

EVALUACIÓN DE ALGORITMOS DE OPTIMIZACIÓN METAHEURÍSTICA APLICADOS A LA PLANIFICACIÓN DE CAMINOS DE UN AGV

EVALUATION OF METAHEURISTIC OPTIMIZATION ALGORITHMS APPLIED TO THE PATH PLANNING OF AN AGV

Dizahab Sehuveret Hernández

Tecnológico Nacional de México / IT de Celaya, México
m1903074@itcelaya.edu.mx

Jorge Alberto García Muñoz

Tecnológico Nacional de México / IT de Celaya, México
jorge.garcia@itcelaya.edu.mx

Recepción: 14/mayo/2021

Aceptación: 30/noviembre/2021

Resumen

En este artículo se desarrolla la idea de utilizar métodos metaheurísticos para determinar una ruta óptima entre dos puntos de un entorno de cuadrícula, por parte de un vehículo autónomo terrestre. El robot móvil debe encontrar una trayectoria que reduzca el número de pasos necesarios para desplazarse desde el punto de inicio al punto final, mientras evita colisiones con obstáculos. Se explican tres de los métodos metaheurísticos más utilizados en la bibliografía para resolver problemas de esta índole y se realiza una comparación entre ellos, a partir de experimentos simulados.

Palabras Clave: algoritmos genéticos, búsqueda armónica, optimización por colonia de hormigas, planificación de caminos, vehículo autónomo terrestre.

Abstract

In this study it is presented the idea for using metaheuristic methods to determine an optimal path between two points in a grid environment, by an autonomous ground vehicle. The mobile robot must find a path that reduces the number of steps needed to travel from the starting point to the target ending point, while avoiding allisions with obstacles. Three of the most widely used metaheuristic methods in the literature

to solve problems of this nature are explained and a comparison is made between them, based on simulated experiments.

Keywords: *ant colony optimization, autonomous ground vehicles, genetic algorithms, harmony search, path planning.*

1. Introducción

La planificación de caminos se considera uno de los problemas más relevantes para el correcto funcionamiento de cualquier sistema robótico. En particular, para un vehículo autónomo terrestre (AGV), el éxito de la tarea que realice puede ser comprometido si el camino no es correctamente diseñado o se encuentran dificultades para seguirlo, resultando en colisiones que dañen el equipo o gastos innecesarios de energía. En la literatura existen numerosas alusiones alrededor del tema [Elhoseny et al., 2018], [Sørensen et al., 2017].

Para realizar un seguimiento de caminos en el que no sean involucradas restricciones temporales, el diseño debe considerar dos objetivos fundamentales. El primero y más importante es la evasión de los obstáculos que presente el terreno que se atraviesa; este objetivo está directamente relacionado con la seguridad y la integridad física de la máquina.

El segundo objetivo es la minimización de la distancia por recorrer. Planteado de esta manera, el problema de la planificación puede ser tratado como un proceso de optimización [Kamel et al., 2011]. A fin de resolver este tipo de cuestiones se emplean numerosas estrategias con diferentes niveles de complejidad, precisión y aplicabilidad. Entre estas estrategias se destacan los métodos metaheurísticos, derivados de conceptos relacionados con inteligencia artificial, adaptaciones biológicas y mecanismos estadísticos, entre otros. Por regla general, estos métodos pueden presentar algunos inconvenientes computacionales, como son: demasiados parámetros de control, convergencia prematura y tiempo de ejecución del método [Kaboli et al., 2017]. En ocasiones, también pueden favorecer el diseño del controlador evitando cálculos exhaustivos y aportando ventajas tales como robustez, amplia capacidad de búsqueda global [Xin et al., 2019] y búsqueda paralela [Xiaolin et al., 2019]. Resulta de especial interés una comparación entre los

diferentes métodos, con la finalidad de determinar en qué situaciones es preferible su utilización.

Este artículo presenta los aspectos básicos de tres de los métodos metaheurísticos más utilizados en la bibliografía para la resolución de problemas de optimización: Algoritmos Genéticos (GAs, *Genetic Algorithms*) [Xin et al., 2019], Búsqueda Armónica (HS, *Harmony Search*) [Li-min, 2015], [Alomoush et al., 2019] y Optimización por Colonia de Hormigas (ACO, *Ant Colony Optimization*) [Xiaolin et al., 2019], [Deng et al., 2019]. Además, somete a escrutinio su desempeño al ser aplicados en módulos de generación de trayectorias formadas por segmentos rectos para AGV, a partir de experimentos simulados en diferentes escenarios de control.

2. Métodos

Para la implementación de los métodos metaheurísticos a la planificación de caminos, es necesario determinar con anterioridad una manera conveniente de modelar el entorno en el que el AGV se desplaza. La generación de caminos se realiza a partir de una vista superior del terreno a transitar, por lo que la información adquirida de dicha imagen puede representarse en un mapa bidimensional. Se adopta el método de cuadrícula para modelar el medio ambiente [Zeng et al., 2016], que consiste en dividir el área de la imagen en celdas a lo largo de ambos ejes direccionales. Por lo tanto, el camino se describe como un conjunto finito de celdas conectadas. Las celdas con valor nulo representan zonas libres a transitar, mientras que las celdas con valor unitario representan zonas de obstáculos, figura 1.

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	0	1	0	0
1	0	1	1	0	0
0	0	0	0	0	0

Figura 1 Modelo de celdas para el entorno.

Otro aspecto a tener en cuenta para la correcta implementación de los algoritmos es la representación de las posibles trayectorias para su análisis. La estructura de las soluciones debe corresponder con los objetivos de la optimización. Para ello, es suficiente incluir en la estructura características espaciales, como son las propiedades geométricas del terreno y del camino deseado y la posición inicial y final del vehículo. Para representar una trayectoria compuesta por segmentos de líneas rectas basta con obtener un vector S_i que contenga las coordenadas $\{x, y\}$ de cada nodo del camino.

Algoritmos genéticos

Los Algoritmos Genéticos son conocidos como métodos adaptativos de optimización basados en las premisas de la mecánica y genética natural. Estos algoritmos combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorio [Goldberg, 1989]. Según los principios teorizados de la adaptación evolutiva, una especie puede adaptarse progresivamente al ambiente donde vive a partir de cambios generacionales [Beasley et al., 1993]. De manera que las combinaciones de la información genética de los progenitores, debidas a la reproducción, tienden a dotar a los descendientes de rasgos que benefician la explotación del ambiente.

En analogía a los procesos biológicos antes mencionados, el GA parte de un conjunto inicial de individuos, llamado población. Cada individuo supone una solución al problema de optimización. El algoritmo simula cambios generacionales a la vez que prueba la eficacia de los individuos para resolver el problema. Con el paso del tiempo, los mejores genes se propagan a un número cada vez mayor de individuos de las sucesivas generaciones, obteniendo así soluciones más prometedoras.

El algoritmo comienza con la generación de un conjunto de individuos, establecidos en un primer momento de manera aleatoria. Una población de N individuos puede ser representada como se indica en la ecuación 1. Los individuos I_i cumplen con la estructura de las soluciones descritas, S_i .

$$Pob_{inicial} = [I_1 I_2 \dots I_i \dots I_N]^T \quad (1)$$

Con la finalidad de decidir cuáles son las mejores soluciones propuestas y cuáles las peores, es necesario evaluar el desempeño de cada individuo. Como criterio para la evaluación, se establece una función de aptitud o función fitness. La función de aptitud propuesta se muestra en la ecuación 2.

$$F_i = \omega_l \sum_{s=1}^m d_s + \omega_c c \quad (2)$$

Donde d_s es la distancia euclídea entre dos nodos consecutivos, m es la cantidad de tramos del camino y c , la cantidad de colisiones que presenta la ruta. ω_l y ω_c son pesos que indican la prioridad de cada uno de los parámetros en la evaluación final de la solución.

Una vez determinados la población inicial y el método de evaluación, se procede a simular los cambios generacionales. Las poblaciones siguientes se conforman a partir de un proceso de reproducción y remplazo iterativo: se seleccionan dos individuos de la población actual para realizar el papel de padres y se les aplican procedimientos reproductivos para obtener un par de individuos nuevos. Las operaciones que determinan estos procesos son el cruce o copia y la mutación:

- **Cruce o copia:** Esta operación produce dos descendientes a partir de los progenitores seleccionados. Los padres pueden ser copiados y transmitidos sin modificación alguna a la siguiente generación o, por el contrario, la información genética de los padres puede ser dividida y recombinada para obtener la descendencia. En la figura 2 se muestra el proceso de fragmentación y recombinación relativo a la operación de cruce.
- **Mutación:** En ciertas ocasiones, algunos elementos del código de los individuos se alteran a propósito, en lo que constituye el símil probabilístico de una mutación.

De esta manera, el valor de un gen de la cadena puede ser sustituido por otro valor originado aleatoriamente.

Estos procedimientos se ejecutan una y otra vez hasta obtener una cantidad de descendientes igual al número de individuos de la población. A continuación, se

llevan a cabo procedimientos de migración y elitismo para reemplazar las peores soluciones resultantes de la reproducción:

- Elitismo: La operación de elitismo almacena la mejor solución de la población actual y la clona directamente en la siguiente generación.
- Migración: Con el objetivo de incrementar la diversidad de la población y evitar el estancamiento, se utilizan procedimientos de migración. Cada vez que se crea una nueva población, se construye un individuo y se compara con el peor descendiente. El individuo más apto permanece en la población.

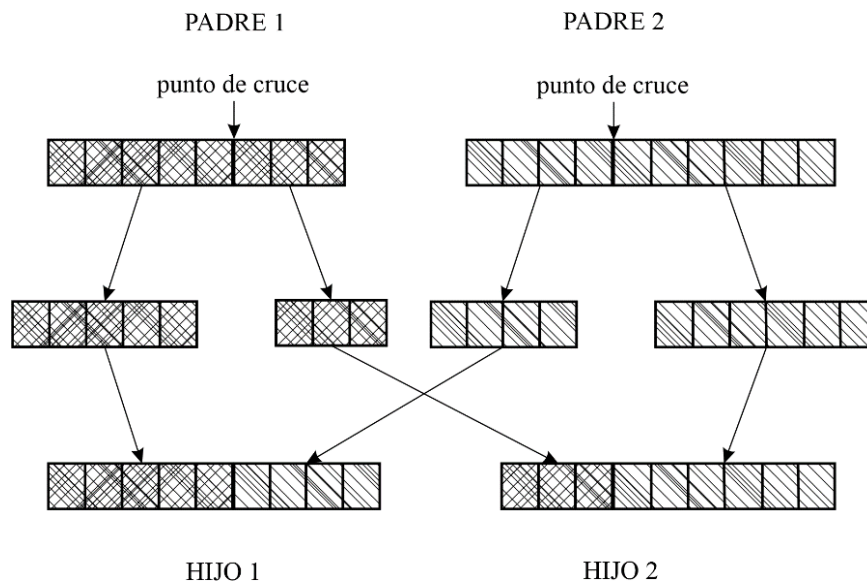


Figura 2 Operación de cruce.

El algoritmo continúa generando y evaluando poblaciones hasta que se encuentra una solución con un valor fitness establecido como adecuado para el camino. En caso de que este valor no pueda ser alcanzado, se debe establecer un número máximo de iteraciones para la finalización del algoritmo.

Búsqueda armónica

El algoritmo de Búsqueda Armónica es un método de búsqueda heurística que imita el proceso natural de improvisación musical. Cuando se improvisa una melodía, cada músico toca en cualquier tono dentro de un rango posible. Si todos

los tonos están en armonía, la experiencia se almacena en la memoria de todos los músicos y la posibilidad de mejorar la composición aumenta. De igual manera, en el modelo básico de HS, cada solución candidata es considerada una “armonía”. El algoritmo genera en cada iteración una nueva armonía y la somete a evaluación con respecto a otras almacenadas en una memoria armónica, si la armonía es suficientemente buena, la memoria es actualizada para incluirla.

El algoritmo comienza determinando e inicializando una memoria armónica (**HM**) de tamaño HMS , que contiene las mejores armonías improvisadas. La estructura de la memoria se muestra en la ecuación 3.

$$\mathbf{HM} = [\mathbf{H}_1 \ \mathbf{H}_2 \ \dots \ \mathbf{H}_i \ \dots \ \mathbf{H}_{HMS}]^T \quad (3)$$

Donde \mathbf{H}_i representa la armonía i almacenada. Estas armonías son vectores que cumplen con la estructura propuesta, \mathbf{S}_i .

Cada valor de coordenada en el vector es considerado una variable de decisión o nota. Al inicio, las notas en la memoria son inicializadas de manera aleatoria dentro de un rango establecido para ellas; o sea, los límites del terreno.

Se establece, además, un criterio de evaluación para cada armonía. Similar al estándar estético que espera la audiencia, en términos de ingeniería, este criterio se aplica al calcular un valor de aptitud o fitness. Aunque la definición conceptual sea distinta, la abstracción matemática de la función fitness puede ser idéntica tanto para el GA como para el HS. Lo anterior se justifica debido a que los objetivos de optimización son los mismos y la estructura de los individuos y las armonías presentan similitud evidente. Se propone utilizar la ecuación 3 como función de aptitud para el HS.

A partir de este momento, se realiza un proceso de improvisación iterativo: nuevas armonías son generadas atendiendo a las siguientes operaciones principales:

- Examinado de memoria o reinicialización aleatoria: En el paso de examinado de memoria, el valor de una variable de decisión de la nueva armonía h'_k es elegido del conjunto $\{h_{1k}, h_{2k}, \dots, h_{(HMS)k}\}$, donde se recogen todos los valores de esa variable de decisión ya existentes dentro de **HM**. A partir de consideraciones probabilísticas, la variable h'_k puede ser generada usando

el método discutido; de otro modo, se obtiene a partir de una reinicialización aleatoria entre las fronteras establecidas para dicha variable de decisión.

- Ajuste de tono: El ajuste de tono se realiza para cada una de las notas ya determinadas de la nueva armonía. Si se cumplen determinadas condiciones, la nota es ajustada como se muestra en la ecuación 4. De lo contrario, el valor de la nota permanece invariable.

$$h'_k = h'_k + U(-BW, BW) \quad (4)$$

Donde $U(.)$ hace referencia a una función que genera un valor entero aleatorio comprendido en el rango que indica su argumento. BW es llamado ancho de banda del ajuste y controla el tamaño máximo de la perturbación aplicada a la variable. Es importante asegurar que las soluciones perturbadas se encuentren dentro de los límites del terreno.

- Finalmente, si la nueva armonía generada es mejor, en términos del valor de aptitud, que la peor armonía almacenada en la memoria, entonces la peor armonía es remplazada por la nueva.

El algoritmo continúa improvisando nuevas armonías hasta que se encuentra una solución con un valor de aptitud adecuado. En caso de que este valor no pueda ser alcanzado, se debe establecer un número máximo de improvisaciones permitidas.

Colonia de hormigas

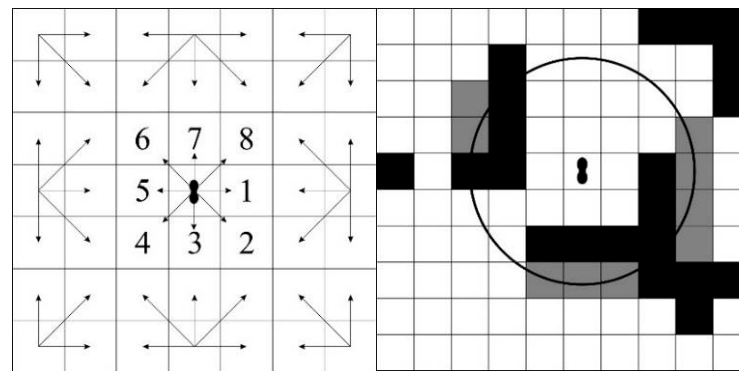
El algoritmo de colonia de hormigas es una técnica de búsqueda heurística que pertenece a la familia de inteligencia de enjambre. Fue desarrollado en base al comportamiento de búsqueda de alimento de las hormigas en la naturaleza [Osman et al., 2004].

Cuando una hormiga explora un camino en busca de alimento, va depositando feromonas y, simultáneamente, olfatea el entorno y corrige su dirección de movimiento cuando detecta una dirección con alta concentración de esta sustancia. Las hormigas que eligen los caminos más cortos alcanzan el alimento y regresan al nido rápidamente, por lo que la cantidad de hormigas que transitan por esos caminos resulta ser mayor y también la cantidad de feromonas depositadas. Como

esta sustancia se volatiliza con el tiempo, el proceso eventualmente excluye los caminos de mayor longitud y termina produciendo hileras de hormigas que recorren el camino más corto.

De forma similar, el ACO genera varias entidades llamadas hormigas artificiales y las hace moverse desde la posición del vehículo a la posición deseada, mientras evitan los obstáculos que puedan encontrarse. Estas hormigas modifican un mapa de feromonas artificiales a su paso. Luego de varias iteraciones, la distribución de las feromonas en el mapa hace converger a todas las hormigas a un mismo camino, de menor longitud.

Para simplificar los cálculos en la ejecución del ACO, es necesario limitar el movimiento de las hormigas artificiales. Cada hormiga, desde su posición, posee ocho direcciones en las que puede avanzar, siempre y cuando existan celdas disponibles para su movimiento, figura 3a.



a) Direcciones permitidas b) Rango y campo visual
Figura 3 Restricciones al movimiento de las hormigas.

Defínase como rango visual de una hormiga al alcance establecido para realizar un movimiento inmediato, sin la presencia de obstáculos alrededor y, como campo visual de la hormiga, al conjunto de celdas a las que puede acceder. En la figura 3b, por ejemplo, el área dentro de la circunferencia puede ser establecido como rango visual, pero las cuadrículas grises no se encuentran en su campo visual, debido a que las bloquea un obstáculo.

Siguiendo las direcciones de movimiento permitidas, la hormiga puede caminar a tantas celdas libre de obstáculos como se incluyan en su campo visual.

En una iteración t , una hormiga g se mueve de su celda actual e a una celda f según la regla de probabilidad de transición detallada en la ecuación 5.

$$p_{ef}^g = \begin{cases} \frac{[\tau_{ef}(t)]^\alpha [\eta_{ef}(t)]^\beta}{\sum_{s \in \mathcal{B}} [\tau_{es}(t)]^\alpha [\eta_{es}(t)]^\beta} & j \in \mathcal{B} \\ 0 & j \notin \mathcal{B} \end{cases} \quad (5)$$

Donde α y β son coeficientes del algoritmo. \mathcal{B} representa el conjunto de todos los nodos candidatos para los que la hormiga tiene la oportunidad real de trasladarse, según las limitaciones anteriormente impuestas a su movimiento.

$\tau_{ef}(t)$ representa la concentración de feromonas que conecta los puntos e y f . Luego de la representación del terreno en celdas, el entorno se limita a un conjunto de puntos discretos. Para reducir el gasto de recursos de ejecución, se utilizan dichos puntos como portadores de feromonas [Wang et al., 2019]. Por lo que la notación τ_{ef} queda reducida a τ_f .

La función $\eta_{ef}(t)$, llamada función de visibilidad, función heurística que recoge el conocimiento del entorno adquirido y se calcula como una división, donde el numerador representa la distancia entre el nodo actual y el posible nodo y el denominador, la distancia entre el posible nodo y la posición final del vehículo.

Después de que toda la colonia finaliza sus recorridos, los niveles de feromonas en cada celda se volatilizan y se agregan nuevos depósitos por las casillas por las que transitaron las hormigas. La actualización de feromonas se realiza según la ecuación 6.

$$\tau_f(t+1) = (1 - \rho)\tau_f(t) + \sum_{g \in \wp} \Delta\tau_g \quad (6)$$

Donde ρ es llamado coeficiente de disipación ($0 < \rho < 1$) y \wp representa el conjunto de todas las hormigas que pasan por el punto f . $\Delta\tau_g$ es el aporte de feromonas de la hormiga g que pasa por el punto y es inversamente proporcional a la longitud total del camino generado.

El algoritmo se detiene al ejecutarse un número de iteraciones anteriormente establecido.

Corrección de la trayectoria

Encontrar trayectorias acertadas a partir de métodos que se basan en la exploración de un espacio de soluciones en vez de reglas deterministas puede disminuir notablemente el esfuerzo realizado; sobre todo cuando dichas reglas incorporan desarrollos matemáticos complejos. Sin embargo, el propio concepto del método por búsqueda implica que la respuesta encontrada puede mejorarse en algunos casos, sobre todo frente a espacios de búsqueda que presentan tamaños considerables. Para caminos formados por segmentos de líneas rectas, es posible idear algunos esquemas de corrección, como la generación de atajos. Este procedimiento considera la posibilidad de que se pueda acortar la longitud de la ruta, conectando dos nodos con una línea de visión libre de obstáculos entre ellos. Como se observa en la figura 4, los nodos intermedios son eliminados.

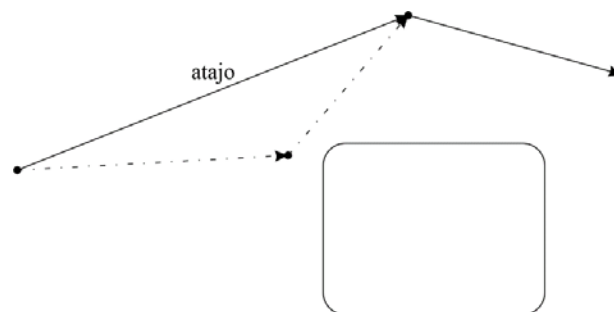


Figura 4 Generación de atajos.

3. Resultados

Para comprobar la efectividad de los algoritmos en la generación de trayectorias, se realizan los siguientes experimentos simulados en la herramienta de software Matlab. Cada método de generación es ejecutado un total de cien veces en tres ambientes diferentes, para evitar aleatoriedad en los resultados. Los ambientes se diferencian en la cantidad de obstáculos presentes y las distribuciones que poseen:

- Escenario 1. Este escenario posee solo tres obstáculos que presentan forma rectangular.
- Escenario 2. Este escenario presenta una complejidad moderadamente compleja. Posee un total de cinco obstáculos, de formas regulares.

- Escenario 3. Este escenario es el más complejo de los tres. Presentando ocho obstáculos de diversas formas.

Cada escenario fue procesado como un mapa de 100x100 celdas. Las posiciones inicial y final del vehículo son las mismas en todos los casos: el vehículo comienza su recorrido cerca de la esquina superior izquierda del terreno y termina cerca de la esquina inferior derecha.

En la tabla 1 se recogen los parámetros de diseño fundamentales utilizados para cada algoritmo en la realización de los experimentos. La cantidad de iteraciones que ejecutan los algoritmos y los valores fitness de terminación, si corresponde, varían según cada escenario.

Tabla 1 Parámetros de diseño seleccionados para el experimento.

GA			
Parámetros	Escenario 1	Escenario 2	Escenario 3
<i>Individuos</i>	300	500	3700
<i>CROSS</i>	0.8		
<i>MUT</i>	0.05		
<i>MIG</i>	0.1		
HS			
Parámetros	Escenario 1	Escenario 2	Escenario 3
<i>HMS</i>	50		
<i>HMCR</i>	0.77		0.95
<i>PAR</i>	0.01		
<i>BW</i>	5		
ACO			
Parámetros	Escenario 1	Escenario 2	Escenario 3
<i>Hormigas</i>	20		
α	1		
β	6		
ρ	0.3		

Por su parte, en la tabla 2 se detallan los valores estadísticos fundamentales del conjunto de datos mencionado y el tiempo medio de ejecución. La longitud del camino se representa en unidades referentes a la cantidad de celdas que ocupa y el tiempo, en segundos.

La figura 5 muestra la mejor trayectoria generada por cada algoritmo en cada uno de los tres escenarios discutidos.

Tabla 2 Resultados obtenidos.

Escenario 1					
Algoritmo	t_{medio}	l_{min}	l_{medio}	l_{max}	σ_l
GA	0.43	191.06	198.54	214.46	4.23
HS	0.65	191.31	201.11	251.01	6.31
ACO	6.93	188.40	199.58	211.01	4.74
Escenario 2					
Algoritmo	t_{medio}	l_{min}	l_{medio}	l_{max}	σ_l
GA	0.78	169.88	172.68	181.19	2.36
HS	0.68	169.51	173.46	184.94	2.88
ACO	2.06	168.64	172.58	177.83	2.14
Escenario 3					
Algoritmo	t_{medio}	l_{min}	l_{medio}	l_{max}	σ_l
GA	18.58	180.58	186.96	200.86	2.81
HS	79.09	174.56	184.84	187.80	2.46
ACO	4.54	194.66	202.44	214.48	4.72



Figura 5 Caminos generados.

En la figura 6 se presentan los resultados comparativos del conjunto de datos de distancias óptimas generadas durante el experimento, agrupados en diagramas de caja.

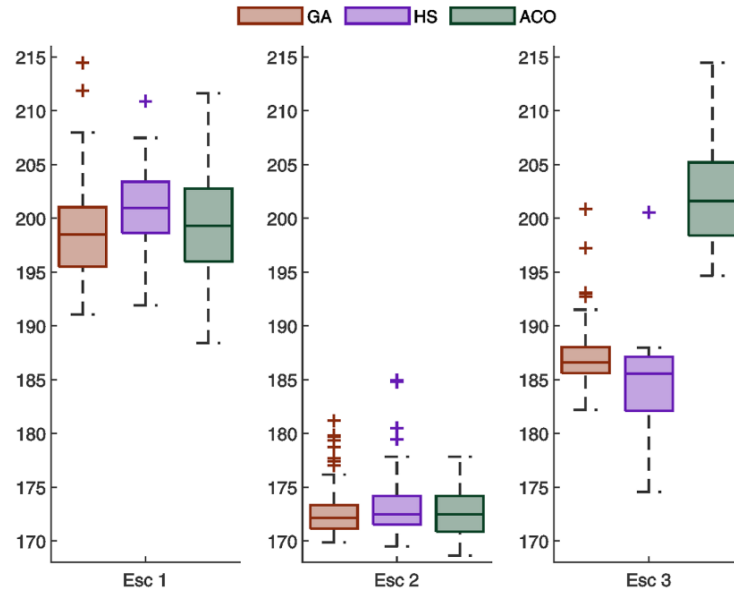


Figura 6 Diagramas de caja de los datos de longitud obtenidos.

4. Discusión

En cada diagrama de la figura 6, la dimensión de la caja corresponde al rango entre el primer y el tercer cuartil del conjunto de datos y puede ser interpretado como una medida gráfica de la dispersión del conjunto. El valor de la mediana es indicado por un segmento recto dentro de la caja y los bigotes del diagrama se extienden hasta los datos extremos. Además, los valores considerados atípicos en la muestra son representados por cruces, [Xiaolin et al., 2019].

En los escenarios más simples, todos los algoritmos proveen longitudes medias y grados de dispersión similares. El grado de dispersión es justificado, además, si se analizan los valores de desviación estándar encontrados en el experimento. Para el primer escenario, la desviación estándar sugiere que cerca del 90% de ocurrencias de soluciones se encuentran en los rangos $\{190, 207\}$, $\{188, 214\}$ y $\{190, 209\}$ para los algoritmos GA, HS y ACO, respectivamente; mientras que, para el segundo escenario, los rangos son $\{167, 178\}$, $\{167, 180\}$ y $\{168, 177\}$. Es posible observar la semejanza entre dichos conjuntos, pues la diferencia entre los valores frontera de una misma naturaleza para algoritmos diferentes no representa más de 7 unidades. No obstante, cabe mencionar la presencia de algunos datos atípicos generados en GA y HS, cuyos valores sobrepasan los límites superiores correspondientes. Lo

anterior indica que los resultados para estos dos algoritmos tienden a ser un poco menos consistentes que los generados por ACO.

Para el tercer escenario, el ACO obtiene rutas más largas; en tanto que HS y GA presentan un rendimiento ligeramente superior en cuanto a reducción de la distancia a recorrer. La distancia media obtenida por el ACO es de 202.44 unidades, que representa un aumento del 9.9% con respecto a la menor de las distancias medias generadas para este escenario. El hecho de que el ACO converja a soluciones levemente mayores que los otros dos algoritmos no necesariamente implica que cae en un mínimo local. Nótese que no existe evidencia en el diagrama de soluciones atípicas por debajo del mínimo estadístico, lo que sugiere que la diferencia mencionada puede ser resultado de otros factores, como las restricciones de movimiento impuestas al algoritmo. Como muestra la información estadística mencionada, todos los algoritmos poseen la bondad, en mayor o menor medida, de minimizar la distancia correctamente si se le permite una cantidad de iteraciones suficientemente grande. Sin embargo, aun cuando una de las características de la generación propuesta es precisamente no atender a restricciones temporales en la misión, se debe tener en cuenta que una demora excesiva por parte del módulo de planificación de trayectorias puede influir negativamente en el desempeño y estabilidad de los subsecuentes módulos de control implementados en el vehículo. Por lo que es importante realizar un análisis de rendimiento relativo al tiempo de ejecución. Una representación gráfica de los datos de tiempo adquiridos en el experimento se presenta en la figura 7.

Realizando una comparación entre los tiempos de ejecución de los algoritmos, se ha de notar que el ACO posee un tiempo medio de ejecución mayor que los otros métodos para los escenarios más sencillos. Esto se debe a la elevada cantidad de operaciones de cálculos implicadas en este algoritmo para la determinación de las probabilidades de selección de nodos. Por otra parte, el ACO se asegura de utilizar para la búsqueda solo caminos sin colisiones, tratando de manera separada los dos objetivos de la planificación. Un tratamiento de este tipo perjudica, en general, el tiempo de ejecución, pero le aporta robustez ante cambios en la complejidad del camino. Lo que se evidencia al poseer valores similares de tiempos de ejecución en

los tres escenarios. De hecho, un terreno con alta concentración de obstáculos favorece la utilización del ACO, pues al restringir la cantidad de caminos sin colisiones, las feromonas se concentran en la mejor ruta con mayor rapidez.

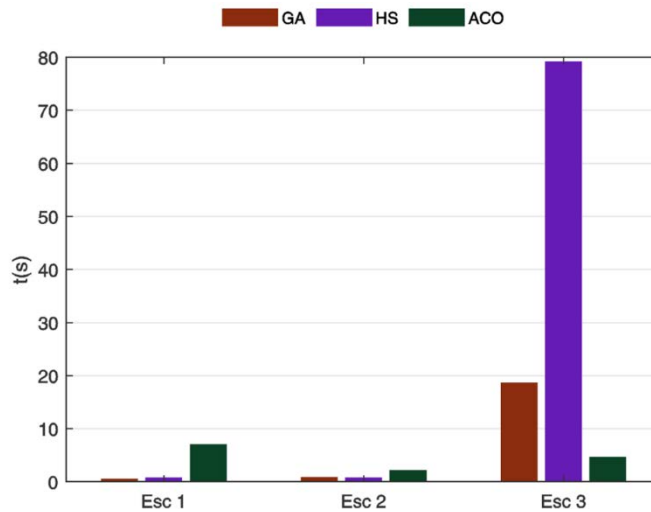


Figura 7 Tiempos de ejecución.

El tiempo de ejecución medio de los algoritmos GA y HS, por su parte, aumenta considerablemente a medida que aumenta la complejidad del escenario. Para el GA, el tiempo medio de ejecución aumenta alrededor de 40 veces del escenario 1 al escenario 3. En cuanto al HS, el tiempo medio de ejecución aumenta más de 100 veces del escenario 1 al 3, llegando a alcanzar un valor medio de 79.09s.

Estos métodos incluyen en la función fitness los dos objetivos de control. Si la función fitness se determina correctamente, la eliminación de colisiones ocurre de manera inherente, sin necesidad de incluir otros procedimientos de discriminación. Sin embargo, al aumentar la zona del entorno que representa los obstáculos también aumenta la cantidad de soluciones inválidas para su implementación real, puesto que es imposible considerar una ruta que presente la más mínima colisión. Este fenómeno dificulta la búsqueda para los algoritmos en cuestión, que están obligados a repetirse si no logran eliminar las colisiones en un primer intento. Lo anterior se traduce en un mayor gasto de tiempo y recursos.

Para el caso de escenarios complejos es necesario, entonces, tener en cuenta el nivel de precisión requerido por la misión impuesta al vehículo y la dinámica del

sistema de guiado y control implementado. En general, el aumento en el tiempo de ejecución observado en GA y HS es sustancialmente mayor que la reducción de la distancia que se logra con respecto al ACO. Por lo que este último representa, entre los tres algoritmos analizados, el recomendado en la planificación de trayectorias para escenarios de tipo complejo, a menos que las exigencias específicas de la misión indiquen lo contrario.

5. Conclusiones

La tarea de planificación de caminos para un AGV puede ser considerada como un problema de minimización de longitudes y evasión de obstáculos. En este artículo se compara la implementación de varios métodos metaheurísticos de optimización para resolver este problema en un ambiente 2D. Se utiliza un modelo del terreno basado en celdas y se realizan varios experimentos simulados en distintos escenarios de operación. Al estudiar las diferentes estrategias de optimización implementadas se advierte que el concepto de los métodos simples puede ser adaptado con éxito a la resolución del problema de planificación, al obtener rutas válidas de longitudes aceptables en todos los casos. No obstante, los experimentos muestran que, comparados con el ACO, los algoritmos GA y HS presentan un mejor desempeño en la búsqueda para escenarios de poca o media complejidad, en términos de tiempo de ejecución. Al aumentar la zona de obstáculos el desempeño anteriormente mencionado se deteriora con facilidad, siendo preferible la utilización de ACO para escenarios de complejidad alta.

6. Bibliografía y Referencias

- [1] Elhoseny, M., Tharwat, A., & Hassanien, A. E., "Bezier curve based path planning in adynamic field using modified genetic algorithm," *Journal of Computational Science*, vol.25, pp.339-350, 2018.
- [2] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [3] Zeng, M. R., Xi, L., & Xiao, A. M., "The free step length ant colony algorithm in mobile robot path planning," *Advanced Robotics*, vol.30, pp.1–6, Oct. 2016.

- [4] Kamel, Mohamed S., Miao, Yun-Qian, Khamis, Alaa, & Karray, Fakhri, "A novel approach to path planning for autonomous mobile robots," *Control and Intelligent Systems*, vol.39, no.4, 2011.
- [5] Kaboli, S. H. A., Selvaraj, J., & Rahim, N. A., "Rain-fall optimization algorithm: A population based algorithm for solving constrained optimization problems," *Journal of Computational Science*, vol.19, pp.31 – 42, 2017.
- [6] Sørensen A. J., Candeloro, M., & Lekkas, A. M., "A voronoi-diagram-based dynamicpath-planning system for underactuated marine vessels," *Control Engineering Practice*, vol.61, pp.41 – 54, 2017.
- [7] Xin, J., Zhong, J., Yang, F., Cui, Y., & Sheng, J., "An improved genetic algorithm for path-planning of unmanned surface vehicle," *Sensors*, vol.19, p.2640, Jun19.
- [8] Xiaolin, D., Long, S., Zhang, Z., & Gong, D., "Mobile robot path planning based on ant colony algorithm with A* heuristic method," *Frontiers in neurorobotics*, vol.13, no.15, Apr. 2019.
- [9] Li-min, Z., "An improved adaptive harmony search algorithm," in *Proc. International Conference on Computational Science and Engineering*. Atlantis Press, Jul. 2015, pp.54–59.
- [10] Alomoush, A., Alsewari, A., Alamri, H., & Zamli, K., "Comprehensive review of the development of the harmony search algorithm and its applications," *IEEE Access*, vol.7, pp.14233 – 14245, Ene. 2019.
- [11] Deng, W., Xu, J., & Zhao, H., "An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem," *IEEE Access*, vol.7, pp.20281 – 20292, 2019.
- [12] Beasley, D., Bull, D. R., & Martin, R. R., "An overview of genetic algorithms: Pt1, fundamentals," *University Computing*, vol.15, no.2, pp.58 – 69, 1993.
- [13] Osman, M., Abo-Sinna, M., & Mousa, A., "A solution to the optimal power flow using genetic algorithm," *Applied Mathematics and Computation*, vol.155, no.2, pp.391 – 405, 2004.
- [14] Wang, Lanfei, Kan, Jiangming, Guo, Jun, & Wang, Chao, "3d path planning for the ground robot with improved ant colony optimization," *Sensors*, vol.19, p.815, Feb. 2019.