

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



Aplicaciones Prácticas para Algoritmos de Inteligencia y  
Robótica de Enjambre

Protocolo de trabajo de graduación presentado por Daniela María  
Baldizón García, estudiante de Ingeniería Mecatrónica

Guatemala,

2021

## Resumen

El objetivo principal de este proyecto es la implementación de algoritmos de inteligencia computacional y robótica de enjambre, y desarrollar aplicaciones para dichos algoritmos. Para ello se implementarán los algoritmos Particle Swarm Optimization y Ant Colony Optimization modificados, para que estos resuelvan problemas de exploración de terrenos, planificación de trayectorias y evasión de obstáculos, y problemas de procesamiento de imágenes biomédicas. Asimismo, para estos algoritmos se realizarán simulaciones realistas y en los casos que aplique se tomarán en cuenta restricciones físicas que puedan afectar el funcionamiento del algoritmo, como lo son los obstáculos.

Para los problemas de exploración de terrenos, planificación de trayectorias y evasión de obstáculos se proponen dos algoritmos. El primero se enfoca en la planificación de trayectorias y evasión de obstáculos, y el segundo en explorar terrenos desconocidos y realizar mapeos de estos. Para el algoritmo de planificación de trayectorias, se validará el funcionamiento en la plataforma Webots utilizando robots móviles. Para los problemas de procesamiento de imágenes médicas se propone un algoritmo que realiza registro de imágenes médicas y otro que realiza segmentación de imágenes médicas.

La implementación de estos algoritmos brindará herramientas que dan solución a evitar poner en riesgo la vida de las personas. Con las aplicaciones biomédicas se brindará una herramienta que sea capaz de dar resultados más exactos para el procesamiento de imágenes médicas, y que no siempre se pueden obtener únicamente mediante el ojo humano.

## Antecedentes

### Robotarium de Georgia Tech

El proyecto de Robotarium del Tecnológico de Georgia en Estados Unidos proporciona una plataforma de investigación de robótica de enjambre accesible de forma remota de libre acceso. El objetivo de este proyecto es que cualquier persona pueda cargar y probar sus ideas en un hardware robótico real, y así librar las inversiones significativas en mano de obra. Para utilizar el Robotarium solo se necesita descargar el simulador de MATLAB o Python y registrarse en la página para recibir la aprobación de la administración del Robotarium [1].

### Algoritmo D\*

Es un algoritmo popular para planificación de trayectorias de robots. Este algoritmo encuentra la mejor trayectoria a través de un grafo, para esto primero calcula el grafo, el cual corresponde a la cuadrícula de ocupación de entrada [2].

Primero, el algoritmo generaliza la cuadrícula de ocupación a un mapa de costo que representa el costo de moverse entre cada celda en dirección horizontal o vertical. Segundo, el algoritmo permite la replanificación incremental. Esto último es importante si mientras se están moviendo los robots, se descubre que el lugar es diferente al mapa que se tiene. Si se

descubre que una ruta tiene más costo del esperado o está bloqueada, se puede replanificar para encontrar una mejor trayectoria [2].

D\* encuentra el camino que minimiza el costo total de moverse. Si se está interesado en el tiempo mínimo que para alcanzar una meta entonces el costo es el tiempo que toma cruzar una celda [2].

## Aplicaciones de Ant Colony Optimization (ACO)

En [3] se encuentra un trabajo en el que se utiliza el ACO para registro de imágenes médicas. Toman una imagen a la que nombran imagen plantilla y el algoritmo se encarga de realizar transformaciones en dicha imagen para que sea lo más parecida posible a una imagen de referencia. La imagen generada por el algoritmo es la imagen registrada, y se puede ver de qué forma se realizó la transformación de la imagen por medio de un mapa de cuadrícula de deformación generado por el algoritmo. Este algoritmo es utilizado para el registro de imágenes médicas.

En [4] proponen un algoritmo basado en el ACO que garantiza que los robots encuentran un camino satisfactorio en presencia de pasillos estrechos. Este trabajo se centra específicamente en un caso de dos dimensiones, y consiste en crear el ambiente en el que se desea que los robots encuentren la trayectoria hasta el punto especificado, y si dentro de esta trayectoria se encuentra un pasillo estrecho, este pueda ser tomado como parte de la trayectoria o esquivado. Además dentro del ambiente se encuentran obstáculos que son evadidos de forma satisfactoria. En este trabajo comparan este algoritmo con algoritmos similares dedicados a la generación de trayectorias con ACO, y se obtuvo que este converge más rápido.

## Aplicaciones de Particle Swarm Optimization (PSO)

En [5] se propone un algoritmo que mezcla el PSO con campos de Markov ocultos (HMRF por sus siglas en inglés). El objetivo de este algoritmo es implementar un método para la segmentación de imágenes médicas, pues este es uno de los problemas fundamentales en el campo de la segmentación de imágenes. Por medio de los HMRF se encuentra la función de costo a optimizar por cada segmento de la imagen, y luego se optimiza la función encontrada mediante el PSO. En este artículo se realiza la segmentación específicamente para imágenes cerebrales.

## Implementación de Algoritmos de Inteligencia de Enjambre en UVG

En la fase anterior a este trabajo [6] se implementó el algoritmo *Ant System* (AS) (o *Ant Colony* (ACO)) como planificador de trayectorias. Dicho algoritmo se basa en el comportamiento de las hormigas para buscar y hallar alimento. Con esta implementación se obtuvo una alternativa de planificación de trayectorias a parte de la del *Modified Particle Swarm Optimization* (MPSO), desarrollada en el proyecto Robotat [7]. Estos algoritmos de inteligencia de enjambre tienen diferentes enfoques, el de AS es basado en teoría de grafos y el del MPSO es en funciones de costo.

Se encontraron los parámetros para el correcto funcionamiento del AS y se validaron por medio de simulaciones computarizadas que permiten visualizar el comportamiento de las feromonas depositadas por la colonia. Se adaptaron los modelos de movimiento y de cinemática de los Bitbots de UVG al ACO, y así se logró realizar la implementación en Webots para comparar su desempeño con el del MPSO.

Se implementó distintos controladores para considerar que los motores que controlan al robot tienen un límite de velocidad y garantizar que este haga lo deseado. Entre los controladores que se implementaron están: Transformación de unicycle (TUC), PID de velocidad lineal y angular, PID de acercamiento exponencial, de pose, de pose de Lyapunov, Closed-loop steering, Regulador cuadrático lineal (LQR) y Integrador cuadrático lineal (LQI). Se concluyó que los modelos de movimiento de los Bitbots desarrollados en el proyecto Robotat [7] funcionan tanto en AS como en MPSO.

Se realizaron pruebas con computación paralela para agilizar la planeación de trayectorias de los robots con el Ant System. También se realizaron pruebas con algoritmos genéticos para explorar si estos pueden ser alternativa al AS y MPSO, pero los resultados no fueron satisfactorios.

En los trabajos anteriores no se especificó ninguna tarea concreta para llevar a cabo durante la ejecución de los algoritmos, es decir, únicamente se implementaron los algoritmos como tal, sin asignarles alguna aplicación específica. Algunas de las simulaciones de estos trabajos tampoco toman en cuenta obstáculos que puedan presentarse en las trayectorias de los robots.

## Justificación

En los trabajos previos se implementaron los algoritmos *Ant Colony* y *Modified Particle Swarm Optimization* como planificadores de trayectorias. También se realizaron simulaciones utilizando la plataforma de Webots para validar el funcionamiento de dichos algoritmos tomando en cuenta las restricciones físicas de los robots, sin embargo, en estas simulaciones únicamente se validó el funcionamiento de los algoritmos sin tomar en cuenta obstáculos y restricciones que pudiera presentar una aplicación real. Debido a que no se tomaron en cuenta estas restricciones, se desearía implementar los algoritmos de tal forma que se puedan lograr simulaciones más realistas, además se quisiera orientar la implementación de los algoritmos a aplicaciones prácticas.

Uno de los campos en los que se quiere explorar aplicaciones es el de biomédica. Esto debido a que en la actualidad algunos métodos utilizados en el campo de la medicina aún son muy rudimentarios y requieren de la intervención de médicos o especialistas, por lo que al explorar aplicaciones biomédicas se puede obtener una herramienta que facilite el trabajo en el área de la medicina y brinde resultados más exactos.

Otro campo en el que se quiere implementar aplicaciones es el de exploración de terrenos, planificación de trayectorias y evasión de obstáculos. Esto se debe a que el trabajo de exploración y planificación puede llegar a ser muy riesgoso para las personas y se exponen vidas humanas al realizar estos trabajos, ya que en algunos casos puede llegar a ser muy peligroso. Entonces, es más práctico que estos problemas sea resueltos por un algoritmo que

puede brindar resultados óptimos y sin exponer vidas.

## Objetivos

### Objetivo General

Implementar algoritmos de inteligencia computacional y robótica de enjambre, y desarrollar escenarios prácticos simulados y aplicaciones para dichos algoritmos.

### Objetivos Específicos

- Implementar algoritmos para exploración de terrenos, planificación de trayectorias, y evasión de obstáculos.
- Implementar algoritmos para aplicaciones biomédicas.
- Validar la implementación de los algoritmos mediante simulaciones realistas y restricciones físicas.

## Marco teórico

### Particle Swarm Optimization (PSO)

Es un algoritmo de optimización meta-heurístico ya que utiliza analogías con otros procesos para resolver un problema. El algoritmo se inspira en la evolución del comportamiento colectivo, trata de imitar el comportamiento social de grupos de animales como manadas, cardúmenes, parvadas, etc. Los métodos meta-heurísticos no se enfocan en resolver un problema en particular, por lo que estos pueden emplearse en cualquier problema y obtener un resultado aceptable. El PSO no es determinista, lo que significa que los resultados obtenidos no siempre serán los mismos aunque se trate de una misma función. Los algoritmos de enjambre de partículas se caracterizan por ser eficientes y de bajo costo computacional [8].

A lo largo del tiempo se han propuesto variaciones para el PSO, esta sección se enfoca en la versión inercial o clásica. El objetivo de un problema de optimización es encontrar un vector  $X = [x_1, x_2, x_3, \dots, x_n]$  que minimice o maximice cierta función  $f(X)$ . El vector variable  $X$  es el vector de posición que representa un modelo variable y es de dimensión  $n$ , donde  $n$  es la cantidad de variables que hay que encontrar en el problema. La función  $f(x)$  es la función peso, esta evalúa que tan bien o mal es la posición  $X$  [8].

Tomando un enjambre con  $P$  cantidad de partículas, hay un vector posición  $X_i^t = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{in})^T$  y un vector velocidad  $V_i^t = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{in})^T$  a una  $t$ -iteración para cada una de las partículas que compone al enjambre. Estos vectores son actualizados a través de una dimensión  $j$  de acuerdo a las siguientes ecuaciones [8]:

$$V_{ij}^{t+1} = wV_{ij}^t + c_1r_1^t (pbest_{ij} - X_{ij}^t) + c_2r_2^t (gbest_j - X_{ij}^t) \quad (1)$$

$$X_{ij}^{t+1} = X_{ij}^t + V_{ij}^{t+1} \quad (2)$$

Donde  $i = 1, 2, \dots, P$  y  $j = 1, 2, \dots, n$ .

La ecuación (1) actualiza la velocidad y (2) actualiza la posición de las partículas.  $w$  es la constante de inercia peso. El segundo término representa la cognición individual y es calculado por medio de la diferencia entre la mejor posición de la propia partícula (*pbest*) y su posición actual. La idea detrás de este término es que mientras la partícula está más lejos de su mejor posición la diferencia aumenta; por consecuencia, el término aumenta atrayendo a la partícula a su mejor posición. El parámetro  $c_1$  multiplica al término como una constante positiva y es un parámetro individual-cognición, y marca la importancia de las experiencias anteriores de cada partícula. El parámetro  $r_1$ , es de valor aleatorio con rango  $[0,1]$  y evita las convergencias prematuras. [8].

El tercer término de (1) es el de aprendizaje social. Esto es así debido a que todas las partículas del enjambre pueden intercambiar información sobre el mejor punto alcanzado sin importar cual lo encontró (*gbest*). La diferencia que se muestra en este término actúa como atractor para las partículas al mejor punto global encontrado en una iteración  $t$ . Similarmente,  $c_2$  es el parámetro de aprendizaje social, y su valor representa la importancia del aprendizaje global del enjambre. Y  $r_2$  tiene el mismo rol que  $r_1$  [8].

## Ant Colony Optimization (ACO)

Marco Dorigo desarrolló el primer algoritmo que modela el comportamiento de las hormigas al buscar alimento, dicho algoritmo fue llamado como Ant Colony Optimization Meta-Heuristic (ACO-MH). Después de esto, se han desarrollado numerosas variantes para dicho algoritmo, como lo es el Ant System (AS) [9].

Las hormigas reales empiezan la búsqueda de alimento comportándose inicialmente de forma aleatoria o con un patrón de actividad caótica. Al encontrar una fuente de alimento, los patrones de actividad se vuelven más organizados con más hormigas siguiendo el mismo camino hacia la fuente de alimento. Lo que ocurre es que las hormigas que encuentran el alimento influyen a las demás hacia la comida. La mayoría de las especies de hormigas utilizan como forma de reclutamiento la comunicación indirecta, la cual hacen por medio de rastros de feromonas y se denomina *stigmergy*. Cuando una hormiga encuentra una fuente de alimento, esta regresa al hormiguero por el mismo camino llevando alimento, al hacer esto deja rastros de feromonas en el camino para indicar a las demás hormigas qué camino seguir. Los caminos con mayor concentración de feromonas tienen una mayor probabilidad de ser elegidos. [9].

Deneubourg estudió el comportamiento de las hormigas por medio de un experimento, en el que el hormiguero está separado de la fuente de alimento por un puente con dos caminos de la misma longitud. Al pasar el tiempo, las hormigas eligieron de manera aleatoria uno de los dos caminos para que fuera la ruta hacia la fuente de alimento. Este experimento es conocido como “El puente binario”. [9].

## Simple Ant Colony Optimization (SACO)

El SACO es una implementación algorítmica del experimento del puente binario de Deneubourg. Considerando el problema general de encontrar el camino más corto entre dos nodos en un grafo  $G = (V, E)$ , donde  $V$  es el conjunto de vértices o nodos y  $E$  es la matriz que representa la conexión entre los nodos. El largo  $L^k$  del camino construido por la hormiga  $k$  es calculado como la cantidad de saltos en el camino desde el nodo que representa al nido hasta el que representa el destino con la comida. Cada arista  $(i, j)$  del grafo tiene una concentración  $\tau_{i,j}$  asociada de feromona [9].

En el algoritmo, a cada arista se le asigna un valor aleatorio pequeño de feromona ( $\tau_{ij}(0)$ ). Luego, las hormigas deciden de forma aleatoria que camino/arista seguir.  $k$  hormigas se colocan en el nodo fuente. En cada iteración del algoritmo cada hormiga construye un camino (solución) hacia el nodo de destino. En cada nodo  $i$ , cada hormiga  $k$  selecciona el siguiente nodo  $j$ , basándose en la probabilidad de transición,

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t)}{\sum_{j \in \mathcal{N}_i^k} \tau_{ij}^\alpha(t)} & \text{if } j \in \mathcal{N}_i^k \\ 0 & \text{if } j \notin \mathcal{N}_i^k \end{cases} \quad (3)$$

donde  $\mathcal{N}_i^k$  es el conjunto de nodos viables conectados al nodo  $i$ , respecto a la hormiga  $k$ .  $\alpha$  es una constante positiva que amplifica la influencia de las concentraciones de feromona. [9].

Una vez todas las hormigas han construido un camino completo desde el nodo de origen al nodo de destino, cada hormiga vuelve a recorrer su camino hacia el nodo fuente, y deposita una cantidad de feromona,

$$\Delta\tau_{ij}^k(t) = \frac{Q}{L_k}(t) \quad (4)$$

en cada arista  $(i, j)$  del correspondiente camino.  $L^k(t)$  es el largo del camino construido por una hormiga  $k$  en el tiempo  $t$ . Donde  $Q$  es una constante y  $L$  es el costo del trayecto, como el largo de este. El resultado de esto representa el cambio de feromona entre el nodo  $i$  y  $j$  que la hormiga visitó en la iteración  $t$  [10].

## Ant System (AS)

Este algoritmo es una mejora del SACO (A pesar que el AS se desarrolló antes). La probabilidad de transición cambia para incluir la información heurística y agregar capacidad de memoria con una lista tabú, la cual previene que una hormiga visite el mismo nodo dos veces [10]. La ecuación de probabilidad de transición es,

$$p_{(i,j)}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij}(t))^\beta}{\sum_{k \in J_k} (\tau_{ij}(t))^\alpha \cdot (\eta_{ij}(t))^\beta} & \text{si } j \in N_i^k \\ 0 & \text{si } j \notin N_i^k \end{cases} \quad (5)$$

donde  $\eta_{ij}$  representa la efectividad *a priori* del movimiento desde  $i$  a  $j$ , o el inverso del costo de la arista [9].

Por lo tanto,

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (6)$$

donde  $d_{ij}$  es la distancia (o costo) entre los nodos  $i$  y  $j$ [9].

## Aplicaciones del Ant Colony Optimization

Los algoritmos de ACO se utilizan normalmente para resolver problemas no determinísticos, los cuales son aquellos que no se pueden solucionar por algoritmos con estructura polinomial, por lo que requieren del tipo exponencial. Para la solución, se realizan iteraciones con las que se pueden encontrar respuestas globales y particulares [11].

Algunos campos de aplicación de los algoritmos por optimización por colonia de hormigas son: redes neuronales, inteligencia artificial, optimización de funciones numéricas, sistemas difusos, procesamiento de imágenes, control de sistemas, problemas del hombre viajero, enrutamiento de vehículos y líneas de producción de carros [11].

## Registro de Imágenes Médicas

El registro de imágenes es un proceso que superpone dos o más imágenes de varios equipos de imágenes o sensores tomadas en diferentes momentos y ángulos, o de la misma escena para alinear geométricamente las imágenes para su análisis [12].

En [3] se plantea un algoritmo basado en el ACO que realiza registro de imágenes. En el algoritmo desarrollado se toman los píxeles de las imágenes como el nido de un enjambre de hormigas, y las hormigas son capaces de buscar la “comida” en su memoria. Luego, las hormigas depositan feromonas en los píxeles, que afectan el movimiento de las hormigas. El proceso de registro de actualización de la feromona, la dirección y la distancia de avance se repite hasta que el coeficiente de correlación entre las imágenes registradas y de referencia alcanza un máximo. Este algoritmo puede ser utilizado en diversas áreas, una de ellas es el registro de imágenes médicas.

El proceso de registro consiste en tomar una imagen de referencia, que es respecto a la cual se miden los cambios en la otras imágenes, y tomar la imagen plantilla, que es la que tiene cambios respecto a la referencia, y con el algoritmo transformar la imagen plantilla en la referencia. La modificación de la plantilla es la imagen registrada y esta es casi idéntica a la referencia. Con el algoritmo se genera un mapa de cuadrícula de deformación y con este se pueden ver las transformaciones aplicadas a la plantilla. En la Figura 1 se muestran un resultado del algoritmo [3].





Figura 1: Ejemplo de registro de imágenes con ACO. Primera columna: imagen plantilla, segunda columna: imagen de referencia, tercera columna: imagen registrada, cuarta columna: mapa de cuadrícula de deformación. [3]

## Planificación de Trayectorias para Robots y Evasión de Obstáculos

En [4] se muestra una mejora al ACO que tiene como fin garantizar que un grupo de robots encuentren una trayectoria satisfactoria en un ambiente que contiene pasillos estrechos. En este documento, se presenta una nueva matriz de adyacencia ponderada para determinar la dirección de paso y, por lo tanto, los pasillos estrechos se evitan rediseñando las reglas de paso. También, se introducen la mejor y la peor hormiga para el ajuste de feromonas para facilitar el proceso de búsqueda.

Los pasos de este algoritmo modificado se muestran a continuación [4]:

*Paso 1.* Recopilar la información del entorno y obtener el mapa de cuadrícula del entorno. Luego, establecer el punto de inicio y final. Inicializar los parámetros de la colonia de hormigas.

*Paso 2.* Determinar la ubicación actual y si hay un camino estrecho alrededor de la cuadrícula.

*Paso 3.* Calcular la distancia entre todas las cuadrículas y las cuadrículas circundantes de acuerdo a las reglas de caminar y construir una nueva matriz de adyacencia ponderada.

*Paso 4.* Colocar las  $m$  hormigas en el punto de inicio y elegir el siguiente nodo- $j$  basándose en la distancia de la nueva matriz de adyacencia ponderada con la distribución de feromonas en diferentes posiciones.

*Paso 5.* Añadir el nodo- $j$  a la lista tabú de la  $k$ -hormiga.

*Paso 6.* Repetir los pasos 4 y 5. La hormiga alcanza el punto objetivo, y guarda la ruta y la longitud de ruta de cada hormiga, de las cuales se selecciona la ruta óptima y la longitud de ruta óptima. La longitud de ruta promedio de las  $m$  hormigas se calcula en este ciclo.

*Paso 7.* Actualizar las feromonas de las rutas.

*Paso 8.* Determinar si el algoritmo alcanza el máximo número de iteraciones, luego terminar el algoritmo y generar la ruta óptima. Si esto no se cumple, repetir los pasos 3 al 7.

## Aplicaciones del Particle Swarm optimization

### Segmentación de imágenes médicas

“La segmentación de imágenes es una técnica de tratamiento digital que permite extraer información de los objetos a partir unas escenas dadas y plasmar esta información en un sistema estructurado” [13].

El modelo de campos aleatorios ocultos de Markov (HMRF) se deriva de los modelos de Markov ocultos (HMM), que se definen como procesos estocásticos generados por una cadena de Markov cuya secuencia de estados no se puede observar directamente, únicamente por medio de una secuencia de observaciones. Se supone que cada observación es una función estocástica de la secuencia de estados. La cadena de Markov subyacente cambia su estado de acuerdo con una matriz de probabilidad de transición  $l \times l$ , donde  $l$  es el número de estados [14].

En [5] se plantea un algoritmo para segmentar imágenes cerebrales que mezcla al PSO con el modelo HRMF. El modelo HRMF provee una manera de modelar el problema de segmentación de imágenes. El proceso de segmentación consiste en encontrar información oculta (la imagen segmentada) mediante la observación de los datos de la imagen original. La imagen segmentada se busca mediante el criterio de Máximo a Posteriori (MAP), sin embargo, este problema no tiene solución computacional por lo que es necesario implementar técnicas de optimización.

Una imagen está formada por un conjunto finito  $S$  de espacios que corresponden a los píxeles. Los espacios en  $S$  están relacionados por un sistema de vecindario  $V(S)$ . La imagen a segmentar o la imagen observada es vista por el modelo HMRF como una realización de una familia de variables aleatorias definidas en  $S$ . Cada variable aleatoria toma valores entre 0 y 255. La imagen segmentada es vista como la realización de otro campo aleatorio de Markov definido en la misma red  $S$ . Sus valores toman valores entre 1 y  $K$ , donde  $K$  representa el número de clases en la imagen. A partir de esto se puede calcular la función costo, para la cual los resultados se obtienen a partir del PSO [5].

## Metodología

Para cumplir con los objetivos planteados se llevará a cabo lo siguiente metodología:

### Levantar la fase previa

Se tomarán como base de los algoritmos a implementar, los algoritmos implementados en las fases previas, ya que los nuevos algoritmos son modificaciones de estos. Se utilizarán los controladores propuestos en las fases previas para las simulaciones de los algoritmos a implementar.

## **Implementación de algoritmos para exploración de terrenos, planificación de trayectorias, y evasión de obstáculos**

Para cumplir con este objetivo se propone lo siguiente:

1. Implementar en Matlab el algoritmo de planificación de trayectorias para robots y evasión de obstáculos propuesto en [4] y verificar su correcto funcionamiento con las constantes dadas en el documento.
2. Añadir condiciones de finalización del algoritmo antes mencionado dada la convergencia de cierta cantidad de hormigas.
3. Realizar pruebas al algoritmo anterior con diferentes entornos, variando la cantidad de hormigas y el máximo de iteraciones.
4. Implementar en Matlab un algoritmo que modifique el funcionamiento del ACO de tal forma que se pueda realizar exploración de terrenos y el mapeo de estos.
5. Realizar pruebas con diferentes terrenos de exploración y analizar los resultados devueltos por el algoritmo mencionado en el inciso 4.
6. Comparar los mapas devueltos por el último algoritmo mencionado con los mapas de exploración originales.

## **Implementar algoritmos para aplicaciones biomédicas**

Para cumplir con este objetivo se propone:

1. Implementar en Matlab el algoritmo propuesto en [3] para registro de imágenes. Asimismo, encontrar las constantes requeridas para el funcionamiento del algoritmo y probar las constantes encontradas en la fase previa.
2. Realizar pruebas del algoritmo utilizando imágenes sencillas.
3. Validar el funcionamiento del algoritmo con imágenes médicas. Por ejemplo, imágenes con tumores, deformaciones, etc.
4. Implementar en Matlab el algoritmo para segmentación de imágenes médicas y verificar su correcto funcionamiento con las constantes dadas en el documento.
5. Realizar pruebas al algoritmo utilizando distintas imágenes médicas que sean válidas para segmentar.

## **Validar la implementación de los algoritmos mediante simulaciones realistas y restricciones físicas**

Para cumplir con este objetivo se propone:

1. Recrear en Webots los entornos creados para el algoritmo de planificación de trayectorias para robots y evasión de obstáculos, y probar las trayectorias encontradas utilizando un robot móvil.
2. Realizar pruebas con los controladores propuestos en las fases anteriores de implementación de algoritmos de inteligencia de enjambre UVG.

## Cronograma de actividades

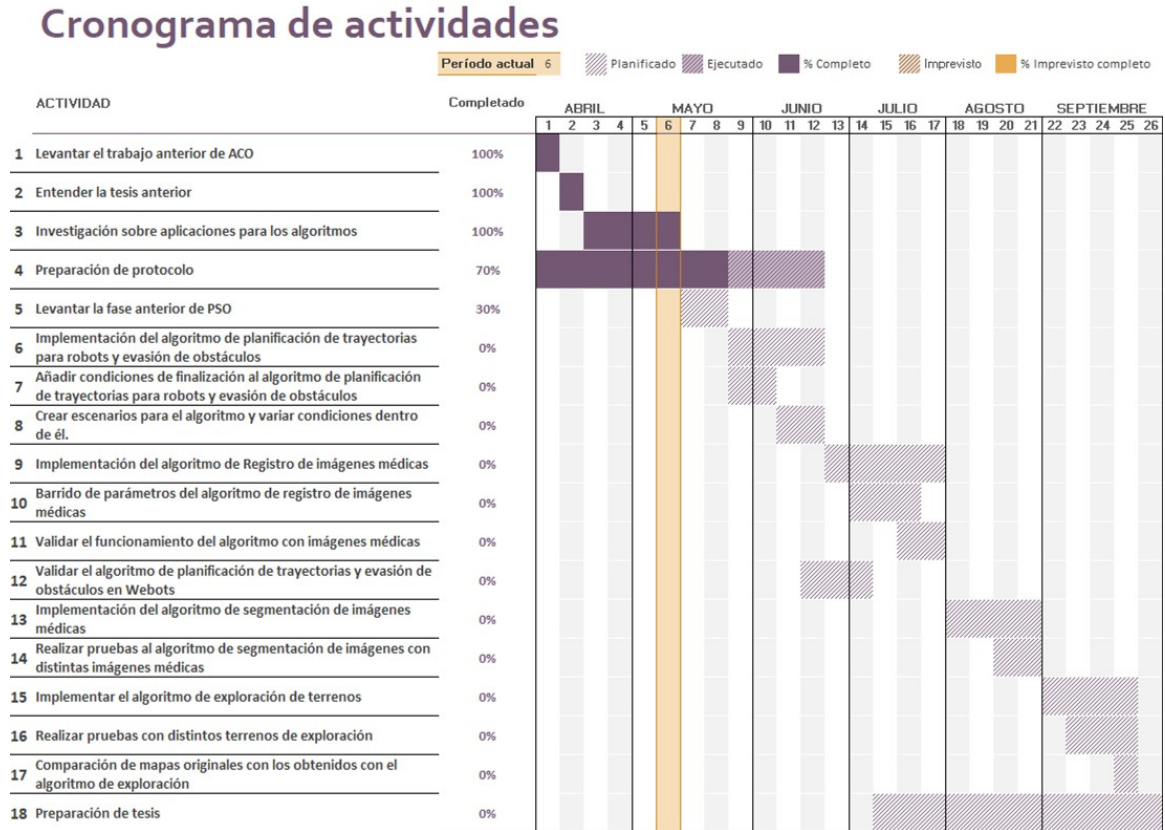


Figura 2: Cronograma de Actividades

## Descripción de Actividades

1. Instalar y correr todos los archivos y programas de la fase Aprendizaje Automático, Computación Evolutiva e Inteligencia de Enjambre para Aplicaciones de Robótica.
2. Entender a profundidad el trabajo realizado en la fase anterior.
3. Búsqueda de aplicaciones para implementar en el trabajo actual.
4. Redacción del protocolo.
5. Instalar, correr y probar los archivos y programas de la fase anterior.

6. Implementación del algoritmo descrito en [4] y hacer pruebas para verificar su funcionamiento.
7. Mejorar las condiciones de paro del algoritmo.
8. Crear diferentes entornos para probar el algoritmo, modificar la cantidad inicial de hormigas y el número máximo de iteraciones.
9. Implementación del algoritmo descrito en [3] y verificar su funcionamiento con las constantes dadas. Hacer pruebas del algoritmo con imágenes sencillas.
10. Barrido de parámetros para encontrar las constantes que logran en el mejor comportamiento del algoritmo.
11. Hacer pruebas con imágenes médicas, por ejemplo, tumores, deformaciones, desgarres, etc.
12. Recrear los entornos creados para el algoritmo y probar las trayectorias encontradas utilizando un robot móvil. Probar los controladores propuestos en las fases anteriores.
13. Implementación del algoritmo descrito en [5] y verificar su funcionamiento con las constantes dadas en el documento.
14. Realizar pruebas y validar el funcionamiento del algoritmo mediante distintas imágenes médicas que sean comunes en la segmentación de imágenes.
15. Implementar en Matlab un algoritmo que tome como base al ACO y que lleve a cabo la exploración de terrenos. Los resultados deben ser el mapeo de los terrenos por medio del algoritmo.
16. Modificar los terrenos a explorar y analizar si los resultados obtenidos son satisfactorios
17. Realizar la comparación del terreno original y el mapeado por el algoritmo.
18. Redacción del documento de tesis.

## Índice preliminar

## Referencias

- [1] *Robotarium / Institute for Robotics and Intelligent Machines*. dirección: <http://www.robotics.gatech.edu/robotarium> (visitado 29-03-2021).
- [2] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, en. Springer Science & Business Media, nov. de 2011, ISBN: 978-3-642-20143-1.
- [3] T. X. Lin y H. H. Chang, “Medical Image Registration Based on an Improved Ant Colony Optimization Algorithm,” en, vol. 5, n.º 1, pág. 6, 2016.
- [4] H.-J. Wang, Y. Fu, Z.-Q. Zhao e Y.-J. Yue, “An Improved Ant Colony Algorithm of Robot Path Planning for Obstacle Avoidance,” *Journal of Robotics*, vol. 2019, K. Watanabe, ed., pág. 6 097 591, jun. de 2019, Publisher: Hindawi, ISSN: 1687-9600. DOI: 10.1155/2019/6097591.

- [5] E.-H. Guerrou, R. Mahiou y S. Ait-Aoudia, "Hidden Markov Random Fields and Particle Swarm Combination for Brain Image Segmentation," en, vol. 15, n.º 3, pág. 7, 2018.
- [6] G. Iriarte, "Aprendizaje Automático, Computación Evolutiva e Inteligencia de Enjambre para Aplicaciones de Robótica," en *Trabajo de Graduación, Modalidad Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2020.
- [7] A. Aguilar, "Algoritmo Modificado de Optimización de Enjambre de Partículas (MPSO)," en *Trabajo de Graduación, Modalidad Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2019.
- [8] B. S. G. d. Almeida y V. C. Leite, "Particle Swarm Optimization: A Powerful Technique for Solving Engineering Problems," en, *Swarm Intelligence - Recent Advances, New Perspectives and Applications*, dic. de 2019, Publisher: IntechOpen. DOI: 10.5772/intechopen.89633.
- [9] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, 2nd. Wiley Publishing, 2007, ISBN: 0470035617.
- [10] M. N. Ab Wahab, S. Nefti-Meziani y A. Atyabi, "A Comprehensive Review of Swarm Optimization Algorithms," *PLOS ONE*, vol. 10, n.º 5, págs. 1-36, mayo de 2015. DOI: 10.1371/journal.pone.0122827.
- [11] C. A. R. Algarín, "Optimización por colonia de hormigas:" ES, *Ingeniería Solidaria*, vol. 6, n.º 10-11, págs. 83-89, 2010, Number: 10-11, ISSN: 2357-6014. dirección: <https://revistas.ucc.edu.co/index.php/in/article/view/454> (visitado 29-03-2021).
- [12] *Image Registration - an overview | ScienceDirect Topics*. dirección: <https://www.sciencedirect.com/topics/neuroscience/image-registration> (visitado 03-05-2021).
- [13] C. Arenal, "SEGMENTACIÓN DE IMÁGENES MÉDICAS," es, pág. 25,
- [14] *Hidden Markov Random Field Model*. dirección: <https://www.fmrib.ox.ac.uk/datasets/techrep/tr00yz1/tr00yz1/node5.html> (visitado 11-05-2021).