

Crowd evacuation: Analysis with Artificial Intelligence techniques

Kevin Novoa Franco, and Javier Andrés Agreda

Abstract— *The crowd evacuation simulation is a very interesting area in the scientific activity, because it presents a situation in which multiple variables should be evaluated in order to avoid setbacks that endanger each human who attends the event. To avoid this problem, the use of multi-agent systems to simulate the human beings' behavior in a specific context, allows the analysis of the interaction of the event attendee with the environment and the impact of their decisions on it.*

In this paper, it applies artificial intelligence to this problem by means of the optimization and learning techniques: Neural Networks and Decision Trees, in a multi-agent system based simulation implemented with the BESA framework. The comparative results of using each technique are presented by means of an experiment that concludes with the definition of the best technique to solve the problem.

Keywords— Intelligent systems, artificial intelligence, evacuation simulation, machine learning, neural networks, multi-agent systems, Decision trees, C4.5, weka.

I. INTRODUCCIÓN

El movimiento de multitudes humanas es conducido principalmente por un proceso auto-organizado basado en las interacciones locales entre los peatones [1]. Esta problemática, se ha convertido en un área de estudio que abarca temas desde el pronóstico del comportamiento humano en situaciones de emergencia hasta física de partículas y análisis sociológicos. Al parecer, el primero en modelar multitudes fue el francés Eric Bouvier en 1983, basado en un sistema de partículas.

La simulación de multitudes implica el modelar el comportamiento de un gran número de objetos o personas, esto tomando en cuenta que la interacción del comportamiento humano observado se debe replicar en el comportamiento colectivo. En este contexto, Curtis, emplea un método que parte del precepto en el cual los parámetros peatonales constantes no son suficientes para capturar este fenómeno complejo [2]. Además, la mayoría de los sistemas simulan un gran número de individuos con el mismo comportamiento, algunos ofrecen una limitada variedad de comportamientos basados en la edad y el género, pero la mayor parte de los sistemas actuales carecen de una amplia variedad de comportamientos basados en formas de actuar validadas demográfica y culturalmente, como lo propone Pelechano [3].

Para proponer una solución a esta problemática, en esta investigación se desarrolla un sistema multi-agente, SMA por sus siglas, con un modelo de toma de decisiones obtenido mediante técnicas de inteligencia artificial, IA por sus siglas, que simule el comportamiento de un grupo multitudinario de personas en el proceso de evacuación del lugar de realización de un evento, el cual puede contar con diversas configuraciones

de las vías de egreso. En el alcance, no se contempla personas lesionadas durante el desalojo ni daño de la edificación, por lo que no se requerirá del auxilio a personas para efectuar el desalojo, mediante camillas, sillas de ruedas o que sean cargados o arrastrados por otros, esto se convierte en el trabajo futuro del proyecto.

El presente artículo inicia con un análisis orientado a agentes. Después, se analiza la problemática por medio de las técnicas de Inteligencia Artificial: Redes Neuronales Artificiales y Árboles de Decisión, que se aplican al algoritmo de toma de decisiones de los agentes que representan a las personas. Para finalizar, se concluye mediante un análisis comparativo cuál es la mejor aproximación para la optimización de esta solución. Cabe aclarar, que el presente artículo es una extracción de un análisis con dos técnicas más: Algoritmos Genéticos y Lógica Difusa, el cual es un análisis más extenso de la problemática comparativa de técnicas IA de los mismos autores.

II. ANÁLISIS ORIENTADO A AGENTES

Primero, se analiza el problema mediante la metodología AOPOA[6] obteniendo un sistema de simulación para evacuación de eventos con asistentes y personal de logística[4], pero del cual se simplifica solo al estudio del rol de asistente, para analizar solo los efectos de la inteligencia artificial y no el de la cooperación de agentes, resumiéndolo en la tabla 1.

Rol	Asistente.
Objetivo	Salir del evento.
Habilidades	Fijar Ruta, Cambiar Ruta, Conocer Distancia de Puerta, Conocer Densidad de Puerta, Conocer Casillas Vacías Alrededor.
Recursos	Puertas: Metas a alcanzar por los asistentes, Fija y Determinada. Casillas: Espacio que puede ser ocupado por un agente o por un obstáculo. Fija y Determinada. Conflicto: Entre dos asistentes la resolución se realiza mediante la velocidad del agente.
Otras entidades	Obstáculo: ocupa casillas durante todo el evento.

Tabla 1 Resumen Análisis de Agentes.

En el SMA, el evento se diseñó como una matriz de casillas, la cual se parametriza mediante variables que ingresa el usuario: alto, ancho, cantidad de asistentes, intención de acercarse, secciones del evento o localidades, tiempo máximo y mínimo de respuesta de cada agente, ubicación de puertas y ubicación de obstáculos. Luego, se asigna un mapa de pesos

por cada puerta para cada casilla, cuyo peso es la medida con la cual se determina la cantidad de pasos que hay que dar para que un agente llegue a la puerta, esta se obtiene mediante la máxima diferencia entre “x” o “y” de la casilla actual a la casilla más cercana de la salida. Otro término importante es la densidad, la cual se refiere a la relación porcentual entre casillas ocupadas y las totales en un radio de 4 casillas alrededor de salida (que tienen peso de 1 a 4).

En la figura 1, se puede ver un mapa de simulación con un área de evento de 30x50 casillas con 3 puertas, una a cada lado y una atrás la cual se encuentra centrada, hay un pequeño obstáculo ubicado a 4 casillas de la salida. Los colores de los agentes, varían según su tiempo de respuesta.

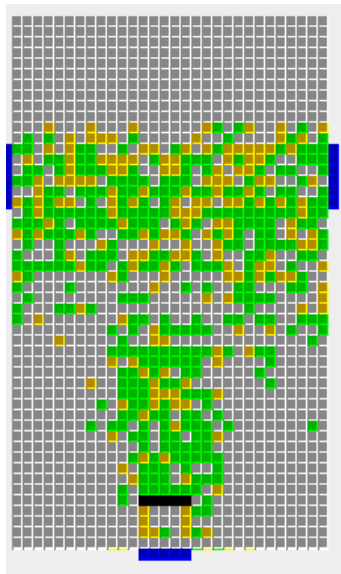


Figura 1. Mapa de simulación

Se crean y organizan los agentes en el entorno utilizando el *framework* BESA [8]. Para su localización, se modelan distintos tipos de distribución aleatoria en todo el área del evento o más o menos cerca del área superior (como si fuera una tarima). Después, en el momento en que el agente selecciona la puerta de salida se le crea el comportamiento de salir del evento.

El comportamiento se define por medio de los siguientes estados del agente inteligente:

- **Selección de puerta:** método por el cual se escoge la puerta de salida. Esta puede ser la que está más cerca o aleatoriamente.
- **Momentos de espera:** momentos que espera el agente antes de cambiar de puerta cuando no se puede mover debido a un obstáculo. Puede esperar entre 1 o 4 momentos.
- **Dirección del cambio:** método por el cual se escoge la dirección para cambiar de casilla cuando se encuentra ante un obstáculo. Puede seleccionar la de menor relación densidad-peso, la de menor densidad, la de menor peso o simplemente una puerta aleatoria.

- **Selección de casilla:** método por el cual escoge la siguiente casilla. Esta puede ser en orden descendente o aleatorio.
- **Densidad máxima:** valor porcentual máximo para que el agente continúe hacia la puerta. Este valor se comprende entre 40 y 70 por ciento.
- **Tiempo de respuesta:** tiempo que tarda el agente en reaccionar frente a sus acciones. Se asigna aleatoriamente y no cambia durante la simulación.

El comportamiento de salir del agente se ejecuta mediante una temporización, cuyo tiempo de iteración es el tiempo de respuesta que se le agregó aleatoriamente como estado al agente y cuya variable de estado es la única que no va a depender del modelo de toma de decisiones. El agente, tiene la función de buscar una casilla siguiente teniendo en cuenta sus demás estados, por ejemplo: cuando se dirige a una puerta evalúa si esta está densa según su estado de comportamiento, en ese caso cambia de curso según la dirección de cambio que ha seleccionado, y así con sus demás estados de comportamiento.

El conflicto se maneja mediante la sincronización del atributo de la casilla que determina si está ocupada o no, de tal manera que si varios agentes pretenden la misma casilla, la seleccionara quien tenga menor tiempo de respuesta.

III. TÉCNICA DE IA: REDES NEURONALES ARTIFICIALES

El modelo de Redes Neuronales, RN por sus siglas, a utilizar para solución propuesta es: perceptrón simple. En la figura 2 se exponen las siguientes variables en las neuronas siguiendo el modelo:

- **Variables de entrada:**
 - Distancia: cantidad de casillas entre la posición actual del agente y la puerta de salida.
 - Velocidad: Cantidad de milisegundos en que un agente tarda en dar el siguiente paso: 500ms, 1000ms y 1500ms.
- **Variables de salida:** creencias del agente (Selección de puerta, Momentos de espera, Dirección del cambio, Selección de casilla, Densidad máxima y Tiempo de respuesta).

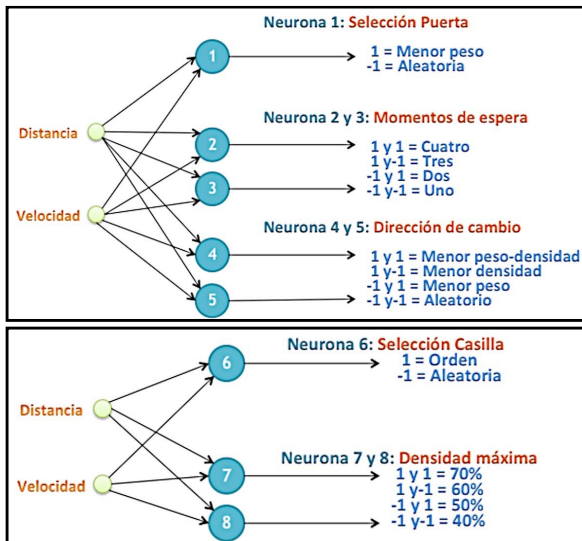


Figura 2. Neuronas del modelo.

Se establece el set de ejemplos para la simulación, que se puede ver en la figura 3.

DISTANCIA	VELOCIDAD	SELECCIÓN DE PUERTA	MOMENTOS DE ESPERA	MOMENTOS DE ESPERA	DIRECCIÓN DE CAMBIO	DIRECCIÓN DE CAMBIO	SELECCIÓN DE CASILLA	DENSIDAD MÁXIMA	DENSIDAD MÁXIMA
5	500	1	1	-1	-1	1	-1	1	1
5	1000	1	1	1	-1	-1	-1	1	1
5	1500	1	1	1	-1	1	-1	1	1
10	500	1	1	-1	1	1	-1	1	1
10	1000	1	1	1	1	1	-1	1	1
10	1500	1	1	1	1	1	-1	1	1
15	500	1	1	-1	1	1	-1	1	1
15	1000	1	1	1	1	1	-1	1	1
15	1500	-1	1	1	1	1	-1	1	1
20	500	-1	-1	-1	1	-1	-1	-1	1
20	1000	-1	-1	1	1	-1	-1	-1	1
20	1500	-1	-1	1	1	-1	-1	-1	1
25	500	-1	-1	-1	1	1	-1	-1	-1
25	1000	-1	-1	1	1	-1	1	-1	-1
25	1500	-1	-1	1	1	-1	1	-1	-1

Figura 3. Ejemplos para entrenamiento.

Función de activación: Recibe el valor de las dos entradas definidas y el umbral (-1) y retorna 1 o -1 como lo define la regla de la figura 4.

$$y = \begin{cases} 1 & \text{si } w_1x_1 + w_2x_2 \geq \theta \\ 0 & \text{si } w_1x_1 + w_2x_2 < \theta \end{cases}$$

Figura 4. Función de activación.

Función de aprendizaje: Se utiliza la función de la figura 5, en donde los resultados fueron satisfactorios: en la iteración 20 del experimento, ya que el error llega a cero por ciento.

$$w(n+1) = w(n) + \eta[d(n) - y(n)]x(n)$$

$$d(n) = \begin{cases} +1 & \text{if } x(n) \text{ belongs to class } \mathcal{C}_1 \\ -1 & \text{if } x(n) \text{ belongs to class } \mathcal{C}_2 \end{cases}$$

Figura 5. Función de cálculo de pesos (aprendizaje).

```
public void calculaPesos(float[] entradas, float salidas, float calculado) {
    for (int i = 0; i < entradas.length; i++) {
        this.pesos[i] = pesos[i] + (factorAprendizaje) * (salidas - calculado) * entradas[i];
    }
}
```

Figura 6. Función de activación implementada en código Java.

Función de entrenamiento: En la figura 7, se muestra la función de entrenamiento (implementada en código java).

```
public float funcionEntrenamiento(float[] entradas, float salida) {
    entradas[0] = entradas[0] / 25;
    entradas[1] = entradas[1] / 1500;
    float funcionEntrenamiento = 0.0f;
    //System.out.println("metodo funcionActivacion");
    for (int i = 0; i < entradas.length; i++) {
        // se multiplica cada peso por cada entrada y se suma
        funcionEntrenamiento += pesos[i] * entradas[i];
    }
    //se determina el valor de la salida
    if (funcionEntrenamiento >= 0) {
        funcionEntrenamiento = 1;
    } else if (funcionEntrenamiento < 0) {
        funcionEntrenamiento = -1;
    }

    if (salida != funcionEntrenamiento) {
        error = 1;
        calculaPesos(entradas, salida, funcionEntrenamiento);
    } else {
        error = 0;
    }
    return funcionEntrenamiento;
}
```

Figura 7. Función de entrenamiento (código java).

IV. TÉCNICA DE IA: APRENDIZAJE DE MÁQUINA POR MEDIO DE ÁRBOLES DE DECISIÓN

La aplicación de la técnica de Aprendizaje de Máquina se realiza por medio de Árboles de Decisión, ya que es una solución que se ajusta al modelo del SMA manteniendo un fundamento comparativo frente a la otra técnica de IA anterior que analiza esta investigación.

La técnica de aprendizaje de máquina seleccionada es Árboles de Decisión de tipo C4.5[12]. Para el modelamiento de la solución, se definieron las mismas variables de entrada: *Distancia* y *Velocidad*. Para la definición de los árboles, se tiene en cuenta el mismo set de ejemplos generado en el capítulo anterior, de los cuales se obtuvo un conjunto de árboles de decisión que se puede ver en la figura 8.

Selección de puerta:

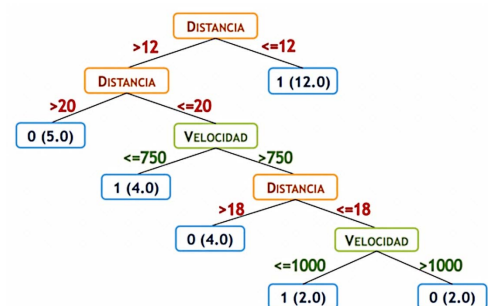
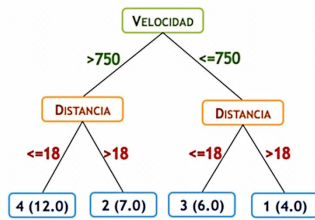
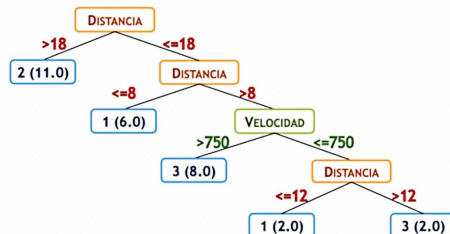


Figura 8. Conjunto de Árboles de Decisión 1

Momentos de espera:



Dirección de cambio:



Selección de casilla:



Densidad máxima:

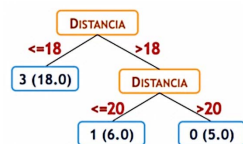


Figura 9. Conjunto de Árboles de Decisión 2

Para la implementación e integración del algoritmo de árboles de decisión C4.5, se utiliza la librería WEKA[14] desarrollada en lenguaje Java, lo que permite fácil integración con el SMA propuesto en esta investigación. En la figura 9, se muestra un ejemplo del mapeo de la información con WEKA para el estado de conducta *Selección de Puerta*:

<pre>@relation arbol @attribute distancia numeric @attribute velocidad numeric @attribute resultado {0,1,2,3,4} @data 5,500,1 5,1000,1 5,1500,1 10,500,1 10,1000,1 10,1500,1 15,500,1</pre>	<pre>Árbol: Selección de Puerta Entrenamiento: 38 Testing: 38 Accuracy of 348: 66,67% 348 pruned tree ----- distancia <= 12: 1 (12.0) distancia > 12 distancia <= 20 velocidad <= 750: 1 (4.0) velocidad > 750 distancia <= 18 velocidad <= 1000: 1 (2.0) velocidad > 1000: 0 (2.0) distancia > 18: 0 (4.0) distancia > 20: 0 (5.0)</pre>
--	---

Figura 10. Ejemplo de Entrenamiento y Resultado (WEKA+Java).

V. EXPERIMENTO Y ANÁLISIS DE RESULTADOS

Se realizó un experimento para cada técnica y posteriormente un experimento para un análisis comparativo. A todos los experimentos, se les realiza Análisis de Varianza, ANOVA por sus siglas en inglés, con diseños 2^k con la técnica de contrastes para obtener los efectos más significativos, todos se analizan con un nivel de significancia estadística de 5% con 3 réplicas.

En general, la variable de salida es el Tiempo Total de Evacuación. Además, Se tienen en cuenta las variables Área de Campo para dos niveles (-)30x17 y (+)30x30, así como Densidad de Agentes (-)40% y (+)70%, las cuales se espera que siempre arrojen un resultado significativo ya que inciden directamente en la variable dependiente. Finalmente, en cada experimento se busca variables específicas de cada tipo de algoritmo.

A. Experimento Análisis IA: Redes Neuronales Artificiales

El factor específico de análisis de las redes neuronales es la cantidad de ejemplos con niveles (-)15 y (+)30.

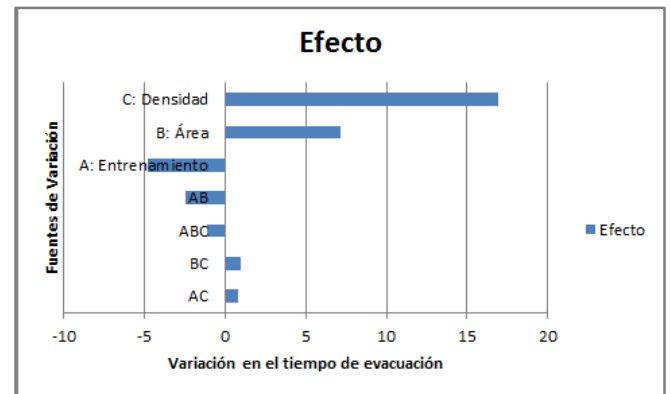


Figura 11. Efectos Redes Neuronales Artificiales.

ANOVA			
FUENTE	Fo	Ft :	0,05
A: Entrenamiento	4,929074432	4,49	RECHAZADA
B: Área	10,88599019	4,49	RECHAZADA
C: Densidad	61,1021245	4,49	RECHAZADA
AB	1,337097014	4,49	NO RECHAZADA
AC	0,148566335	4,49	NO RECHAZADA
BC	0,213935522	4,49	NO RECHAZADA
ABC	0,274788293	4,49	NO RECHAZADA

Tabla 2. ANOVA Redes Neuronales Artificiales.

Como se ve en la figura 10 y en el ANOVA de la tabla 2, los efectos más significativos para incrementar el tiempo son la densidad y el área, pero para reducirlo, la cantidad de datos de entrenamiento es significativa, sin ninguna interacción significativa. Por lo tanto, se puede concluir que entre más datos de entrenamiento se mejora el tiempo de salida de los agentes y se optimiza el algoritmo.

B. Experimento Análisis IA: Árboles de Decisión

Los factores específicos de análisis de Árboles de Decisión son: el número de datos de entrenamiento del árbol con niveles (-)15 y (+)30, así como el tipo de árbol el cuál puede ser (-)ID3 o (+)C4.5.

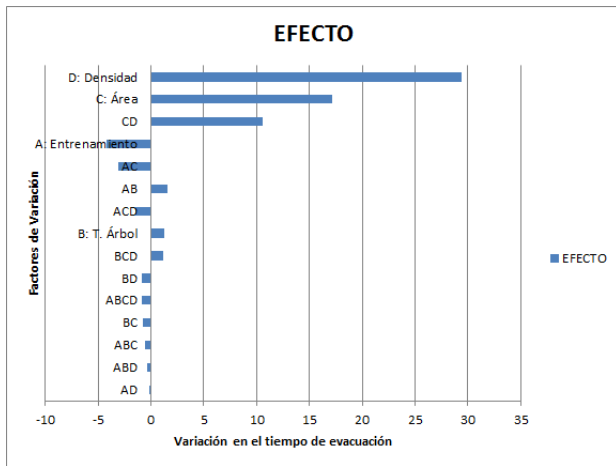


Figura 12. Efectos de Árboles de Decisión.

ANOVA			
FUENTE	Fo	F :	0,05
A: Entrenamiento	8,69	4,15	RECHAZADA
B: T. Árbol	0,84	4,15	NO RECHAZADA
C: Área	144,63	4,15	RECHAZADA
D: Densidad	426,08	4,15	RECHAZADA
AB	1,19	4,15	NO RECHAZADA
AC	4,79	4,15	RECHAZADA
AD	0,01	4,15	NO RECHAZADA
BC	0,30	4,15	NO RECHAZADA
BD	0,39	4,15	NO RECHAZADA
CD	55,50	4,15	RECHAZADA
ABC	0,18	4,15	NO RECHAZADA
ACD	1,07	4,15	NO RECHAZADA
ABD	0,06	4,15	NO RECHAZADA
BCD	0,72	4,15	NO RECHAZADA
ABCD	0,33	4,15	NO RECHAZADA

Tabla 3. ANOVA Árboles de Decisión.

Como se puede ver en la figura 11 y en el ANOVA de la tabla 3, los efectos más significativos para el incrementar el tiempo son la densidad, el área y en menor medida del tipo de árbol, mientras que la reducción está a cargo de la cantidad de términos de entrenamiento. Además, las interacciones entre datos de entrenamiento y el área, así como el área y densidad también son significativas. Por lo tanto, se puede concluir que entre mayor número de datos de entrenamiento se optimiza la evacuación de los agentes, y el tipo de árbol no es significativo.

C. Análisis comparativo de las 2 técnicas de IA

El factor específico de este experimento es la técnica de IA aplicada, este factor es de 2 niveles: (-) Redes Neuronales y (+) Árboles de Decisión.

ANOVA			
FUENTE	Fo	F :	0,05
A: Técnica IA	90,23	2,90	RECHAZADA
B: Área	57,41	4,15	RECHAZADA
C: Densidad	203,78	4,15	RECHAZADA
AB	4,30	2,90	RECHAZADA
AC	12,07	2,90	RECHAZADA
BC	21,66	4,15	RECHAZADA
ABC	4,32	2,90	RECHAZADA

Tabla 3. ANOVA Análisis Comparativo.

Según el ANOVA, todos los factores como son la técnica, el área, la densidad y todas las posibles interacciones son significativas. Principalmente, el factor de estudio relevante, la Técnica IA se puede expresar en la siguiente gráfica en la cual se muestran los valores de evacuación para cada combinación de valores de Área y Densidad que siempre demostraron ser significativos pasando del mínimo hasta el máximo en las dos variables.

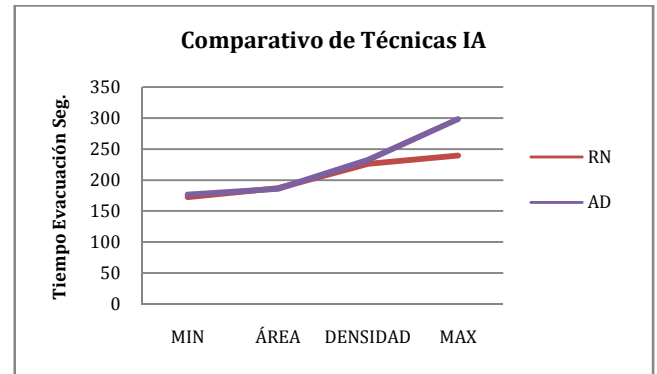


Figura 13. Análisis Comparativo.

Del gráfico del análisis comparativo, presentado en la figura 12, se puede concluir que para la optimización del sistema, la técnica de IA más estable es Redes Neuronales, aunque no existe una gran diferencia, pero se puede apreciar que los Árboles de Decisión incrementan su valor cuanto más se incrementa el área y la densidad, dando un resultado significativo en el experimento.

VI. CONCLUSIONES

En esta investigación se aplican dos técnicas de inteligencia artificial: Redes Neuronales Artificiales y Árboles de Decisión, con una simulación por medio de un SMA para el problema de optimización de Evaluación de Multitudes, realizando un análisis individual y comparativo de las técnicas. En el análisis con Redes Neuronales, entre más datos de entrenamiento mejor es el tiempo de salida de los agentes. En la aplicación de Árboles de Decisión se coincide que entre mayor número de datos de entrenamiento mejor es el tiempo de la evacuación de los agentes, mientras que el tipo de árbol no es un factor significativo.

En el análisis comparativo de las dos técnicas, se puede concluir que para la optimización del sistema, la técnica de IA más estable es Redes Neuronales, aunque no existe una gran diferencia, se puede apreciar que los Árboles de Decisión incrementan el tiempo de evacuación de los agentes cuando más se incrementa el área y la densidad.

REFERENCIAS

- [1] Moussaïd M, Perozo N, Garnier S, Helbing D, Theraulaz G (2010). The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics. PLoS ONE 5(4): e10047. doi:10.1371/journal.pone.0010047.
- [2] Curtis S, Zafar B, Guy S, Manocha D (2010). Virtual Tawaf: A Case Study in Simulating the Behavior of

- Dense, Heterogeneous Crowds. <http://gamma.cs.unc.edu/LARGE>. Accedido el 10 de marzo de 2014.
- [3] Pelechano N, O'Brien K, Silverman B, Badler N (2012). Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication. http://www.lsi.upc.edu/~npelechano/Pelechano_V_CRO_WDS05.pdf. Accedido el 10 de marzo de 2014.
- [4] Agreda J, Sprockel J, Giraldo H,. Modelado de la Evacuación de Eventos Mediante el Uso de Sistemas Multi-Agentes. 2013. Actualmente en estudio.
- [5] Multiobjective Shortest Path Problem. http://www.mat.uc.pt/%7Ezeluis/INVESTIG/MSPP/mspp.htm#Public_code. Accedido el 11 de marzo de 2014.
- [6] González E., Bustacara C. "Desarrollo de Aplicaciones Basadas en Sistemas Multi-Agentes". Libro Producto de Investigación – Grupo SIDRe, Departamento de Ingeniería de Sistemas, Facultad de Ingeniería, Pontificia Universidad Javeriana. 2007.
- [7] Cervantes, A., Galvan, I., and Isasi, P. A Comparison between the Pittsburgh and Michigan Approaches for the Binary PSO Algorithm. IEEE, pp. 290–297. 2005.
- [8] González, E., Avila, J., & Bustacara, C. BESA: Behavior-oriented, event-driven and social-based agent framework. In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (Vol. 3, pp. 1033–1039). 2003.
- [9] Javier Barrachina, Piedad Garrido, Manuel Fogue, Francisco J. Martinez, Juan-Carlos Cano, Carlos T. Calafate, Pietro Manzoni, Reducing emergency services arrival time by using vehicular communications and Evolution Strategies, Expert Systems with Applications, Volume 41, Issue 4, Part 1, March 2014, Pages 1206-1217, ISSN 0957-4174, <http://www.sciencedirect.com/science/article/pii/S0957417413006155>.
- [10] Hong Qu, Ke Xing, Takacs Alexander, An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots, Neurocomputing, Volume 120, 23 November 2013, Pages 509-517, ISSN 0925-2312, <http://www.sciencedirect.com/science/article/pii/S0925231213005195>
- [11] Lim Eng Aik, Exit Selection by Occupant during Building Evacuation Using Neural Network. Institut Matematik Kejuruteraan, Universiti Malaysia Perlis, 02000 Kuala Perlis, Perlis, Malaysia.
- [12] X. Llorá, J. Garrel, Evolution of Decision Trees. Universidad Ramon Llull.
- [13] S. Russell, P. Norvig, Inteligencia Artificial. Prentice-Hall, México.
- [14] Data Mining: Practical Machine Learning Tools and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems).