

Instituto Tecnológico de Costa Rica
Área Académica de Ingeniería en Computadores

Programa de Licenciatura de Ingeniería en Computadores



**Proyecto de Diseño de
Ingeniería en Computadores**

Entregable 1

Selección del algoritmo de enjambre

Proyecto de Diseño

Diseño de pruebas en un entorno aleatorio para exploración
por medio de drones con inteligencia de enjambre

Elaborado por

Jose Ignacio Granados Marín

Ingeniería en Computadores

No. Carné: 2018319698

Fecha del informe: 24/03/2023

Supervisado por: MSc. Luis Alberto Chavarría Zamora

1. Actividad 1 (A1): Realizar investigación sobre algoritmos de enjambre y hacer una lista de los que se adapten a la solución del problema

1.1. Algoritmo de optimización de enjambre de partículas (PSO)

La optimización de enjambre de partículas (PSO) es un método de inteligencia de enjambre basado en la capacidad de los grupos. El algoritmo del enjambre de partículas simula el comportamiento del proceso de búsqueda de alimento y cortejo de los pájaros [1].

El PSO representa una técnica de optimización metaheurística bien conocida debido a su fácil implementación en problemas complejos y no supervisados. Es una técnica confiable que se ha utilizado para tratar varios problemas de optimización. Dicho algoritmo, inicializa un enjambre de partículas en el que atraviesan el espacio de búsqueda en busca de una mejor posición global óptima. De hecho, cada partícula representa una solución potencial al problema planteado [2].

En la vida real, las bandadas de pájaros cambian constantemente la posición y la dirección del vuelo. Según su información global y ajustando la velocidad y la posición, consiguen la mejor posición del individuo y consiguen que todos los equipos mantengan el estado óptimo [1].

En la optimización del enjambre de partículas, cada partícula es equivalente un pájaro, actualizan constantemente su posición y velocidad. Logran la solución óptima a través de la iteración continua. Determinando su propia posición y la calidad de la velocidad por dos valores .^{extremos.}en cada iteración. El primer valor extremo es la solución óptima encontrada por la propia partícula, llamada mejor personal, y el otro extremo es la solución óptima encontrada por toda la población actualmente, llamada máximo global. De modo que se pueda asegurar su posición y velocidad óptimas [1].

Algunas de sus ventajas son [1]:

- Es fácil de implementar.
- El principio de la optimización del enjambre de partículas es simple, sus parámetros son menores y su efecto de aplicación es obvio.
- Se utiliza principalmente para manejar problemas de optimización no lineal, funciones multipico complejas y no diferenciables.
- Es un proceso de resolución a partir de soluciones aleatorias, a través de cálculos iterativos para encontrar la solución óptima y luego evaluar la calidad de la solución a través de la aptitud.
- Este algoritmo puede ser flexible para satisfacer las necesidades de las aplicaciones prácticas a través de mejoras, mostrando su gran practicabilidad.

1.2. Algoritmo de colonia de hormigas (ACO)

El algoritmo de colonia de hormigas tiene una gran ventaja para abordar los problemas de optimización combinatoria. Varios investigadores han estudiado las tareas de programación utilizando el algoritmo de colonia de hormigas en el entorno de computación en la nube. Normalmente se clasifican de acuerdo con el enfoque de sus objetivos, como la eficiencia de la programación, el rendimiento del sistema o el costo. ACO representa un algoritmo inteligente para la planificación de rutas, el cual dispone de un fuerte mecanismo de cálculo. En general, se utiliza para la optimización mediante la actualización de los rastros de feromonas y la orientación de las hormigas alrededor del espacio de búsqueda mediante el cual, cada hormiga genera una nueva función de acondicionamiento físico que se utilizará para generar una aptitud global general. Además, una estrategia de este algoritmo está determinada por el método de la rueda de la ruleta en el que se repetirá hasta que se haya alcanzado el punto objetivo [2].

El ACO fue inspirado en el comportamiento de las hormigas en la búsqueda de alimento en el proceso. La hormiga individual tiene una clara división del trabajo, puede completar de manera eficiente las tareas asignadas a través de la colaboración entre sí [1].

Un estudio encontró que la transmisión de información entre hormigas individuales es por medio de una sustancia especial llamada feromona, el comportamiento de las hormigas se completa bajo la guía de la feromona. La concentración de feromonas es mayor, cuantas más hormigas se juntan, más feromonas quedan para entonces, lo que muestra un mecanismo de información de retroalimentación positiva. La mayor concentración de camino de feromonas a menudo significa el camino más corto, por lo que el camino atrae a más hormigas, el resultado final es que todas las hormigas han elegido este camino [1].

Algunas de sus ventajas son [1]:

- Tiene una gran robustez y la capacidad de buscar una mejor solución para resolver el rendimiento.
- Es un algoritmo evolutivo basado en la población fácil de implementar en paralelo.
- Es muy fácil de combinar con una variedad de algoritmos heurísticos para mejorar el rendimiento del algoritmo.
- El algoritmo de la colonia de hormigas converge lentamente y fácilmente para caer en el óptimo local.

Algunas de sus desventajas son [1]:

- La feromona inicial del algoritmo de colonia de hormigas es deficiente y generalmente requiere tiempos de búsqueda más largos, y este método es propenso al estancamiento, a saber, la búsqueda procede hasta cierto punto.
- La solución es casi idéntica para todos los individuos encontrados.
- La falta de un proceso de búsqueda adicional en el espacio de soluciones no conduce a encontrar una mejor solución.

1.3. Algoritmo de optimización de Drosophila (FOA)

El algoritmo de optimización de Drosophila (FOA) es un nuevo método inspirado en el comportamiento de búsqueda de alimento global de Drosophila y la evolución de la optimización. Una vez que se descubre la ubicación de un individuo, toda la drosófila volará en esa dirección [1].

El algoritmo de optimización de Drosophila primero inicializa las coordenadas de posición inicial de las poblaciones de Drosophila, inicializando una posición inicial de acuerdo con el rango de variación individual, luego proporciona la dirección y distancia aleatorias de acuerdo con el comportamiento de la mosca de la fruta en la búsqueda de alimento y valor de determinación de concentración de sabor introducido. Este valor es el recíproco de la posición de origen de drosophila a una distancia para representar, luego tome el valor de decisión de concentración de sabor a la función de determinación de densidad y encuentre el sabor de la concentración individual de drosophila, identificando la mejor concentración de sabor de grupos de drosophila y encontrando el valor óptimo. Lo último que se guarda es la concentración óptima del valor de drosófila y sus coordenadas de posición. En este momento la población de moscas volará a este lugar [1].

Algunas de sus ventajas son [1]:

- Es simple y fácil de implementar, y otros algoritmos no tienen la ventaja.
- Se aplica más fácilmente para resolver problemas prácticos.
- Tiene una fuerte capacidad de optimización global y alta precisión, y al mismo tiempo es de gran utilidad para resolver los parámetros de optimización del problema real no negativo.

Algunas de sus desventajas son [1]:

- El algoritmo de Drosophila también tiene sus propios inconvenientes, por ejemplo, el algoritmo de Drosophila no es adecuado para resolver el problema del argumento que es negativo, porque la función de determinación de la densidad del sabor solo puede procesar el argumento que es positivo.

- El algoritmo carece de estabilidad cuando se trata de problemas complejos del algoritmo.

1.4. Colonia de abejas artificiales (ABC)

El algoritmo de la colonia de abejas artificiales (ABC) simula el comportamiento de recolección de miel del enjambre de abejas, y las abejas tienen diferentes comportamientos de acuerdo con su respectiva división del trabajo y realizan la comunicación de información y el intercambio de enjambre de abejas para lograr la solución óptima. Dicho algoritmo divide el enjambre de abejas artificiales en tres categorías: abejas empleadas, espectadores y exploradores. En cada proceso de búsqueda, guiar a la abeja y seguir la fuente de alimento de la mina de abejas sucesivamente, es decir, buscar la solución óptima. Mientras explora, la abeja observa si está atrapada en un óptimo local y busca aleatoriamente otras posibles fuentes de alimento si está atrapada en un óptimo local. Cada fuente de alimento representa una posible solución al problema, y la cantidad de néctar de la fuente de alimento corresponde a la calidad de la solución [3].

El ABC representa un enfoque metaheurístico para investigar el comportamiento de las abejas. Las aplicaciones típicas de ABC se pueden encontrar en varias áreas, como la computación en la nube, el procesamiento de imágenes, el análisis de big data y las redes neuronales. Actualmente, el ABC es considerado como uno de los algoritmos de inteligencia de enjambre más exitosos en términos de optimización [2].

Algunas de sus ventajas son [3]:

- Las ventajas de ABC radican en la fuerte capacidad de búsqueda global y la rápida velocidad de convergencia. de estabilidad cuando se trata de problemas complejos del algoritmo.

Algunas de sus desventajas son [3]:

- Sus desventajas incluyen la diversidad de la raza es pobre.
- Cuando la solución está cerca del óptimo global, es fácil caer en el óptimo local, lo que resulta en un fenómeno de estancamiento.

1.5. Colonia de abejas artificiales (ABC)

La idea principal del algoritmo de optimización de luciérnagas (FA) es considerar la solución en el espacio como un individuo de luciérnaga, juzgando la calidad de las soluciones por dos indicadores de radio llamados el valor de fluorescencia y el radio de percepción. El tamaño del valor de la fluorescencia se utiliza para juzgar los méritos de las posiciones individuales, que la solución de los pros y los contras. El tamaño del radio de percepción se utiliza para medir la búsqueda individual [1].

El algoritmo implica dos factores clave, a saber, la intensidad de fluorescencia relativa y el atractivo relativo. El brillo de los fluorescentes se ve afectado por las luciérnagas que emiten de la ubicación del objetivo, el brillo más alto ilustra mejor la ubicación, ese es el mejor objetivo. El atractivo y el brillo se refieren. Cuanto más brillantes tengan las luciérnagas, mayor será la atracción. Las luciérnagas más brillantes pueden atraer la debilidad del brillo de las luciérnagas para que se muevan en esta dirección a la vista del brillo. Si el brillo de la luminiscencia es el mismo, su movimiento de luciérnagas es aleatorio. Inversamente proporcional a la distancia entre el grado de brillo y atracción, ambos disminuyen al aumentar la distancia [1].

Algunas de sus ventajas son [1]:

- El algoritmo no solo puede optimizar la función unimodal y encontrar la solución óptima global, sino que también puede optimizar la función multimodo y encontrar varios óptimos locales.
- Es fácil de operar, implementar y tiene menos parámetros de impacto y menos parámetros.

Algunas de sus desventajas son [1]:

- Deficiencias de la detección de picos bajos, la convergencia lenta, la solución sigue siendo inexacta.

2. Actividad 2 (A2): Documentar los criterios tomados en cuenta para la selección del algoritmo

El algoritmo que se utilizará para modelar el comportamiento de los drones en el entorno de pruebas aleatorio, será el algoritmo de optimización de enjambre de partículas (PSO). Dicha selección se debe a las siguientes razones:

- Los algoritmos de ACO y ABC se encuentran orientados principalmente para la solución de problemas relacionados con la computación en la nube, procesamiento de imágenes, análisis de big data y redes neuronales. Por lo que, su respectiva codificación puede llegar a ser más elaborada y brindar funcionalidades adicionales que no sean necesarias, impactando negativamente la cantidad de iteraciones y cálculos involucrados.
- Dado que el algoritmo de PSO fue diseñado para solventar problemas complejos y no supervisados, el objetivo actual de exploración de territorios puede llegar alcanzado ya que este algoritmo permitirá buscar la posición óptima global de todo el entorno de pruebas aleatorio, por medio de la comunicación y comportamiento colectivo de los drones.
- El hecho de que en el algoritmo PSO, cada partícula dentro del conjunto represente una solución potencial al problema de exploración planteado para el producto actual, es considerado como una ventaja ya que, al ser una cantidad limitada de drones, la respectiva revisión y comparación de posición y ruta de exploración no representaron una tarea compleja ni demandante de realizar.
- A pesar de que los algoritmos de FOA y FA, sean posibles candidatos a ser seleccionados, el hecho de que el valor de la drosófila, fluoresceína y radio de percepción deban ser calculados y configurados para que la solución brinde los resultados esperados, esto puede llegar a involucrar más complejidad de la estipulada debido a que dichas cantidades deben ser estudiadas y probadas para que el algoritmo se comporte como espera.

- La característica de que el PSO sea flexible y práctico para satisfacer cualquier tipo de necesidad, hace que dicho algoritmo sea ideal ya que el mismo puede ser adaptado a la solución en cuestión, dada su gran practicabilidad.

3. Actividad 3 (A3): Analizar con detenimiento el algoritmo seleccionado para comprender su funcionamiento

En el algoritmo de PSO, es necesario crear una población de partículas, que se desenvolverá en un determinado espacio de búsqueda, en el cual, se medirá la aptitud individual de cada una de dichas entidades con base en una función de aptitud del problema. Las partículas se mueven de su posición actual a la siguiente en función de su mejor ubicación personal y de la mejor ubicación del enjambre hasta el momento. Al iterar los movimientos, velocidades y aceleraciones de las entidades, el enjambre tiende a alcanzar gradualmente un punto óptimo de la función de aptitud durante a través de ciclos iterativos [4].

El ajuste de velocidad se basa en 3 factores que son [4]:

- Velocidad anterior (componente de inercia).
- Mejor posición de la partícula individual (componente cognitivo).
- Mejores posiciones del enjambre (componente social).

Adicionalmente, existen ciertos coeficientes o pesas que afectan cada una de las futuras direcciones que pueden tomar las partículas, estos son [4]:

- Coeficiente W (peso de la inercia) es una fuerza que mantiene la partícula en movimiento en la misma dirección que la generación anterior.
- C1 y C2 son valores de aceleración constante donde $C1 = C2$, los cuales proviene del algoritmo original.

- R1 y R2 denotan hiperparámetros y causan algunas perturbaciones aleatorias. El valor más alto de dichos cantidades, da como resultado un movimiento más sensible de las partículas.

Dado lo anterior, el pseudocódigo del algoritmo de optimización de enjambre de partículas, se muestra a continuación [4]:

1. Inicializar la matriz de población de partículas.
2. Bucle iterativo.
3. Para cada partícula, calcular la aptitud usando la función de aptitud.
4. Comparar el valor de aptitud actual con la mejor posición individual. Reemplazar dicho valor, en caso de que sea mejor que lo mejor.
5. Verificar la mejor partícula del enjambre con base en su respectiva posición individual y asignarla como mejor entidad de todo el enjambre.
6. Calcular la nueva velocidad y actualizar la posición de cada partículas.
7. Si se cumple un criterio de salida, salir del bucle iterativo.
8. Finalizar el bucle iterativo.

4. Actividad 4 (A4): Crear repositorio en GitHub con la adaptación del algoritmo seleccionado a la solución del problema

Link de repositorio: <https://github.com/NachoGranados/ComputerEngineeringDesignProject.SwarmIntelligence>

Referencias

- [1] Yu, T., Wang, L., Han, X., Liu, Y., Zhang, L. (2015). Swarm intelligence optimization algorithms and their application. WHICEB 2015 Proc, 3. <https://core.ac.uk/download/pdf/301366614.pdf>
- [2] Qawqzeh, Y. (2021). Una revisión de la implementación de algoritmos de inteligencia de enjambre para la programación y optimización en entornos de computación en la nube. PeerJ. Informática, 7, E696. <https://doi.org/10.7717/peerj-cs.696>
- [3] Sun, W., Tang, M., Zhang, L., Huo, Z., Shu, L. (2020). A survey of using swarm intelligence algorithms in IoT. Sensors, 20(5), 1420. <https://www.mdpi.com/1424-8220/20/5/1420>
- [4] Moriwaki, K. (2022, 15 junio). Swarm Intelligence: Coding and Visualising Particle Swarm Optimisation in Python. Medium. <https://towardsdatascience.com/swarm-intelligence-coding-and-visualising-particle-swarm-optimisation-in-python-253e1bd00772>