



DESCRIPCIÓN DE COMPORTAMIENTOS ANIMALES QUE SE UTILIZAN PARA ALGORITMOS DE OPTIMIZACIÓN E INTELIGENCIA ARTIFICIAL

*DESCRIPTION OF ANIMAL BEHAVIORS USED FOR OPTIMIZATION AND
ARTIFICIAL INTELLIGENCE ALGORITHMS*



TRABAJO DE FIN DE GRADO
MIGUEL ÁNGEL ZARZOSA GÓMEZ

Tutorizado por Hipólito Hernández Pérez. Grado en Biología. Septiembre 2017

ÍNDICE

<u>1. Resumen</u>	1
<u>2. Abstract</u>	1
<u>3. Introducción</u>	2
<u>3.1. Definiciones Matemáticas</u>	3
<u>3.2. Clasificación de Algoritmos</u>	4
<u>3.2.1. Basados en la Inteligencia de Enjambres</u>	4
<u>3.2.2. Inspirados en otros Factores Biológicos</u>	5
<u>3.2.1. Basados en la Física y la Química</u>	6
<u>4. Algoritmos Bio-Inspirados</u>	6
<u>4.1. Colonias de Hormigas (<i>Ant Colony Optimization</i>)</u>	6
<u>4.2. Enjambres de Abejas (<i>Bee Colony Optimization</i>)</u>	12
<u>4.3. Destellos de Luciérnagas (<i>Firefly Algorithm</i>)</u>	15
<u>4.4. Búsqueda del Cuco (<i>Cuckoo Search</i>)</u>	17
<u>4.5. Manadas de Lobos (<i>Wolf Search</i>)</u>	18
<u>4.6. Ecolocalización de Murciélagos (<i>Bat Echolocation Algorithm</i>)</u>	20
<u>4.7. Ecolocalización de Delfines (<i>Dolphin Echolocation Algorithm</i>)</u>	22
<u>5. Conclusiones</u>	24
<u>6. Conclusions</u>	25
<u>7. Bibliografía</u>	26

1. RESUMEN

Muchas especies en la naturaleza se caracterizan por el comportamiento de enjambre. Los bancos de peces, bandadas de aves y manadas de animales terrestres se forman como resultado de necesidades biológicas para permanecer juntos. Los individuos en un rebaño, un banco de peces, o una bandada de aves tienen una mayor probabilidad de mantenerse con vida, ya que los depredadores suelen atacar individuos aislados. El movimiento colectivo caracteriza a estos grupos de animales. Los individuos que los forman, responden rápidamente a los cambios en la dirección y velocidad de sus vecinos. El comportamiento de enjambre es también una de las principales características de los insectos sociales (abejas, hormigas, etc). La comunicación entre individuos en una colonia de insectos sociales ha sido ampliamente estudiada. Los sistemas de comunicación entre insectos individuales contribuyen a la configuración de la "inteligencia colectiva" de las colonias de insectos sociales. La inteligencia de enjambre es la parte de la Inteligencia Artificial basada en el estudio de las acciones de los individuos en varios sistemas descentralizados. Estos están compuestos de individuos físicos (robots, por ejemplo) o "virtuales" que se comunican entre sí, cooperan, colaboran, intercambian información y conocimiento y realizan algunas tareas en su entorno.

2. ABSTRACT

Many species in the nature are characterized by swarm behavior. Fish schools, flocks of birds, and herds of land animals are formed as a result of biological needs to stay together. Individuals in herd, fish school, or flock of birds has a higher probability to stay alive, since predator usually assault only one individual. A collective movement characterizes them. Herds of animals respond quickly to changes in the direction and speed of their neighbors. Swarm behavior is also one of the main characteristics of social insects (bees, wasps, ants, termites). Communication between individual insects in a colony of social insects has been well known. The communication systems between individual insects contribute to the configuration of the "collective intelligence" of the social insect colonies. Swarm Intelligence is the part of Artificial Intelligence based on study of actions of individuals in various decentralized systems. These decentralized

systems are composed of physical individuals (robots, for example) or “virtual” ones that communicate among themselves, cooperate, collaborate, exchange information and knowledge and perform some tasks in their environment.

3. INTRODUCCIÓN

La inteligencia en enjambre y los algoritmos inspirados en la biología forman un tema candente en el desarrollo de nuevos algoritmos también inspirados en la naturaleza. Estos algoritmos pueden basarse en la inteligencia de enjambres, sistemas biológicos, y sistemas físico-químicos, dependiendo de las fuentes en que se basen. No todos ellos son útiles, aunque algunos algoritmos han demostrado ser muy eficientes y por lo tanto se han convertido en herramientas populares para resolver problemas del mundo real. Algunos algoritmos están muy poco estudiados [1,2,3,5,11], pero otros como el de colonias de hormigas han sido estudiados por numerosos autores.

Los problemas del mundo real son, a menudo, muy difíciles de resolver, y muchas aplicaciones tienen que lidiar con problemas complicados. Para resolver estos problemas, se deben utilizar herramientas de optimización, aunque no hay garantía de que se pueda obtener la solución óptima. De hecho, para problemas muy difíciles, no hay algoritmos en absoluto eficientes. Como resultado, muchos problemas tienen que ser resueltos por ensayo y error, usando varias técnicas de optimización. Además, se han desarrollado nuevos algoritmos para ver si pueden hacerles frente.

En la literatura actual, hay unos 40 algoritmos diferentes basados en la naturaleza. Es realmente difícil clasificar estos algoritmos sistemáticamente. Obviamente, las clasificaciones pueden depender en gran medida de los criterios, y no existe una guía fácil para establecerlos. La naturaleza ha inspirado a muchos investigadores de muchas maneras y por lo tanto es una fuente rica de inspiración. Hoy en día, muchos de los algoritmos que se van desarrollando están inspirados en ella.

Incluso conociendo la fuente de inspiración, todavía existen diferentes niveles de clasificaciones, dependiendo de cómo se detallen y cuántas fuentes se utilicen. Para simplificar, suelen utilizarse las fuentes de más alto nivel como la biología, la física o la química.

Cabe destacar que, la mayoría de los algoritmos inspirados en la naturaleza, se basan en algunas características exitosas del sistema biológico. Por lo tanto, la mayor fracción de algoritmos inspirados en la naturaleza, están inspirados en la biología.

Entre los algoritmos inspirados en la biología, hay una clase especial de algoritmos que se han desarrollado inspirándose en la inteligencia de enjambre. De hecho, están entre los más populares. Algunos buenos ejemplos son la optimización de colonias de hormigas, la búsqueda del cuco, el algoritmo de murciélagos y el algoritmo de luciérnagas. Obviamente, no todos los algoritmos están basados en sistemas biológicos. Muchos algoritmos se han desarrollado usando la inspiración de sistemas físicos y químicos. O algunos incluso pueden basarse en la música.

3.1. DEFINICIONES MATEMÁTICAS

Antes de entrar en detalle sobre todo este tema que se acaba de comentar, es preciso definir algunos conceptos que se usaran mucho a lo largo de este escrito.

Algoritmo: Conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problemas.

Heurística: Es un algoritmo que busca buenas soluciones del problema a optimizar, sin poder garantizar que llegue a la óptima.

Metaheurística: Es una heurística compleja. Generalmente, combina varios procesamientos usando parámetros dados por el usuario.

Optimización: En matemáticas e informática, método para determinar los valores de las variables que intervienen en un proceso o sistema para que el resultado sea el mejor posible.

Optimización Combinatoria: Rama de la optimización en matemáticas aplicadas y en ciencias de la computación, relacionada con la teoría de algoritmos y de la complejidad computacional. También está relacionada con la inteligencia artificial e ingeniería de software. Los algoritmos de optimización combinatoria resuelven problemas que se creen difíciles, explorando el espacio de soluciones para estas instancias.

3.2. CLASIFICACIÓN DE ALGORITMOS

Los algoritmos pueden dividirse en tres grandes grupos: basados en la inteligencia de enjambres (colonias, manadas, grupos de animales), inspirados en otros factores de la biología, y basados en sistemas físico-químicos [22, 25, 30].

Vale la pena señalar que las clasificaciones no son únicas, ya que algunos algoritmos pueden clasificarse en diferentes categorías al mismo tiempo. Hablando francamente, las clasificaciones dependen en gran medida del enfoque o el énfasis y de su perspectiva. Esto demuestra claramente que las clasificaciones dependen de la perspectiva real y las motivaciones.

3.2.1. Basados en la inteligencia de enjambres. La inteligencia de enjambre se refiere al comportamiento colectivo emergente de múltiples agentes que interactúan y que siguen algunas reglas simples. Mientras que cada agente puede ser considerado como inteligente, todo el sistema de múltiples agentes puede mostrar algún comportamiento de organización propia y por lo tanto puede comportarse como una especie de inteligencia colectiva. Muchos algoritmos han sido desarrollados inspirándose en los sistemas de inteligencia de enjambres en la naturaleza. El término "inteligencia de enjambre", que denota esta "inteligencia colectiva" entró en uso entre 1988 y 1999 [1-6, 8-11]. Están inspirados en el comportamiento colectivo de insectos sociales, como hormigas, termitas, abejas y avispas, así como de otras sociedades animales como bandadas de aves o bancos de peces. La optimización clásica utiliza el comportamiento de enjambre de peces y pájaros, mientras que los algoritmos más novedosos utilizan el comportamiento intermitente de luciérnagas enjambradas.

La búsqueda del cuco se basa en el parasitismo de algunas especies de cuco, mientras que el algoritmo de murciélagos utiliza la eco localización de los murciélagos para alimentarse. La optimización de las colonias de hormigas utiliza la interacción de insectos sociales, mientras que las clases de algoritmos de abejas se basan en el comportamiento de forrajeo de la abeja. Están entre los más populares y ampliamente utilizados, y es que estos algoritmos comparten información entre múltiples agentes, de manera que la auto organización, la evolución conjunta y el aprendizaje durante las iteraciones pueden ayudar a proporcionar la alta eficiencia de la mayoría. Como aditivo,

el agente múltiple puede ser paralelizado fácilmente para que la optimización a gran escala se vuelva más práctica desde el punto de vista de la implementación.

Algoritmos Inspirados en la Inteligencia de Enjambres	
Ant colony optimization	<i>Artificial bee colony</i>
<i>Bacterial foraging</i>	<i>Bacterial-GA foraging</i>
Bat algorithm	Bee colony optimization
<i>Bee hide system</i>	<i>Bees algorithms</i>
Wolf search	<i>Termite colony optimization</i>
<i>Flower pollination algorithm</i>	<i>Bumblebees</i>
<i>Cat swarm</i>	Cuckoo search
<i>Eagle Strategy</i>	<i>East bacterial swarming algorithm</i>
Firefly algorithm	<i>Fish swarm/school</i>
Dolphin echolocation	<i>Glow worm swarm optimization</i>
<i>Hierarchical swarm model</i>	<i>Krill Herd</i>
<i>Monkey search</i>	<i>Weightless swarm algorithm</i>
<i>Virtual ant Algorithm</i>	<i>Virtual bees</i>

Tabla 1: algoritmos inspirados en la Inteligencia de Enjambres [35].

3.2.2. Inspirados en otros factores biológicos. Obviamente, estos algoritmos pertenecen a una clase más amplia de algoritmos, llamados bio-algoritmos. No todos los algoritmos inspirados por la naturaleza, son bio-inspirados, y algunos son puramente físicos o químicos. Muchos bio-algoritmos no utilizan directamente el comportamiento de los enjambres. Por lo tanto, es mejor llamarlos bio-inspirados (inspirados en la biología, pero no en la inteligencia de enjambres). Sin embargo, no es fácil clasificar ciertos algoritmos como la evolución diferencial. Estrictamente hablando, no es bio inspirado porque no hay ningún vínculo directo con ningún comportamiento biológico, aunque como tiene cierta similitud con los algoritmos genéticos y también tiene la palabra clave "evolución", se puede nombrar como bio-inspirado.

Algoritmos Bio-Inspirados	
<i>Atmosphere clouds model</i>	<i>Biogeography-based optimization</i>
<i>Brain storm optimization</i>	<i>Differential evolution</i>
<i>Japanese tree frogs calling</i>	<i>Eco inspired evolutionary algorithm</i>
<i>Egyptian vulture</i>	<i>Gene expression</i>
<i>Great salmon run</i>	<i>Group search optimizer</i>
<i>Human-inspired algorithm</i>	<i>Invasive weed optimization</i>
<i>Paddy field algorithm</i>	<i>Roach infestation algorithm</i>
<i>Marriage in honey bees</i>	<i>Queen-bee optimization</i>

Tabla 2: algoritmos bio-inspirados [35].

3.2.3. Basados en la física y la química. Otros algoritmos que no son bio-inspirados, han sido desarrollados imitando ciertas leyes físicas y/o químicas, incluyendo cargas eléctricas, gravedad, sistemas fluviales, etc. Aunque la física y la química son dos temas diferentes, no es útil subdividirlos. Después de todo, muchas leyes fundamentales son las mismas.

Algoritmos Inspirados en Física/Química	
<i>Big bang-big crunch</i>	<i>Black hole</i>
<i>Central force optimization</i>	<i>Charged system search</i>
<i>Electro-magnetism optimization</i>	<i>Galaxy-based search algorithm</i>
<i>Gravitational search</i>	<i>Harmony search</i>
<i>Intelligent water drop</i>	<i>River formation dynamics</i>
<i>Self-propelled particles</i>	<i>Simulated Annealing</i>
<i>Spiral Optimization</i>	<i>Water cycle algorithm</i>
<i>Particle swarm algorithm</i>	<i>Self-propelled particles</i>

Tabla 3: algoritmos inspirados en la física y la química [35].

4. ALGORITMOS BIO-INSPIRADOS

A partir de aquí, se resumen algunos de los principales algoritmos bio-inspirados basados en la inteligencia de enjambres, por ser de los más estudiados hasta la actualidad, o por encontrarse como modelos novedosos, surgidos recientemente. Además, nos centramos en ellos porque son los que se definen mejor con el título, significado y finalidad de este trabajo.

4.1. Colonias de Hormigas (*Ant Colony Optimization*)

Las hormigas son insectos sociales que viven en colonias y que, debido a su colaboración mutua, son capaces de mostrar comportamientos complejos y realizar tareas difíciles desde el punto de vista de una hormiga individual.

Un aspecto interesante del comportamiento de muchas especies de hormigas es su habilidad para encontrar los caminos más cortos entre su hormiguero y las fuentes de alimento. Este hecho es especialmente interesante si se tiene en cuenta que muchas de las especies de hormigas son casi ciegas, lo que evita el uso de pistas visuales.

Mientras se desplazan entre el hormiguero y la fuente de alimento, algunas especies de hormigas depositan feromonas (una sustancia química que puede “olerse”). Si no se

encuentra ningún rastro de feromona, las hormigas se mueven de manera básicamente aleatoria, pero cuando existe feromona depositada, tienen mayor tendencia a seguir el rastro. De hecho, algunos experimentos han demostrado que las hormigas prefieren de manera probabilística los caminos marcados con concentraciones mayores de feromona.

En la práctica, las hormigas eligen el camino a seguir con una decisión probabilística sesgada por la cantidad de feromona: cuanto más fuerte es el rastro de feromona, mayor es la probabilidad de elegirlo. Esto se produce cuando se encuentran con distintos cruces en el camino.

Puesto que las hormigas depositan feromonas en el camino que siguen, este comportamiento lleva a un proceso de auto refuerzo que concluye con la formación de rastros señalados por una concentración de feromona elevada. Este comportamiento permite además a las hormigas encontrar los caminos más cortos entre su hormiguero y la fuente del alimento. Inicialmente no existe ningún rastro de feromona en el medio y, cuando una hormiga llega a una intersección, elige de manera aleatoria una de las bifurcaciones posibles. Según transcurre el tiempo y mientras que las hormigas están recorriendo los caminos más prometedores, estos van recibiendo una cantidad superior de feromona. Esto ocurre gracias a que, cuando hay caminos más cortos, las hormigas que los siguen consiguen encontrar la comida más rápidamente, por lo que comienzan su viaje de retorno antes. Entonces, en el camino más corto habrá un rastro de feromona ligeramente superior y, por lo tanto, las decisiones de las siguientes hormigas estarán dirigidas en mayor medida a dicho camino. Además, este camino recibirá una proporción mayor de feromona por las hormigas que vuelven por él que por las que vuelven por el camino más largo.

Este proceso finaliza haciendo que la probabilidad de que una hormiga escoja el camino más corto aumente progresivamente y que, al final, el recorrido de la colonia converja al más corto de todos los caminos posibles. Esta convergencia se complementa con la acción del entorno natural que provoca que la feromona se evapore transcurrido un cierto tiempo. Así, los caminos menos prometedores pierden progresivamente feromona porque son visitados cada vez por menos hormigas.

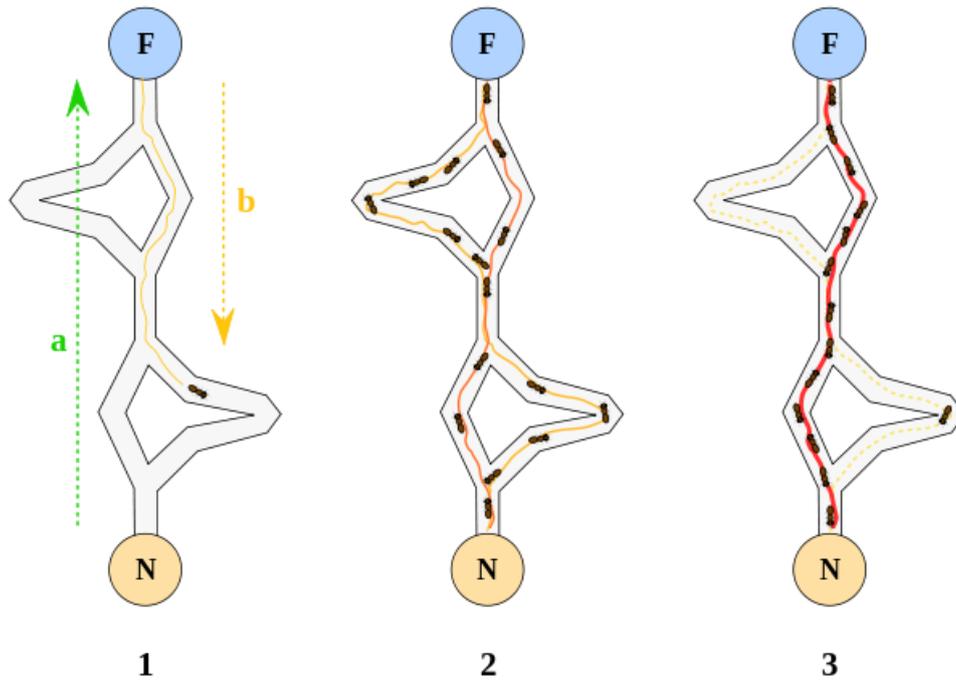


Figura 1: Ejemplo del recorrido de las hormigas en base a las feromonas colocadas en el camino.

Sin embargo, algunos estudios biológicos han demostrado que los rastros de feromona son muy persistentes -la feromona puede permanecer desde unas pocas horas hasta varios meses dependiendo de varios aspectos distintos, como la especie de las hormigas, el tipo de suelo, etc, lo que provoca una menor influencia del efecto de la evaporación en el proceso de búsqueda del camino más corto.

Numerosos experimentos muestran que, debido a la gran persistencia de feromona, es difícil que las hormigas “olviden” un camino que tiene un alto nivel de feromona, aunque hayan encontrado un camino aún más corto. Hay que tener en cuenta que si se traslada este comportamiento directamente al ordenador para diseñar un algoritmo de búsqueda puede quedarse rápidamente estancado en un óptimo local. Los algoritmos ACO (*Ant Colony Optimization*) son modelos inspirados en el comportamiento de colonias de hormigas.

El primer algoritmo basado en la optimización mediante colonias de hormigas fue aplicado al TSP (*Travelling Salesman Problem*) [6], obteniéndose unos resultados bastante alentadores, ya que este problema de camino único es una instancia bien conocida y se incluye de manera inmediata, haciendo por tanto que su adaptación al

comportamiento real de las hormigas para resolverlo sea una tarea casi instantánea. A partir de dicho algoritmo, se han desarrollado diversos heurísticos que incluyen varias mejoras, y han sido aplicados no solo al TSP, sino también a problemas como el VRP (*Vehicle Routing Problem*) y el QAP (*Quadratic Assignment Problem*) entre otros [11].

Los algoritmos ACO son procesos iterativos. En cada iteración se "lanza" una colonia de hormigas y cada una de las hormigas de la colonia construye una solución al problema. Las hormigas construyen las soluciones de manera probabilística, guiándose por un rastro de feromona artificial y por una información calculada a priori de manera heurística.

Cuando todas las hormigas han construido una solución debe actualizarse la feromona en cada arco. Se obliga a que el nivel de feromona permanezca en un rango. Estos límites se imponen con el objetivo de evitar el estancamiento en la búsqueda de soluciones. Tras la actualización de la feromona puede comenzarse una nueva iteración. El resultado final es la mejor solución encontrada a lo largo de todas las iteraciones realizadas.

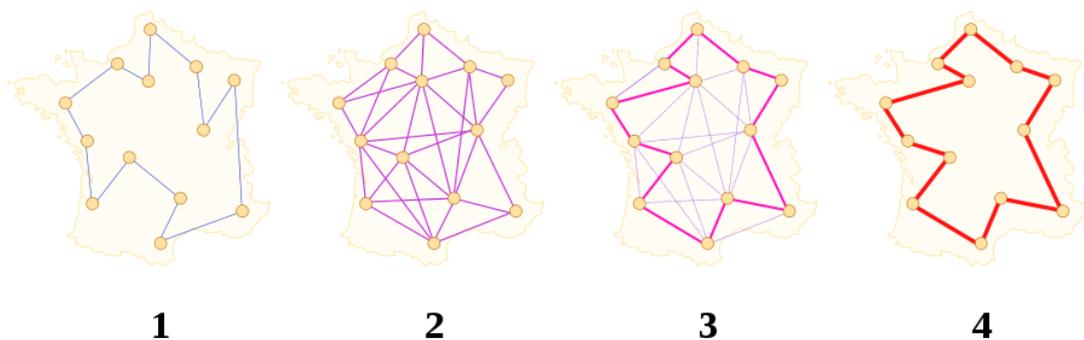


Figura 2: Ejemplo de cómo desaparecen las feromonas y solo se mantienen las más visitadas.

Este tipo de algoritmos se han aplicado a un gran número de problemas de optimización combinatoria diferentes. Sus aplicaciones actuales se distribuyen dentro de dos clases fundamentales. La primera clase de problemas está compuesta por los problemas de optimización combinatoria, para los que las técnicas clásicas ofrecen a menudo un comportamiento pobre. Una característica común a casi todas las aplicaciones exitosas de ACO es la combinación de las hormigas con algoritmos de búsqueda local que refinan las soluciones ofrecidas por las hormigas. La segunda clase de aplicaciones se compone

de problemas dinámicos de caminos mínimos, donde la instancia del problema que hay que solucionar cambia durante la ejecución del algoritmo. Esta última clase incluye aplicaciones de la OCH al enrutamiento en redes de comunicaciones.

Desde la primera aplicación del SH (sistema de hormigas) [8], se convirtió en un problema estándar realizar pruebas en otros modelos posteriores que ofrecían un mejor rendimiento que él. Cronológicamente, las tres aplicaciones siguientes fueron QAP, JSP (*Jobshop Scheduling Problem*) y el enrutamiento en redes [9,10]. Otras aplicaciones incluyen problemas clásicos de enrutamiento de vehículos, de ordenación secuencial, de secuenciación, etc.

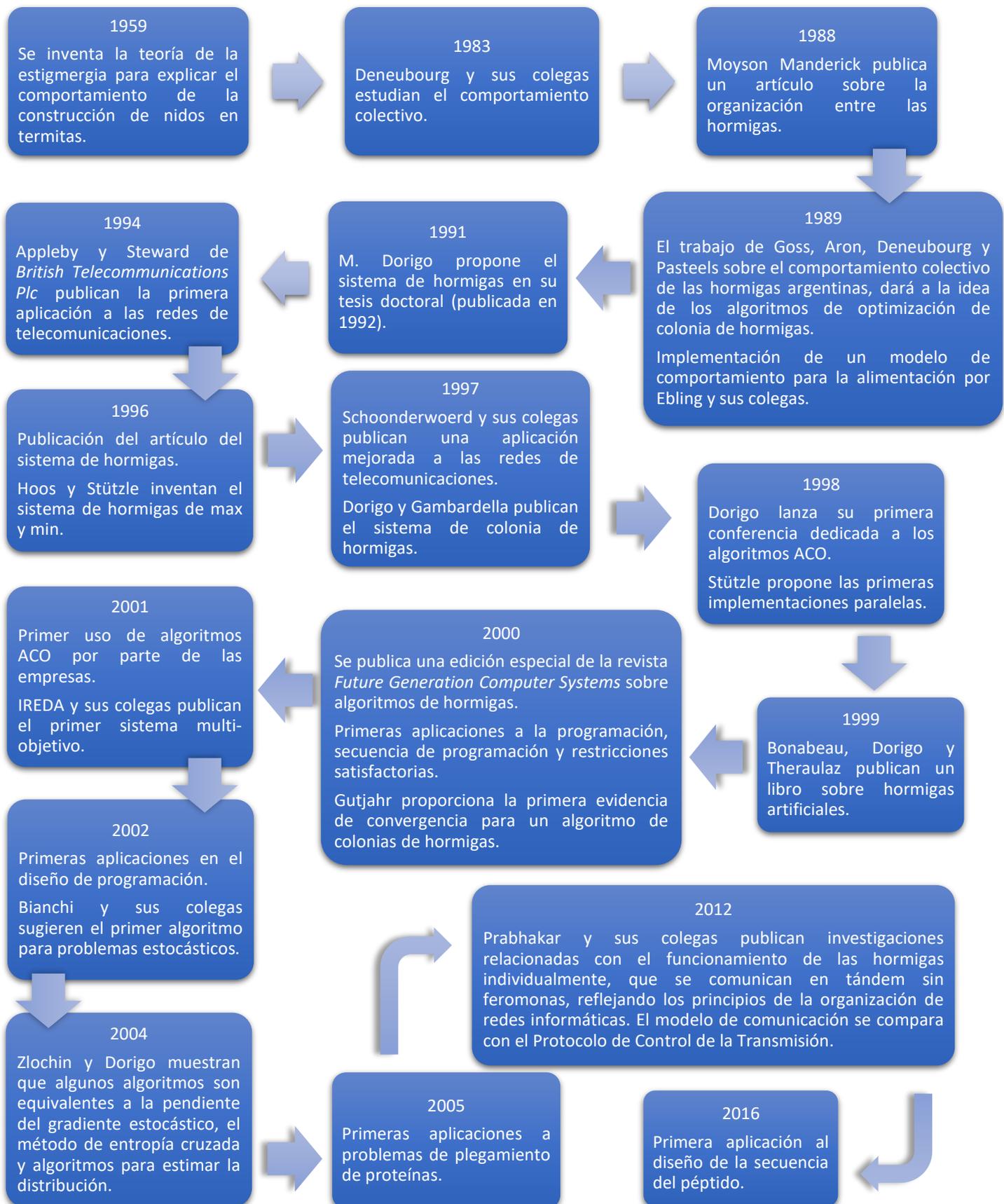
Muchos autores distintos han usado la metaheurística ACO para solucionar un gran número de problemas de optimización combinatoria como la supersecuencia común más corta, la asignación generalizada, la cobertura de conjuntos y varios problemas de satisfacción de restricciones, entre otros.

Aparte de las aplicaciones anteriores, ACO ha sido usada recientemente para aprendizaje automático ("*machine learning*"), concretamente para el diseño de algoritmos de aprendizaje para estructuras de representación del conocimiento como las clásicas reglas lógicas, reglas difusas y redes bayesianas, demostrando resultados bastante prometedores.

Los resultados computacionales para otros muchos problemas son muy buenos y cercanos a los mejores posibles, lo cual es remarcable, ya que muchos de esos problemas han atraído una gran cantidad de esfuerzo e investigación. Por otro lado, esta metaheurística está siendo aplicada a nuevos problemas reales con resultados prometedores (por ejemplo, la aplicación al diseño de circuitos lógicos combinatorios).

A continuación, se representa la cronología de los avances más importantes que han ido aconteciendo en la historia de los algoritmos ACO desde su aparición hasta la actualidad.

Cronograma sobre los Algoritmos de Optimización ACO



4.2. *Enjambres de Abejas (Bee Colony Optimization)*

Las abejas son uno de los insectos sociales más estudiados. Poseen muchas características que pueden ser usadas como modelos de comportamiento inteligente, entre las cuales están la división de labores, la comunicación entre un individuo y en grupos, el comportamiento cooperativo y su estrategia reproductiva. Estos comportamientos realizan una secuencia de acciones producto de su potencialidad genética, ambientes ecológicos y fisiológicos, las condiciones sociales de la colonia, y otras prioridades.

Una aproximación algorítmica al comportamiento de las abejas usando su estrategia reproductiva, se presenta cuando las reinas de la colonia se aparean con varios zánganos, obteniendo una biblioteca genética, y donde se extraen nuevos individuos que representan las reinas de la siguiente iteración. Dicho algoritmo tiene mecanismos muy similares a los algoritmos genéticos, por lo cual no puede ser considerado de Inteligencia de Enjambres.

Un comportamiento que se puede modelar de forma algorítmica es el sistema de búsqueda y recolección de Néctar, también conocido como forrajeo. El algoritmo que aparece [13], se puede utilizar para optimización, haciendo uso de la información local y un balance entre la exploración y la explotación. Tal vez, lo más particular de este comportamiento es la forma como las abejas se comunican a través de danzas para informar de posibles fuentes de alimento a toda la colonia. Cada abeja sigue el mismo conjunto de reglas comportamentales, las cuales consisten en un conjunto de condiciones que definen una acción a ser desarrollada. Las condiciones incluyen información externa e interna, como la danza de otras abejas y la localización memorizada de una fuente de alimento, respectivamente [4].

La colonia de abejas tiene que adaptarse continuamente a cada nueva situación, procurando tener la cantidad de néctar necesario para el mantenimiento de la colmena, por medio de la división apropiada de la fuerza de trabajo entre la exploración del campo de nuevas fuentes de alimento y la explotación de las fuentes conocidas. El comportamiento de forrajeo de cada abeja como individuo es regulado, y la información externa e interna juegan roles importantes. Las reglas incluyen especificaciones para el

viaje de partida de la abeja, del nido a la fuente, la búsqueda de la fuente por una abeja, la recolección de néctar de la fuente, el viaje de regreso al nido, el medio por el cual la información acerca de la fuente se transmite a otras abejas en el nido, etc. [17].

La información representada por las danzas indica la localización de las fuentes en la distancia (por medio de la duración), la posición (por medio de la dirección) y la productividad de la fuente (por medio de la insistencia y vigorosidad del baile). Cuanto más vigorosos y alargados sean los bailes más seguidores podrían ser atraídos. Es seguro que la distancia y la dirección de las fuentes de alimento codificado en las danzas no son el único tipo de información que puede guiar el encuentro de las abejas a encontrar fuentes provechosas de alimento. El olor influye en la colmena y en el campo, y los estímulos visuales influyen en el comportamiento de búsqueda y hallazgo de las abejas. Además, las abejas tienen una especie de memoria, la cual guarda información que percibe del ambiente para usarla posteriormente.

La cantidad de agentes iniciales para la búsqueda aleatoria de fuentes, o mejor, la cantidad de individuos que se convierten automáticamente en forrajeros exploradores, es un parámetro del algoritmo. La ubicación inicial de estos forrajeros se hace de manera aleatoria sobre el espacio de búsqueda. La oportunidad de encontrar una fuente rica de alimento es limitada por la “motivación a casa”, la cual se aumenta con cada paso que estos den sobre un espacio desconocido [32]. Los forrajeros exploradores cambian la dirección y el tamaño del desplazamiento si es necesario, una vez hayan alcanzado un tope de tiempo de búsqueda.

Particularmente, el algoritmo intenta darle características de exploración/explotación a los forrajeros que hacen la búsqueda de las fuentes de néctar. Los agentes que se encuentran explorando suelen seguir una sola dirección y un desplazamiento de acuerdo al paso aleatorio adquirido. El giro y desplazamiento de los agentes durante la fase explosiva de las fuentes varía en cada agente de forma aleatoria si el paso realizado por el agente lo lleva a un punto de menor concentración que el anterior, y de paso le brinda la posibilidad de regresar a su posición anterior para iniciar la búsqueda del gradiente desde el punto ha donde había llegado.

Es importante resaltar que es poco probable que un explorador pueda llegar directamente al punto de mayor concentración de nutrientes y que podría ser efecto de agentes recolectores, debido al error en la transmisión de la información a la que es sometido el algoritmo en las danzas. Una vez alguna fuente es encontrada y se ha cumplido su tiempo de búsqueda, los forrajeros regresan a la colmena para informar al resto de la población la ubicación relativa y lo provechoso de la fuente hallada.

Otra característica del forrajeo que permite la rápida convergencia, es la capacidad de detener la danza de una de sus compañeras al informar un punto de mejores características, evitando que algunos agentes presentes en la colmena sigan la danza con información que ahora es poco adecuada. Una vez el forrajero logra esto, regresa al punto donde se encuentra su nueva fuente y explora un poco más tratando de encontrar un gradiente positivo de alimento, como el objetivo es que regrese rápido con la información de la fuente hallada y que además intente llegar a una mejor posición. Una vez se ha completado, regresa a la colmena para informar la ubicación de la fuente. Adicionalmente, cuando el agente intenta detener al forrajero que le informó la posición de la fuente anterior, podría interrumpir a una abeja que informa de una fuente muy cercana ocasionando una pérdida de información [16]. Por lo tanto, el algoritmo de abejas puede cometer errores.

Los resultados obtenidos en experimentos sobre el algoritmo BNSO (*Bee Natural System Optimization*), presentan características que le permiten obtener un buen acercamiento a su contraparte real, además de la convergencia que estos agentes presentan en cuanto al propósito de explotar fuentes con mayor provecho. Este algoritmo es robusto, ya que al simular el comportamiento biológico siempre se acondiciona a los ambientes virtuales a los que es sometido, logrando el propósito básico del bienestar de la colmena. Además, puede aplicarse con cambios mínimos, a otro problema de optimización combinatoria y, por ende, mostrando características de versatilidad. Los mejores resultados se observan cuando hay mayor concurrencia sobre la fuente más provechosa y, por lo tanto, mayor probabilidad que los movimientos realizados por los agentes lo lleven a ubicarse sobre el punto máximo de la superficie [18]. Caso contrario ocurre con las otras fuentes, ya que si no hay buena concurrencia de los agentes pocas posibilidades existen para que estos puedan ubicarse en mejores posiciones.

Actualmente, el algoritmo es estudiado para darle mayores posibilidades en términos de exploración y explotación, ya que el camino seguido por los agentes es errático e impide que un agente que se dirija a una fuente informada pueda explorar un poco su camino de ida a la fuente y, por lo tanto, impidiendo mayor posibilidad de encontrar una fuente más provechosa [20].

4.3. *Destellos de Luciérnagas (Firefly Algorithm)*

La luz que destellan las luciérnagas es una visión asombrosa en los cielos de verano de regiones tropicales y templadas. Hay cerca de dos mil especies de luciérnagas, y la mayoría de ellas producen destellos cortos y rítmicos. El patrón de destellos es, a menudo, único para una especie particular.

La luz intermitente es producida por un proceso de bioluminiscencia, y las verdaderas funciones de tales sistemas de señalización todavía se encuentran como objeto de debate. Sin embargo, dos funciones fundamentales de estos destellos sí que se conocen de manera clara: atraer a una pareja o atraer a presas potenciales. Además, el parpadeo también puede servir como un mecanismo de advertencia de protección. El flash rítmico, la velocidad de parpadeo y la cantidad de tiempo forman parte del sistema de señales que reúnen ambos sexos.

Las hembras responden al patrón único del macho de la misma especie, mientras que, en algunas especies, las luciérnagas femeninas pueden imitar el patrón de destello que aparece a otras especies, para atraer y alimentarse de las luciérnagas masculinas que pueden confundir los flashes como los de un compañero potencial de su misma especie.

Sabemos que la intensidad de la luz a una distancia particular decrece a medida que la distancia aumenta. Además, el aire absorbe la luz que se hace más débil y más débil a medida que aumenta la distancia. Estos dos factores combinados hacen que la mayoría de las luciérnagas sean visibles sólo a una distancia limitada, por lo general varios cientos de metros por la noche, lo que suele ser suficiente para que las luciérnagas se comuniquen entre sí. La luz intermitente puede formularse de tal manera que se asocie con la función objetivo a optimizar, lo que permite formular nuevos algoritmos de optimización.

Actualmente se pueden idealizar algunas de las características destellantes de las luciérnagas para desarrollar algoritmos inspirados en ellas. Para ello, se usan las siguientes tres reglas idealizadas: 1) todas las luciérnagas son unisex para que una luciérnaga sea atraída a otras luciérnagas sin importar su sexo; 2) La atracción es proporcional a su brillo, así que para cualesquiera dos luciérnagas que destellan, la menos brillante se moverá hacia la más brillante; 3) El brillo de una luciérnaga es afectado o determinado por el paisaje objetivo [23]. Algo a tener en cuenta, es que los individuos tienen visibilidad ajustable y más versátil en las variaciones de atracción, lo que generalmente conduce a una mayor movilidad y por lo tanto el espacio de búsqueda se explora de manera más eficiente.

El algoritmo básico de la luciérnaga es muy eficiente. A pesar de que se han hecho muchos progresos en algoritmos basados en él desde 2008, todavía se necesitan esfuerzos significativos para mejorar su rendimiento. Algunas de sus utilidades son: análisis teórico para convergencia de trayectorias; deducción de las condiciones suficientes y necesarias para la selección de coeficientes de control; mecanismos o estrategias eficaces para la selección de parámetros de control; reglas no-homogéneas de actualización para realzar la capacidad de búsqueda [15]. Las variantes clásicas del algoritmo tienen encuadres de parámetros inesperados y leyes de actualización limitada, notablemente la regla homogénea necesita ser mejorada para hacer más búsquedas en escenarios de formas físicas diferentes. Se ha efectuado el análisis de la trayectoria de una sola luciérnaga en el algoritmo tradicional y en una variante adaptativa, respectivamente.

Estos análisis llevan a un modelo general de los algoritmos que incluyen un conjunto de las condiciones de frontera para los parámetros que garantizan las tendencias de convergencia de los dos algoritmos, aunque es posible mejorar la calidad de las soluciones que presenta reduciendo gradualmente la aleatoriedad, lo que podría constituir temas importantes para futuras investigaciones [33]. Además, como una extensión relativamente directa, este algoritmo puede ser modificado para resolver problemas de optimización multiobjetivo. Además, la aplicación de algoritmos de luciérnagas en combinación con otros algoritmos, también puede constituir un área interesante para nuevos estudios [26].

4.4. *Búsqueda del Cuco (Cuckoo Search)*

Existen especies de cucos que presentan parasitismo obligado a la hora de poner sus huevos, recurriendo a dejarlos en los nidos de aves de otras especies, consideradas hospedadoras. Algunas de estas aves huésped, pueden entablar un conflicto directo con los cucos intrusos. Por ejemplo, si un pájaro huésped descubre que los huevos no son los suyos, o bien lanzará estos huevos extraños, o simplemente abandonará su nido y construirá un nuevo nido en otra parte [19]. Es por esto, que hay especies de cucos que han evolucionado de tal manera que los cucos parásitos femeninos suelen especializarse en la mímica de los colores y el patrón de los huevos de unas pocas especies huésped elegidas.

La búsqueda del cuco idealizó este comportamiento a la hora de la cría y, por lo tanto, puede aplicarse para varios problemas de optimización, teniendo en cuenta la siguiente afirmación, y es que, cada huevo en un nido representaría una solución, y un huevo de cuco representaría una nueva solución introducida y distinta a las anteriores [24]. El objetivo es utilizar las nuevas y potencialmente mejores soluciones (las que aportarían los cucos) para reemplazar una solución no tan buena dentro de los nidos. En la forma más simple, cada nido tiene un huevo. El algoritmo puede extenderse a casos más complicados en los que cada nido tiene múltiples huevos que representan un conjunto de soluciones.

Obviamente, la búsqueda del cuco tiene ciertas reglas o restricciones, que también deben ser contempladas a la hora de resolver un problema de optimización: cada cuco pone un huevo a la vez, y lo deposita en un nido elegido al azar; los mejores nidos, con una alta calidad para sus huevos, permite que sobrevivan, y que ese conocimiento se transfiera a la próxima generación; el número de nidos disponibles por los anfitriones es fijo, y el huevo puesto por un cuco presenta cierta probabilidad de ser descubierto por el pájaro anfitrión, sin embargo, operando en nidos peores, se descubren soluciones que fueron desechadas en cálculos anteriores.

Con el algoritmo basado en la búsqueda del cuco, se pueden solucionar tareas tales como: problemas de optimización de ingeniería; problemas de optimización combinatoria; fusión de datos en redes de sensores inalámbricos; nanoelectrónica

basada en la tecnología de amplificación; trabajar en mejorar el tren de red neuronal; o programación para sistemas de fabricación en serie [30]

4.5. *Manadas de Lobos (Wolf Search)*

Los lobos son depredadores sociales que cazan en manadas. Normalmente, viajan como una familia nuclear, que por lo general se mueven en grupos relativamente pequeños. Permanecen en silencio y usan el sigilo cuando cazan presas juntos. A diferencia de las hormigas, que utilizan feromonas para comunicarse con sus compañeros sobre los rasgos de los alimentos, los lobos renuncian a este tipo de comunicación, lo que acorta el tiempo de ejecución al buscar [12].

Ellos, a diferencia, han desarrollado características únicas y semi-cooperativas, es decir, se mueven en grupo en una formación libre, pero acoplada, aunque tienden a tomar presas individualmente. Este detalle es importante porque algunos algoritmos de optimización, como los que se basan en el enjambre, se centran en la coordinación del grupo, mientras que los algoritmos que hacen hincapié en los movimientos individuales caen en el otro extremo del espectro.

Cuando cazan, los lobos tratan de ocultarse a medida que se acercan a su presa. Y cambian de posición continuamente a otras mejores, con más refugio, menos obstáculos en el terreno o menos vulnerabilidad. Mientras están cazando, se mantienen atentos tanto a la búsqueda de posibles presas, como a tener cuidado con las amenazas que pueden aparecer, como son los cazadores humanos. Cada lobo en el grupo elige su propia posición, moviéndose continuamente a un lugar mejor y observando amenazas potenciales. Cuando estas aparecen, el lobo se desvía a una gran distancia de su posición actual, lo que ayuda a romper el punto muerto de vulnerabilidad. La dirección y la distancia que recorren al alejarse de una amenaza son aleatorias.

Los lobos tienen un excelente sentido del olfato y, a menudo, ubican a sus presas por el olor. Tienen una distancia de detección que crea un área de cobertura, que generalmente se denomina distancia visual. Esta distancia visual se aplica a la búsqueda de alimento (el óptimo global), a la conciencia de sus compañeros (con la esperanza de

moverse en una mejor posición) y a la señalización de los enemigos que pueden estar cerca (para saltar fuera del alcance visual).

Una vez que perciben que la presa está cerca, van rápidamente hacia ella, aunque en silencio y con mucha cautela porque no desean revelar su presencia. Durante la búsqueda, cuando ninguno de los elementos mencionados se detecta dentro del rango visual, los lobos se mueven en movimiento browniano, que imita la derivación aleatoria de partículas suspendidas en el fluido.

Recientemente, se ha propuesto un nuevo algoritmo de optimización heurística, basado en este comportamiento por parte de los lobos, el WSA (*Wolf Search Algorithm*). Este algoritmo es diferente de otras metaheurísticas inspiradas en la biología ya mencionadas anteriormente, porque posee tanto capacidad de búsqueda local individual como movimiento de grupo autónomo simultáneamente. En otras palabras, cada lobo en WSA caza independientemente, recordando su propio rastro, y sólo se fusiona con su compañero cuando este se encuentra en una mejor posición. De esta manera, WSA elimina la intercomunicación de largo alcance entre los lobos, que representan los puntos de búsqueda de soluciones, ahorrando tiempo.

El montaje depende del alcance visual. Por lo tanto, el comportamiento de enjambre de WSA, a diferencia de la mayoría de los algoritmos bio-inspirados, se delega a cada lobo individual en lugar de a un solo líder. Es decir, WSA trabaja como si hubiese múltiples líderes moviéndose en múltiples direcciones para hallar la mejor solución, en lugar de tratarse como un solo grupo que busca un óptimo en una única dirección a la vez. Además, WSA está equipado con una probabilidad de amenaza, que simula incidentes de lobos encontrándose con sus enemigos. La aparición de un cazador para cada lobo, se añade al azar, y al conocer a su cazador, cada lobo salta lejos de la gama visual de su cazador para evitar ser atrapado por el diseño del algoritmo.

En particular, se investiga el requisito de memoria de WSA. Se sabe que los lobos de la naturaleza tienen una capacidad de memoria sin precedentes, ya que pueden esconder alimentos en escondites, perciben y rastrean a una presa a millas de distancia, y establecen marcadores en su territorio orinando en las fronteras. Por lo tanto, se cree que los lobos recuerdan bien sus rasgos.

En el contexto de la programación de computadoras, se supone que los lobos que funcionan como agentes de búsqueda en el algoritmo de optimización de WSA, están habilitados por cachés de memoria que almacenan las posiciones visitadas previamente [29]. Esta característica hace que los agentes busquen y se muevan siempre a una mejor posición de la misma manera que los lobos cambian continuamente sus posiciones.

WSA ha mostrado ventajas únicas en eficiencia porque cada agente de búsqueda realiza simultáneamente la búsqueda y fusión de soluciones autónomas. Los niveles óptimos son superados cuando los agentes de búsqueda se alejan mucho al ser provocados por la aparición aleatoria de un enemigo. Sin embargo, las contribuciones de la WSA para encontrar soluciones en aplicaciones, presentan algunos inconvenientes, tales como: problemas en la prueba del vendedor ambulante, problemas de asignación cuadrática, problemas de programación de trabajo y problemas de orden secuencial [28].

4.6. *Ecolocalización de Murciélagos (Bat Echolocation Algorithm)*

Los murciélagos son animales fascinantes. Son los únicos mamíferos con alas y también presentan una habilidad avanzada, la ecolocalización. Se estima que hay alrededor de 996 especies diferentes que representan hasta el 20% de todas las especies de mamíferos [7,12]. Sus tamaños oscilan entre el del murciélago minúsculo del abejorro (de cerca de 1.5 a 2g), al de los murciélagos gigantes con una envergadura de aproximadamente 2 m y cerca de 1 kg de peso. Los micro murciélagos, típicamente tienen una longitud de antebrazo de aproximadamente 2,2 a 11 cm.

La mayoría de los murciélagos utilizan la ecolocalización hasta cierto grado, como sonar para detectar presas, evitar obstáculos y localizar sus hendiduras en la oscuridad, al emitir un pulso de sonido muy fuerte y escuchar el eco que se recupera al rebotar en los objetos circundantes [21]. La mayoría de los murciélagos usan señales cortas moduladas en frecuencia para barrer a su alrededor, mientras que otras utilizan continuamente señales de frecuencia constante para la ecolocalización. El ancho de banda de la señal varía dependiendo de la especie. Si idealizamos algunas de las características de ecolocalización de los micro murciélagos, podemos desarrollar varios algoritmos inspirados en murciélagos.

Por simplicidad, se usan las siguientes reglas aproximadas o idealizadas: todos los murciélagos usan la ecolocalización para sentir la distancia, y también "conocen" la diferencia entre las barreras de alimento/presa y de fondo de alguna manera "mágica"; los murciélagos vuelan aleatoriamente a una velocidad determinada y con una frecuencia fija, longitud de onda variable y sonoridad exacta para buscar presas. Pueden ajustar automáticamente la longitud de onda (o frecuencia) de sus impulsos emitidos y ajustar la velocidad de emisión de impulsos, dependiendo de la proximidad a su objetivo; aunque la sonoridad puede variar de muchas maneras, se asume que varía de un tono grande (positivo) a un valor constante mínimo.

Otra simplificación obvia es que no se usa trazado de rayos para estimar el retardo de tiempo y la topografía tridimensional. Aunque esto podría ser una buena característica para la aplicación en la geometría computacional, no se utiliza para los algoritmos, ya que es más computacionalmente extensa en casos multidimensionales. Para un problema dado, se puede usar cualquier longitud de onda para facilitar la implementación del algoritmo. En la implementación real, también se puede ajustar el rango, ajustando las longitudes de onda (o frecuencias), y el rango detectable (o la longitud de onda más grande) se elige de tal manera que sea comparable al tamaño del dominio de interés, o a intervalos más pequeños. Además, no necesariamente tenemos que usar las propias longitudes de onda, sino que también podemos variar la frecuencia mientras se fija la longitud de onda. Esto se debe a que ambas están relacionados debido al hecho de que la frecuencia es constante. Las frecuencias más altas tienen longitudes de onda cortas y viajan a una distancia más corta. Para los murciélagos, los rangos típicos son de unos pocos metros. La velocidad del pulso puede estar simplemente en un rango de (0, 1) donde 0 significa que no hay pulsos en absoluto, y 1 significa la velocidad máxima de emisión de impulsos.

El algoritmo de murciélagos es un algoritmo realmente prometedor, en lo que se refiere a problemas de optimización con restricciones continuas. Aunque su implementación es más complicada que la de muchos otros algoritmos metaheurísticos, demuestra que utiliza una combinación equilibrada de las ventajas de los algoritmos exitosos ya

existentes, con la característica innovadora basada en la ecolocalización de los murciélagos.

Las nuevas soluciones se generan ajustando las frecuencias, la sonoridad y las tasas de emisión de impulsos, mientras que para que la solución propuesta sea aceptada o no, dependerá de la calidad de las soluciones controladas o caracterizadas por la sonoridad y la frecuencia del pulso, que a su vez están relacionadas con la cercanía o la aptitud de las ubicaciones finales. Una extensión interesante será utilizar esquemas diferentes de longitud de onda o variaciones de frecuencia en lugar de la implementación lineal actual. Además, las velocidades de emisión de impulsos y sonoridad también se pueden variar de una manera más sofisticada. Otra extensión para problemas discretos es usar el retardo de tiempo entre la emisión de impulsos y el eco recuperado. Por ejemplo, en el problema del vendedor ambulante, la distancia entre dos nodos/ciudades adyacentes puede codificarse fácilmente como retardo de tiempo. A medida que los micro murciélagos usan la diferencia de tiempo entre sus dos oídos para obtener información tridimensional, pueden identificar el tipo de presa y la velocidad de un insecto volador. Por lo tanto, una extensión natural adicional al algoritmo del murciélago actual sería usar la ecolocalización direccional y el efecto Doppler, lo que puede conducir a variantes aún más interesantes y nuevos algoritmos [27].

4.7. *Ecolocalización de Delfines (Dolphin Echolocation Algorithm)*

La ecolocalización de los delfines es un nuevo método de optimización. Este método imita las estrategias utilizadas por los delfines para su proceso de caza. Los delfines producen una especie de voz llamada sonar para localizar el objetivo, haciendo este sonar de cambio de delfines para modificar el objetivo y su ubicación. Este hecho se imita aquí como la característica principal del nuevo método de optimización.

Un delfín es capaz de generar sonidos en forma de clics. Los clics son direccionales y sirven para la ecolocalización, que a menudo ocurren en una serie corta llamada tren de clics. La tasa de clics aumenta al acercarse a un objeto de interés. Su frecuencia es mayor que la de los sonidos utilizados para la comunicación y difiere entre especies. Cuando el sonido golpea un objeto, parte de la energía de la onda sonora se refleja hacia el delfín.

Tan pronto como se recibe un eco, el delfín genera otro clic. El lapso de tiempo entre el clic y el eco permite al delfín evaluar la distancia desde el objeto; la fuerza variable de la señal que se recibe en los dos lados de la cabeza del delfín le permiten evaluar también la dirección. Al emitir continuamente clics y recibir ecos de esta manera, el delfín puede rastrear objetos y el lugar en que se encuentran.

Aunque los murciélagos también utilizan la ecolocalización, difieren de los delfines en su sistema de sonar. Los murciélagos utilizan su sistema de sonar a intervalos cortos de hasta aproximadamente 3-4 m, mientras que los delfines pueden detectar sus objetivos en rangos que varían de unas decenas de metros a más de cien. Muchos murciélagos buscan insectos que se lanzan rápidamente hacia adelante y atrás, lo que lo hace muy diferente al comportamiento de huida de un pez perseguido por un delfín [14]. Además, la velocidad del sonido en el aire es aproximadamente una quinta parte de la del agua, por lo que la tasa de transferencia de información durante la transmisión de sonar para los murciélagos es mucho más corta que para los delfines. Estas y muchas otras diferencias en el medio ambiente y la presa requieren tipos totalmente diferentes de sistema de sonar, lo que, naturalmente, hace una difícil comparación directa.

En cuanto los problemas de optimización, se puede entender que la ecolocalización es similar a la optimización en algunos aspectos; el proceso de forrajear presas utilizando la ecolocalización en delfines es similar a encontrar la respuesta óptima de un problema. Los delfines, inicialmente, buscan en todo el espacio de búsqueda para encontrar a la presa. Tan pronto como un delfín se acerca al objetivo, el animal restringe su búsqueda, y aumenta gradualmente sus clics para concentrarse en la ubicación de la presa.

El método simula la ecolocalización de delfines limitando su exploración proporcional a la distancia del objetivo [31]. Se pueden identificar dos fases: en la primera fase, el algoritmo explora todo el espacio de búsqueda para realizar una búsqueda global, por lo que se deben buscar regiones inexploradas, localizando lugares aleatorios en el espacio de búsqueda; la segunda fase se concentra en la investigación en torno a mejores resultados obtenidos desde la etapa anterior.

Para controlar el valor de las respuestas creadas al azar con el fin de establecer la relación de los resultados mediante el algoritmo de ecolocalización de delfines, el

usuario podría cambiar la relación de las respuestas de acuerdo con una curva predefinida. En otras palabras, una búsqueda global, cambia gradualmente a una local en un estilo definido por el usuario.

Este nuevo método de optimización basado en la ecolocalización de los delfines tiene la ventaja de trabajar de acuerdo con el esfuerzo computacional que el usuario puede permitirse para su optimización. En este algoritmo, el factor de convergencia viene definido y es controlado con el fin de realizar una optimización adecuada [34].

5. CONCLUSIONES

Los métodos de optimización heurística tienen una ventaja sobre los exactos porque pueden inducir una solución globalmente óptima de manera rápida y con un resultado más que conforme. Un tipo especial, conocida como optimización inspirada por la naturaleza o “metaheurística” está ganando popularidad sustancial en la comunidad de investigación debido a sus ventajas, que son aplicables en la inteligencia computacional, la minería de datos y sus aplicaciones.

Tomados de las maravillas de la naturaleza, tales algoritmos optimizan computacionalmente los problemas de búsqueda complejos, con rendimiento y eficiencia de búsqueda superiores en comparación con las técnicas de optimización que se usaban anteriormente. Cabe añadir que, basándose en muchos estudios literarios, está claro que algunos son más eficientes y populares que otros. Por ello, sería útil llevar a cabo más estudios para cada tipo.

En la actualidad, puede haber cierta confusión y distracciones en la investigación de algoritmos metaheurísticos. Por un lado, los investigadores se han centrado en ideas novedosas importantes para resolver problemas difíciles. Por otra parte, algunos investigadores inventan artificialmente nuevos algoritmos en beneficio de las publicaciones, con poca mejora y sin novedad.

Los investigadores deben ser alentados a llevar a cabo verdaderamente nuevos e importantes estudios, que sean realmente útiles para resolver problemas difíciles y, con el tiempo, es muy probable que todos estos métodos nuevos terminen utilizándose en

el día a día, como programación de nuestros ordenadores o móviles, nuestros vehículos, todos nuestros aparatos electrónicos en general. Esto conllevará a que la ciencia y la tecnología sigan avanzando de la mano hasta un futuro realmente prometedor para el ser humano.

6. CONCLUSIONS

Heuristic optimization methods have an edge over exact methods because they can incrementally induce globally optimum solution by using heuristics to efficiently search a large space. A special kind of heuristic optimization is known as nature-inspired optimization or metaheuristics gaining substantial popularity in the research community due to its advantages, which are applicable in computational intelligence, data-mining and their applications.

Borrowed from the wonders of nature, such algorithms computationally optimize complex search problems with superior performance and search efficiency compared to earlier optimization techniques. Based on many studies in the literature, some algorithms are more efficient and popular than others. It would be helpful to carry out more studies.

Currently, there may be some confusion and distraction in the research of metaheuristic algorithms. On the one hand, researchers have focused on important novel ideas for solving difficult problems. On the other hand, some researchers artificially invent new algorithms for the sake of publications with little improvement and no novelty.

Researchers should be encouraged to carry out truly novel and important studies that are really useful to solve hard problems. Over time, it is very likely that all these new methods end up being used in everyday life, such as programming our computers or mobiles, our vehicles, and all our electronic devices, in general. This will mean that science and technology continue to advance hand to hand, to a really promising future for humans.

7. BIBLIOGRAFÍA

1. Beni, G. 1988. The concept of cellular robotic system. In: Proceedings of the 1988 IEEE International Symposium on Intelligent Control, pp. 57–62. IEEE Computer Society Press, Los Alamitos.
2. Beni, G. y Wang, J. 1989. Swarm intelligence. In: Proceedings of the Seventh Annual Meeting of the Robotics Society of Japan, pp. 425–428. RSJ Press, Tokyo.
3. Dorigo, M. 1991. Optimization, Learning and Natural Algorithms, PhD thesis, Polytechnic di Milano, Italy.
4. Camazine, S. y Sneyd, J. 1991. A Model of Collective Nectar Source by Honey Bees: Self organization Through Simple Rules. *J. Theor. Biol.* 149, 547–571.
5. Beni, G. y Hackwood, S. 1992. Stationary waves in cyclic swarms. In: Proceedings of the 1992 International Symposium on Intelligent Control, pp. 234–242. IEEE Computer Society Press, Los Alamitos.
6. Dorigo, M., Maniezzo, V. y Colorni, A. 1996. Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics--Part B*, volume 26, número 1, pp. 29-41.
7. Altringham, J.D. 1996. *Bats: Biology and Behaviour*, Oxford University Press.
8. Caro, D. y Dorigo, M. 1998. "Extending AntNet for best-effort quality-of-service routing," Proceedings of the First International Workshop on Ant Colony Optimization (ANTS'98).
9. Caro, D y Dorigo, M. 1998. "AntNet: a mobile agents approach to adaptive routing," Proceedings of the Thirty-First Hawaii International Conference on System Science, vol.7, pp.74-83.
10. Caro, D. y Dorigo, M. 1998. "Two ant colony algorithms for best-effort routing in datagram networks," Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98), pp.541-546.
11. Bonabeau, E., Dorigo, M. y Theraulaz, G. 1999. *Swarm Intelligence*. Oxford University Press, Oxford.
12. Colin, T. 2000. *The Variety of Life*. Oxford University Press.
13. Lučić, P. y Teodorović, D. 2001. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In: Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, Sao Miguel, Azores Islands, Portugal, pp. 441–445.
14. Thomas, J.A., Moss C.F. y Vater, M. 2002. *Echolocation in bats and dolphins*. University of Chicago Press, Chicago.
15. Lučić, P. y Teodorović, D. 2002. Transportation modeling: an artificial life approach. In: Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence, Washington, DC, pp. 216–223.
16. Lučić, P. y Teodorović, D. 2003. Computing with bees: attacking complex transportation engineering problems. *Int. J. Artif. Intell. T.* 12, 375–394.
17. Lučić, P. y Teodorović, D. 2003. Vehicle routing problem with uncertain demand at nodes: the bee system and fuzzy logic approach. In: Verdegay, J.L. (ed.) *Fuzzy Sets in Optimization*, pp. 67–82. Springer, Heidelberg.
18. Teodorović, D. y Dell'Orco, M. 2005. Bee colony optimization – a cooperative learning approach to complex transportation problems. In: *Advanced OR and AI Methods in Transportation*. Proceedings of the 10th Meeting of the EURO Working Group on Transportation, Poznan, Poland, pp. 51–60.

19. Payne, R.B., Sorenson, M.D. y Klitz, K. 2005. *The Cuckoos*, Oxford University Press, Oxford.
20. Teodorović, D., Lučić, P., Marković, G. y Dell' Orco, M. 2006. Bee colony optimization: principles and applications. In: Reljin, B., Stanković, S. (eds.) *Proceedings of the Eight Seminar on Neural Network Applications in Electrical Engineering – NEUREL 2006*, University of Belgrade, Belgrade, pp. 151–156.
21. Richardson, P. 2008. *Bats*. Natural History Museum, London.
22. Yang, X.S. 2008. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press.
23. Yang, X.S. 2009. Firefly Algorithms for Multimodal Optimization. In: O. Watanabe and T. Zeugmann (Eds.): *SAGA 2009*, LNCS 5792, University of Cambridge, Cambridge, pp. 169–178.
24. Yang, X.S. y Deb., S. 2009. Cuckoo search via Lévy flights. *World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*. IEEE Publications. World Congress on, pp. 210–21.
25. Yang, X.S. 2010 *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley, Chichester.
26. Yang, X.S. 2010. Firefly algorithm, stochastic test functions and design optimisation. *Internacional Journal of Bio-Inspired Computation*. 2(2):78-84.
27. Yang, X.S. 2010. A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. pp. 65-74.
28. Kaveh, A. y Farhoudi, N. 2011. A unified approach to parameter selection in metaheuristic algorithms for layout optimization. *J Constar Steel Res* 2011; 67:15453–62.
29. Tang, R., Fong, S., Yang X.S y Deb, S. 2012. Wolf search algorithm with ephemeral memory. In *Digital Information Management (ICDIM)*. 2012 Seventh International Conference on. pp. 165-172.
30. Yang, X.S. y Deb, S. 2013. Multiobjctive cuckoo search for design optimization, *Computers & Operations Research*, 40(6), pp. 1616-1624.
31. Kaveth, A., Farhoudi, N. 2013. A new optimization method: Dolphin Echolocation, *Advances in Engineering Software*, 59 (2013) 53-70.
32. Davidović, T., Teodorović, D. y Selmić, M. 2014. Bee colony optimization: The Algorithm Overview.
33. Yang, X.S. 2014. *Nature-Inspired Optimization Algorithms*, Elsevier.
34. Kaveth A., Jafari L. y Farhoudi N. 2014. Truss optimization with natural frequenxy constraints using a dolphin echolocation algorithm, *Asian Journal of Civil Engieering (BHRC)* Vol. 16, No. 1 (2014) pp. 29-46
35. Fister Jr, I., Yang, X.S., Fister, I., Brest, J. y Fister, D. 2015. A Brief Review of Nature-Inspired Algorithms for Optimization. *ArXiv:1307.4186v1 [cs.NE]*.