

## PLANIFICACIÓN DE CAMINOS PARA MÚLTIPLES ROBOTS BASADO EN ALGORITMO GENÉTICO

### GENETIC ALGORITHM BASED MULTIROBOT PATH PLANNING

**M. Estefanía Pereyra, Gonzalo Perez Paina, Martín Pucheta y R. Gastón Araguás**

*Centro de Investigación en Informática para la Ingeniería (CIII), Universidad Tecnológica Nacional, Facultad Regional Córdoba (UTN-FRC), Argentina, mepereyra@frc.utn.edu.ar*

**Palabras clave:** Algoritmo genético; planificador de caminos; operadores evolutivos; robot móvil; algoritmo de Dijkstra, Voronoi.

**Resumen.** Un planificador de caminos para múltiples robots (MPP, del inglés Multirobot Path Planning) debe encontrar el conjunto de caminos libre de colisiones, óptimo (o casi-óptimo) en términos de algún criterio en particular, desde una posición inicial a una posición final en un entorno con obstáculos. Dada la complejidad del problema de MPP, usualmente se utilizan algoritmos heurísticos para encontrar una solución óptima, especialmente en entornos complejos y con un número elevado de robots. En este trabajo se presenta un algoritmo de planificación de caminos para sistemas multirobots el cual utiliza como método de exploración el conocido Algoritmo Genético Simple (AGS). Dado un grafo representativo del espacio de búsqueda el algoritmo encuentra los caminos de una formación de robots teniendo en cuenta la posibilidad de separación de la formación y su posterior reunión. Partiendo de una población inicial con gran diversidad de individuos y utilizando una codificación de cromosomas y operadores evolutivos especialmente diseñados para el problema de múltiples robots, el algoritmo evoluciona generación tras generación devolviendo los mejores caminos para la formación de robots. Los resultados de las simulaciones demuestran que el algoritmo es apto para ser utilizado en entornos con obstáculos y para una formación de decenas de robots.

**Keywords:** Genetic algorithm; path planning; crossover operator; mobile robots, dijkstra's algorithm, Voronoi.

**Abstract.** A multirobot path planner (MPP) should find an optimal (or near-optimal) collision-free path, optimal in terms of a particular criterion, from the starting position to the goal position in an environment with obstacles for each robot in the formation. Because of the MPP complexity, heuristic algorithms usually are used to find an optimal solution, specifically in complex environments with many robots. In this work, we present a multirobot path planning algorithm using a search method based on the Simple Genetic Algorithm (SGA). Given a graph of the working environment, the algorithm finds a path for a formation taking into account possible split of the formation and its consecutive merge. From an initial population of suboptimal feasible paths, a particular encoding of chromosomes and genetic operators specially designed for the MPP problem, the algorithm evolves through generations, to find the best solution for the robots formation. Results from the simulations show that the algorithm performs well in environments with obstacles and multiple robots formation.

## 1. INTRODUCCIÓN

La planificación de caminos puede considerarse uno de los pilares fundamentales en la navegación autónoma de robots, junto con la percepción, localización y generación de trayectorias, cuyo interés ha ido en constante crecimiento dentro de la comunidad, especialmente para sistemas multirobot y robots en formación. En robótica la planificación de caminos consiste en encontrar un conjunto de caminos libre de colisión a través de un entorno con obstáculos, desde una ubicación inicial a un destino final deseado. Los caminos se obtienen optimizando algún parámetro específico (como distancias, consumos de energía, tiempo de operación, etc.) y aplicando luego restricciones respecto del tamaño y/o geometría de la formación (Sharir, 1989). Dependiendo del conocimiento que tiene el robot sobre su entorno la planificación puede ser clasificada en dos categorías: planificación global, si el robot tiene un total conocimiento del entorno o planificación local, si el robot conoce solo una parte del mismo. Además ese entorno puede ser estático o dinámico. En este trabajo se considera el problema de planificación global de caminos en un entorno estático representado mediante un grafo ponderado. Los métodos de planificación global de caminos se dividen principalmente en dos categorías: métodos clásicos y métodos heurísticos. Los métodos clásicos son variantes o combinaciones de métodos como Roadmap (RM), Campos Potenciales y descomposición en celdas. En estos métodos primero se modela el espacio de trabajo como una red o grafo de posibles caminos, uniendo los puntos inicial y final a dicho grafo, para después aplicar un algoritmo de búsqueda tales como A\* Hart P. E. (1968), el algoritmo de Dijkstra Dijkstra (1959); Yüksel y Sezgin (2008) u otros. Estos métodos clásicos son exactos, efectivos y determinísticos. Sin embargo estos enfoques presentan ciertas debilidades en entornos muy complejos donde el número de nodos y aristas del modelo crece exponencialmente, y en sistemas con muchos robots. En estos casos la aplicación de estos algoritmos se hace inviable computacionalmente (Pereyra et al., 2017a; P. Raja, 2012). Otra debilidad presente en alguno de estos métodos clásicos es que pueden quedar atrapados en soluciones óptimas locales, muy distantes de la solución óptima global (Altaharwa et al., 2008; Miao et al., 2011). Debido a estas debilidades presentes en los planificadores clásicos se han propuesto los métodos heurísticos para resolver el problema de planificación de caminos. Entre los métodos heurísticos se incluyen algoritmos como optimización de colonia de hormigas (ACO, ant colony optimization) Noormohammadi Asl et al. (2014), optimización de enjambre de partículas (PSO, particle swarm optimization) Bai et al. (2009), algoritmos genéticos (AG) Qu H. (2013), búsqueda local iterada (ILS, iterated local search), algoritmo de recocido simulado (SA, simulated annealing) Hussein et al. (2012) y redes neuronales (NN, neural network) (Yang y Meng, 2000), entre otros. Una de las principales características de los métodos heurísticos es el uso de decisiones probabilísticas durante la búsqueda. Si bien los métodos heurísticos son aproximados y usualmente no determinísticos, proveen formas inteligentes de explorar el espacio de búsqueda, lo cual disminuye significativamente el tiempo de cómputo. Una de las técnicas que sobresale sobre otras en la planificación de caminos son las basadas en algoritmos genéticos, debido a su capacidad de explorar el espacio de búsqueda preservando la mejor solución ya encontrada (Tang et al., 1996). En los últimos años muchos planificadores de caminos basados en AG han sido implementados Castillo O. (2007); Qu H. (2013); R. Kala (2010); Hoshier et al. (2016). Una de las limitaciones presente en estos trabajos es que todos comienzan con una solución inicial aleatoria, o adoptan un método aleatorio para generar una solución inicial válida. Este enfoque no es eficiente con respecto al tiempo y al costo computacional. Además utilizan operadores evolutivos comunes o convencionales, que aunque su rendimiento ha sido probado en numerosos problemas, la eficiencia del algoritmo evolu-

tivo puede mejorarse considerablemente mediante el uso de operadores evolutivos diseñados específicamente para el problema planteado (Tuncer A., 2012). En este trabajo se presenta un método de planificación de caminos global para sistemas multi-robots que combina técnicas de búsqueda clásicas y heurísticas basadas en algoritmos genéticos. Para esto, primero se propone realizar una búsqueda secuencial de caminos mediante el algoritmo presentado en Pereyra et al. (2017b), el cuál extiende el bien conocido algoritmo de Dijkstra para ser aplicado a sistemas multirobot, que además considera la posibilidad de separación y unión de la formación. Este algoritmo encuentra los caminos entre las posiciones inicial y final que conformarán los individuos de la población inicial de caminos. Luego, partiendo de la población inicial generada por el algoritmo secuencial, se buscan los caminos óptimos de la formación de robots aplicando operadores evolutivos diseñados para el problema de multi-robots. Esta combinación evita que el planificador quede atrapado en mínimos locales, y permite computar rápidamente soluciones óptimas o cercanas a óptimas. Los resultados de simulación muestran que mediante la integración de ambos planificadores se obtienen soluciones óptimas en tiempos considerablemente menores comparados con el método de búsqueda exhaustivo (Pereyra et al., 2017a).

## 2. PLANIFICADOR DE CAMINOS PARA MULTI-ROBOTS

El problema de planificación de caminos para múltiples robots (MPP, *multirobot path planning*) se puede interpretar de forma general como la búsqueda de una secuencia continua de configuraciones posibles del conjunto de robots, desde una configuración inicial hasta una configuración final dada. Una configuración es considerada posible si sus robots no entran en colisión con el entorno que los rodea ni entre ellos.

En este trabajo se presenta una solución al problema de guiar un conjunto de robots desde un punto inicial a un punto destino dentro de un mapa conocido. Los caminos de cada robot se obtienen minimizando un funcional de costo dado. De ser necesario se permiten divisiones del grupo de robots pero con la condición que se reúnan lo antes posible para mantener la formación. Por lo tanto, cada robot que llega a un punto de unión debe esperar a los demás robots que también contienen ese punto en sus caminos. Consecuentemente, el costo de la solución completa queda determinado por el costo del peor camino del conjunto de caminos individuales.

El MPP debe realizarse evitando las colisiones con los obstáculos del mapa y evitando las colisiones entre robots. Para evitar colisiones con los obstáculos se realiza el MPP utilizando una representación gráfica del mapa basada en su diagrama de Voronoi, y sobre este diagrama se realiza la búsqueda de caminos óptimos. En el Algoritmo 1 se describe en forma general el método propuesto de planificación basado en algoritmos evolutivos que se detalla en las secciones posteriores.

---

### Algoritmo 1: MPP evolutivo.

---

- 1 Modelado del entorno mediante grafo conectado  $\mathcal{G}(\mathcal{V}, \mathcal{E})$
  - 2 Inicialización de la población  $H_i$
  - 3 Búsqueda del conjunto óptimo de caminos en  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  desde el nodo  $Q_{ini} \in \mathcal{V}$  hasta el nodo  $Q_{goal} \in \mathcal{V}$  para cada robot  $r_i \in \mathcal{R}$  mediante algoritmo evolutivo
- 

### 2.1. Modelado del entorno y propiedades de los caminos

El modelado del entorno consiste en una representación mediante un grafo ponderado como se muestra en la Fig. 1(a). Este grafo se obtiene a partir del diagrama de Voronoi de los

obstáculos del entorno. Aquí se consideran obstáculos estáticos los cuales pueden ser modelados mediante polígonos. Las aristas de este grafo forman los caminos libres de colisiones por donde pueden circular los robots y cada una de ellas tiene asociado un valor que representa el costo (tiempo, energía, etc.) de ser atravesada por un robots. Para el caso de múltiples robots el costo de la arista puede ser variable, dependiendo de la cantidad de robots que la atraviesan. En este trabajo el costo representa el tiempo necesario para atravesar la arista considerando velocidades constantes de movimiento. El valor representativo de este tiempo puede estimarse mediante algún algoritmo de planificación de movimientos local, o puede aproximarse mediante una función de costo. Aquí se utiliza una función de costo que considera la longitud y ancho del corredor (arista), y la cantidad de robots que lo atraviesan. Representado mediante un vector de costos  $\mathbf{c} = (c_1, c_2, \dots, c_R)$ , donde  $c_r$  corresponde al costo de atravesar con  $r$  robots dicha arista. Sobre el grafo con costos ya asignados,  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ , se realiza posteriormente la búsqueda de los  $\mathcal{R}$  caminos mediante el planificador evolutivo. Cada uno de estos caminos es una secuencia de vértices de forma  $p = \{v_0 = Q_{\text{ini}}, v_1, v_2, \dots, v_n = Q_{\text{goal}}\}$ , que cumple con las propiedades y restricciones mencionadas a continuación.

Sea  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  el grafo conectado que representa el entorno de trabajo, con vértices  $\mathcal{V} = \{v_1, v_2, \dots, v_i\}$  y aristas  $\mathcal{E} = \{e_1, e_2, \dots, e_j\}$ , y una flota de robots  $\mathcal{R} = \{r_1, r_2, \dots, r_R\}$ , el conjunto de caminos que guiará a los robots por el mapa se denota  $\mathcal{P} = \{p_1, p_2, \dots, p_R\}$ , con  $p_i : \mathbb{Z}^+ \rightarrow \mathcal{V}$ . Además, el camino de un robot individual  $r_i$  es una secuencia de vértices  $p_i = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}$  tal que  $(v_{i_j}, v_{i_{j+1}})$  es una arista del grafo.

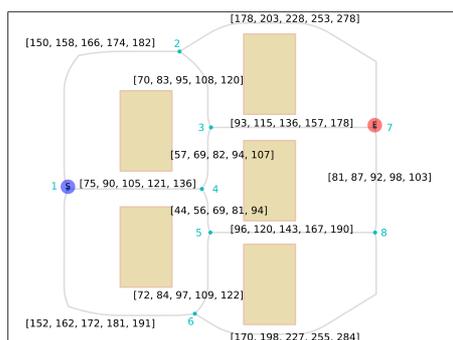
- en un inicio, todos los robots de la formación se encuentran en la posición inicial, esto es  $p_i(0) = Q_{\text{ini}}, \forall p_i \in \mathcal{P}$ .
- para cada camino existe un estado  $k_{\text{min}} \in \mathbb{Z}^+$  tal que  $p_i(k_{\text{min}}) = Q_{\text{goal}}$ , donde el robot  $r_i$  alcanzó su objetivo utilizando el camino de menor costo.
- dos caminos de  $\mathcal{P}$  no colisionan, es decir, dados dos estados intermedios cualquiera  $m, l \in \langle 0, k_{\text{min}} \rangle$ , los caminos  $p_i, p_j$  se encuentran en colisión si  $(p_i(m), p_i(m+1)) = p_j(l+1), p_j(l)$ .
- sean dos caminos  $p_i, p_j$  y dos estados  $m, l \in \langle 0, k_{\text{min}} \rangle$ , si  $p_i(m) = p_j(l)$  entonces  $m = l \leftarrow \max(m, l)$ , lo cual significa que el primer robot en alcanzar un vértice compartido espera por el segundo, con el fin de mantener la formación unida tanto como sea posible.

## 2.2. Algoritmo Genético propuesto

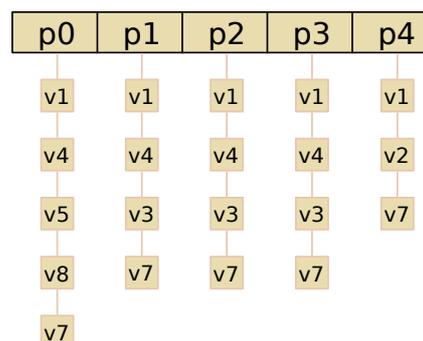
Existen múltiples propuestas y variantes de algoritmos genéticos, en este trabajo se utilizó como base el Algoritmo Genético Simple (AGS) descrito en Algoritmo 2. Sobre este algoritmo base se realizan modificaciones tanto en la representación del cromosoma, como en los operadores evolutivos de cruzamiento y mutación con el fin de poder adaptar el algoritmo al problema planteado.

### 2.2.1. Inicialización

La población inicial a partir de la cual el algoritmo evoluciona está formada por un conjunto de soluciones subóptimas válidas que llevan al conjunto de robots desde la posición inicial a la final. Este conjunto de posibles soluciones se obtiene aplicando el algoritmo de búsqueda



(a) Representación gráfica del entorno con obstáculos.



(b) Representación de un individuo de la población (cromosoma).

Figura 1: Modelado del entorno y del cromosoma.

**Algoritmo 2:** Algoritmo Genético Simple (AGS).

---

```

1 Inicializar(P(0));
2 generación = 0;
3 while (no CriterioParada) do
4   Padres = Seleccionar(P(generación));
5   Hijos = aplicar Operadores Evolutivos(Padres);
6   NuevaPoblación = Reemplazar(Hijos, P(generación));
7   Evaluar(NuevaPoblación);
8   generación++;
9   P(generación) = NuevaPoblación;
10 retornar MejorSolución

```

---

secuencial presentado en [Pereyra et al. \(2017b\)](#). Esta inicialización permite reducir el número de generaciones necesarios para obtener la solución óptima y con ello el tiempo y costo computacional del proceso de búsqueda que se obtiene con una inicialización aleatoria.

**2.2.2. Codificación del cromosoma**

La elección de la codificación del cromosoma es de gran importancia debido a la relación directa con la complejidad que puede presentar luego el diseño de los operadores evolutivos. En nuestro caso se propone como representación de la solución al problema un cromosoma que codifica los caminos individuales de cada robot de la formación. Esto es, cada cromosoma estará formado por  $R$  alelos, donde  $R$  es el número máximo de robots de la formación. A la vez, cada alelo contiene la lista de vértices que conforman el camino individual de cada robot desde el nodo inicial al nodo final, siendo su longitud variable con el número de vértices que conforman el camino. En la Fig. 1(b) se muestra un ejemplo de la representación de un individuo mediante un cromosoma de cinco alelos,  $\mathcal{P} = \{p_0, p_1, p_2, p_3, p_4\}$ , con  $p_0 = \{v_0, v_4, v_5, v_8, v_7\}$ ,  $p_1 = \{v_1, v_4, v_3, v_7\}$ ,  $p_2 = \{v_1, v_4, v_3, v_7\}$ ,  $p_3 = \{v_1, v_4, v_3, v_7\}$ ,  $p_4 = \{v_1, v_2, v_7\}$ .

### 2.2.3. Función de fitness

La siguiente se propone como función de fitness:

$$F(x) = C - c_{\max}$$

donde,  $C$  es el costo que corresponde a llevar la totalidad de los robots por un mismo camino (camino óptimo de un robot considerando el costo de llevar los  $R$  robots por las aristas involucradas), y  $c_{\max}$  el costo del camino de mayor costo del conjunto.

$$c_{\max} = \max\{c_1, c_2, \dots, c_R\}.$$

### 2.2.4. Operadores evolutivos

Los operadores evolutivos encargados de alterar la composición de los individuos de la población a través de las generaciones, son similares a los propuestos en AGS con algunas modificaciones, a saber:

*Operador de selección:* el tipo de selección utilizada es la selección proporcional o rueda de ruleta (Holland), donde la selección de un individuo se realiza teniendo en cuenta una probabilidad de selección la cual es proporcional a su fitness relativo. Además se utiliza la selección por elitismo, la cual garantiza que el mejor individuo permanezca dentro de la población generación tras generación.

*Operadores de cruzamiento:* Con el fin de obtener diversidad en la población y de generar caminos nuevos se proponen dos tipos de cruzamientos, uno a nivel de caminos o cromosomas y el otro a nivel de aristas o alelos.

- *Cruzamiento a nivel de caminos o cromosomas:* Dados dos individuos de la población se genera un hijo considerando los  $R$  caminos de menor costo y otro con los  $R$  caminos restantes. La Fig. 2 muestra un ejemplo de este tipo de cruzamiento para individuos de 5 caminos, donde el primero de los hijos se obtiene con 3 caminos de uno de los padres y 2 del otro. Luego, los caminos restantes conforman el segundo hijo. Notar que luego del cruzamiento los costos de cada camino deben recalcularse debido a que los cambios de configuración pueden modificar los costos de las aristas individuales.

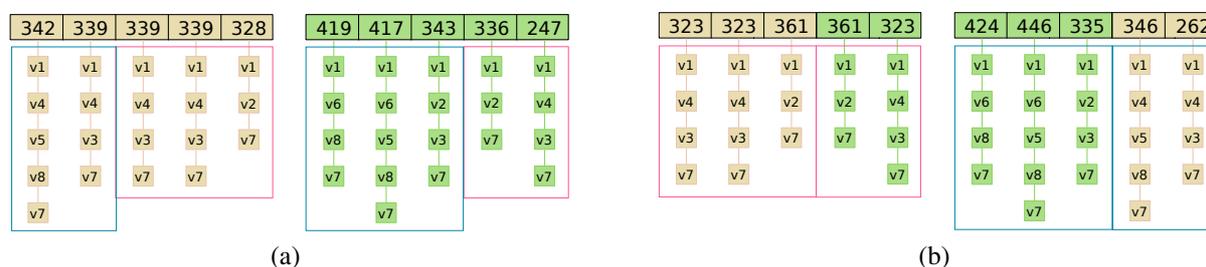


Figura 2: Cruzamiento de cromosomas a nivel de caminos.

- *Cruzamiento a nivel de aristas o alelos:* Dados dos individuos de la población se toman de a uno los caminos del primer individuo y se combinan con uno de los caminos del otro individuo siempre que sea posible. Dos caminos se pueden combinar siempre que contengan una arista intermedia en común. Un ejemplo de este cruzamiento se muestra

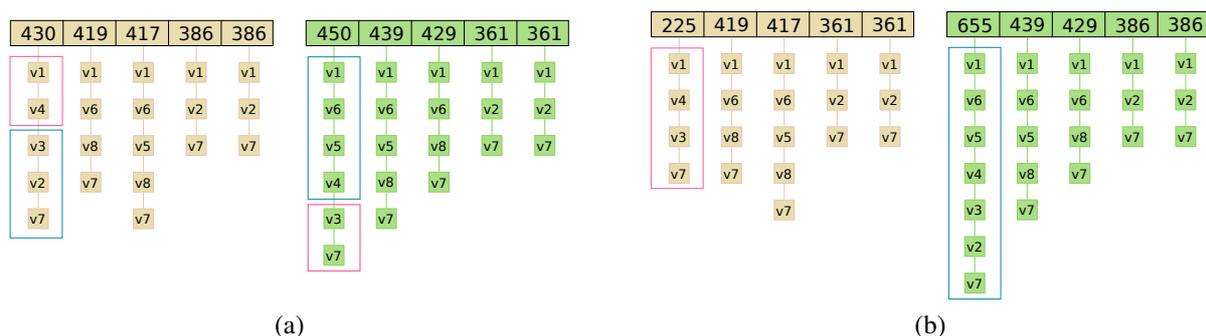


Figura 3: Cruzamiento de cromosomas a nivel de aristas.

en la Fig. 3. En este caso el único camino combinable fue el primero de ambos individuos, con arista en común ( $v_4 - v_3$ ).

*Operadores de mutación:* Se proponen dos tipos de mutaciones, una a nivel de aristas o alelos y otra a nivel de caminos o cromosomas.

- *Mutación a nivel de aristas o alelos:* por cada camino de una solución se elimina de forma aleatoria una de sus aristas del grafo y se recalcula el camino mediante el algoritmo de Dijkstra para llegar al nodo final desde el vértice correspondiente a la arista eliminada.
- *Mutación a nivel de caminos:* dado un individuo de  $R$  caminos se toman aleatoriamente uno de ellos y se copian  $R - 1$  veces, reemplazando  $R - 1$  caminos del individuo, seleccionados aleatoriamente.

### 2.2.5. Parámetros de control y condición de parada

La elección de los parámetros de control del algoritmo evolutivo tales como el tamaño de la población, la probabilidad de cruzamiento y mutación son de fundamental importancia ya que afectan de forma directa el desempeño del algoritmo. Por ejemplo un tamaño de población grande implica una gran diversidad de individuos pero a la vez implica un mayor requerimiento de capacidad de cómputo y tiempo de ejecución. En nuestro caso nos enfocamos en obtener una población inicial del orden  $10 \cdot R$ , lo más diversa posible.

La elección de los parámetros de probabilidad cruzamiento y mutación se basaron en la experimentación del método y en la bibliografía consultada. Usualmente se toma una probabilidad de cruzamiento elevada con el fin de obtener nuevos individuos, los valores típicos son de 0,6 a 1,0. Con respecto a la probabilidad de mutación se toman valores pequeños entre 0,05 y 0,15, este operador es fundamental para generar nuevos individuos que aporten diversidad en la población. Como condición de finalización se utilizó un número máximo de iteraciones basado también en experimentación. Con un número de 100 iteraciones fue suficiente para obtener los resultados que se detallan en la siguiente sección.

## 3. SIMULACIÓN

Los experimentos fueron realizados sobre una serie de mapas de diferentes complejidades en cuanto al número de obstáculos que contienen. Estos mapas son *gaps*, *dense*, *staggered\_brick\_wall*, *potholes*, *var\_density*, *var\_density2* y *var\_density3* de <http://imr.ciirc.cvut.cz/planning/maps.xml>. El algoritmo de búsqueda de caminos fue escrito en Python y se utilizó

una notebook con un procesador Intel CORE i5 y 4GB de RAM, bajo el sistema operativo Debian GNU/Linux. Los caminos óptimos sobre los que se realiza la evaluación del algoritmo propuesto se obtienen a partir de un método de búsqueda exhaustivo. Este algoritmo devuelve el camino óptimo para cada configuración (mapa, punto inicial  $Q_{ini}$ , punto final  $Q_{goal}$ , cantidad de robots  $R$ ). Este método de búsqueda exhaustivo tiene una alta complejidad computacional, y se incrementa exponencialmente con el número de robots y el número de vértices del grafo. Lo cual limita el análisis del algoritmo a no más de tres robots en mapas cuyos grafos correspondientes tienen hasta cincuenta vértices. Cada evaluación se realiza corriendo el algoritmo de búsqueda con optimización de tipo secuencial (Pereyra et al., 2017b), luego corriendo el algoritmo de búsqueda propuesto con optimización basada en algoritmos genéticos y comparando ambos resultados con el obtenido mediante el algoritmo de búsqueda exhaustivo (Pereyra et al., 2017a).

La Fig. 4 muestra los resultados de las simulaciones, en la Fig. 4a se muestra una comparación entre los caminos óptimos (línea negra) y los caminos generados por el algoritmo secuencial (línea roja) y en la Fig. 4b se muestra la comparación entre los caminos óptimos (línea negra) y los caminos generados por el algoritmo genético (línea roja). En ambos casos las gráficas son *muestras vs. costo*, donde una *muestra* hace referencia a un camino retornado por el algoritmo para cada configuración (mapa,  $Q_{ini}$ ,  $Q_{goal}$ ,  $R$ ) y *costo* es el costo del camino encontrado. Como se puede ver en las figuras la evaluación del algoritmo se realizó sobre un total de aproximadamente 2000 caminos, de los cuales el 92 % de los caminos encontrados por el algoritmo secuencial resultaron en caminos óptimos y el 97 % de los caminos encontrados por el algoritmo genético fueron óptimos. La Fig. 4c muestra la diferencia porcentual entre los algoritmos con respecto a la solución óptima. Estos resultados fueron obtenidos configurando los parámetros del algoritmo genético de la siguiente manera. El tamaño de la población se fijó en 20 individuos, la probabilidad de cruzamiento se fijó en 0,9 y la de mutación en 0,3. Con esta configuración se obtuvieron soluciones óptimas dentro de las 20 generaciones en promedio. Y los tiempos de cómputo fueron de  $\approx 0,2$  segundos por generación, dependiendo del tamaño del mapa y número de robots en la formación. La Fig. 5 muestra un ejemplo de MPP, donde el algoritmo de optimización propuesto genera una solución óptima al igual que la generada por el algoritmo de búsqueda exhaustiva.

#### 4. CONCLUSIONES

En este trabajo se presentó un algoritmo de planificación de caminos basado en algoritmos genéticos y aplicado al problema de múltiples robots. En base al análisis de los resultados obtenidos y del comportamiento del algoritmo se pudo comprobar que tanto la inicialización de la población como los operadores evolutivos propuestos influyen de forma considerable en el rendimiento del algoritmo. Uno de los operadores que resultó ser de gran importancia fue el operador de mutación propuesto. Este operador tiene la capacidad de generar caminos nuevos, lo que aumenta la diversidad de la población y en conjunto con una inicialización de la población restringida a caminos válidos, se logra una convergencia a la solución óptima en muy pocas generaciones. Si bien los tiempos que se lograron son de un orden de magnitud mayor a los obtenidos mediante el algoritmo secuencial, el número de caminos óptimos encontrados fue un 5 % mayor. Como trabajo futuro se espera poder investigar nuevos tipos de operadores evolutivos tendiendo a disminuir los tiempos de cómputo con el fin de poder aplicar el algoritmo en tiempo real.

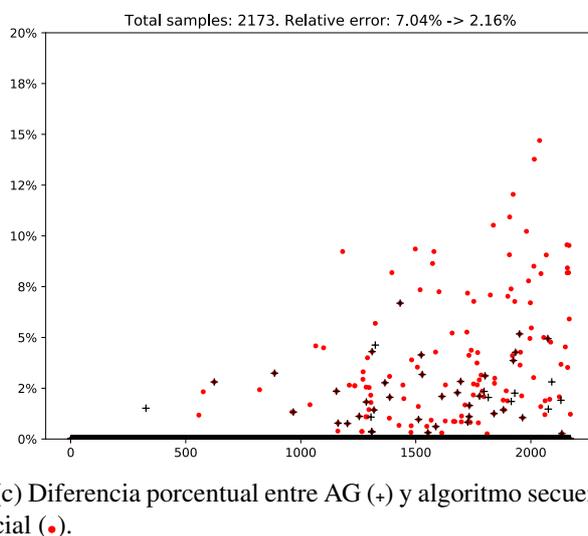
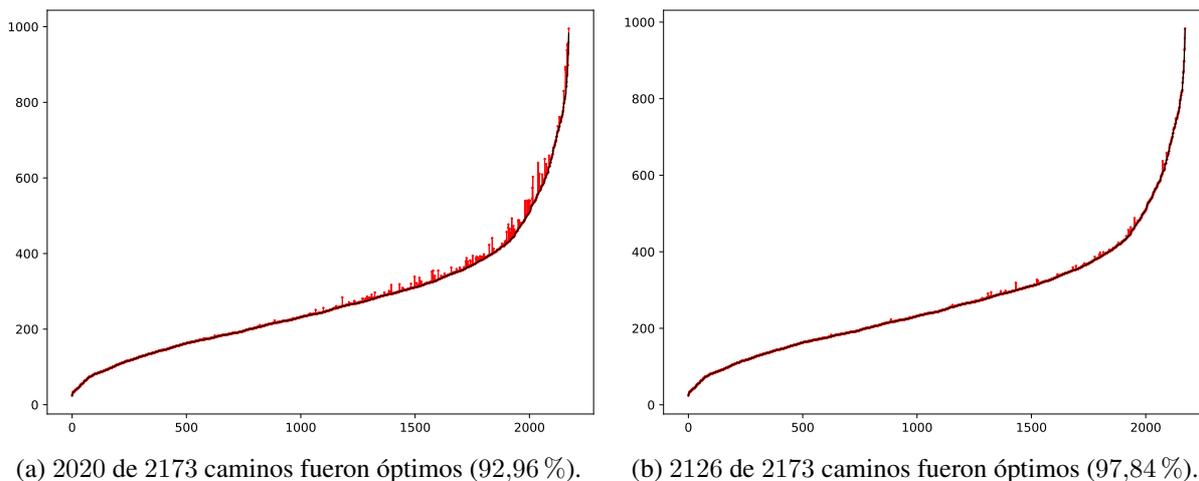


Figura 4: Las primeras figuras muestran la comparación entre la solución óptima (negro) y la solución dada por el algoritmo secuencial (rojo) en (a) y el algoritmo genético (rojo) en (b), eje- $x$  = número de experimentos y eje- $y$  = costo de los caminos correspondientes. (c) es la diferencia porcentual entre la optimización evolutiva (+) y la optimización secuencial (•).

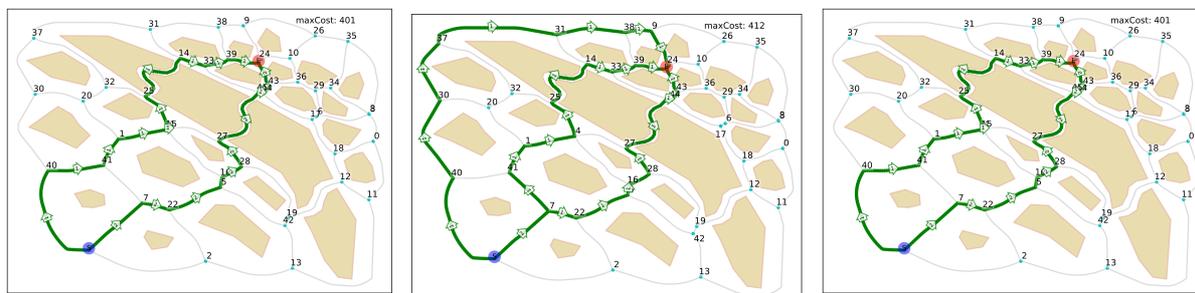


Figura 5: MPP para  $R = 3$ . Izquierda, caminos óptimos dados por el algoritmo exhaustivo,  $C_{\max} = 401$ . Centro, caminos encontrados por el algoritmo secuencial,  $C_{\max} = 412$ . Derecha, caminos dados por el algoritmo evolutivo,  $C_{\max} = 401$ .

**REFERENCIAS**

Altaharwa I., Sheta A., y Alweshah M. A mobile robot path planning using genetic algorithm in static environment. *Journal of Computer Science*, 4:341–344, 2008.

- Bai C., Duan H., Li C., y Zhang Y. Dynamic multi-uavs formation reconfiguration based on hybrid diversity-pso and time optimal control. En *Intelligent Vehicles Symposium, 2009 IEEE*, páginas 775–779. 2009. ISSN 1931-0587. doi:10.1109/IVS.2009.5164376.
- Castillo O. Trujillo L. M.P. Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots. *Soft Computing*, 11(1):269–279, 2007.
- Dijkstra E.W. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.
- Hart P. E. Nilsson N. J. R.B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- Hoshiar A.K., Kianpour M., Nazarahari M., y H.Korayem M. Path planning in AFM nanomanipulation of multiple spherical nano particles using a co-evolutionary Genetic Algorithm. En *International conference on manipulation, automation and robotics at small scales (MARSS)*. 2016.
- Hussein A., Mostafa H., Badrel-din M., Sultan O., y Khamis A. Metaheuristic optimization approach to mobile robot path planning. En *2012 International Conference on Engineering and Technology (ICET)*, páginas 1–6. 2012. doi:10.1109/ICEngTechnol.2012.6396150.
- Miao Y., Khamis A., y Karrayand M. Kamel F. A novel approach to path planning for autonomous mobile robots. *Control and Intelligent Systems*, 39(1):235–244, 2011.
- Noormohammadi Asl A., Menhaj M.B., y Sajedin A. Control of leader-follower formation and path planning of mobile robots using asexual reproduction optimization (aro). *Appl. Soft Comput.*, 14:563–576, 2014. ISSN 1568-4946. doi:10.1016/j.asoc.2013.07.030.
- P. Raja S.P. Optimal path planning of mobile robots. *International Journal of Physical Sciences*, 7(1):1314–1320, 2012.
- Pereyra M.E., Araguás R.G., y Kulich M. Sequential path planning for a formation of mobile robots with split and merge. En *Modelling and Simulation for Autonomous Systems*, volumen 10756, páginas 59–71. Springer International Publishing, 2017a. ISBN 978-3-319-76071-1.
- Pereyra M.E., Araguás R.G., y Kulich M. Sequential path planning for a formation of mobile robots with split and merge. En *4th IEEE Latin American Conference on Computational Intelligence, LA-CCI 2017*. IEEE, 2017b.
- Qu H. Xing K. A.T. An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing*, 120(1):509–517, 2013.
- R. Kala A. Shukla R.T. Dynamic environment robot path planning using Hierarchical Evolutionary Algorithms. *Cybernetics and Sys: An International Journal*, 41(1):435–454, 2010.
- Sharir M. Algorithmic motion planning in robotics. *IEEE Computer Society*, 1(1):9–19, 1989.
- Tang K.S., Man K.F., Kwong S., y He Q. Genetic algorithms and their applications. *IEEE Signal Processing Magazine*, 13(6):22–37, 1996. ISSN 1053-5888. doi:10.1109/79.543973.
- Tuncer A. Y.M. Dynamic path planning of mobile robots with improved genetic algorithm. *Computers and Electrical Engineering*, 38(1):1564–1572, 2012.
- Yang S. y Meng M. An efficient neural network approach to dynamic robot motion planning. *Neural Networks*, 13(2):143–148, 2000.
- Yüksel T. y Sezgin A. An implementation of path planning algorithms for mobile robots on a grid based map. 2008.