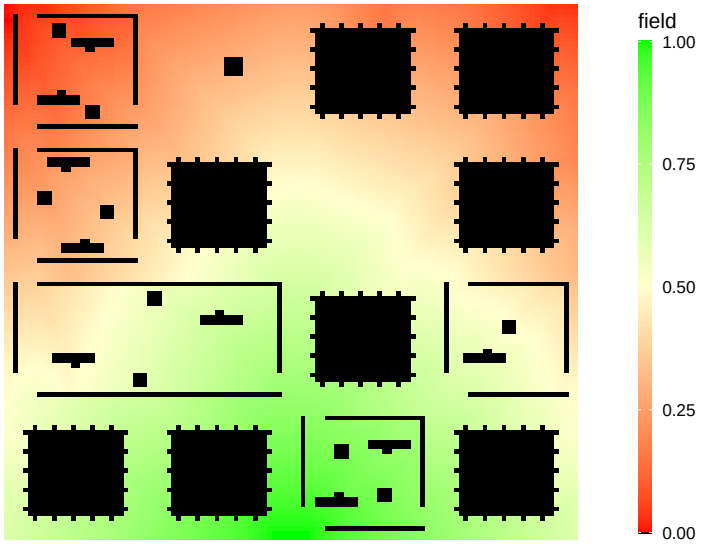


Figura 1. Representación del campo atractivo en un entorno que representa una oficina. Hay una única salida (no mostrada explícitamente) situada en la parte central del lado inferior del entorno.

$[1/\eta, 1]$, siendo 1 el valor correspondiente a una casilla completamente rodeada de posiciones no transitables (bloqueadas u ocupadas por otra persona), y $1/\eta$ (donde η es el tamaño de la vecindad, $\eta = 8$ en nuestro caso) corresponde a que todas las posiciones adyacentes están libres.

- La atracción y la repulsión representan dos vertientes del comportamiento de la persona, objetivo y subjetivo, respectivamente. Ambos se combinan dentro de una única medida de *deseabilidad* a través de la fórmula

$$\mathcal{D}_{ij} = \exp(\phi \mathcal{A}_{ij} - \psi \mathcal{R}_{ij})$$

  donde ϕ y ψ son coeficientes que ponderan la importancia de cada factor.
- Una vez conocida la deseabilidad de todas las celdas a las que una persona puede moverse, se determina el exceso de deseabilidad de cada una (la diferencia con el peor valor entre las mismas), y se elige un destino de manera proporcional a dicho exceso.

El proceso anterior se repite para todas las personas (excepto aquellas que, lógicamente, lograron salir del entorno en pasos anteriores). La simulación termina al alcanzar un tiempo de evacuación máximo $T_{\text{máx}}$, momento en el que se puede evaluar la calidad de la evacuación. El código completo del modelo está disponible en nuestro repositorio de código[3].

[3] https://github.com/Bio4Res/pedestrian-evacuation-simulator-cellular-automaton

4. Diseño e implementación

El sistema de visualización de las simulaciones ha sido desarrollado en Unity, el motor de creación líder en la industria del desarrollo de videojuegos[4]. La arquitectura principal utilizada es una arquitectura basada en notificaciones, puesto que se requiere una interacción con el usuario en todo momento, independientemente de si se está ejecutando la simulación o no. Un beneficio adicional es el asincronismo que proporciona, permitiendo ejecutar de forma inmediata las tareas (*callbacks*) solicitadas por el usuario. Para ello, se hace uso de los delegados, eventos y acciones de Unity, aplicando en múltiples ocasiones distintos patrones de diseño [10]. Por ejemplo, el Observador (*Observer*) en casos como la apertura del explorador de archivos y la selección de una ruta en este, el cambio de vista o la interacción con cualquier elemento de la interfaz que requiera una notificación inmediata (como los botones o la barra que indica la velocidad de la simulación).

También se ha empleado el patrón Constructor (*Builder*). Así, los parámetros tanto del autómata celular como de los Agentes (Peatones) se reciben mediante una clase inicializada a través de una clase *Builder*, ya que permite asignar únicamente los parámetros deseados desde el principio.

Por último, para la creación de los agentes se ha utilizado el patrón de factoría (*Factory*), ya que en el futuro estos podrían presentar características diferenciadas (e.g., distintas velocidades de desplazamiento o edades).

5. Visualización e interacción con el usuario

Un elemento central de la herramienta de visualización es la definición del escenario de simulación, es decir, la especificación de la geometría del entorno donde se pretende evacuar personas (véase la Sect. 3). Cada celda puede contener un obstáculo, una salida o simplemente estar vacía, mostrados en color rojo, verde o blanco, respectivamente. Estos elementos se diferencian de los peatones, representados en color azul, tal como se puede apreciar en la Fig. 2. A nivel de funcionalidad, existen dos maneras de cargar un escenario en la herramienta: mediante un fichero JSON que especifica su estructura o por generación aleatoria. En ambos casos, la herramienta adapta la visualización al sistema de cálculo de distancias entre celdas. Esto es necesario para que los movimientos de los peatones dependan de dicha distancia, haciendo que las celdas almacenen un orden de prioridad y apoyándose en la vecindad de Moore para indicar cuáles son las celdas vecinas [6].

Internamente, cada escenario está definido en un fichero JSON que contiene la especificación de los obstáculos y las puertas de salida, además de la altura y anchura de estos. Para futuras extensiones, la arquitectura contempla que estos elementos pueden tener formas distintas (e.g., rectángulo, círculo, etc.). En todos los casos, las posiciones de las celdas se transforman para que concuerden con el sistema del autómata celular original.

[4] https://unity.com/

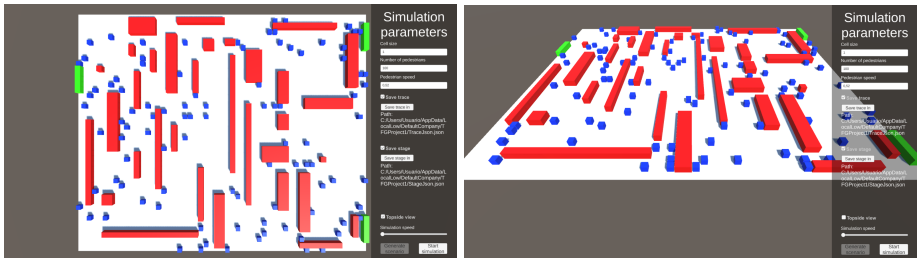


Figura 2. (Izquierda) Vista cenital. (Derecha) Vista libre

La herramienta está diseñada para que todo el flujo de interacción del usuario con el simulador se lleve a cabo de forma intuitiva. Así, lo primero que el usuario puede observar en esta pantalla es una interfaz/panel situada a la derecha con diversos elementos interactivos (véase las Figs. 2 y 3), que contienen, entre otros, parámetros de la simulación que deben definirse antes de iniciarla (e.g., el tamaño de las celdas, el número de peatones y la velocidad de estos). En este panel se encuentran igualmente otros elementos como las opciones de guardado y carga tanto del escenario como de la simulación de los movimientos de los agentes, la elección del modo de visualización (véase la Fig. 2), la generación aleatoria de escenarios y la configuración de la velocidad de reproducción de la simulación, entre otros.

Profundizando en los modos de visualización, existen dos vistas posibles: vista cenital y vista libre. La *Vista Cenital* está controlada por una cámara que observa la simulación desde arriba con un ángulo de 90 grados. La cámara puede desplazarse mediante las teclas de dirección o WASD, y permite hacer zoom con el ratón. En la *Vista libre*, la cámara se sitúa inicialmente en una posición que proporciona una imagen en perspectiva. La cámara puede desplazarse y rotarse en cualquier dirección, pero no se dispone de zoom. Esta cámara puede rotarse mediante el movimiento del ratón, permitiendo observar la simulación desde cualquier ángulo deseado.

Una vez terminada la simulación, aparece un panel que muestra algunas de las estadísticas más relevantes, como la media y la mediana del número de pasos realizados por los agentes, la media y la mediana del tiempo de evacuación, así como el número de evacuados y no evacuados (véase la Fig. 3).

6. Caso de estudio: optimización del emplazamiento de salidas de emergencia

Como caso de estudio, consideramos la optimización de la colocación de las puertas de emergencia en espacios cerrados en los cuales hay un movimiento dinámico y constante de personas [6]. La colocación óptima de esas puertas facilita la evacuación de personas que se encuentran dentro de espacios cerrados (i.e., los escenarios) ante situaciones de emergencia. En este sentido, la función objetivo

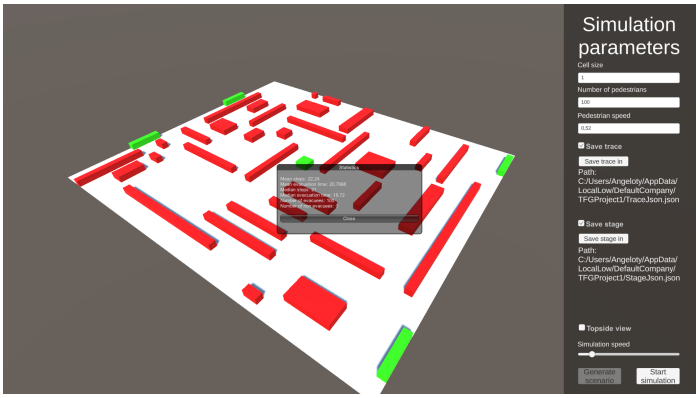


Figura 3. Estadísticas principales de la simulación

trata de cuantificar la bondad de un diseño particular (un emplazamiento específico de las salidas de emergencia para un cierto entorno) mediante una serie de objetivos estratificados:

- El objetivo principal es minimizar el número de personas que permanecen en el recinto tras un tiempo máximo $T_{\text{máx}}$.
- En caso de que la evacuación sea completamente satisfactoria (todo el mundo logra abandonar el recinto), se intenta minimizar el tiempo de salida de la última persona y, secundariamente, el tiempo promedio de evacuación.
- En caso de que la evacuación no sea completamente satisfactoria (haya peatones que no logren salir), se intenta minimizar la distancia a la salida más cercana de cualquier persona en el interior y, secundariamente, la distancia promedio.

Para resolver el problema de optimización así definido se emplea un algoritmo evolutivo (EA) con la configuración indicada en [6] (i.e., $maxevals = 20000$, $\mu = 100$, $p_X = 0{,}9$, $p_M = 1/\ell$, donde ℓ es el número de puertas, y amplitud de mutación $\gamma = 0{,}05$). La función objetivo es la anteriormente descrita, tomando valores medios sobre un conjunto de m configuraciones iniciales de la multitud (las mismas en todas las evaluaciones, para poder establecer una comparativa válida). Otros parámetros de la simulación son tiempo máximo $T_{\text{máx}} = 60$s, tamaño de celda $c = 0{,}5$m, parámetros de deseabilidad $\phi \in [1{,}5, 2], \psi \in [0{,}5, 1]$, y velocidad de los peatones $v = 1{,}3$m/s.

Como ejemplo de aplicación, se han considerado instancias de un conjunto de datos públicos [7]; véase la Fig. 4. El EA se emplea para optimizar el emplazamiento de tres puertas de emergencia. La Fig. 5 (Izquierda) muestra la evolución del fitness en el EA con respecto al número de evaluaciones. La Fig. 5 (Derecha) ilustra la mejor solución encontrada por el EA, donde cada una de las tres puertas de emergencia se representa en color verde.

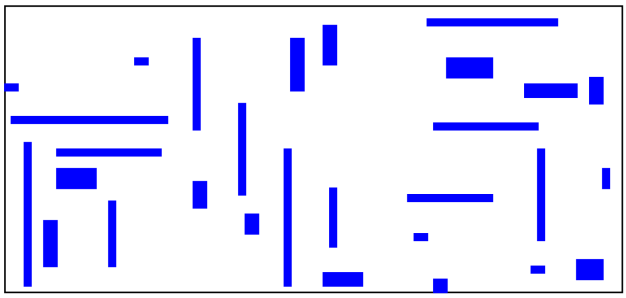


Figura 4. Escenario base para el EA

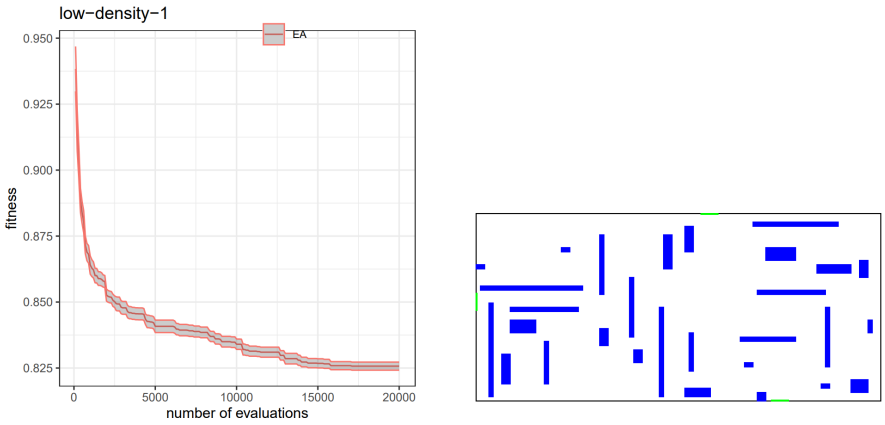


Figura 5. (Izquierda) Los mejores fitness del EA. (Derecha) La mejor solución encontrada.

6.1. Modelo subrogado

Uno de los grandes problemas principales de los métodos de optimización bioinspirada (MOBs) en el problema descrito en este artículo es su elevado coste computacional, debido principalmente a la simulación del movimiento de personas. Por ello, probamos a sustituir el cálculo de la calidad, es decir, el valor de fitness en los MOBs empleados, por modelos subrogados automáticos [13].

A continuación, describimos resultados preliminares en esta dirección. Se ha considerado el diseño y desarrollo de modelos subrogados mediante técnicas de aprendizaje. En concreto, hemos empleado una red neuronal basada en aprendizaje supervisado. La red se compone de 5 capas: la de entrada, 3 ocultas y la de salida. La capa de entrada está formada por 3 neuronas, una por cada salida de emergencia. Cada capa oculta se compone de 512 neuronas (elección realizada de forma intuitiva). La capa de salida es de una neurona asociada al valor de fitness de la calidad de la simulación, considerando las tres puertas de entrada como la

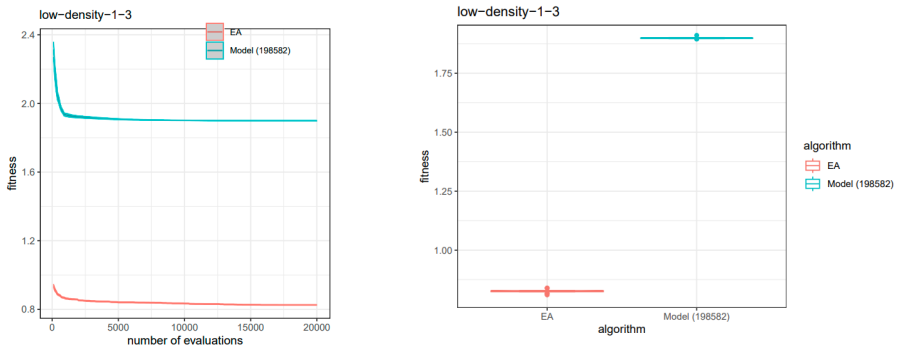


Figura 6. (Izquierda) Los mejores fitness de cada algoritmo. (Derecha) Los fitness de las mejores soluciones de cada algoritmo.

solución a la instancia del problema. La idea es que la red nos defina una función que devuelva un valor de fitness cercano al que generaría la simulación, mediante el autómata celular descrito en la Sección 3, del movimiento de peatones en ese escenario y con las puertas de emergencia colocadas en él. El objetivo es reducir el coste computacional de la simulación a una única llamada a una función.

Como función de activación, empleamos la Unidad Lineal Rectificada (ReLU) en toda la red neuronal. Como función de optimización, de pérdida y de rendimiento se utilizan, respectivamente, la Propagación de Raíz Cuadrática Media (RMSProp), el Error Cuadrático Medio (mse) y el Error Absoluto Medio (mae). El modelo se entrena usando un conjunto de datos para el entrenamiento y otro para la validación.

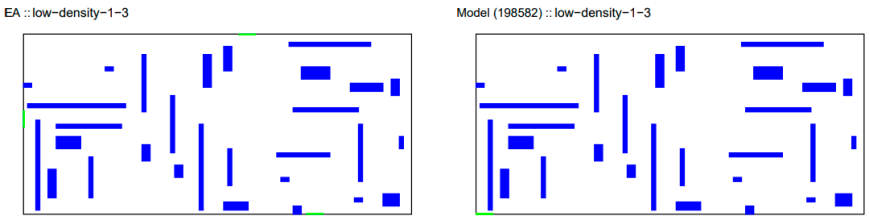
Se ha considerado una instancia específica del problema (o sea, sobre un único escenario prefijado denominado 'low-density-1-3') para valorar la efectividad que tiene el modelo subrogado con respecto al rendimiento de nuestro EA. La comparación se realiza sobre 20 ejecuciones de cada algoritmo. Las configuraciones del EA y de la simulación coinciden con las descritas anteriormente. En el caso de la configuración del modelo subrogado, como el modelo trata de imitar al EA, se usa la misma configuración para este. La instancia sobre la que se ha realizado la comparación es la mostrada en la Fig. 4.

Los resultados de cada algoritmo para la búsqueda de la colocación óptima de 3 puertas se muestran en la Figura 6. A la izquierda, observamos la evolución del fitness conforme el número de evaluaciones aumenta; A la derecha, se encuentran las distribuciones de los valores de fitness de las mejores soluciones entre todas las evaluaciones.

La Fig. 7 contiene una tabla que muestra la media y la desviación típica. Hemos aplicado igualmente el test de Wilcoxon [10]. En la tabla, el símbolo de una estrella ($\star$) identifica la mejor solución por instancia, y el símbolo de ■ indica que existen diferencias estadísticamente significativas entre los algoritmos al nivel de $\alpha = 0,01$. Observamos que el EA nos proporciona la mejor solución

	EA				Model (198582)			
instance	x^*	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$		x^*	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$	
low-density-1-3	0.81	0.827	0.826 ± 0.002	$\star$	1.895	1.899	1.899 ± 0.001	■

Figura 7. Tabla con estadisticas de cada algoritmo en cada instancia



(a) (b)

Figura 8. La mejor solución encontrada por cada algoritmo en la instancia del problema

y que existen diferencias significativas. Finalmente, la Fig. 8 visualiza la mejor solución encontrada por cada algoritmo en esa instancia del problema.

7. Conclusiones

Hemos presentado un marco de trabajo integrado para la optimización de problemas de evacuación de multitudes en recintos interiores. La propuesta combina un modelo de simulación basado en autómatas celulares probabilísticos, técnicas de optimización evolutiva y un entorno de visualización interactivo desarrollado en Unity. Esta integración permite no solo evaluar automáticamente distintas configuraciones de diseño, sino también facilitar el análisis visual de los resultados y la participación activa de expertos humanos en el proceso de toma de decisiones. De este modo, la herramienta constituye una plataforma flexible para el estudio de estrategias de evacuación y para el análisis de escenarios complejos en los que intervienen múltiples factores geométricos y comportamentales.

Como demostración de las capacidades del sistema, se ha considerado el problema de la optimización del emplazamiento de salidas de emergencia mediante algoritmos evolutivos. Asimismo, se ha explorado de forma preliminar el uso de modelos subrogados con el objeto de reducir el coste computacional asociado a la evaluación basada en simulación. Aunque los resultados obtenidos muestran que esta aproximación todavía requiere un mayor desarrollo para alcanzar el rendimiento del algoritmo evolutivo original, ponen de manifiesto el potencial

de la plataforma como banco de pruebas para la experimentación con nuevas estrategias de optimización. Como líneas futuras de trabajo se contempla la incorporación de modelos de comportamiento más sofisticados, el soporte para escenarios de mayor escala y el desarrollo de mecanismos avanzados de asistencia a la decisión que refuercen el enfoque *human-in-the-loop* que inspira este marco de trabajo.

Agradecimientos. Este trabajo cuenta con el apoyo parcial del Ministerio de Economía de España, en el marco del proyecto Bio4Res (PID2021-125184NB-I00 – http://bio4res.lcc.uma.es), y de la Universidad de Málaga, a través del Campus de Excelencia Internacional Andalucía Tech.

Declaración de Intereses. Los autores no tienen ningún conflicto de intereses que declarar que sea relevante para el contenido de este artículo.

Referencias

1. Bellomo, N., Bellouquid, A., Knopoff, D.: From the Microscale to Collective Crowd Dynamics. Multiscale Modeling & Simulation **11**(3), 943–963 (Jan 2013). https://doi.org/10.1137/130904569
2. Bueno, D., Cotta, C., Gallardo, J.: Using an agent-based cellular automaton for emergency evacuation simulation. In: Mathieu, P., la Prieta, F.D. (eds.) Practical Applications of Agents and Multi-Agent Systems. Lecture Notes in Artificial Intelligence, vol. 15157, pp. 324–329. Springer Nature, Switzerland (2024). https://doi.org/10.1007/978-3-031-70415-4_28
3. Cao, R.F., Lee, E.W.M., Xie, W., Gao, D.L., Chen, Q., Yuen, A.C.Y., Yeoh, G.H., Yuen, R.K.K.: Development of an agent-based indoor evacuation model for local fire risks analysis. Journal of Safety Science and Resilience **4**(1), 75–92 (Mar 2023). https://doi.org/10.1016/j.jnlssr.2022.09.006
4. Chen, J., Shi, T., Li, N.: Pedestrian evacuation simulation in indoor emergency situations: Approaches, models and tools. Safety Science **142**, 105378 (2021). https://doi.org/https://doi.org/10.1016/j.ssci.2021.105378
5. Cotta, C.: Optimization-driven workplace evacuation using evolutionary algorithms and derivative-free methods. In: 8th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence. pp. 86–91. ACM, Singapore (2024). https://doi.org/10.1145/3665065.3665079
6. Cotta, C., Gallardo, J.: Evolutionary algorithms for optimizing emergency exit placement in indoor environments. In: Smith, S., Correia, J., Cintrano, C. (eds.) Applications of Evolutionary Computation – EvoApplications 2024. Lecture Notes in Computer Science, vol. 14634, pp. 194–208. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-56852-7_13
7. Cotta, C., Gallardo, J.E.: Instance dataset for the pedestrian evacuation problem (Jan 2025). https://doi.org/10.17605/OSF.IO/CNH7U, osf.io/cnh7u
8. Diffey, B.: An overview analysis of the time people spend outdoors. British Journal of Dermatology **164**(4), 848–854 (2011). https://doi.org/10.1111/j.1365-2133.2010.10165.x
9. Dong, H., Zhou, M., Wang, Q., Yang, X., Wang, F.Y.: State-of-the-Art Pedestrian and Evacuation Dynamics. IEEE Transactions on Intelligent Transportation Systems **21**(5), 1849–1866 (May 2020). https://doi.org/10.1109/TITS.2019.2915014

10. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley (1994)
11. Golas, A., Narain, R., Lin, M.C.: Continuum modeling of crowd turbulence. Physical Review E **90**(4), 042816 (Oct 2014). https://doi.org/10.1103/PhysRevE.90.042816
12. Haghani, M.: Optimising crowd evacuations: Mathematical, architectural and behavioural approaches. Safety Science **128**, 104745 (Aug 2020). https://doi.org/10.1016/j.ssci.2020.104745
13. He, C., Zhang, Y., Gong, D., Ji, X.: A review of surrogate-assisted evolutionary algorithms for expensive optimization problems. Expert Systems with Applications **217**, 119495 (May 2023). https://doi.org/10.1016/j.eswa.2022.119495
14. Huang, K., Zheng, X., Yang, Y., Wang, T.: Behavioral evolution in evacuation crowd based on heterogeneous rationality of small groups. Applied Mathematics and Computation **266**, 501–506 (2015). https://doi.org/10.1016/j.amc.2015.05.065
15. Juan, A., et al.: A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. Operations Research Perspectives **2**, 62–72 (Dec 2015). https://doi.org/10.1016/j.orp.2015.03.001
16. Klepeis, N.E., Nelson, W.C., Ott, W.R., Robinson, J.P., Tsang, A.M., Switzer, P., Behar, J.V., Hern, S.C., Engelmann, W.H.: The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants. Journal of Exposure Science & Environmental Epidemiology **11**(3), 231–252 (2001). https://doi.org/10.1038/sj.jea.7500165
17. Kouskoulis, G., Spyropoulou, I., Antoniou, C.: Pedestrian simulation: Theoretical models vs. data driven techniques. International Journal of Transportation Science and Technology **7**(4), 241–253 (Dec 2018). https://doi.org/10.1016/j.ijtst.2018.09.001
18. Li, Z., Xu, C., Bian, Z.: A force-driven model for passenger evacuation in bus fires. Physica A: Statistical Mechanics and its Applications **589**, 126591 (Mar 2022). https://doi.org/10.1016/j.physa.2021.126591
19. Louis, P.Y., Nardi, F.R. (eds.): Probabilistic Cellular Automata: Theory, Applications and Future Perspectives, Emergence, Complexity and Computation, vol. 27. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-65558-1
20. Martinez-Gil, F., Lozano, M., García-Fernández, I., Fernández, F.: Modeling, Evaluation, and Scale on Artificial Pedestrians: A Literature Review. ACM Computing Surveys **50**(5), 72:1–72:35 (Sep 2017). https://doi.org/10.1145/3117808
21. Schönfisch, B., De Roos, A.: Synchronous and asynchronous updating in cellular automata. Biosystems **51**(3), 123–143 (Sep 1999). https://doi.org/10.1016/S0303-2647(99)00025-8
22. Shi, M., Lee, E.W.M., Ma, Y.: A dynamic impatience-determined cellular automata model for evacuation dynamics. Simulation Modelling Practice and Theory **94**, 367–378 (Jul 2019). https://doi.org/10.1016/j.simpat.2019.04.003
23. Zheng, Y., Li, X.G., Jia, B., Jiang, R.: Simulation of pedestrians' evacuation dynamics with underground flood spreading based on cellular automaton. Simulation Modelling Practice and Theory **94**, 149–161 (Jul 2019). https://doi.org/10.1016/j.simpat.2019.03.001