FUNDAMENTOS Y PERSPECTIVAS DE ALGORITMOS INSPIRADOS EN SISTEMAS BIOLÓGICOS

MÓNICA BARONA FIGUEROA
ALONSO ESTEBAN ZAPATA DÍAZ

UNIVERSIDAD ICESI
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA INDUSTRIAL
CALI
NOVIEMBRE 2015

FUNDAMENTOS Y PERSPECTIVAS DE ALGORITMOS INSPIRADOS EN SISTEMAS BIOLÓGICOS

MÓNICA BARONA FIGUEROA
ALONSO ESTEBAN ZAPATA DÍAZ

Proyecto de Grado para optar el título de Ingeniero Industrial

Director proyecto
FERNANDO QUINTERO

UNIVERSIDAD ICESI
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA INDUSTRIAL
CALI
NOVIEMBRE 2015

CONTENIDO **Pág.**

Lista de Figuras

Listado de Tablas

Listado de ecuaciones.

RESUMEN

Desde la década de 1940 se hizo una primera apuesta de sugerir la computación natural para el desarrollo de algoritmos en la resolución de diversos problemas complejos. Desde entonces, a la medida que se desarrolló la programación matemática y las cualidades computacionales hicieron posible las soluciones de modelos que dan soporte a decisiones organizacionales, se fue teniendo como referente a los sistemas naturales, contenidos dentro la biomímesis, para desarrollar algoritmos, inspirados biológicamente, que brinden soluciones a problemas más complejos, a través de la creación de métodos meta heurísticos, análogos a funcionalidades de redes neuronales, procesos evolutivos, sistemas sociales (enjambres) y al sistema inmunológico, todo dado a que la programación matemática queda imposibilitada en dar solución en tiempos razonables para problemas combinatorios. Es entonces que el objetivo de este trabajo parte del interés de conocer meta heurísticas inspiradas en la biología, partiendo del origen de la biomímesis y conociendo sus múltiples posibilidades metafóricas que sirven de fuente de inspiración en la búsqueda de soluciones reales en ingeniería. Describir la taxonomía de la biomímesis, estudiar el camino hacia la computación y poder comprender sus estructuras algorítmicas en pro de la optimización, son los motivos de este trabajo que contribuye a difundir conocimiento del inmenso universo creado de las metaheuristicas desde hace un poco más de dos décadas, específicamente a partir de la inspiración en sistemas biológicos. Al final, se vive la experiencia de conocer y comparar la solución de un problema de agente viajero (TSP, *travelling salesman problem*) aplicando un algoritmo de ACO (Ant Colony Optimization) y programado desde Matlab frente a heurísticas tradicionales.

Palabras claves: Biomimetísmo, Biomimicry, Biomimética, Computación evolutiva, Algoritmo genético, meta heurística, sistemas naturales, computación.

INTRODUCCIÓN

En la últimas dos décadas se ha incrementado el número de publicaciones descriptivas y de casos aplicados de metaheurísticas, que surgen de la razón de ser de la biomimética, siendo esta una aproximación a la innovación que busca soluciones sostenibles a los problemas humanos mediante la simulación de estrategias y patrones probados por la naturaleza. Lo anterior, conducente a la creación de algoritmos inspirados en comportamientos de especies y la genética. Los practicantes y teóricos de la optimización han encontrado en la naturaleza una enorme fuente de inspiración para construir métodos de búsqueda de soluciones a problemas de programación combinatoria, cuya solución entra en la categoría NP, es decir, que su complejidad requiere de un tiempo de solución computacional polinomial en una máquina de Turing no determinística (término abstracto para idealizar una máquina no existente). Muchos de los problemas abordados con modelos meta heurísticos se caracterizan por ser *ad-hoc*, es decir, adecuados para un fin específico, lo que obliga a crear definiciones y conceptos abstractos de las técnicas meta heurísticas, para que dado el planteamiento específico de un problema, se desarrolle un algoritmo que salvaguarde las funcionalidades de la técnica pero con adaptación a las características propias del problema.

Frente a este panorama, hay un universo de posibilidades en el rango de problemas, y por ende de soluciones ya propuestas y otras aún por hallar. Ante el tradicional esfuerzo de plantear soluciones con modelación y programación matemática, este trabajo es un esfuerzo por conocer las bases funcionales de varias metaheuristicas. Se restringe al trazo inspirador proveniente de los conocimientos surgidos en la biología, que han permitido aplicar conceptos a diversas áreas de la ingeniería, para luego si entrar al campo específico de la investigación de operaciones, apoyada en la teoría de la programación, con el interés de abordar soluciones computacionales a problemas de diversos ámbitos. En el caso específico de este proyecto a problemas reales de la cadena de abastecimiento y la logística.Si bien el marco teórico de la biomímesis y su relación con la ingeniería aglutinan un gigantesco horizonte, este trabajo realiza una síntesis descriptiva y aclaradora de la metáfora entre las dos ciencias, para luego centrarse en su aplicación en la optimización combinatoria y por último, experimentar un caso aplicado de una de las técnicas.

1 CAPÍTULO I. DEFINICIÓN DEL PROBLEMA

1.1 Contexto del Problema

La programación matemática con modelos exactos se originó de manera formal dentro de la investigación de operaciones, con la factibilidad computacional de aplicar el método simplex para la solución de problemas con estructuras lineales. Luego con el desarrollo de algoritmos avanzados de ramificación y acotación para obtener soluciones en programación entera, y posteriormente, con aplicaciones de programación binaria y mixta entera, se identificaron complejidades computacionales en la búsqueda de soluciones óptimas en problemas denominados combinatorios, llamados así, dadas las numerosas posibilidades de posibles soluciones entre los elementos de decisión. Estos problemas se ubican con aplicaciones en sistemas de la cadena de abastecimiento, la logística y en la programación de trabajos. *Traveling salesman problem* (TSP) en todas sus aplicaciones y extensiones a problemas de ruteo de vehículos (VRP), *bin packing*, *hubs and spokes*, y muchos entornos de máquinas, tipos de restricciones y criterios a optimizar en *job scheduling* (flujos intermitentes, continuos, talleres abiertos, sin y con restricciones).

En los problemas combinatorios, a la medida que se escala el tamaño del problema, se configuran tiempos exponenciales de ejecución, considerando una máquina determinística de Turing (computadores convencionales), lo que ha conllevado a la iniciativa por determinar métodos alternativos de búsquedas de solución, con algún detrimento tolerable en la calidad de la solución, pero con eficiencia en el tiempo de búsqueda de una buena solución.

Los antecedentes de esta iniciativa, se remontan al mismo tiempo de la formalización del área de estudio de *Operations Research*, cuando Alan Turing, pionero de los algoritmos computacionales, visionó la computación evolutiva en relación a la teoría de la evolución. A finales de los años cuarenta del siglo XX, Turing propuso la “investigación genética o evolucionaria”, a la que siguió el trabajo de Bremermann en 1962, quien realizó experimentación computacional a través de “optimización en evolución y recombinación”. Se dieron simultáneamente trabajos en Estados Unidos de programación evolutiva, en Holanda de Algoritmos Genéticos y en Alemania de la estrategia de evolución. La consolidación se da en

1990 con la conformación de la “Programación Evolucionaria”. (A.E.Eiben & Smith, 2003).

Tener como referente los sistemas naturales como fuente de inspiración para el desarrollo algorítmico es un brazo adicional de las muchas posibilidades que ofrece la biomímesis. La imitación de estructuras anatómicas y genéticas, comportamientos sociales, métodos evolutivos, y en general, toda clase de procesos y elementos naturales, ha despertado interés en múltiples disciplinas, para enriquecer la capacidad para el diseño de productos y procesos. Investigar el origen y la estructura taxonómica de la biomímesis, identificar el enlace con la programación computacional y comprender las metaheuristicas surgidas desde la inspiración natural, es el problema visto como oportunidad de investigación, que sirve de base en el estado del arte para futuros trabajos con interés en la biomímesis, computación evolutiva y la metaheurística.

1.2 Análisis y Justificación

La ingeniería industrial como disciplina que busca métodos de administración eficiente de recursos y mejoramiento de procesos, hace uso de metodologías cuantitativas que representen realidades sistémicas y den soporte a propuestas de solución a problemas diversos. Estas metodologías enmarcan técnicas de modelado de tres diferentes modalidades: descriptivas, prescriptivas y predictivas. Las técnicas prescriptivas de optimización han sido históricamente las que mayor atención investigativa han recibido dentro de la investigación de operaciones. En estas, el arte de modelado son representaciones abstractas de la realidad empleando relaciones matemáticas entre entradas y salidas, y con ayuda de procesos algorítmicos se activa la búsqueda de soluciones exactas. Estos procesos algorítmicos requieren medios computacionales rápidos y confiables.

Se ha dispuesto entonces el camino para el desarrollo de sistemas computacionales caracterizados por patrones complejos, que han encarado barreras aún por superar en problemas combinatorios, un desafío parcialmente satisfecho con el desarrollo de métodos heurísticos, que con pasos o procedimientos simples aproximan la búsqueda a una solución subóptima pero con riesgo de detenerse en óptimos locales, según características del problema a resolver y del procedimiento empleado. Intensificar la búsqueda de mejores soluciones y en procura de óptimos globales, ha sido el motivante para la creación de las metaheuristicas (“más allá de la heurística”), con procedimientos

innovadores que imitan mecanismos metalúrgicos (simulado de recocido), memorias condicionadas y selectivas (búsqueda tabú), sistemas neuronales, comportamientos de sociedades de insectos y enjambres de especies, mecanismos genéticos y evolutivos en general.

Conocer el camino inspirado desde la biología, con una breve introducción a todo lo que representa la biomímesis, sus posibilidades en campos de aplicación relacionados con la ingeniería, y luego centrarse en el campo de la programación computacional, con el fin último de la optimización de problemas combinatorios, constituye el interés de este trabajo, para sentar una base de conocimiento significativo y oportuno para trabajos posteriores en docencia e investigación.

1.3 Formulación del Problema

Identificar el árbol de conocimiento de aplicaciones inspiradas en sistemas naturales, y conocer la lógica y conceptualización de la ramificación conducente a la contribución de técnicas metaheuristicas en problemas combinatorios en campos de estudio de la ingeniería industrial, con previo trámite en la programación computacional y con comprensión en la metáfora biológica, es el elemento motivador al proceso investigativo, descriptivo y analítico por desarrollar en este trabajo.

2 CAPITULO II. Objetivos

2.1 Objetivo General

Describir la Biomímesis como fuente de inspiración en la búsqueda de soluciones de problemas reales a través de las técnicas de Optimización.

2.2 Objetivo del Proyecto

Describir las técnicas meta heurísticas desde la metáfora de los sistemas naturales.

2.3 Objetivos Específicos

- Identificar y describir la taxonomía de las funcionalidades de los sistemas biológicos que han servido como inspiración para aplicaciones y soluciones diversas de ingeniería.

- Identificar y describir las funcionalidades algorítmicas inspiradas desde la biología en el marco de la ciencia de la computación y en relación a necesidades de optimización.

- Categorizar aplicaciones metaheurísticas inspiradas en la biología, para problemas de la logística y la cadena de abastecimiento.

- Ejemplificar e Ilustrar la aplicación de una técnica metaheurística ad-hoc (inspirada en la biología) para un caso en el área de logística o cadena de abastecimiento.

3 CAPÍTULO III. Marco de Referencia

A partir del enfoque teórico del proyecto, en su fin de conocer el camino recorrido desde la biomímesis hasta la metaheurística inspirada en la biología, el marco de referencia introduce los fundamentos básicos y el contexto esencial de estos dos conceptos, al igual que algunos fundamentos y principios sobre la evolución y la genética que pueden ser relevantes para la comprensión de ideas posteriores en el trabajo. Los conceptos de optimización combinatoria serán ampliados en el desarrollo del trabajo.

3.1 BIO-MÍMESIS

(Del griego *bios*, vida, y *mímesis*, imitación)

La población humana ocupa hoy todo el espacio del planeta, con su número creciente y con el mayor e inadecuado uso del territorio, ha sobrepasado los límites de la sostenibilidad y esto conlleva a una necesidad por aprender a renovarse como especie en su interacción con la naturaleza, obligando a cambiar su propio nicho ecológico y a cuestionar su profesión (rol) dentro del ecosistema. Es aquí donde la biomímesis aparece como una nueva manera de contemplar y valorar la naturaleza, que enmarca a la humanidad empezando a vivir una era basada no en lo que pueda extraer del mundo natural, sino en lo que éste puede enseñar.

La biomímesis es la ciencia que estudia los modelos de la naturaleza para imitar los diseños y procesos biológicos en pro de hallar soluciones a los problemas de la especie humana. Se pueden mencionar ejemplos como el diseño de una célula fotovoltaica inspirada en una hoja, fibras que imitan las telas de araña, cerámica irrompible derivada de la madreperla, granos perennes inspirados en las gramíneas pratenses, y otros innumerables casos de imitación de elementos y procesos de la naturaleza. El fin es estudiar obras maestras de la naturaleza (fotosíntesis, autoensamblaje, selección natural, ojos y oídos, pieles y caparazones, neuronas parlantes, medicinas naturales entre otras) para luego replicar esos diseños y procesos, y hacer la emulación consciente del ingenio de la vida o la innovación inspirada en la naturaleza. Ésta imitación tiene múltiples posibilidades específicas, tales como la de cambiar maneras de obtener alimento,

desarrollar materiales y modos de generación de energía, descubrir mecanismos de curación y de almacenar información, entre otras.

Siendo difícil mencionar una época o edad de la historia de la humanidad como punto de referencia para el origen de la biomímesis, es remarcable encontrar ejemplos de imitaciones en siglos pasados. Leonardo Da Vinci aplico biomimética al estudio de las aves buscando permitir el vuelo humano. Él observo muy de cerca la anatomía y el vuelo de las aves, realizo numerosas notas y bocetos (**Figura 1**) de observaciones e imitaciones, e innumerables planos de las "maquinas voladoras" que propuso. A pesar de que no tuvo éxito con su propia maquina voladora, sus ideas viven y fueron fuente de inspiración para los hermanos Wright que se inspiraron en las aves, específicamente en las palomas (**Figura 2**), en su vuelo. Finalmente tuvieron éxito en la creación y pudieron volar el primer avión en 1903.

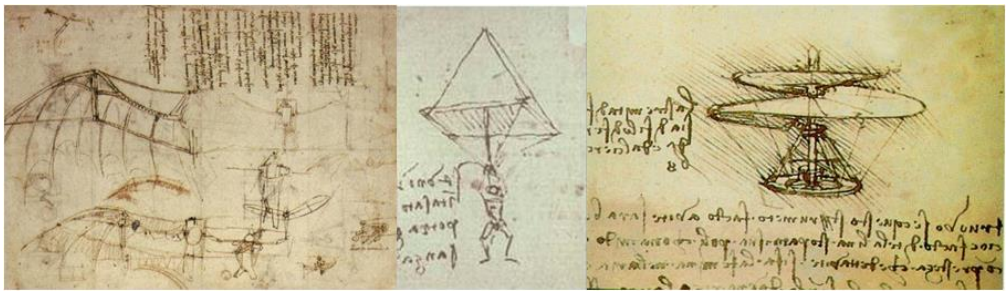


Figura 1. Bocetos "maquinas voladoras" Da Vinci

Figura 2. Izquierda: diseño Leonardo Da Vinci como una maquina voladora (1488) inspirada en el vuelo de las aves, en el Medio y Derecha: las palomas influyeron en el diseño de los hermanos Wright para el primer avión (1903).

Al llegar al siglo XX la inercia del desarrollo tecnológico fijó un enfoque primordial en estudiar con mayor ahínco las estructuras y funcionalidades de sistemas biológicos, desarrollados desde hace 3.8 billones de años, en el contexto del manejo eficiente de recursos, uso energético, reutilización de desperdicios, y en general, la administración del entorno.

En un mundo biomimético se produciría tal y como los animales y las plantas hacen empleando energía solar y compuestos simples, fibras totalmente biodegradables, materiales cerámicos, plásticos y productos químicos. Las explotaciones agrícolas, inspiradas en las praderas, se auto-abonarían y serían resistentes a plagas. En el momento de buscar nuevos medicamentos o cultivos, se tomarían como referencia a los animales, que durante millones de años han empleado plantas para mantenerse sanos y nutridos. Hasta la computación imitaría la naturaleza, con un *software* cuyas soluciones evolucionan y un *hardware* que aplica el paradigma de llave y cerradura. La metáfora de la llave cerradura se relaciona con el sistema inmunológico humano, en donde los anticuerpos (cerradura) quienes reconocen sustancias extrañas para ser eliminadas por el cuerpo y el antígeno (llave) que es cualquier sustancia que estimula la producción de un anticuerpo (**Figura 3**). En el caso del Hardware (cerradura) y el Software (llave) la metáfora se realiza con respecto a su funcionamiento

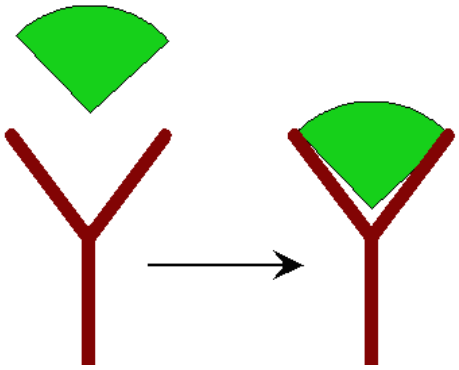


Figura 3. Metáfora de Llave Cerradura.

3.2 Computación Natural

Siendo la programación computacional el medio eficiente de solución de las técnicas de optimización se hace relevante introducir lo que se conoce como computación natural, donde se han extraído ideas del mundo natural para ser aplicadas en soluciones computacionales.

El significado de vida como el procesamiento de información en un cuerpo, se antepone a todas las inspiraciones que han servido como base del desarrollo de la computación evolutiva. (Banzhaf, 2012). Haciendo referencia a la computación natural al área donde convergen modelos y técnicas inspirados en los sistemas biológicos, se cuentan varios ejemplos: la computación neuronal que se inspira en el funcionamiento del cerebro; la computación evolutiva fundamentada en la evolución de las especies de la teoría Darwinista; la teoría "autómata celular" basada en la comunicación intercelular; la inteligencia de enjambre inspirada por el comportamiento de grupos de organismos; el sistema inmune artificial apoyado en el sistema inmune natural; los sistemas de vida artificial fundamentados por las propiedades de la vida natural en general; la computación con membranas que recoge elementos de las vías de compartimentar –células con divisiones internas que trabajan independientemente y aumentan su diversidad funcional– en las cuales las células procesan información; y la computación amorfa inspirada por la morfogénesis (desarrollo de la forma en un organismo).

Las múltiples investigaciones son interdisciplinarias y forman un puente entre las ciencias naturales y la computación. Este puente conecta el nivel de información tecnológica y el nivel de la investigación básica, debido a que la computación natural cubre un gran espectro de metodologías de investigación, desde la investigación teórica, algorítmica y aplicaciones de software en ingeniería, hasta investigaciones experimentales en biología, química y física. (Bck, Kok, & Rozenberg, 2010).

Los algoritmos inspirados en la Biología son una rama proveniente desde la Biomimética, pasando por la Computación Natural, que comprenden a las Redes Neuronales, la Computación Evolutiva, y esta a su vez a los algoritmos genéticos, la programación genética, la programación evolutiva, la evolución diferencial y las estrategias de la evolución; también comprende a los algoritmos inspirados en sistemas sociales (enjambres) y una adaptación del sistema inmune humano. En el desarrollo del trabajo, se sintetizan todas las ramas que derivan de los algoritmos inspirados en sistemas biológicos para describir las estrategias encaminadas a encontrar soluciones óptimas a los problemas que esta teoría

afronta en el campo de la complejidad computacional. La manera como se representa la información de la población de problemas y de candidatos a solución, logrando la evaluación de estos para encontrar los mejores candidatos, hace referencia a como se desprenden los tipos de computación dentro de los algoritmos inspirados biológicamente.

La importancia de esta investigación es generar un documento que sea fuente de consulta para cualquier practicante de modelado que requiera un punto de partida que visualice desarrollos de ingeniería desde la biología, y específicamente en el entendimiento de los fundamentos de algoritmos inspirados en biología, y así reconocer su alcance y posibilidades en campos de aplicación de la ingeniería industrial.

3.3 Algoritmos Bio-inspirados

Continuamente se están afrontando viejos y nuevos problemas de ingeniería, disponiendo de mejores recursos computacionales, nuevos tipos de computadoras, redes, y lo que se ha denominado el internet de las cosas. Plataformas tecnológicas, nuevas formas de programación, aprendizaje de máquinas, e inteligencia artificial entre otras, encierran un imaginario algorítmico que asume el reto de resolver problemas aún complejos, de manera eficaz, confiable y eficiente (en el menor tiempo computacional posible).

En relación a los problemas de optimización, la actividad investigativa empezó explorando el planteamiento de algoritmos exactos en heurística *ad hoc,* y luego hizo tránsito hacia las metaheurísticas. Los algoritmos exactos garantizan encontrar para cada instancia de un problema de tamaño finito, una solución óptima en tiempo limitado. Sin embargo, en los problemas de programación matemática combinatoria (*NP-hard y NP-complete*) se presenta un fuerte inconveniente ya que el tiempo de ejecución crece de forma exponencial con el tamaño del problema, extendiéndose a soluciones que tardarían años, décadas y hasta siglos alcanzarlas. Ante esto, los algoritmos *ad hoc* tienen como objetivo encontrar soluciones en un tiempo significativamente reducido, pero sin garantía de hallar soluciones óptimas o convergentes dentro de un nivel de tolerancia sub-óptimo. Tampoco son fáciles de definir en determinados problemas. En este sentido, las metaheurísticas ofrecen un balance adecuado, basadas en definiciones o conceptos abstractos pero robustos, adaptables a contextos y características de problemas, para así lograr soluciones de buena calidad (óptimo global en muchos casos) en un tiempo moderado. El diseño de estas técnicas tiene como punto central, el desarrollar estrategias eficientes de búsqueda de

soluciones factibles al problema, haciendo exploraciones aleatorias en diferentes módulos del espacio factible, y de manera iterativa busca mejorar la solución del problema, hasta acercarse a un nivel de convergencia adecuado, condicionado a un tiempo razonable de ejecución.

La hibridación y el paralelismo son dos de las ramas más exitosas para el diseño de algoritmos. La hibridación permite integrar información del problema en el algoritmo de resolución para trabajar en contacto con sus características diferenciadoras, tiene la posibilidad de involucrar varios algoritmos distintos en el proceso de búsqueda de una mejor solución. Los algoritmos paralelos permiten mitigar los problemas vinculados al tiempo de ejecución intensivos y requerimientos de memoria importantes para problemas complejos y elaborados de interés actual. Los modelos de paralelización donde varios hilos de búsqueda simultánea usan estrategias y valores de parámetros que posibilitan una exploración más profunda que conducen a soluciones más precisas.

3.4 Principios biológicos de interés.

Todos los organismos vivos están constituidos por células, éstas contienen cromosomas en su núcleo, conformados por cadenas de ADN, las cuales contienen la mayor parte de la información genética y sirve de plano de construcción para el organismo. Un cromosoma está constituido de genes y cada uno de estos codifica para una proteína en particular (esto es llamado, expresión del gen). El gen codifica, o hace, para un rasgo específico, por ejemplo el color azul de los ojos o el color rojo del cabello, mientras que los diferentes juegos de genes son llamados alelos, estos determinan, por ejemplo el tipo de sangre. Situaciones como las que el padre tenga un color de ojos y el hijo nazca con otro color, estas son formas alternativas del gen y cada gen está situado en un locus (posición) en particular.

Muchos organismos poseen múltiples cromosomas en sus células, la colección de todo el material genético se llama genoma. Los organismos con los cromosomas organizados en pares son denominados diploides, y los que poseen la organización en impares son haploides. Las especies que se reproducen sexualmente son diploides, incluyendo a los seres humanos. Durante la reproducción sexual, en la recombinación (cruce) se produce un intercambio genético entre los padres. (Melanie, 1996). Esta información es importante debido a que sirve como base para entender parte de la metáfora de la computación evolutiva que está desarrollada más adelante en el cuerpo del trabajo.

3.5 Principio de Selección Natural de Darwin.

Los algoritmos evolutivos siguen el principio diferencial de la selección natural propuesta por Charles Darwin: una explicación de la diversificación biológica como una visión macroscópica de la evolución, resultado de un ambiente que puede alojar un número limitado de individuos. Ante el instinto básico de reproducción de las especies, la selección natural se vuelve inevitable si el tamaño de la población está creciendo de forma exponencial. A continuación se detallan los fundamentos de la teoría de la selección natural, explicando cada componente desde la unidad básica -individuo- y sus relaciones en una comunidad.

a) Existen unas entidades llamadas **individuos** (de las mismas características) los cuales forman una **población**, estos individuos pueden reproducirse o pueden ser objeto de la reproducción.
b) Existe **herencia** en la **reproducción**, y los individuos de esta población producen nacimientos similares.
c) Durante la reproducción, existe **variabilidad** que afecta la probabilidad de **supervivencia**, y así, la reproducibilidad de los individuos. Esta variedad es producida por efectos estocásticos (como la recombinación y la mutación aleatoria).
d) Al existir recursos finitos, se causa la **competencia** entre individuos.
e) No todos los individuos pueden sobrevivir.
f) Cada individuo representa una combinación única de características fenotípicas que son evaluadas por el ambiente.

La selección natural favorece a aquellos individuos que compiten con mayor eficiencia por los recursos dados, en otras palabras, estos son los mejores **adaptados** respecto a las condiciones del ambiente (***Fitness***). (Darwin, 1859). La competencia y la medida del *fitness* son las dos piedras angulares del progreso evolutivo; las fuerzas para la selección identificadas por Darwin son los caracteres fenotípicos, tanto las características físicas como los factores comportamentales que afectan directamente al individuo en relación con el ambiente.

4 METODOLOGÍA

4.1 Para el Objetivo Específico 1.

i) Identificar la taxonomía de la biomimética y de aquellas áreas de estudio de la ciencia de la computación que han servido de fuente de inspiración para logros en áreas diversas de las ingenierías.
ii) Investigar tipos de productos y técnicas desarrolladas que han dado valor a soluciones de problemas reales no algorítmicos, inspirados por la biología.

4.2 Para el Objetivo Específico 2.

iii) Estudiar y sintetizar los fundamentos y propiedades, de las redes neuronales, la computación evolutiva y los métodos de hormigas además sus clases y características diferenciadoras.
iv) Investigar, listar y definir las técnicas más usadas de modelado inspiradas en la biología.

4.3 Para el Objetivo Específico 3.

v) Investigar y comprender diversas aplicaciones desde los sistemas biológicos con el uso de Heurísticas y Metaheurísticas realizadas hasta el momento en áreas de estudio de la ingeniería industrial.
vi) Comprender y describir las dificultades técnicas computacionales en la solución de modelos de optimización NP-hard (no polinomiales).

4.4 Para el Objetivo Específico 4.

vii) Seleccionar un área logística e investigar y describir en detalle una aplicación algorítmica realizada en dicha área.

5 RESULTADOS

5.1 Fundamentos de la Biología Aplicados a la Ingeniería

Objetivo Específico 1. "*Describir y analizar la taxonomía de las funcionalidades de los sistemas biológicos que han servido como inspiración para aplicaciones y soluciones diversas de ingeniería."*

El surgimiento de la nanotecnología, los microprocesadores con alta capacidad y gran velocidad, el almacenamiento eficaz de energía, los dispositivos de almacenamiento de datos, así como memorias de acceso rápido y de gran compactabilidad, la comunicación inalámbrica, han sido entre tantos desarrollos tecnológicos, imitaciones exitosas inspiradas en las capacidades de la naturaleza para resolver problemas en los sistemas vitales en la especie humana. Puesto que son soluciones efectivas, éstas involucran desarrollos de nuevos métodos en ingeniería, procesos, materiales e incluso algoritmos de solución. Dicha efectividad se ha dado durante billones de años, con procesos naturales desarrollando mecanismos de uso eficientes de recursos limitados perdurando en el tiempo. Con esta evidencia, se ha determinado la creación de la ciencia que estudia a la naturaleza desde su funcionalidad, con el fin de entender y trasladar sus mecanismos y elementos estructurales, para iniciar un proceso de creación de nuevas tecnologías, que resuelven problemas desde la misma manera en que los organismos lo han podido resolver frente a los retos que propone el entorno y la supervivencia. Esta ciencia es llamada "Biomímesis" o "Biomimética" (Biomimicry). (Bar-Cohen, 2006).

El punto de partida de la biomimética es la funcionalidad de la estructura celular dada su propiedad de crecer con tolerancia a fallos y ejecutar la auto-reparación para corregirlos (Bar-Cohen, 2006). De hecho, una de las características más importantes es aquella en la que es posible identificar la capacidad de operar autónomamente en ambientes complejos. Parte de esta capacidad es la adaptación dentro de la biología que conlleva a los individuos a copiar apariencias completas y funciones específicas de criaturas.

Las comunidades y los organismos evolucionan respondiendo a distintas necesidades requeridas en pro de su supervivencia y hacia el dominio de su entorno, todas aquellas soluciones obtenidas desde su evolución perdurarán en el tiempo puesto que su meta es establecerse en la próxima generación (Bar-cohen,

2012). Mediante la identificación de mecanismos y procesos de la naturaleza, los enfoques científicos se han dado a la tarea de comprender las causas y relaciones de los fenómenos vinculados a la evolución, además de sus principios fundamentales, con el fin de diseñar y operar dispositivos con mayor y mejor funcionalidad.

5.1.1 Origen y Taxonomía de la Biomimética

En sus orígenes, el término toma auge a principios de los años noventa, con el desarrollo de la robótica y la investigación en nuevos materiales inspirados desde la biología. Sin embargo, éste fue acuñado a finales de los años 60s por Otto H. Schmitt, solo unos años después que surgiera el concepto de biónica. Si bien, siempre se vincula a la biología con la ingeniería por el desarrollo tecnológico, la biomímesis comprende la imitación de métodos, diseños y procesos en cualquier orden de la vida. Incluso, las primeras aproximaciones entre la tecnología y la biología, se dan en una glosa común para procesos similares: hibernar, cuarentena, virus, ciclo de vida, fatiga y otras tantas.

Investigar sobre biomímesis, es explorar un vasto mundo de publicaciones que reseñan sus orígenes, sus vínculos con la tecnología, sus relaciones con campos específicos de la ingeniería, sus posibilidades de reintegrar la vida humana con la biosfera, y descripciones futuristas de potencialidades con la bio-nanotecnología. Como punto de partida, es importante establecer una distinción que Yoseph Bar-Cohen resalta en su libro "*Biomimetics, biologically inspired technologies*": imitar e inspirar. Existen creaciones cuyas funcionalidades son inspiradas desde la biología -las celdas solares respecto a la fotosíntesis-, y otras que son imitaciones de diseños o estructuras de sus equivalentes en la naturaleza, por ejemplo las aletas submarinas, los módulos espaciales con formas hexagonales de panal de abejas, y proteínas a partir de aminoácidos sintéticos.

Lógica Taxonómica.

Frente a un campo gigantesco de aplicaciones, la mayoría de investigadores y autores hacen despliegue metodológico para describir avances tecnológicos a partir de la biomímesis pero poco se ha publicado sobre un criterio lógico que determine la taxonomía de esta ciencia. Asknature es una organización que ha estructurado una biblioteca virtual que reúne un catálogo de libre acceso con centenares de casos inspirados en la biología (www.asknature.org), e información

de casi dos mil fenómenos naturales. Su propósito es servir de plataforma para todo público en general, no solo investigadores o diseñadores de la biomímesis, en búsqueda de referencias de aplicaciones inspiradas desde la biología y explicaciones de fenómenos naturales. Su lógica taxonómica (subjetiva) es un esfuerzo de biólogos, quienes parten de una razón funcional basada en las formas como los organismos y los sistemas naturales direccionan sus desafíos, y para ello clasifica según las estrategias (cómo) y las funciones (por qué) con que el organismo dispone su realización o comportamiento. Estas estrategias adaptativas acerca del “cómo” y la función específica desde su especialización anatómica acerca del “por qué”, proporciona una manera de aproximarse al reto taxonómico a nivel de grupos (8), subgrupos (30) y en el nivel inferior, las estrategias empleadas (más de 1600).

En la **Tabla 1**, se exhibe un ejemplo que estructura la función específica y por ende el grupo o categoría a la que pertenece la función, para el caso de un escarabajo que requiere de un mecanismo de captura de agua en medio del desierto, logrado con el comportamiento de sus alas.

Tabla 1. Clasificación en la Taxonomía de la Biomimética del escarabajo de Namibia.

Organismo	*¿Qué organismo es?*	Escarabajo de Namibia, vive en el desierto.
Reto	*¿Qué reto debe afrontar este escarabajo?*	Capturar agua en un clima muy árido
Estrategia	*¿Cómo los escarabajos afrontan este reto?*	La cubierta de las alas del escarabajo recoje el agua del aire a traves de golpes a nano escala.
Función	*¿Por qué el escarabajo necesita de esta estratégia?*	Para la captura de líquido. Esto está representado en la taxonomía biomimética así: **Grupo:** Get, store, or distribute resources **Sub-Grupo**: Capture, absorb, or filter **Función**: Liquids

Figura 4. Escarabajo de Namibia.

En este ejemplo, el grupo de su categoría es la función de conseguir, almacenar o distribuir recursos (agua), dentro del subgrupo específico de capturar, absorber y filtrar los recursos, y el elemento clave: líquidos, con la estrategia de uso nanoescalar del movimiento de sus alas. Como se puede apreciar, la lógica taxonómica no surge del campo aplicado sino del propósito de la fuente de inspiración y su estrategia diseñada.

Esta lógica planteada por el equipo de asknature, si bien es subjetiva, usa simplicidad en los términos que definen los grupos y los subgrupos, dado que pueden ser de uso común para cualquier disciplina y tan solo es asociarlo con la necesidad exacta del diseñador que explora la biología como fuente de inspiración. Hay otros esquemas desarrollados por Glier et al. 2011; Vattam et al. 2010; Vincent et al. 2006; Yen et al. 2011, que han probado sus impactos en los usuarios, pero en publicaciones accesibles solo con pago previo. La proyección del portal asknature.org se acercó a los 500.000 visitantes anuales y 2 millones de visitas para el año 2013, lo que podría considerarse como un paso hacia la validez de su base de datos y el impacto en la comunidad del diseño bio-inspirado.

De acuerdo con todo lo anterior, a continuación se presenta la **Tabla 2**, estructurada por *asknature,* donde las categorías están descritas con palabras referentes a funciones, equivalentes a las necesidades del diseñador que busca inspiración en la biología, pero donde se omite el nivel jerárquico inferior de las estrategias.

Tabla 2. Estructura taxonómica de la biomímesis. (Deldin & Schuknecht, 2014)

Grupo	**SubGrupo**	**Función**
Mover o Retener	Adjuntar o Atar	Permanentemente Temporalmente
	Mover	En/sobre Líquidos En/sobre Sólidos En gases
Mantener Integridad Física	Proteger de los factores bióticos	Animales Plantas Hongos Microbios
	Proteger de los factores abióticos	Exceso de líquido/ Pérdida de Líquido Viento Gases/ Pérdida de gases Suciedad/sólidos Luz / Químicos / Fuego /Hielo Temperatura Radiación Nuclear
	Manejar las fuerzas estructurales	Corte Choque térmico Impacto Tensión Turbulencia

		Daño mecánico Desgaste químico Arrastre Compresión.
	Regulas procesos fisiológicamente	Procesos Celulares Homeostasis Reproducción o Crecimiento
	Prevenir fallos en estructuras	Pandeo Deformación Fatiga Fusión
Conservar la comunidad	Coordinar	Grupos (autoorganización) Actividades Sistémas
	Cooperar y competir	Dentro de las mismas especies Entre diferentes especies Dentro de un (eco)sistema Entre ecosistemas
	Proveer servicios ecosistémicos	Regular la respuesta del hábitat a la perturbación Regular los flujos hídricos Polinizar Generar suelo/renovar fertilidad Desintoxicación/purificación de aire/agua/residuos Control de erosión y sedimentos Regular el almacenamiento de agua Ciclo de nutrientes Regular la composición atmosférica Regular el Clima Dispersión de semillas Mantener biodiversidad Control biológico de poblaciones, pestes, enfermedades.
Modificar	Modificar el estado físico	Tamaño/forma/masa/volumen Presión Densidad Fase Flotabilidad Luz/Color Características Materiales Cantidad Velocidad Posición

	Modificar el estado químico/eléctrico	Estado energético Reactividad de radicales libres Concentración Potencial químico Reactividad con el agua Estado de oxidación Carga eléctrica Conductividad Tensión superficial pH Solubilidad Transporte de electrones Generación química de flujo de electrones
	Adaptar/optimizar	Adaptar el genotipo Adaptar el fenotipo Coevolución Comportamientos adaptativos Optimizar espacio/materiales
Hacer	Reproducir	Autoreplicación
	Montaje físico	Estructura
	Generar/convertir energía	Energía eléctrica Energía magnética Energía química Energía mecánica Energía térmica Radiación (luz)
	Montar químicamente	Polímeros Compuestos a base de metal Esteroisómeros específicos Cristales minerales Bajo demanda Compuestos inorgánicos Compuestos orgánicos Ataque a un grupo funcional Separar un grupo funcional Catalizar reacciones químicas Dispositivos moleculares
Procesar Información	Navegar	A través del aire A través del agua A través de la tierra A través de sólidos
	Envío de señales	Luz (espectro visible) Luz (espectro no visible)

		Sonido Táctil Químico (odor, sabor, etc) Vibraciones Eléctrico/magnético
	Procesamiento de señales	Diferentes señales desde el ruido Trasducción/convertir señales Respuesta a señales
	Señales sensoriales/ Señales ambientales	Luz (espectro visible) Luz (espectro no visible) Electricidad/magnetismo Tocar y fuerzas mecánicas Químicos (odor, sabor, etc) Condiciones atmosféricas Sonidos y otras vibraciones Temperatura Enfermedad Movimiento Dolor Balance/gravedad/orientación Forma y patrón Tiempo y duración del día Conciencia corporal
	Calcular Aprender Codificar Decodifcar	
Descomposición	Descomposición Química	Metales pesados escinden a partir de compuestos orgánicos Halógenos escinden a partir de compuestos orgánicos Otros componentes inorgánicos Polímeros Otros componentes orgánicos Catalizar reacciones químicas
	Descomposición Física	Materiales abióticos Materiales bióticos
Conseguir, Almacenar o Distribuir Recursos	Capturar, absorber o filtrar	Organismos Partículas sólidas Solidos a granel Gases Líquidos Energía Entidades químicas.

	Almacena	Sólidos a Granel Gases Entidades químicas Partículas sólidas Energía Líquidos
	Distribuye	Sólidos Líquidos Gases Energía
	Expulsar	Solidos Líquidos Gases

5.1.2 Campos Tecnológicos con Diseño Bio-Inspirado

Con el propósito de ilustrar los principales campos en la ingeniería de diseños bio-inspirados, se resumen y describen algunos de ellos, que según Cohen en su libro *Biomimetics, biologically inspired technologies*", subrayan los avances tecnológicos más sobresalientes con diseños inspirados desde la biomímesis. El directamente relacionado con el objetivo de este proyecto, procesar información, será analizado en detalle en el siguiente capítulo.

Vida Artificial (*Artificial Life/ A-Life*)

En el desarrollo del concepto de la vida artificial, ésta sugiere la síntesis de la vida a partir de componentes no vivientes. Este campo, consiste en una amplia gama de temas relacionados con la síntesis y simulación de sistemas vivos en forma de código computacional, auto-replicables. Permite aprender acerca de los aspectos fundamentales del proceso evolutivo tanto en su impacto como en su contexto ecológico. Cubre la simulación o la emulación de los sistemas vivos o parte de ellos, con la intensión de comprender su comportamiento y otras propiedades emergentes de las poblaciones: tamaño (número de organismos que componen la población), densidad (número de organismos por unidad de área), patrón de distribución, tasa de crecimiento poblacional, estructura poblacional, etc.

La *A-life* es comparable en su funcionamiento con la programación genética, que consiste en un grupo de ecuaciones y operaciones donde un *software* computacional evalúa la capacidad del programa de resolver un problema

específico, y las cadenas o líneas de código que no logran su propósito son eliminadas, con las cadenas restantes se permite la creación de nuevas fuentes de código a través del proceso de recombinación, o preferiblemente utilizando la mutación –un cambio puntual en el código-. Por otro lado, al igual que las soluciones producidas en la programación genética, están las generadas por la programación evolutiva que emulan las soluciones en el mundo real para el problema que se está resolviendo (Bar-Cohen, 2006).

Inteligencia Artificial (IA)

La inteligencia artificial es una rama de la ciencia computacional que estudia los requerimientos necesarios para tareas como la percepción, el razonamiento y el aprendizaje (Siang, 2013), contiene un número creciente de técnicas entre las cuales se encuentran las redes neuronales artificiales, los sistemas expertos, la lógica difusa y los algoritmos genéticos.

El cerebro es un increíble controlador en los sistemas biológicos y a la medida que se entiende su funcionamiento, se busca su imitación. Los primeros desarrolladores de la automatización – entendida como la aplicación de máquinas o de procedimientos automáticos en la realización de un proceso o en una industria-- se ayudaron en la utilización de *software* para controlar sus sistemas, además de la inclusión de funciones para la acción y reacción en ellos, utilizando algoritmos de inteligencia artificial como la captura de conocimientos, la representación y el razonamiento, la planificación, el razonamiento bajo incertidumbre, la visión sensorial, el seguimiento de la función del rostro, el procesamiento del lenguaje natural, la cartografía y la navegación, y el aprendizaje de máquina. Se observa IA en motores de búsqueda web, captura de datos, la representación del razonamiento, la planificación, la visión, la robótica, el procesamiento del lenguaje natural y el aprendizaje automático. Finalmente los modelos cognitivos desarrollados desde de la Inteligencia Artificial, sugieren principios para el apoyo efectivo desde modelos educativos en el aprendizaje humano.

Mecanismos biológicos como modelos de mimetismo

Desde el siglo XV se ha realizado un esfuerzo por copiar aspectos de la locomoción de los animales, muchos de los diseños desarrollados no obtuvieron

un logro considerable respecto a la agilidad y el rendimiento, debido a que fueron utilizados materiales con propiedades muy diferentes en comparación con los materiales biológicos. Solo recientemente y gracias al desarrollo de polímeros blandos y flexibles, ha sido posible imitar la actuación locomotriz biológica. Como lo anterior, el reino animal expone ejemplos de inspiraciones, al igual que los sistemas naturales estructurados como una montaña, meseta o valle, impredecibles como una isla volcánica, en constantes cambios como un arrecife de coral. Los diseños naturales se basan en su historia evolutiva, en el cambio que han experimentado en toda su historia, y es allí donde pueden mostrarse restricciones considerables, el proceso de diseño natural es considerablemente bueno. El punto de partida para cualquier diseño biomimético deberá ser la función por la cual este será emulado, por ejemplo, para la copia de las extremidades inferiores de un robot biomimético sería interesante emular las características elásticas de masa y péndulo que son exploradas por los animales respecto a las extremidades y su función.

La naturaleza como modelo para estructuras y herramientas

Las criaturas biológicas construyen formas y estructuras sorprendentes utilizando como base las formas, estructuras y materiales que el entorno produce. Las formas y estructuras a menudo tienen un tamaño significativamente mayor que la especie que lo construyó. Por ejemplo, el castor construye represas como parte de su hábitat en ríos y arroyos.

El uso de reglas basadas en la biología permite la fabricación de dispositivos e instrumentos que son fáciles de usar obteniendo la manera de operarlos con el menor número de instrucciones. La construcción de estructuras a partir de las células adopta características que ofrecen ventajas, como la capacidad de crecer con tolerancia a fallos, la auto-replicación, la termorregulación y la especialización en diferentes ambientes. Los avances en la tecnología están permitiendo la fabricación de células artificiales de igual modo que es apreciable que la mayoría de las piezas y estructuras que utilizamos hoy tienen un modelo biológico de inspiración. Estas características de inspiración biológica son empleadas mediante el uso del auto-ensamblaje hacia el desarrollo de herramientas de ingeniería para producir específicamente estructuras, dispositivos y sistemas (*hardware* y *software*). Se espera que estos métodos permitan la integración, desde la fabricación de diferentes dispositivos en un solo sistema, uno de los principales desafíos para la manufactura en este siglo.

Bio-sensores

Los seres vivos están equipados con un sistema sensorial que permite la traducción de señales proveniente del ambiente en una acción motora, ésta recepción de señales es captada por un sistema periférico neuronal, luego se hace una integración de la señal por el sistema nervioso central, y finalmente se transmite de vuelta hacia la acción motora. En la mayoría de los animales el sistema nervioso central se encuentra localizado desde el cerebro pasando por la médula espinal, de ahí en adelante se encuentra bifurcándose por cada parte del cuerpo el sistema nervioso periférico, finalmente son los músculos quienes reciben la orden de generar la acción motora a partir del análisis de la información recibida. (Solomon, Berg, & Martin, 2008).

Los sistemas biológicos sensoriales son extremadamente sensibles. Toda la red sensorial, es decir, los mecanoreceptores como la piel, los electroreceptores que ilustran a la recepción eléctrica de los tiburones, propioceptores que permiten a un animal percibir la orientación de su cuerpo, quimiorreceptores como la lengua y finalmente los fotoreceptores como el ojo, son imitados cada vez más, permeando un entorno común lleno de sensores. Mediciones de la presión y la temperatura con sensores ópticos, acústicos y sensoriales son útiles para mejorar la capacidad de detección y reducir el tamaño y potencial requerido. También incluye la adaptación de los principios oculares y de visión a cámaras, de los bigotes de roedores a los sensores de prevención de colisiones y de los murciélagos a los detectores acústicos que imitan su sonido.

Interconexión para la Biología y las máquinas

La interconexión entre los animales y los seres humanos con las máquinas para complementar o sustituir nuestros sentidos biológicos tiene gran importancia en las aplicaciones médicas. Existe una importante conexión entre las máquinas y el cerebro humano. Se han desarrollado chips que pueden reconocer las señales del cerebro para el movimiento y convertirlos en acciones. Los avances en este campo han logrado que la “Food and Drug Administration” (FDA) de los Estados Unidos, apruebe de manera limitada, la realización de tales experimentos en seres humanos con el objetivo a corto plazo de desarrollar prótesis neuronales controladas. Los chips actuales duran hasta un año y se hacen esfuerzos permanentes para desarrollar una capacidad inalámbrica de mayor duración. Los logros en estos estudios superarían los problemas de las discapacidades.

Para ilustrar el tipo de investigación en esta rama, está el trabajo de científicos de la Universidad de Duke, quienes experimentan con electrodos conectados al cerebro de un mono, y con el uso de ondas cerebrales permite al mono operar un brazo robótico a nivel local y de forma remota a través de la Internet. En este sentido, los investigadores buscan crear implantes que ayuden a las personas con problemas de visión a recuperar esta capacidad. Todavía se presentan errores de reconocimiento a niveles inaceptables.

Materiales y Procesos en la biología.

El cuerpo de cualquier ser vivo es un laboratorio químico que procesa los productos adquiridos de la naturaleza y los convierte en: energía, materiales de construcción, residuos, y varias estructuras multifuncionales. Los materiales provenientes de la naturaleza han sido bien reconocidos y caracterizados por los seres humanos como fuentes de alimento, vestido, comodidad, y así sucesivamente. Estos incluyen entre otros, pieles, cuero, miel, cera, leche, y la seda. A pesar que algunas de las criaturas e insectos productores de materiales son relativamente pequeños, pueden producir cantidades de materiales que son suficientes para satisfacer el consumo humano a una escala de producción en masa (por ejemplo, miel, seda y lana). El uso de materiales naturales se remonta a miles de años. La seda, que se produce para proteger el capullo de la polilla de seda, tiene grandes propiedades que incluyen la belleza, la fuerza y durabilidad. Algunas de las capacidades fascinantes de los materiales naturales incluyen la auto-sanación, auto-replicación, reconfiguración, equilibrio químico, y la multifuncionalidad.

Muchos materiales hechos por el hombre son procesados a través de calentamiento y presurización (uso de presión en el proceso), en contraste con la naturaleza que utiliza las condiciones ambientales en la producción de los materiales. Materiales como el hueso, el colágeno, o la seda, son procesados y realizados dentro del cuerpo del organismo sin el duro proceso que se utiliza para hacer materiales sintéticos o hechos por el hombre. La fabricación de materiales derivados biológicamente produce una mínima cantidad de residuos y no hay impacto al ambiente por contaminación, donde el resultado es biodegradable, y puede ser reciclado por la naturaleza. Aprender a procesar estos materiales permitiría hacer unas mejores opciones de material, mejorando nuestra capacidad de crear materiales reciclables, que pueden proteger mejor el medio ambiente y disminuir el impacto por contaminación. También existen estudios hacia el

mejoramiento de las prótesis, las cuales incluyen las caderas, dientes, soporte estructural de los huesos, y otros.

Beneficios en el estudio de la biomimética pueden verse en muchas aplicaciones, incluyendo la fibra más fuerte, los materiales multifuncionales, el mejoramiento de los medicamentos, los robots superiores, y muchos otros. Otro aspecto de la biomimética es el reconocimiento de la importancia de proteger las especies en peligro de extinción, con el fin de no perder las soluciones de la naturaleza que han logrado sobrevivir, pero que todavía no se han estudiado o no se entienden.

5.1.3 Ilustración de un caso aplicado: el pez globo ((Tinsley, Midha, Nagel, & McAdams, 2007)

A continuación se describe la metáfora de un diseño inspirado desde la biología para la solución de un problema de seguridad en los autos, lo cual dentro de la taxonomía vista sería una inspiración del grupo de mantener integridad física. La descripción del proceso natural versus la adaptación a la solución de tecnología, ilustra el proceso de búsqueda de solución a partir de la necesidad de una función en un contexto de problema de “parada funcional”.

El pez globo es conocido por su habilidad de absorber agua o aire para inflar su estómago con el fin de asustar y ahuyentar a sus predadores cuando se siente amenazado. El pez hace crecer su estómago muchas veces en relación a su tamaño normal, y hace parar al depredador o retroceder y obtiene como resultado su huida. Cuando el pez no se siente amenazado, el agua o aire es expulsado del pez a través de su boca. Aplicando la concepción de proceso con entradas y salidas, se tiene que el mecanismo de defensa del pez globo tiene una funcionalidad de “parada” (**Figura 5**. **Modelo general del Pez Globo. (Tinsley et al., 2007)Figura 5**), requiere un enemigo en asocio al miedo y demanda aire, agua, epidermis (piel) y la energía biológica para expandirlo. La comparación biomimética está en la bolsa de aire o “airbag” de un automóvil, en esta se permite ver la misma función de “parada” (**Figura 6**) que en el pez globo, donde el impacto (miedo) se materializa en el peligro que amenaza la protección de la cabeza del pasajero (enemigo), y requiere de energías químicas que permitan llenar la bolsa (piel).

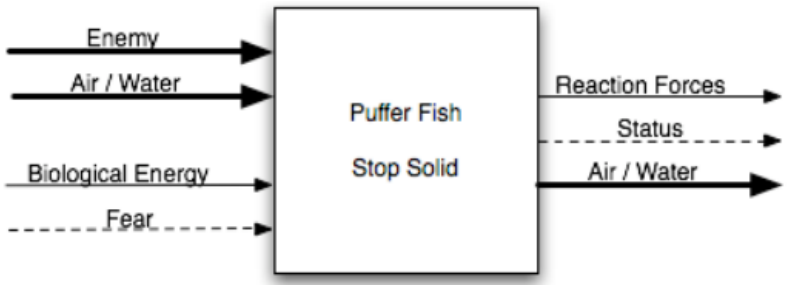


Figura 5. Modelo general del Pez Globo. (Tinsley et al., 2007)

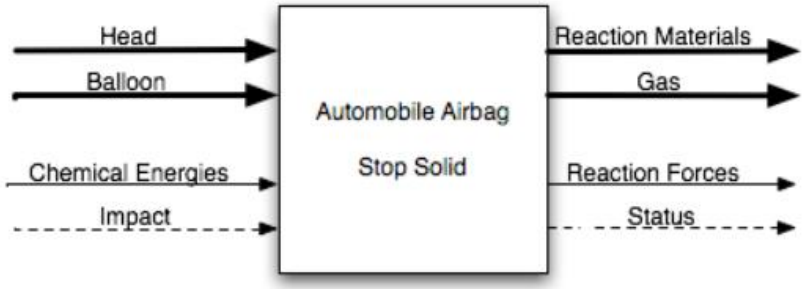


Figura 6. Modelo general de la bolsa de aire en un Automóvil.(Tinsley et al., 2007)

Siendo análogas las entradas, el proceso requiere de un mecanismo activador, es decir, un control de señales a través del estímulo del miedo a partir de un enemigo, este estímulo activa su mecanismo de defensa. El control de señales es enviado al cerebro donde se procesa y se envía como una señal a los músculos. La energía biológica es transferida a los músculos del pez, donde es convertida en energía mecánica, finalmente el aire o agua son importados al estómago rellenándolo y la epidermis detiene el llenado apartando al enemigo. En este sentido, la bolsa de aire de los automóviles están diseñadas para evitar que los pasajeros y el conductor se golpeen con el habitáculo del vehículo en caso de una fuerte colisión. En el momento del choque, éstas se activaran automáticamente, puesto que cuenta con unos sensores y actuadores automáticos que se activan de inmediato cuando se cumplen las siguientes dos condiciones de forma simultánea: una desaceleración brusca en un segundo de más de 16 kilómetros por hora y un impacto de frente con un elemento contundente. La bolsa es hecha de una fibra de nylon delgada y una vez se activa inicia el llenado con aire de la bolsa. Después que la bolsa ha sido inflada, el gas y las partículas de polvo comienzan a escaparse a través de los respiraderos ubicados en los costados de ellas.

A grandes rasgos, los dos tipos de modelos funcionales, el del pez globo y la bolsa de aire del automóvil tienen la misma funcionalidad en el momento de parada ante un sólido (**Figura 7**). Sin embargo, cada ejecución en el subnivel funcional tiene una meta diferente. El pez globo requiere energía biológica, en cambio la bolsa de aire, siendo un sistema de ingeniería convierte la energía química de un gas, en energía mecánica para inflar.

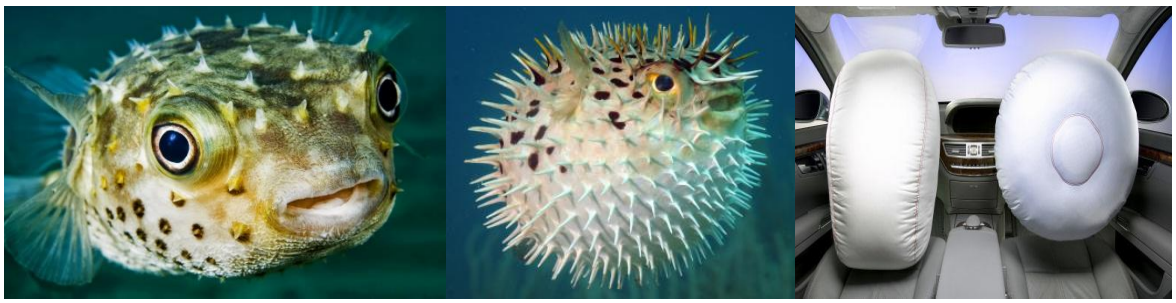


Figura 7. Bolsa de aire y su inspiración del pez Globo.

5.1.4 Conclusión de la Biomímesis y su taxonomía.

Entrando a una era de estímulo para la innovación y la exploración de soluciones tecnológicas creativas, la biomímesis se ha constituido en una fuente enorme de inspiración para la creación de nuevos diseños en productos y procesos con necesidades similares a las que organismos y sistemas naturales se han enfrentado en su adaptación a un entorno. Esfuerzos como el de Asknature.org de crear desde el año 2008 una base de datos organizada con una lógica taxonómica desde la función de adaptación del organismo o sistema natural, permite obtener claridad y rápido acceso, a la identificación del sistema análogo para cualquier diseñador, ofreciendo un catálogo de estrategias que determinan el cómo se logra el fin o solución.

Si bien está por probar el nivel de impacto de la taxonomía en los usuarios, su popularidad marca un éxito en el esfuerzo esquemático por hallar las líneas funcionales comunes y sus niveles jerárquicos inferiores que estructuren la taxonomía. De esta manera, se halla el camino conducente a la bio-inspiración algorítmica desde la función de “proceso de la información”, y el subgrupo de codificar y decodificar. La **Figura 8**. muestra la ramificación exclusiva de los algoritmos inspirados desde la biología en relación a la taxonomía de la biomímesis propuesta por asknature.org.

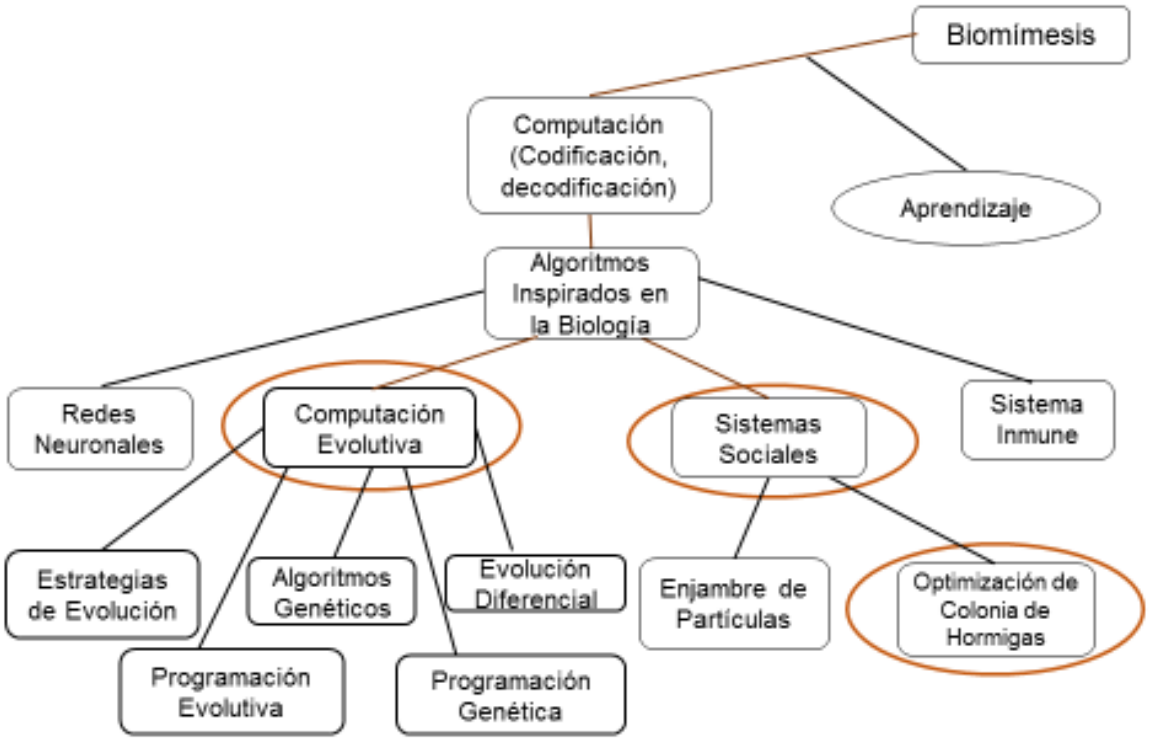


Figura 8. Muestra de la taxonomía referente a los algoritmos inspirados en la Biología. (Adaptado de: (Brabazon et al., 2006))

5.2 Metáfora de los Algoritmos Inspirados Biológicamente.

Objetivo específico 2. "*Identificar y describir las funcionalidades algorítmicas inspiradas desde la biología en el marco de la ciencia de la computación y en relación a necesidades de optimización.*"

Con el acrónimo de BIAs (*Bio-Inspired Algorithms*) se denomina a todos los desarrollos algorítmicos inspirados en la biología. Como bien se plantea en la taxonomía extendida de la biomímesis de la Figura 3, se pueden ramificar 4 tipos de algoritmos inspirados en Biología: redes neuronales, todos los algoritmos asociados a computación evolutiva, los referentes a sistemas sociales inteligentes (enjambres) y los sistemas inmunes. Incluso, se podría expandir una rama adicional de ecología, la cual se explicará más adelante. La idea principal es describir la metáfora biológica sin entrar en definiciones o representaciones matemáticas implícitas en los algoritmos, lo cual implicaría un trabajo de investigación por cada uno. A la fecha son muchas las publicaciones y las aplicaciones que se han desarrollado y aún hay un estado de arte por seguir definiendo a la medida que se establecen clases y subclases de problemas abordados por este tipo de metaheuristicas.

No todos los BIAs son orientados a la optimización. En el caso de redes neuronales, además de ser usadas en la optimización, también apunta a modelos de predicción en aprendizaje de máquinas (sistemas que aprenden de los datos), mientras que las demás áreas competen con la optimización. En las redes neuronales se imita la conexión entre neuronas (nodos) de entrada y salida (*inputs* y *ouputs*) intercambiándose mensajes entre sí que son ponderados y se transforman por una función que los conduce a otro nodos hasta uno final que estima el resultado de la función (**Figura 9**). Muy útiles en una amplia variedad de tareas, incluyendo la construcción de modelos en predicciones, clasificación y análisis de conglomerados de datos (Clustering). (Fogel & Corne, 2002)

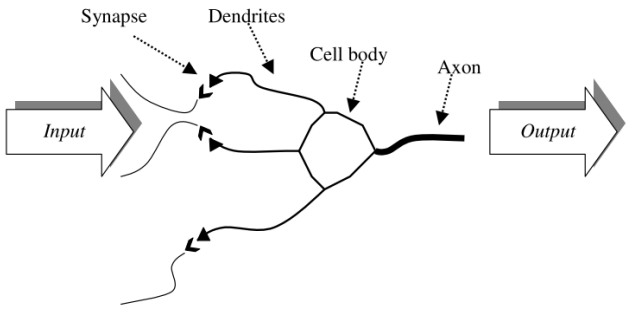


Figura 9. Diagrama de una célula nerviosa. (Brabazon et al., 2006)

Las redes neuronales artificiales comprenden una familia de metodologías de modelado matemático cuya inspiración metafórica ha sido poca, pero extrae y aplica en su mayoría los funcionamientos del cerebro humano y del sistema nervioso central (SNC).

5.2.1 Tipología Básica de los BIAs en Optimización.

Los algoritmos de optimización pueden ser determinísticos o estocásticos en su naturaleza. En el caso de los BIAs, estos se encargan de buscar caminos aleatorios de poblaciones de soluciones y ejercen como alternativas a los métodos exactos que resuelven problemas clásicos, y se determinan con una conveniente representación del problema para luego evaluar la calidad de la solución a través de una función de *fitness*, y generar nuevas soluciones a través de operadores definidos. Los dos grandes campos de BIAs en optimización se pueden considerar como los Algoritmos Evolutivos y los Algoritmos basados en Enjambres. Los primeros hacen una metáfora del proceso evolutivo mientras que los segundos, es a partir del comportamiento de sociedades de animales.

5.2.2 Computación Evolutiva

La teoría de la selección natural de Darwin explica como generaciones de especies han sobrevivido con éxito como resultado del proceso evolutivo. La evolución muestra mecanismos de diversificación de la vida en nuestro planeta y de igual manera, la computación evolutiva aplica principios de búsqueda heurística, inspirados por la evolución natural hacia una variedad de dominios diferentes (diversidad), dándole importancia a la optimización paramétrica y también a otros tipos de resolución de problemas tradicionalmente considerados. (Banzhaf, 2012). El objetivo de estos algoritmos está en la noción de encontrar un comportamiento inteligente en donde exista un tipo de aleatoriedad -que más adelante se aclarará como mutación y recombinación- ligada al proceso de investigación y búsqueda de la soluciones a problemas desde la perspectiva evolutiva, y cómo se aplican estos elementos sobre la naturaleza.

Historia de la Computación Evolutiva.

La idea de aplicar los principios de la teoría de Darwin para la solución de problemas se remonta a la década de 1940, cuando los computadores apenas empezaban a surgir. Alan Turing fue uno de los primeros autores que correctamente identificó el poder de la evolución en la resolución de problemas, propuso la “genetical and evolutionary search” (a. E. Eiben & Smith, 2003) en los años 50 a través de un ensayo llamado “Computing Machinery and intelligence” (Banzhaf, 2012) exhibiendo en todo su contenido a la inteligencia del comportamiento, que considera que las máquinas podrían pensar. A pesar que las computadoras no eran suficientes - para esta época - en producir un comportamiento inteligente, Turing señaló que una computadora digital con un elemento aleatorio debía ser una variante interesante por la cual existiera la posibilidad de encontrar solución a algunos problemas. “El proceso de aprendizaje puede ser considerado como una búsqueda” (a. E. Eiben & Smith, 2003), es aquí donde Turing aclara que las computadoras pueden aprender de manera similar a los niños, dado que es muy probable que exista un gran número de soluciones satisfactorias durante este proceso de aprendizaje, este método aleatorio de encontrar soluciones muestra que es mucho mejor que el sistemático (en donde se trata de encontrar una solución, sin conocer su grado de impacto ya sea positivo o negativo) y es análogo al proceso de evolución. El objetivo de estos algoritmos está en la noción de encontrar un comportamiento inteligente en donde exista un tipo de aleatoriedad -que más adelante se aclarará como mutación y recombinación- ligada al proceso de investigación y búsqueda de la soluciones a

problemas desde la perspectiva evolutiva, y cómo se aplican estos elementos sobre la naturaleza.

En 1962, John Holland describió que el estudio de la adaptación envuelve tanto el sistema como el proceso de adaptación de este a su ambiente. Nombró la función de aptitud (*fitness function*) como un sustituto para el ambiente y creó el concepto de población de programas generada que actúa sobre una población de problemas (el ambiente) intentando producir soluciones de manera eficiente (Banzhaf, 2012). Según Holland, debe existir una adaptación a las diferentes condiciones del ambiente (población de problemas), y una supervisión continúa de estos sistemas, con el fin de observar la población de programas exitosos que tengan la habilidad de producir soluciones óptimas. Los programas que son predominantes dentro de la población inicial, deben convertirse en una nueva población y ahí subyace la aplicación del principio de la selección natural que depende directamente de una generación continua de nuevas variedades de programas a través de los procesos de mutación y recombinación.

Mientras tanto, Bremermann había ejecutado experimentos en "*optimization through evolution and recombination*", en Estados Unidos se hacía una introducción a la *"programación evolutiva"* y en Alemania, dos investigadores, Rechenberg & Schwefel, inventaron las *"Estrategias de Evolución"*. Fue hasta el año de 1990 donde se observó que estas áreas se habían desarrollado individualmente, pero que hacían parte de una disciplina llamada ***Computación Evolutiva***, compuesta de *Algoritmos Evolutivos* en relación a la programación evolutiva, estrategias evolutivas, algoritmos genéticos y la programación genética. (A. E. Eiben & Smith, 2003).

Algoritmo Evolutivo

En la caracterización del algoritmo evolutivo, se encuentra la relación que existe entre la teoría de la evolución desde la ciencia biológica con la ciencia computacional; es aquí donde se muestra la relevancia en la metáfora entre las dos ciencias. En un Algoritmo Evolutivo hay una población de soluciones individuales, generalmente inicializada como una población aleatoria. El siguiente paso está en la determinación del fitness (aptitud) dentro del proceso de evaluación, ésta determinación puede llevar algunos cálculos para llegar al tamaño relativo de una solución individual. El resultado de este cálculo es usado luego para determinar cuál solución individual ha sobrevivido en la competencia (de soluciones) por recursos, el último paso es aplicar la variación a estas soluciones individuales hasta que la población esté completa y lista para la

siguiente ronda de evaluación. En la (**Figura 10**) se muestra el proceso cíclico del Algoritmo Evolutivo.

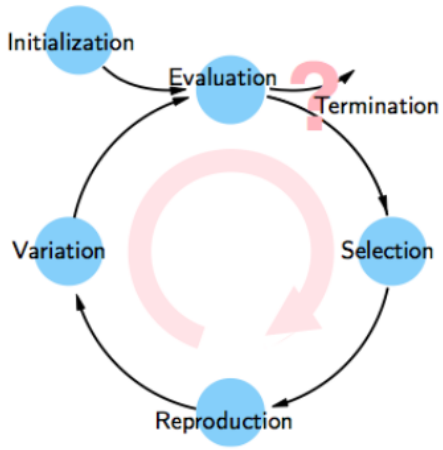


Figura 10. Proceso general de un algoritmo genético. (Banzhaf, 2012)

El Algoritmo Evolutivo tiene en cuenta el *fitness* para que algunas de las mejores soluciones candidatas sean elegidas para engendrar la siguiente generación mediante la aplicación de la recombinación y/o mutación. La recombinación es un operador aplicado a dos o más candidatos seleccionados, que también son llamados parentales (*parents*), resultando en uno o más candidatos nuevos, llamados hijos (*offspring*). Entre tanto, la mutación es aplicada a un candidato y resulta en un nuevo candidato. Al ejecutar la recombinación y la mutación es posible tener nuevos candidatos que compiten por un lugar en la siguiente generación. Si se sigue aplicando este proceso, iterándolo, se llegará a encontrar un candidato con una calidad suficiente para ser reconocido como solución, así que, los operadores de variación (la recombinación y la mutación) crean la diversidad necesaria, mientras que la selección actúa como una fuerza que presiona para obtener la calidad. (A. E. Eiben & Smith, 2003).

Etapas de un Algoritmo Evolutivo

Inicialización

Teniendo en cuenta que, la evolución es un proceso que opera y se desarrolla sobre poblaciones, los mecanismos de selección natural dentro de ella actúan indistintamente sobre las entidades (individuos) una a una. La representación, que hace parte de los mecanismos de selección natural, actúa sobre las entidades y es

partícipe en la elección de un individuo dentro de la población simulada, ésta es clave de éxito para el proceso de optimización a través de la evolución. Las representaciones son traducidas desde el espacio problema, con el fin de buscar dentro de las características (codificaciones) de tipo genético (es decir desde genotipo al fenotipo), cuál iteración puede ser usada por el mecanismo de evolución, también que incluyan una codificación hacia el comportamiento fenotípico, es decir la característica física; estas, tanto genotípicas como fenotípicas, definen un mapa de búsqueda en el espacio de todas las posibles soluciones que pueden ser encontradas por el proceso evolutivo. Algunas de las combinaciones de éstas y de los operadores de variación permiten al algoritmo evolutivo buscar y dar la mejor medida del *fitness* (adaptación). (Fogel & Corne, 2002)

Función de Evaluación de la Adaptación (*Fitness* Function)

La función de evaluación, representa en si misma los requerimientos necesarios para la adaptación de las entidades a las presiones de selección indicadas en la inicialización. Desde la perspectiva de la resolución de un problema, ésta función simboliza la tarea que fue o será resuelta en el contexto evolutivo. Técnicamente, esta función o procedimiento asigna una medida cualitativa al genotipo. Se entiende entonces que el genotipo es el código (el problema), el fenotipo es el resultado de esa evaluación genotípica, es decir, la expresión del genotipo. El problema original es resuelto por un algoritmo evolutivo y se acerca a un problema de optimización, en este caso, la función objetivo es usada como función de evaluación del *fitness*. (A. E. Eiben & Smith, 2003)

Selección.

Se requiere un método de selección con el fin remover de la población las soluciones menos apropiadas. La selección requiere un medio de puntuación para cada solución respecto a un objetivo dado por el *fitness*, y puede minimizar el número de deficiencias y desajustes al tiempo que maximiza el número de coincidencias sobre el alineamiento en torno a las soluciones dadas. El puntaje de adaptación puede ser determinado por un amplio rango de atributos, incluyendo la energía, el error cuadrado medio entre los patrones de actividades predichos y observados, o incluso una interpretación subjetiva. El fitness de cada solución debe ser un reflejo exacto del problema o de lo contrario el proceso evolutivo encontrará la solución adecuada para el problema equivocado. La probabilidad de selección varía directamente proporcional con el fitness. (Fogel & Corne, 2002)

Variación

Algunos procesos de mutación son considerados pequeños (micromutación) y otros grandes (macromutación), respecto a los cambios en el genotipo y/o fenotipo. Las micromutaciones indican una mutación puntual en un gen, la cual se puede tornar en deletérea, trayendo la muerte del individuo, es decir una mutación letal. También puede tornarse en neutral o con cambios benéficos en el comportamiento de quien se expresa. Dentro de las macromutaciones se encuentra la recombinación, que mueve grandes secciones de material genético de un lugar a otro dentro de un genoma pequeño. Analogías de cada uno de los tipos de operadores de variación son utilizadas comúnmente en los algoritmos evolutivos, donde el operador es elegido dependiendo de la representación de cada espacio de interés. La variación dentro de un padre o parental es llamada mutación, pero si es a través de múltiples parentales es llamada recombinación. Cuando se considera un espacio de búsqueda que tiene muchos puntos locales óptimos y un solo óptimo global, las mutaciones a gran escala pueden ser importantes para escapar de un óptimo local, pero no da una resolución fina para encontrar un óptimo global. Las mutaciones menos radicales pueden ser importantes para encaminar el proceso hacia el óptimo global deseado, pero puede tomar mucho tiempo en llegar a estos puntos útiles en el paisaje solución (Fogel & Corne, 2002).

Una posible solución a este dilema está en sugerir un rango de posibles mutaciones (pequeñas a grandes) para un problema dado conectando y sincronizando la tasa de mutación durante el proceso evolutivo. (Fogel & Corne, 2002)

Condición de finalización.

Se pueden distinguir dos casos: 1) Si el problema tiene un nivel óptimo de *fitness*, probablemente venga desde un óptimo conocido dado por la función objetivo, entonces para alcanzar este nivel se debe detener la condición, y 2) los algoritmos evolutivos son estocásticos, y no hay garantías para llegar a un óptimo, por lo que esta condición nunca podría conseguir satisfacerlo y el algoritmo no pararía. El criterio actual de terminaciones en el caso que exista una disyunción es el valor óptimo o condición que sea satisfecha. Si el problema no tiene una solución óptima conocida, entonces, se necesita una condición que garantice detener el algoritmo.

A continuación se presenta el esquema general de un algoritmo evolutivo como un flujograma (**Figura 11**) y en seudocódigo.

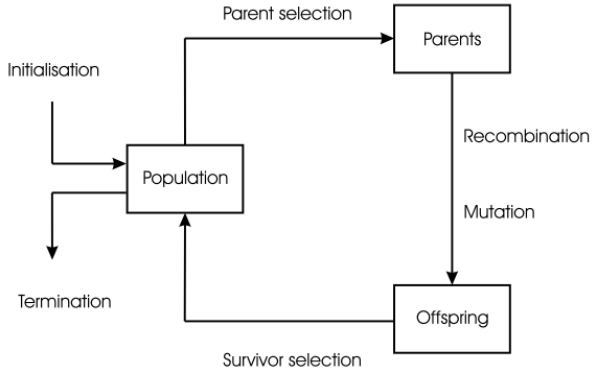


Figura 11. Esquema general de un Algoritmo Evolutivo como un flujograma.

Al observar la **Figura 12**. se puede establecer la relación entre las etapas del algoritmo evolutivo y la teoría de la Evolución de Darwin, este algoritmo recoge todo el principio y el procedimiento iterativo correspondiente al mecanismo de selección.

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

Figura 12. Esquema general de un Algoritmo Evolutivo en pseudocódigo.

5.2.2.3 Algoritmos Genéticos

Los algoritmos genéticos (GA, *Genetic Algorithm*s) están dentro de los algoritmos evolutivos más exitosos. Inicializan la población de solución (cromosomas), y comprime la representación del problema a través un vector de bits o cadenas de binarias que equivalen a las variables de decisión, y por cada cromosoma evalúa el *fitness* para que escoger aquellos que deben ser recombinados y mutados en su proceso de apareamiento. De ahí los tres operadores: selección, recombinación y mutación. Los GA son útiles cuando el espacio de solución es vasto o poco

conocido, y no hay forma matemática de reducir el espacio de búsqueda. En estos, la función de *fitness* debe ser convenientemente definida para que no caer rápidamente en óptimos locales y dados la cantidad de tiempo computacional no son recomendados en optimización restringida.

La forma canónica del algoritmo genético se puede explicar así (Brabazon et al., 2006):

i. Determinar cuál solución va a ser evaluada como una cadena de código, al igual que determinar la función de aptitud.

ii. Constituir una población inicial, con “n” individuos (codificaciones), con una disposición de aleatoriedad (al azar), preferiblemente.

iii. Decodificar (*decode*) cada cadena en una posible solución, calculando la aptitud de cada solución candidata dentro de la población.

iv. Implementar la selección en un par de individuos/codificaciones correspondientes a las soluciones candidatas dentro de la población, siendo éstas los parentales; ajustar/sesgar el proceso de selección a favor de las codificaciones correspondientes a las mejores soluciones.

v. Con una probabilidad “p” de cruzamiento, llevar a cabo el proceso de cruce en las codificaciones parentales seleccionadas, para producir dos nuevas soluciones (los hijos).

vi. Aplicar el proceso de mutación (cambio), con una probabilidad “pmut” de mutación, a cada elemento de los individuos hijos solución.

vii. Guardar las codificaciones hijas (solución) en la nueva población generada.

viii. Repetir los pasos iv al vii, hasta crear una nueva población de “n” soluciones candidatas, luego es desechar la población vieja para así constituir una nueva generación.

ix. Volver al paso iii y repetir hasta que se alcance el nivel de físico adecuado de la población deseada o hasta que hayan transcurrido un número de generaciones predeterminadas.

Programación genética.

La programación genética (GP, *Genetic Programming*) es una extensión de los algoritmos genéticos (GA), inserta dentro de la inteligencia Artificial y aprendizaje

de máquinas (*Machine learning*). Se diferencia de los GA en que es una representación principalmente en forma de árbol (también hay linealidades y grafos), codificada en un programa computacional en lugar de usar cadenas o vectores de bits. Adicionalmente, la representación es de longitud variable y no fija como en el caso de los GA, es decir, se maneja diversidad tanto en los valores de los genes como en la estructura de los individuos. Los pasos son equivalentes a los de GA pero recaen sobre los programas de computador en lugar de las cadenas de binarias propuestas por Holland, pero usando los mismos tipos de operadores. En el caso de inicialización se utilizan métodos básicos o combinados de crear poblaciones de árboles con terminales (ramificaciones) de variables y funciones representadas en nodos por operadores matemáticos que determinan las funciones a optimizar. **Figura 13**.

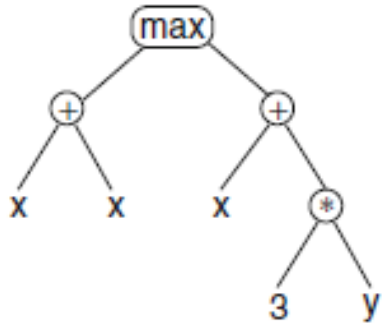


Figura 13. Representación en árbol de la sintaxis en la programación genética.(Hillier, 2010)

La recombinación y la mutación en la programación genética difieren de la mutación y la recombinación para los algoritmos evolutivos. Para la recombinación o el cruce, es usado un subárbol de cruce proveniente desde dos parentales y que permite cruzarlos aleatoriamente seleccionando un punto de cruce (*crossover point*) para dicha acción. Luego, se crean los nacimientos por el reemplazo de la rama en el subárbol del cruce **Figura 14**. (Hillier, 2010)

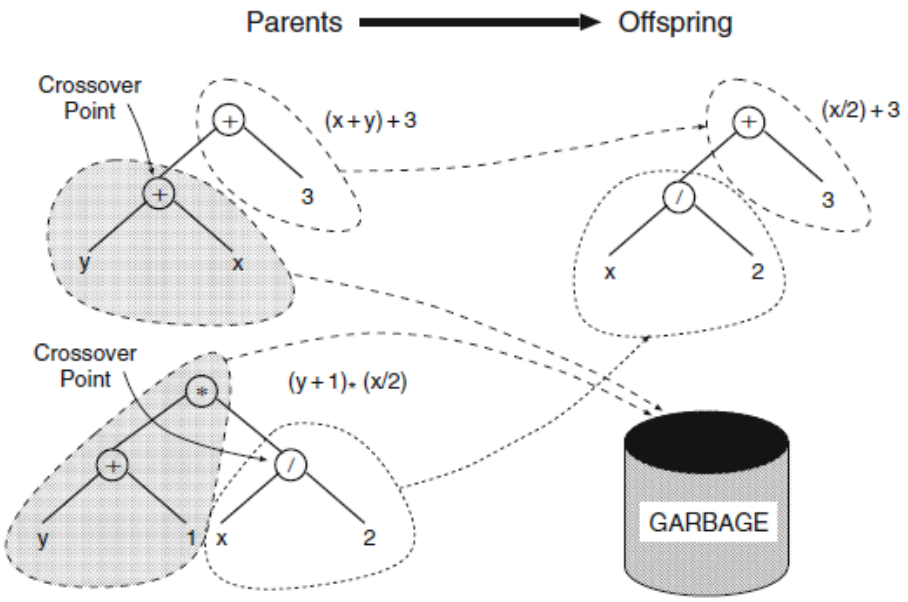


Figura 14. Ilustración de un subárbol de cruce. En la izquierda están los padres y a la derecha el nacimiento (offspring).

Para la mutación, se usa un subárbol de mutación. Esta es una selección aleatoria del punto de mutación y sustituye la rama donde ocurre la mutación y genera un nuevo subárbol. **Figura 15**

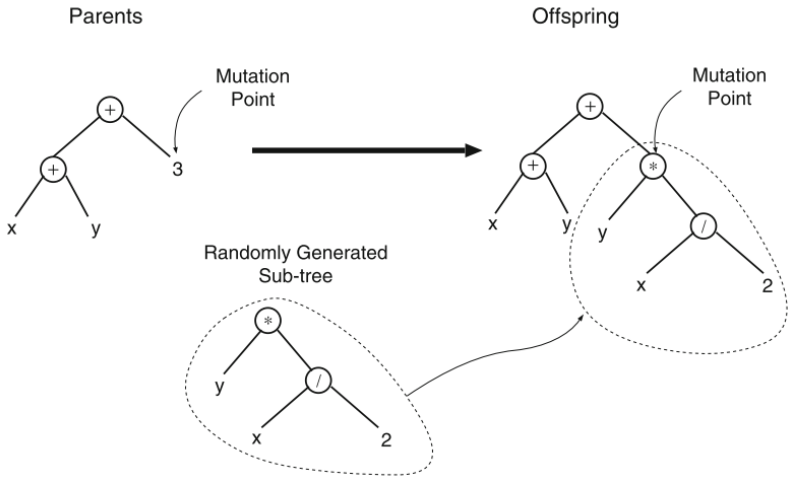


Figura 15. Ejemplo del subárbol de mutación. (Hillier, 2010)

Otros Algoritmos Evolutivos: Estrategias Evolutivas, Evolución Diferencial y Algoritmo del Campo de Arroz.

De creación simultánea a los GA, las ES (*evolution strategies*) simbolizan la evolución a un macro nivel y no al micro nivel, cuando los GA usan conceptos de genotipos, cromosomas, genes y alelos, entre tanto las estrategias evolutivas recurren a fenotipos, herencias y variaciones. Estas incluyen diversos mecanismos de auto adaptación para controlar las mutaciones y la conservación o de permanencia de padres-hijos o solo hijos para la iteración generadoras de nuevas soluciones.

La DE (*differential evolution*) formulada por Storn y Price en 1995, realiza mutaciones a través de combinaciones aritméticas de individuos y no de perturbaciones o cambios en los genes de los individuos. Se caracteriza por su rápida convergencia pero de otro lado su naturaleza codiciosa (*greedy*) puede afectar su desempeño.

En un trabajo más reciente (2009), Premaratne propuso una metáfora de la polinización y dispersión en la evolución de plantas en lo que llamó algoritmo del campo de arroz (PFA, *paddy field algorithm*) que basa la búsqueda en la densidad de la población. El algoritmo “siembra semillas aleatorias” y la planta que mejor desarrollo presente, sirve de referencia (*threshold*). Después de la selección, viene el proceso de dispersión de semillas donde cada planta aporta una proporción acorde a su estado, es decir, aquellas ubicadas en mejores campos, producen más semillas (*fitness*), y luego por procesos de polinización con animales o dispersión por el viento, se amplía la densidad de la población (búsqueda de soluciones).

5.2.3 Inteligencia de Enjambres

Establecida por Kennedy & Eberhart en 2001, implementa sistemas adaptativos como extensión de la computación evolutiva pero esta vez, imitando los comportamientos sociales de organismos. Conocida en Inglés como *Swarm Intelligence* (SI), se inspira en el comportamiento social de agentes que genera una interacción colectiva e inteligente entre ellos y con el ambiente o entorno, y establecen trazados irregulares en sus movimientos, los cuales son una manifestación natural en los procesos de satisfacción de necesidades dentro del problema-espacio. En el caso de sistemas sociales de organismos, se muestra como un proceso de generación de trayectorias de integrantes de una comunidad o rebaño, de manera descentralizada y auto-organizada, logra establecer

mecanismos de optimización en procesos de forrajeo y pastoreo, soportados en comportamientos globales complejos.

Dentro de esta categoría de metaheurística, se ubican tanto el enjambre de partículas (PSO, *particle swarm optimization*), como todos los algoritmos basados en especies. Ambos sistemas exhiben flexibilidad, robustez, auto-organización y una remarcable coordinación en la labor de las actividades entre individuos, esta coordinación no se deriva de un centro de control (*core*) sino que el punto de partida es la auto-organización, del mismo modo da la posibilidad de tener cooperativamente el control del sistema a través de las interacciones entre los organismos para un sistema inicialmente en caos.

Dentro de la lista de algoritmos basados en sistemas sociales, se encuentra colonia de hormigas (ACO, *ant colony optimization*), colonias artificiales de abejas (ABP, *artificial bee colony algorithm*), enjambre de peces (FSA, fish swarm algorithm), caída inteligente de gotas de agua (IWD, *Intelligent Water drops*), forrajeo de bacterias (*bacterial foraging optimization algorithm*), el algoritmo de luciérnaga (*firefly algorithm*), y otros tantos más. La mayoría de estos algoritmos basados en conductas sociales de organismos son creaciones recientes (última década), y son producto del auge y éxito logrado en los primeros desarrollos, como es el caso de colonia de hormigas, y varios han sido propuestos para atender requerimientos de problemas específicos de optimización. Entrar a estudiar, comprender y detallar cada uno de ellos, es extender este trabajo más allá del alcance establecido y por eso mejor se centra en los dos métodos más populares y aplicados hasta el momento: SPO y ACO.

Los cinco principios básicos de la inteligencia de enjambres son:

1) Proximidad: la población debe estar en capacidad de realizar cálculos simples de espacio y tiempo.

2) Calidad: la población debe estar en capacidad de responder a factores de calidad en el entorno.

3) Respuesta Diversa: la población gestiona sus actividades a través de múltiples y amplios canales.

4) Estabilidad: la población no cambia sus comportamientos a la medida que el entorno también cambie.

5) Adaptabilidad: la población podría cambiar su comportamiento cuando enriquece el beneficio computacional.

5.2.4 El modelo de enjambre de partículas.

Inspirado en el comportamiento de manadas de aves que vuelan en busca de comida, hace analogía de partículas con miembros de bajo volumen y masa que se mueven en posiciones y a velocidades y aceleraciones en el espacio de solución. Cada partícula representa una solución en un espacio multidimensional con cuatro vectores: posición actual, la mejor posición hallada hasta el momento, la mejor posición hallada por la vecindad y la velocidad para ajustar su posición en el espacio de búsqueda. En efecto, la partícula tiene guardado en memoria cuál ha sido la mejor posición dentro de la búsqueda general en el espacio (*pbest*), y conoce también la mejor posición encontrada por todas las partículas dentro de la posición guardada para este espacio (*gbest*). Metodológicamente para el desarrollo del algoritmo, las partículas son desplazadas de su posición inicial mediante un vector de velocidad, y la magnitud y dirección del vector está determinado por la iteración inmediatamente anterior en el algoritmo, esto simula el impulso inicial y la ubicación de una partícula adyacente, teniendo en cuenta el *gbest* y *pbest*. (Brabazon et al., 2006). Los cambios de velocidad son resultados aleatorios según parámetros cognitivos y sociales. Esto porque el tamaño y la dirección del movimiento de cada partícula están determinados por su experiencia, además de la influencia social del grupo.

La **Figura 16** permite visualizar un diagrama de flujo del algoritmo, y la forma canónica (Brabazon et al., 2006), del mismo se describe así:

i. Inicializar cada partícula en la población a través de una selección de valores aleatorios para esta locación y sus vectores de velocidad. Número de partículas: entre 20 y 50. Rango del problema: dependiente.

ii. Calcular el valor del *fitness* para cada partícula.

iii. Seleccionar el pbest para cada partícula, y determinar la locación del gbest.

iv. Para cada partícula, calcular la velocidad usando (i).

v. Actualización de la posición de la partícula (iii).

vi. Calcular de nuevo el *fitness* de cada partícula, solo si el valor del fitness en el momento es mayor que su mejor *fitness* anterior, luego de revisar el *pbest*.

vii. Después que se hayan actualizado todas las partículas, se determina la ubicación de la partícula con el mejor *fitness* y si es necesario, se revisa el *gbest.*

viii. Repetir los pasos iv-vii hasta detenerse después que se haya encontrado el criterio.

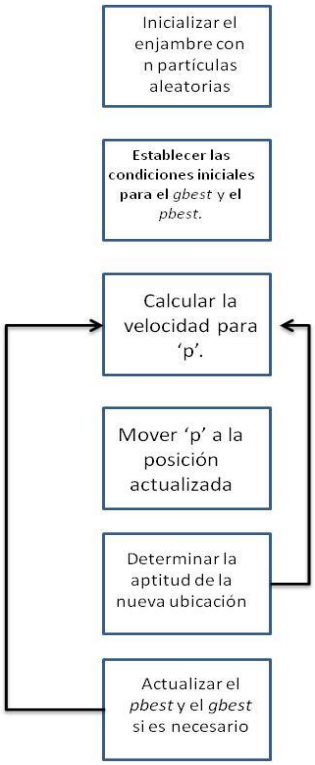


Figura 16. Diagrama de flujo del algoritmo de enjambre de partículas. (Brabazon et al., 2006)

El número de iteraciones permitidas o el lograr una solución de buen *fitness* permite detener el algoritmo. Entre las ventajas del PSO se cuenta su fácil implementación y su no dependencia de parámetros de ajuste, y al mismo tiempo es más eficiente que los GA ya que todas las partículas usan información de la mejor partícula y mejoran su solución a partir de la mejor solución global sin necesitar de operadores de recombinación o mutación. En los GA, las soluciones de pobre desempeño se van descartando.

5.2.5 Modelo de Colonia de Hormigas

Este modelo constituye una familia de algoritmos que son metafóricamente basados en las actividades de insectos como: hormigas, abejas o termitas. El comportamiento de forrajeo, específicamente, de la colonias de hormigas ha sido intensamente estudiado, así como la relación biológica en su comportamiento ha permitido encontrar diferentes soluciones. Cada insecto sigue una serie de reglas limitadas por su rol dentro de la comunidad, dado así que las interacciones de estas actividades da lugar a una estructura auto-sostenible y a un complejo auto-organizado, proporcionando a la colonia la capacidad de adaptarse a las alteraciones de su entorno. Las vías de comunicación de las hormigas son través de las feromonas (*Stigmergy*), que son sustancias químicas utilizadas en la comunicación entre organismos de la misma especie. Existen muchos tipos de feromonas que son producidas por estas, unas son de agregación o de llamado, otras feromonas son de alerta, y también están las feromonas para marcar su camino, etc. El intercambio de información entre hormigas pude ser directo – por un contacto físico o químico – o indirecto – cuando una hormiga modifica su ambiente al encontrarse con sus compañeros. En general los algoritmos son derivados de 4 metáforas del comportamiento de las hormigas: transporte cooperativo, formación de cementerios, clasificación de crías y el de forrajeo. (Xinjie, 2010). Se describirá a continuación solo el de forrajeo.

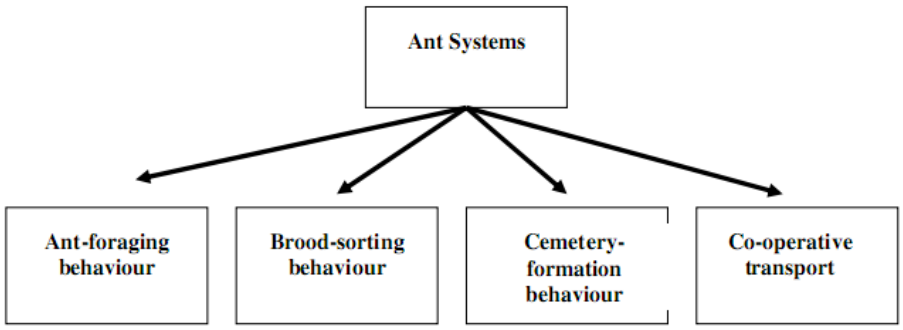


Figura 17. Taxonomía de los algoritmos para la colonia de hormigas. (Brabazon et al., 2006)

Modelo de forrajeo.

Como se observa en la **Figura 17**. En el sistema de hormigas es al modelo de forrajeo el que más se hará énfasis y utilidad. Este modelo está encaminado a cómo las hormigas descubren las fuentes de comida a través del ambiente. Se debe suponer que existen dos hormigas con las mismas características, velocidad y capacidad para producir la feromona, y han encontrado comida al mismo tiempo. Al volver a su nido, ellas depositan en el camino feromonas demarcándolo. La **Figura 18**a. muestra el momento exacto en el que estas hormigas se encuentran en la intersección de los caminos y como no hay información de cuál es el camino más corto la probabilidad de tomar el camino es de 0,5 (50%). (Xinjie, 2010). La hormiga triángulo toma el camino más corto, y la hormiga cuadrado toma el más largo, así que la hormiga triángulo alcanza el nido más rápido. Después de regresar al nido, la hormiga triángulo selecciona la ruta de vuelta al centro de comida de acuerdo a su feromona con probabilidad del 100%. En la **Figura 18**b. la hormiga cuadrado está aún en el camino al nido y cada vez que pasa el tiempo, más feromonas son depositadas en el camino más corto, estimulando a que la demás hormigas tomen éste camino.

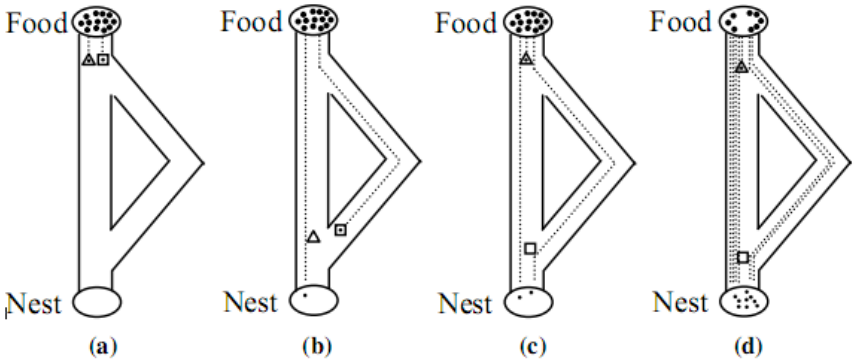


Figura 18. Ejemplo simplificado del forrajeo de las hormigas.(Xinjie, 2010)

Es claro que aquella hormiga que encuentre el camino más corto, tendrá una menor evaporación de su feromona e incentivará a otras hormigas a que sigan su rastro (*self-reinforcing process*), elevando aún más los niveles de feromonas y es ahí donde la alta concentración de estas permite que se tome el camino indicado en la consecución del alimento. En el sentido de la imitación de las hormigas para el diseño de los algoritmos es importante considerar lo siguiente:

i. El modelo probabilístico en la selección de las diferentes opciones, en otras palabras, la forma de buscar una solución.

ii. La manera por la cual se depositan y se evaporan las feromonas, es decir, cómo es el método de la formación de la solución.

Bajo este principio fue que Dorigo & Di Caro en 1999, propusieron un algoritmo metaheurístico de resolución de problemas de optimización combinatoria, después que en 1996, Dorigo introdujera el AS (*ant system*) para hallar una solución eficiente en el problema del agente viajero (TSP, *travelling salesman problem*), el cual se encarga de un solo nodo origen-destino y desde el cual se visita a cada cliente una sola vez, conformando un tour de nodos secuenciados en busca del mínimo costo de recorrido.

Al iniciar el algoritmo, todos los fragmentos solución tiene el valor de la feromona inicial distinto de cero, cada hormiga comienza a reconstruir un recorrido a partir de los fragmentos de soluciones potenciales y parciales, con el fin de construir una solución completa.

Otro comportamientos de de las colonias de hormigas que se han utilizado como metáfora para crear algoritmos están en la metodología del picking y el depósito de materiales. Ejemplos de estos comportamientos se exhiben en la clasificación de las crías, donde las larvas de las hormigas son ordenadas y agrupadas en grupos del mismo tamaño del individuo; así como la construcción de cementerios donde las hormigas muertas son eliminadas de la colonia y son depositadas juntas, en este tipo de algoritmos los vectores son clasificados en grupos similares. (Brabazon et al., 2006)

5.2.6 Otros BIAs Relevantes

Los sistemas artificiales inmunes son otra propuesta de BIA, considerando fortalezas del sistema inmunológico del ser humano que clona anticuerpos y donde los más aptos son los menos mutados y viceversa. De igual manera, se han establecido algoritmos donde los antígenos representan el valor de la función objetivo, y entre mayor *fitness* presenten, mayor clonación sufrirán, y luego los clones serán hipermutados según su calidad.

De otro lado, se han ido desarrollando algoritmos que imitan los equilibrios del ecosistema considerando factores abióticos como la temperatura, la luz, el suelo, etc. Los niveles de interacción entre y dentro de especies, sea de orden colaborativo o cooperativo, ha permitido crear optimizadores multi-especies que

evolucionan simbióticamente dentro del ecosistema. En otros casos, se hace analogía de sistemas de malezas invasivas las cuales entran en competencia a la medida que se amplía la densidad y donde el mayor *fitness* permite la supervivencia.

5.2.7 Conclusiones de la Metáfora Biológica para la Optimización

A la medida que se comprenden los procesos evolutivos y genéticos, de comportamiento social de enjambres e interacción entre-especies y con el entorno, se hace acopio de mayor número de inspiraciones que retan a investigadores de nuevos algoritmos para la optimización de sistemas. Es indudable, que la cascada de innovaciones en BIAs de la última década no permite hacer una consolidación introductoria de todos los algoritmos desarrollados en un trabajo de grado cuyo alcance es determinar el camino recorrido desde la biología hasta las técnicas de optimización inspiradas en la biología. Los niveles de sofisticación y de adaptación de cada algoritmo demandan enormes horas de investigación y estudio para aglutinar todas las posibilidades creadas hasta el momento.

Son incontrovertibles los éxitos registrados por la implementación de estos métodos pero al mismo tiempo aún está por definir los niveles de asertividad y confiabilidad de muchos métodos para problemas específicos que ayuden a determinar la mejor asignación de técnicas a cada tipo de problema. Por ahora, son las metáforas constitutivas de los algoritmos las que permiten alinearlos a problemas reales de optimización para plantear su adaptación y respectiva posibilidad de solución.

Si bien todo el desarrollo algorítmico ha surgido desde la investigación de operaciones, entendiendo el medio resolutivo dentro de la teoría de la programación, muchos de estos métodos han abierto posibilidades extensivas al campo de la minería de datos, la inteligencia artificial y el aprendizaje de máquinas. Aplicaciones en robótica, modelado autónomo, y múltiples sistemas inteligentes de la tecnología actual y futura permitirán motivar mayores desarrollos y aplicaciones que no se centren prioritariamente en requerimientos o necesidades de optimización.

5.3 Metaheurísticas en logística y cadenas de abastecimiento.

Objetivo específico 3: "Categorizar aplicaciones de meta-heurísticas inspiradas en la biología en problemas de la logística y la cadena de abastecimiento."

Las técnicas metaheurísticas en las últimas dos décadas se han convertido en un método capaz de proporcionar soluciones casi óptimas para los problemas que la optimización exacta no podía resolver. Dicha técnica ha sido enfocada a la solución de problemas de tipo logístico y Cadena de suministro. Entre ellas, las más utilizadas son: la optimización de la colonia de hormigas, el algoritmo genético, el recocido simulado y la búsqueda tabú en la metaheurística. Estas técnicas tienen la capacidad para examinar los riesgos en las cadenas de suministros, las interrupciones, las operaciones intermodales, las soluciones óptimas de servicio al cliente, las estrategias de backhaul, la ubicación de instalaciones simultáneas y los problemas de rutas de vehículos. (Suárez, 2011).

Métodos de solución

El modelado de problemas en la cadena de suministro y la logística se ha basado en tres métodos principales: la optimización, la simulación y la heurística. En las últimas décadas la optimización ha sido el método más utilizado para hallar la solución óptima de problemas complejos para la cadena de suministro, a través de modelos matemáticos. En cambio el principal interés del uso de la simulación para la solución de problemas de la cadena de suministro está aplicado a casos de una naturaleza estocástica con el fin de entender y simular la evolución de la capacidad de las diversas prácticas. (Suárez, 2011).

El desequilibrio operacional y de resultados en el uso de metodologías de modelado analítico, limita la capacidad para resolver problemas críticos de logística y cadena de abastecimiento. Para éstos problemas existen métodos más allá de lo común desde la optimización y simulación. En la última generación, la heurística y metaheurística, han ofrecido la capacidad para analizar problemas muy complejos que son casi imposibles de resolver en plazos razonables de tiempo por los métodos de optimización tradicionales. Aunque no se cuenta en la comunidad científica con una definición exacta del termino metaheurística. La más comúnmente usada es examinarla como un conjunto de patrones empleados para definir métodos heurísticos que pueden ser aplicados a determinado tipo de problemas. Se puede interpretar a las metaheurísticas como el modelo general de algoritmos (framework) que se aplica a diferentes problemas de optimización, el cual, con pocas modificaciones logra adaptarse a las particularidades de cada

problema. La aplicación de metaheurística en el análisis y solución de problemas complejos de la cadena de suministro, permitiendo que se seleccione la taxonomía metaheurística que más se ajuste para la resolución del mismo. Los investigadores y los profesionales prefieren métodos de análisis cerrados, donde se evalúan ecuaciones y funciones para resolver los problemas. (Suárez, 2011)

5.3.1 Heurística.

La “Heurística” es un concepto originado de la palabra griega *Heuriskein,* cuyo significado enuncia el concepto de encontrar o descubrir. La heurística se orienta al estudio del descubrimiento y la invención. Ésta se relaciona con la búsqueda constante de resolver los problemas inteligentemente utilizando la información disponible. Los problemas altamente complejos o “Difíciles de resolver” se les denominan problemas NP ya que no se les puede garantizar el encontrar la mejor solución en un tiempo polinómico determinístico. Es en este tipo de problemas donde los métodos heurísticos se convierten en procedimientos eficientes para encontrar “buenas soluciones” con rapidez e intentando garantizar la calidad de la solución obtenida (nivel de sub-optimalidad).

Es posible utilizar métodos heurísticos cuando un problema de optimización determinístico o no, tiene alguna de estas características (Suárez, 2011):

a. El problema es de una naturaleza tal que no se conoce ningún método exacto para su resolución.

b. Aunque exista un método exacto para resolver el problema, su uso es computacionalmente muy costoso o inviable.

c. El método heurístico es más flexible que un método exacto, permitiendo por ejemplo, la incorporación de condiciones de difícil modelación.

d. El modelo matemático es demasiado grande, demasiado No lineal o demasiado complejo desde el punto de vista lógico.

e. El asumir suposiciones o aproximaciones para simplificar el problema, tiende a destruir estructuras del modelo que son vitales en el contexto del mundo real, haciendo la solución no viable.

f. El método heurístico se utiliza como parte de un procedimiento global que garantiza el óptimo de un problema. Existen dos posibilidades:

- El método heurístico proporciona una buena solución inicial de partida
- El método heurístico participa en un paso intermedio del procedimiento como ejemplo de esto están las reglas de selección de la variable a entrar en la base en el método Simpléx.

Se han desarrollado muchos métodos heurísticos de naturalezas muy diferentes, en su mayoría diseñados para problemas específicos, sin posibilidad de aplicación en problemas similares. Es difícil hacer una clasificación de estos métodos, puesto que siguen apareciendo nuevos problemas, dando lugar a la construcción de nuevos algoritmos para tratar de solucionarlos.

Algunos de los métodos heurísticos más conocidos son (Suárez, 2011):

Métodos de Descomposición: El problema original se divide en sub-problemas más sencillos de resolver. Teniendo siempre presente que todo pertenece al mismo problema.

Métodos Inductivos: Busca generalizar las propiedades o técnicas identificadas en los diferentes casos, para posteriormente, aplicarlas en un problema complejo.

Métodos de Reducción: Busca reducir el espacio de solución, simplificando el problema. Evita rechazar soluciones óptimas del problema original, identificando propiedades que se cumplan en la mayoría de las buenas soluciones e introduciéndolas en las restricciones del mismo.

Métodos Constructivos: A través de la construcción de una lista de actividades para solucionar el problema y el uso de métodos deterministas basados en la mejor elección de cada una de las corridas del algoritmo, se llega a la mejor solución.

Métodos de Búsqueda Local: Parte de una solución al problema la cual se mejora progresivamente. A medida de que se avanza en la solución y se logra mejorarla, se selecciona el valor que arroja mejores resultados.

Las Heurísticas permiten una mayor flexibilidad y robustez para la utilización de las características del problema respecto a las técnicas que buscan soluciones exactas. En la aplicación de los métodos heurísticos se enfoca en generar una buena solución inicial a un problema complejo. Después de hallar esta solución, el método inicia un proceso de búsqueda de mejoras.

Al resolver un problema por métodos heurísticos, se debe medir la calidad de los resultados, evaluando la eficiencia del algoritmo para determinar su validez frente a los demás. Un buen algoritmo heurístico debe tener las siguientes propiedades:

a. **Ser Eficiente**: El esfuerzo computacional debe ser realista y adecuado para obtener la solución.

b. **Ser Bueno**: La solución debe estar cerca del óptimo.

c. **Ser robusto**: La probabilidad de obtener una solución errónea debe ser baja.

Las heurísticas de mejoras locales parten de una solución inicial y de forma repetitiva buscan en las soluciones vecinas hasta encontrar una mejor solución. La heurística puede construir soluciones o mejorarlas. Las capacidades de solución están limitadas por el tamaño y la complejidad del problema, se determina un óptimo local de solución, ignorando las mejores soluciones en diferentes regiones del espacio de solución. (Griffis, Bell, & Closs, 2012)

5.3.2 Metaheurísticos

En los últimos veinticinco años la investigación en el diseño de nuevas técnicas orientadas a la elaboración de soluciones heurísticas se ha conocido como Metaheurísticas. Éstas son estrategias para diseñar y/o mejorar los procedimientos heurísticos. El término metaheurística fue introducido en 1986 por Fred Glover, a partir de este acontecimiento se han presentado muchas propuestas para diseñar mejores procedimientos de solución a problemas combinatorios. Posteriormente en 1995 los profesores Kelly y Osman, elaboraron la siguiente definición: ”Los procedimiento metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas de difícil optimización combinatoria, en los que los heurísticos clásicos no son efectivos.” (Griffis et al., 2012)

Los métodos metaheurísticos otorgan un marco general para establecer nuevos algoritmos híbridos, combinando conceptos de diversos campos como la biología, la inteligencia artificial, las matemáticas, la física, la neurología, entre otros. Se definen como métodos de solución que articulan interacciones entre mejoras básicas en heurísticos locales y las estrategias de alto nivel, dirigidas a escapar de óptimos locales en un espacio de soluciones, encaminado a un óptimo global. Éstos métodos difieren de los métodos clásicos porque ellos “realizan una

búsqueda mucho más completa del espacio de soluciones, lo que permite movimientos diferentes, así como recombinaciones de soluciones para crear otras nuevas" (Griffis et al., 2012).

Algunas de las propiedades que buscan alcanzar los métodos metaheurísticos son (Alfonso et al., 2010):

-**Simple**: La metaheurística debe estar fundamentada en un principio de fácil comprensión (sencillez y claridad).

-**Precisa**: Las fases y pasos deben formularse en términos concretos.

-**Coherente**: Los elementos deben inferir de los principios.

-**Eficaz**: La probabilidad de alcanzar soluciones óptimas debe ser alta.

-**Eficiente**: Buen aprovechamiento de los recursos computacionales: tiempo de ejecución y espacio de la memoria

-**General**: Buen rendimiento en una amplia variedad de problemas.

-**Adaptable**: Capacidad de adaptarse a diferentes contextos de aplicación o modificaciones importantes del modelo.

-**Robusta**: La metaheurística debe ser poco sensible a pequeñas alteraciones del modelo o contexto de aplicación.

-**Interactiva**: Posibilita al usuario aplicar sus conocimientos para mejorar el rendimiento del proceso.

-**Múltiple**: Debe tener la capacidad de suministrar diferentes soluciones alternativas de alta calidad para que el usuario pueda elegir.

-**Autónoma**: Permite un funcionamiento libre de parámetros, que se pueda establecer automáticamente.

Áreas de Aplicación para Metaheurísticas

En las cadenas de suministro se han aplicado metaheurísticas a una variedad de problemas tales como rutas para vehículos, enrutamiento de red, planificación del espacio, secuenciación de trabajo y diseño de productos. La tabla 3 reúne los diferentes tipos de problemas relacionados con las cadenas de suministro. Como

se muestra en esta tabla, los usos más comunes están direccionados en rutas para vehículos y modelos de ubicación de instalaciones (distribución de planta). Además, se evidencia que existe una variedad de temas que son adecuados para ser analizados por medio del uso de metaheurísticas.

Metaheurísticas Inspiradas en la Biología

Ésta es la optimización basada en la colonia de hormigas (OCH) ("Ant Colony Optimization" ACO). Como se mencionó anteriormente esta metaheurística es bio-inspirada en el comportamiento estructurado de las colonias de hormigas, donde individuos muy simples se comunican entre sí por medio de sustancias químicas denominadas feromonas, la repetición de recorridos por los individuos instaura un camino adecuado entre su nido y su fuente de alimento. El método consiste en simular computacionalmente la comunicación indirecta que utilizan las hormigas para establecer el camino más corto, guardando la información aprendida en la matriz de feromonas.(Liao, Gen, Tiwari, & Chang, 2013). Este método utiliza el conocimiento utilizado por toda la colonia de hormigas, así que esta es una metaheurística basada en la población. Para el problema de ruteo de vehículos se simulan y modelan a las hormigas con el fin de encontrar soluciones; las hormigas (vehículos) se mueven a través de una red de búsqueda que refleja la situación real, todo esto análogo a la búsqueda de comida (objetivo), depositando feromonas en los caminos de la red (arcos), con el paso del tiempo en los caminos más cortos se acumula feromonas. Una propiedad importante, explicada anteriormente, muestra que las feromonas en los caminos se van evaporando, así que va desapareciendo la conveniencia de un sendero menos visitado. Además las hormigas ignoran al azar los senderos con el más fuerte rastro de feromona y toman un camino diferente con el fin de buscar nuevas fuentes de alimento (Griffis et al., 2012). Si ese camino viene siendo más corto, en comparación con los anteriores, renuevan los rastros de feromonas llamando la atención de otras hormigas para preferir ese nuevo camino que es más corto. Esta metaheurística aprende siguiendo un proceso casi aleatorio, y las hormigas con capaces de confluir en soluciones mejoradas e incluso en óptimos.

Esta metaheurística tiene algunas desventajas, una de ellas es la dificultad de ajuste de varios parámetros que no tienen una base teórica para sus valores. Esto requiere inversión de tiempo por parte de los usuarios para obtener pruebas y ensayos en los parámetros dados para un problema particular; la actualización de las rutas demarcadas por las feromonas con cada red y camino, deben ser logradas aumentando cada vez la dificultad de codificación y las veces de cálculo para los caminos.

Metaheurísticas Inspiradas en la Evolución: Esta técnica, que antes se mencionó, se basa en los mecanismos empleados por la selección natural, en la cual, los individuos más aptos de una población son los que sobreviven, al adaptarse más fácilmente a los cambios que se producen en el entorno. Lo mismo sucede con los genes en un individuo, los atributos más deseables se transmiten a sus descendientes cuando este se reproduce sexualmente. Estos son métodos que van construyendo un conjunto de soluciones diferentes; el procedimiento consiste en generar, seleccionar, combinar y reemplazar un conjunto de soluciones en la búsqueda de la mejor solución. (Griffis et al., 2012). El algoritmo genético posee algunas ventajas, puesto que no requieren una formulación lineal para generar soluciones. Tampoco requiere un amplio conocimiento de las limitaciones y reglas del problema real. Ésta metodología permite que las soluciones mejoradas converjan rápidamente y posteriormente se puedan traducir en un vector de variables de decisión representado en una solución para el problema real. (Griffis et al., 2012).

Algunas desventajas de los algoritmos genéticos son:

En primer lugar, éste acentúa la labor esencial que realiza la naturaleza, dando lugar a que el algoritmo genético no proporciona ninguna garantía de perfección para el óptimo, como lo hace la naturaleza. Además, el desarrollo de una función de aptitud (*fitness*) la cual evalúa nuevas soluciones como resultado de la descendencia dentro de una sola búsqueda en el algoritmo, puede ser complejo. No existen métodos universales para el cruce (recombinación) y la mutación (los cambios) con el fin de crear nuevas soluciones hijas que funcionen correctamente con todo tipo de problemas, y los investigadores a menudo tienen que desarrollar operadores especiales de selección, tamaño de la población y otros parámetros para encontrar la mejor solución posible.

Para el caso puntual de la cadena de suministro, los métodos de solución metaheurísticas, son especialmente adecuados para el análisis dentro de esta. Existen varias ventajas y desventajas al usar los métodos metaheurísticos a comparación con otros métodos analíticos, para el caso de la ilustración en cadenas de abastecimiento y logística los problemas que se afrontan son muy grandes para los problemas de optimización tradicional para garantizar una solución óptima. Los problemas de enrutamiento de vehículos y de distribución de planta dan lugar a infinitas combinaciones que no se pueden resolver en tiempo real con cualquier algoritmo cerrado, esto debido a que se clasifican como problemas de optimización NP-duros. Técnicamente, derivar los límites inferiores para tales problemas resulta difícil y la función de algoritmos exactos como los

branch and bounch es demasiado lenta para converger y no son adecuados para el problema. Sin embargo los modelos metaheurísticos tienen la habilidad de encontrar una solución “cuasi-óptima” (Griffis et al., 2012), además éstos pueden proporcionar pueden proporcionar soluciones dentro de los óptimos esperados y ofrecer una idea de los problemas complicados que la simulación o la programación lineal no pueden analizar por completo. (Liao et al., 2013).

Además del tamaño y la complejidad, las metaheurísticas proporcionan un método robusto que puede ser adaptado a los problemas con diferentes características de la solución. Esta capacidad es importante cuando se enfrenta a problemas de la cadena de suministro, asì que las restricciones y condiciones puedan cambiar con frecuencia. El diseño de algoritmos metaheurísticos modulares los hacen fáciles de actualizar y volver a ejecutar cuando se producen en torno del mundo real. Estas proveen soluciones posibles a un problema que encuentra soluciones alternativas, soluciones óptimas y proporciones alternativas al usuario; todo esto es necesario en los problemas logísticos que poseen varias soluciones alternativas, es decir cada resultado válido proporciona –costo, distancia, velocidad, etc.-. En conclusión los métodos metaheurísticos pueden proporcionar una variedad de soluciones que se utilizarán para desarrollar un diagrama de frontera eficiente, así que el usuario deberá identificar soluciones en torno al servicio al cliente o costo logístico. (Griffis et al., 2012)

Áreas de Aplicación para Metaheurísticas

En las cadenas de suministro se han aplicado metaheurísticas a una variedad de problemas tales como rutas para vehículos, enrutamiento de red, planificación del espacio, secuenciación de trabajo y diseño de productos. La tabla que se muestra a continuación reúne los diferentes tipos de problemas relacionados con las cadenas de suministro

En la Tabla 3 se reúnen algunos tipos de problemas relacionados con las cadenas de suministro; los cuales se han analizado por las diferentes taxonomías metaheurísticas inspiradas en la biología. Como se muestra en esta tabla los usos más comunes dentro de las especificaciones logísticas están direccionados en rutas para vehículos y modelos de ubicación de instalaciones (distribución de planta). Además, se evidencia que existe una variedad de temas que son adecuados para ser analizados por medio del uso de metaheurísticas.

La Tabla 4. presenta una revisión de temas respecto a las potenciales aplicaciones de las metaheurísticas que se han trabajado en el desarrollo de este objetivo y en el cuerpo del trabajo, aunque no son las únicas que se relacionan

con la investigación en general, se presenta un resumen obtenido de diferentes autores encaminándolo hacía el desarrollo de la aplicación, ademas que han sido punto de comparación y metáfora de los sistemas biológicos, para la práctica en la cadena de suministro (Supply Chain), que es el tema que con el que se debe cerrar esta investigación, entre esas aplicaciones se encuentran, distribución de instalaciones, optimización del nivel de servicio, programación de tripulación, y la aplicación que ha sido elegida para desarrollar en el siguiente objetivo, que hace parte de la fundamentación del ruteo de vehículos uniendo la simulación del problema del agente viajero y aplicando el sistema de hormigas y la optimización de colonia de hormigas (ACO).

Tabla 3. Las relaciones entre los métodos metaheurísticos inspirados en sistemas biológicos y las áreas temáticas en la cadena de abastecimiento. (Griffis et al., 2012).

Áreas Temáticas	**Métodos**
Rutas para vehículos	Busqueda Tabú
Ubicación de la instalación /Diseño de la Red	Algoritmos genéticos
Producción/ ops Programación	Recocido Simulado
Diseño y asignación de recursos	Colonia de Hormigas
Programación de vehiculos	GRASP (Procedimiento codicioso, aleatorio adaptativo de condicones de búsqueda)
Metodo y desarrollo Cuantitativo	Otros
Productos Mixtos/ Surtidos	
Dimensionamiento de Flota de Vehiculos	
Diseño de Productos	
Previsión	
Programación de Trabajo	
Selección del mercado de Productos	
Compras	
Programación de Proyectos	

Tabla 4. Aplicación potencial de las metaheurísticas para los problemas de la cadena de suministro. Fuente: Los Autores, adaptado de (Griffis et al., 2012), (Binitha & Sathya, 2012), (Hillier, 2010).

Tipo de Problema	**Potencial Aplicación de las Metaheurísticas inspiradas en sistemas biológicos**	**Problema de Partida**	**Aplicación Practica en la Cadena de Suministro**
Rutas Para Vehículos	**Optimización de Colonia de hormigas (ACO)**	**La flexibilidad, la velocidad y una solución de calidad apropiada para horizontes cortos de planificación de las operaciones de despacho. Minimizar el número de vehiculos y las distancias de viaje totales.**	**La posibilidad de enfrentar las interrupciones de la cadena de abastecimiento y las operaciones de *Cross Docking* dada la capacidad de adaptación de las hormigas.**
Instalaciones / Distribución de planta y almacén	Algoritmos Genéticos	Capacidad de manejar problemas complejos para la acomodación de los recursos y el uso adecuado del espacio físico.	Planificar la ubicación del pasillo y la manipulación de los materiales (*Material Handing*), ubicando equipos en un almacén con la colocación de la grúa en los muelles de carga.
Optimización del nivel de servicio	Optimización de Colonia de hormigas (ACO)	Capacidad de encontrar soluciones factibles a los múltiples problemas de enrutamiento. Tardanzas acumulativas, medición del tiempo a través de las tareas desarrolladas	Analizar el equilibrio de los niveles de servicio a los clientes frente a los objetivos de costes y beneficios de enrutamiento o red de diseño. Asumiendo la demanda estocástica y asignación de la movilidad
	Algoritmos Genéticos	Compensación en el análisis de los objetivos y la capacidad en múltiples problemas.	Direccionar el servicio al cliente, además de direccionar los costos de compensación en un sistema de distribución. Se aplica a sistemas justo a tiempo en un multiproducto y para cadenas de suministro multicanal.

Dimensionamiento de la flota	**Optimización de Colonia de hormigas (ACO)**	**Capacidad de utilizar de manera eficiente los recursos tales como el tiempo, el trabajo y los equipos en problemas de diseño y enrutamiento.**	**Analizar las estrategias de tercerización de rutas para vehículos, priorizar en minimizar el tamaño de la flota.**
Programación de Tripulación	Algoritmos Genéticos	Demostrar la capacidad para resolver problemas con recursos móviles, tales como los operadores de vehículos. Disminución de retrasos, y optimización en acomodación de embarques	Aplicación para camiones, aviones y programación de buques en casos en los que las acciones estables de la tripulación podrían disminuir el rendimiento de la familiarización del servicio con el medio de transporte.
Gestión de Devoluciones	Algoritmos Genéticos	Demostrar la capacidad de optimizar diseños complejos de redes, dadas múltiples restricciones y flujos.	Utilizado para encontrar el enfoque para mejorar la recolección, reparación, reventa y disposición del producto devuelto.
Optimizar el Problema de Transporte	Algoritmos Genéticos	Capacidad para visualizar la compleja combinación de los precios y los problemas de asignación.	Administrar la subasta de contratación de transporte y precios de la oferta de los proveedores con diferentes objetivos y puntos de fijación de precios al comprador.
Problemas Híbridos	Algoritmos Genéticos	Capacidad para crear diseños de multiescalon, capacidad para resolver problemas de inventarios y de enrutamiento.	Decisiones simultaneas sobre el número y planta, distribuidores y puntos de venta en una cadena de abastecimiento. Determinar enrutamiento del nivel de inventarios y red de vehículos.

Conclusión del alcance de las aplicaciones metaheurísticas.

Los algoritmos inspirados en sistemas biológicos son un nuevo reto para la ciencia de la computación, todas las metaheurísticas aplicadas a la ingeniería industrial, tanto a logística como manufactura, están representadas por la variación y la aleatoriedad para encontrar la solución deseada. Todas aquellas aplicaciones metaheurísticas desde la computación natural han permitido impactar áreas como las redes computacionales, los sistemas de control, la bioinformática, entre otras. Sin embargo ésta investigación permite vislumbrar que los BIAs, las metaheurísticas asociadas y sus aplicaciones en la industria y la búsqueda de soluciones son un campo en crecimiento debido a la variabilidad de problemas, su cantidad de parámetros y de combinaciones entre ellos permiten generar alternativas para los algoritmos en el proceso de la optimización de las soluciones o de la interpretación de los problemas generados, el número de metaheurísticas tiene un efecto directo en la complejidad en el uso y aplicación del algoritmo. Con esto, se irán desarrollando nuevas áreas de investigación donde existirá una oportunidad para explorar nuevos alcances de los algoritmos y las metaheurísticas, y esto es posible gracias a una investigación colaborativa desde muchas áreas inmersas desde la ciencia de la computación para la computación natural.

5.4 Optimización por Colonia del Hormigas. (Ant System).

Objetivo específico 4: "Ejemplificar e Ilustrar la aplicación de una técnica metaheurística ad-hoc (inspirada en la biología) para un caso en el área de logística o cadena de abastecimiento".

Como se ha mencionado a lo largo de este trabajo, se ha hecho uso de metaheurísticas en la resolución de problemas de optimización combinatoria NP-Hard; para el desarrollo de este objetivo y cerrar la investigación teórica que se ha realizado se hará la ilustración de uno de los algoritmos inspirados en sistemas biológicos y que han servido para resolver muchos problemas en el área logística. Se hará el estudio del método de Optimización a través de Colonia de Hormigas (Ant Colony Optimization), se presentarán una visión global, los elementos que conforman este algoritmo y una visión general de la red que recorre este método.

Generalidades de la Optimización por Colonia de Hormigas.

En los últimos años se ha acrecentado el interés en la aplicación en las metaheurística para la resolución de problemas de optimización combinatoria de tipo NP-hard. Dentro de las taxonomías de las metaheurística las más comunes son los Métodos basados en trayectoria y Métodos Basados en población. El segundo de los métodos mencionados hace referencia a un conjunto de soluciones potenciales (la población) en lugar de proporcionar una única solución. (Maniezzo, Gambardella, & Luigi, 2007).

La Optimización de colonia de hormigas (ACO) es una metaheurística en la cual se ha trabajado en los últimos 15 años, esta se ha utilizado con éxito para la resolución de problemas muy complejos en la optimización combinatoria. (Maniezzo et al., 2007).

La fuente inspira a la optimización por colonia de hormigas (ACO) ha sido el ensayo a través de caminos marcados con feromonas y el comportamiento que se genera en las hormigas reales, las cuales utilizan las feromonas como un medio de comunicación entre ellas. En analogía con el ejemplo biológico, ACO está basado en la comunicación entre colonias de agentes simples, llamados "hormigas artificiales", mediados por caminos de feromonas artificiales. Estos caminos de feromonas en ACO sirven como información distribuida numéricamente, la cual es usada por las hormigas para construir de manera probabilística las soluciones a problemas en los que se adapta durante la realización del algoritmo para reflejar su experiencia de búsqueda para la solución óptima. (Gendreau & Potvin, 2010).

Antecedentes

La capacidad que tiene la colonia de hormigas para realizar tareas de gran complejidad por medio de la organización de los individuos, permite el diseño de muchos algoritmos. A pesar que las capacidades de trabajo de cada individuo son limitadas, las hormigas utilizan una forma de comunicación indirecta que les permiten lograr coordinación en las tareas que realizan, ésta consiste en realizar modificaciones en su entorno. Debido a que la visión de las hormigas es bastante deficiente, ésta comunicación se basa en la liberación de sustancias químicas (feromonas) que se depositan para marcar rastros en el suelo. (Maniezzo et al., 2007)

Marco Dorigo es quien formuló la heurística ACO basada en población (1991-1992). Él presenta en un solo marco de referencia las diferentes propuestas sobre la utilización de hormigas artificiales para la solución de problemas de optimización, y establece las características fundamentales de este tipo de técnicas:

a) Emplear agentes, que son hormigas artificiales con el fin de construir y lograr soluciones en forma incremental. Cada una de ellas construye una solución independiente sobre una solución parcial.
b) Integración de los componentes por medio de la regla probabilística, que tiene presente la experiencia adquirida en etapas anteriores a la búsqueda además de la información heurística del problema que está siendo resuelto.
c) La experiencia se obtiene en la construcción de una matriz de feromonas. La cual guarda y permite depositar a las hormigas el rastro en la construcción de soluciones de buena calidad. (Maniezzo et al., 2007)

En la Tabla 5 se presenta el planteamiento de los mecanismos para abordar los diferentes tipos de problemas y las características de los mismos.

Tabla 5. Mecanismos para los diferentes problemas en ACO. (Adaptado de: (Maniezzo et al., 2007))

Tipo de Problema	**Descripción – Características**
Estático	Estos están definidos completamente a priori. Se conocen todas las características antes de la resolución y se mantienen constantes durante su solución.

Dinámico	Poseen en ellos características que evolucionan mientras que el problema es resulto. Los algoritmos que abordan este tipo de problemas deben incorporar mecanismos que permiten la adaptación a esos cambios.
Multi-objetivo	Se dispone de varias funciones a optimizar y generalmente implican el balance entre intereses contrapuestos.

(Fuente: Los Autores).

La Metaheurística ACO

Las hormigas artificiales usadas en ACO son construcciones precedentes de una solución estocástica, que implementan una construcción heurística aleatoria para la toma en las decisiones probabilísticas como una función de la hormona artificial de seguimiento, además de la información heurística disponible inicialmente basada en la información del problema que se va a solucionar. La solución se construye probabilísticamente mediante la adición sucesiva de componentes de solución a soluciones parciales, por ejemplo, la actualización de la hormona, por ende se toma en cuenta primero la información heurística acerca del problema que se quiere resolver, si ésta está disponible o no, y segundo las feromonas de seguimiento artificiales que cambian dinámicamente durante la corrida del algoritmo para reflejar la experiencia adquirida por los agentes. (Gendreau & Potvin, 2010)

Un componente estocástico en un ACO permite a las hormigas construir un mayor número de soluciones y hacer énfasis en explorar un número mayor de soluciones. Al mismo tiempo, el uso de la información estocástica que está disponible para muchos problemas puede guiar a las hormigas a través de soluciones mucho más confiables. Adicionalmente lo más importante es que la experiencia de búsqueda de las hormigas pueda ser influenciada para construir una solución con iteraciones futuras del algoritmo. El uso de la colonia de hormigas puede dar al algoritmo una mayor robustez y en muchos casos las aplicaciones del ACO usan iteraciones colectivas de una población de agentes lo

cual es necesario para resolver de forma eficiente un problema. (Gendreau & Potvin, 2010)

El ámbito de aplicación de los algoritmos ACO es muy amplio. Por esta razón ACO puede ser interpretado como una extensión de la construcción heurística tradicional, la cual puede ser utilizada en muchos problemas de optimización combinatoria. A pesar de esto una diferencia importante con las construcciones heurísticas es la adaptabilidad de la feromona de seguimiento durante la ejecución del algoritmo puesto que tiene en cuenta la experiencia de búsqueda acumulada (Gendreau & Potvin, 2010).

Algoritmos de Construcción

Los algoritmos de construcción crean soluciones a un problema bajo consideración de una manera progresiva, empezando con una solución inicial vacía y de manera repetida agrega componentes de la solución sin retroceder al inicio, se logra hasta que una solución completa es obtenida, siendo esta una iteración o corrida del algoritmo de construcción. En el caso más simple, los componentes de la solución son agregados de manera aleatoria. A menudo, se pueden obtener mejores resultados si se toma en cuenta un cálculo heurístico del beneficio obtenido de agregar componentes de la solución. La "*Greedy construction*" o algoritmo de construcción, agrega a cada paso un componente de la solución que logra el beneficio máximo medido por alguna componente de la información heurística. La **Figura 19**. muestra la descripción de un algoritmo de una greedy construction heurística. La función (operación) GreedyComponent da el componente de la solución "e" con el mejor cálculo como una función de la solución parcial actual "sp". Las soluciones que dan los algoritmos greedy son típicamente de mucha mejor calidad que las soluciones generadas al azar. Sin embargo, una de las desventajas de la greedy construction heuristics es que solamente generan un número limitado de soluciones diferentes. Asimismo, decisiones greedy tomadas en etapas tempranas del proceso de construcción limitan las opciones disponibles en etapas posteriores del proceso, llevando a malas (pobres) movidas en las fases finales de la construcción de la solución.

procedure *Greedy Construction Heuristic*
  $s_p = empty\ solution$
  while s_p not_a_complete_solution **do**
    $e = \texttt{GreedyComponent}(s_p)$
    $s_p = s_p \otimes e$
  end
  return s_p
end *Greedy Construction Heuristic*

Figura 19. Estructura del algoritmo heurístico de construcción Greedy. Se adiciona el componente e para la solución parcial sp denotada con ⊗. (Gendreau & Potvin, 2010).

Ant System (AS)

Dorigo en 1991 presentó la primer propuesta de este tipo de algoritmo. Esta constaba de tres propuestas diferentes *ant-density, ant-quality y ant-cycle*. Las dos primeras cayeron rápidamente en desuso, la principal diferencia con la tercera variante es la forma en la cual se actualiza la feromona. Aunque han aparecido otras variantes, en general al hablar de (AS) se refieren a la variante *ant-cycle.* (Maniezzo et al., 2007)

La principal característica de los AS es que la actualización de la feromona se realiza una vez todas las hormigas han completado sus soluciones. El procedimiento es el siguiente: i) Todos los rastros de feromona se reducen en un factor constante, de esta forma implementa la evaporación de la feromona. ii) Cada hormiga deposita una cantidad de feromona que es la función de calidad de su solución. (Atehortúa, 2012).

En una forma de diagrama de flujo, el algoritmo tiene el siguiente desarrollo:

Figura 20. Desarrollo secuencial del algoritmo en el sistema de hormigas.

El algoritmo tiene una ejecución continua hasta cumplir una condición de parada. La condición de parada puede ser establecida de muchas maneras según el objetivo y la disponibilidad del recursos, en ocasiones puede ser un numero de iteraciones específicas.

De acuerdo con la probabilidad resultante de una función heurística y de la cantidad de feromona detectada en ese recorrido concreto (“mover hormiga”), en cada ciclo se crea una hormiga que va cambiando de estado o nodo. Se puede afirmar que la hormiga es un elemento simple que se traslada teniendo presente la información local tanto heurística, como la aportada por la colonia (feromona). Como las ciudades no se pueden repetir cada hormiga tiene una lista denominada Tabú, como método de búsqueda local, de las ciudades ya visitadas. Se determina la probabilidad de las diferentes rutas posibles a tomar. Una vez que se cuenta con estos valores la hormiga decide en función de estos valores (“Elección del

movimiento"). El nodo elegido se agrega a la lista Tabú, éste proceso se repite hasta visitar todos los nodos. Cuando finaliza el ciclo, se realiza la evaporación de feromona depositada en los arcos de la red ("Actualización de Feromona"). El ciclo se repite hasta la condición fin establecida para el algoritmo. (Atehortúa, 2012)

Las hormigas hacen uso del depósito de feromonas para recordar su comportamiento, lo hacen para poder acumular el conocimiento que van adquiriendo del problema a resolver. Al inicio, todos los arcos tienen la misma probabilidad y por esta razón se considera oportuno introducir un pequeño valor de feromona, lo cual permitirá explorar caminos que no han sido recorridos.

La Metaheurística

La estructura general de la metaheurística ACO para problemas de optimización combinatoria se presenta a continuación:

```
procedure ACO algorithm for combinatorial optimization problems
    Initialization
    while (termination condition not met) do
        ConstructAntSolutions
        ApplyLocalSearch        % optional
        UpdatePheromones
    end
end ACO algorithm for combinatorial optimization problems
```

Figura 21. Estructura general de la metaheurística ACO. (Gendreau & Potvin, 2010)

En la **Figura 21** , se exhibe el seudo código esqueleto para los algoritmos ACO aplicado a problemas de optimización combinatoria.

Seguido de la inicialización de parámetros y senderos de feromonas, el código principal consta de tres pasos principales. En primer lugar, 'm' hormigas construyen soluciones a la condiciones dadas por el problema que se estudia, sesgadas por la información de feromonas y, posiblemente, por la información heurística disponibles. Una vez que las hormigas han completado sus soluciones, éstas pueden ser mejoradas en una fase de búsqueda local opcional. Por último, antes del comienzo de la siguiente iteración, los rastros de feromona están adaptadas para reflejar la experiencia de búsqueda de las hormigas. Los pasos principales de la metaheurística ACO se explican con más detalle a continuación.

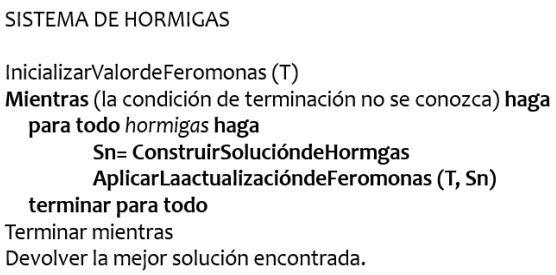


Figura 22. Algoritmo para el sistema hormiga. (Maniezzo et al., 2007)

Initializar, es el comienzo del algoritmo, todos los parámetros se configuran y todas las variables de feromonas a un valor τ_{init}, que corresponde a la concentración inicial de la feromona, este es un parámetro inicial del algoritmo. Se inicializan el rastro de feromona para cada una de las posibles componentes de la solución. Si se utilizan valores muy grandes es necesario hacer varias iteraciones antes de que la evaporación reduzca suficientemente el efecto para la feromona depositada por la hormiga. Contrario a si se utilizan valores muy pequeños, entonces la búsqueda estará fuertemente orientada por las primeras soluciones generadas. Lo ideal es inicializar este valor con una cantidad de feromona que será depositada por la colonia de hormigas en la primera iteración.

Es importante conocer una solución al problema considerado. Normalmente el costo asociado a una solución construida simplemente por medio de la estimación de una buena solución o a través de alguna heurística auxiliar, el valor sugerido para este parámetro es:

$$\forall \tau_{i,x_i} \in T, \tau_{i,x_i} = \tau_{init} = \frac{m}{Costo(SolucionAuxiliar)}$$ (**Ecuación 1**)

ConstructAntSolutions. Un conjunto de 'm' hormigas construye soluciones a las condiciones que el problema ha planteado. Para ello, cada hormiga comienza con una solución inicialmente vacía $s^p = 0$ (vacia). La selección de la próxima componente es realizada de acuerdo con la regla de transición de estados:

$$p(c_{i,x_i}|s^p) = \begin{cases} \frac{[\tau_{i,x_i}]^\alpha [\eta_{i,x_i}]^\beta}{\sum_{c_{j,x_j} \in J(s^p)} [\tau_{j,x_j}]^\alpha [\eta_{j,x_j}]^\beta} & \text{si } c_{i,x_i} \in J(s^p) \\ 0 & \text{en otro caso} \end{cases}$$ **(Ecuación 2)**

(Maniezzo et al., 2007)

En la Ecuación 3, los parámetros β y α son utilizadas para ajustar la influencia relativa de la información heurística ($\eta_{i, xi}$), además de los valores de feromona ($\tau_{i, xi}$) y $J(s^p)$ es el conjunto de componentes que pueden ser agregados a la solución parcial s^p. En la mayoría de los casos se utiliza:

$$\eta_{i,x_i} = \frac{1}{Costo(c_{i,x_i})}$$ (**Ecuación 3**.) Relación heurística

Si se anula el parámetro α, solamente se considera el componente heurístico. Si se anula el parámetro β, rápidamente se observa la aparición del fenómeno conocido como estancamiento que consiste en la situación en la cual todas las hormigas siguen exactamente el mismo camino y construyen exactamente la misma solución

ApplyLocalSearch. Una vez que se obtienen soluciones candidatas completas, éstas se pueden ir mejorado mediante la aplicación de algoritmos de búsqueda local. Además, para una amplia gama de problemas de optimización combinatoria, los algoritmos ACO alcanzan un mejor rendimiento cuando se combina con algoritmos de búsqueda local. (Gendreau & Potvin, 2010)

UpdatePheromones. La actualización de feromona se encarga de actualizar la hormona, esencialmente existen dos mecanismos que se utilizan para lograr este objetivo. El primero es depositar feromona, lo que aumenta el nivel de la feromona en componentes de la solución que se asocian con un conjunto elegido de buenas soluciones. El segundo es indicar el rastro de evaporación de la feromona, que es el mecanismo que disminuye con el tiempo la feromona depositada por las hormigas anteriores. Desde un punto de vista práctico, se necesita la evaporación de feromonas para evitar una rápida convergencia del algoritmo hacia una región subóptima, menos que la óptima. Esta clase en el algoritmo implementa una forma útil de olvido, lo que favorece la exploración de nuevas áreas del espacio de búsqueda. (Gendreau & Potvin, 2010). La actualización de la feromona se implementa comúnmente como la regla de actualización conocida: “online delayed pheromone rule”:

$$\tau_{i,x_i} \leftarrow (1-\rho) * \tau_{i,x_i} + \sum_{a \in A} \Delta\tau_{i,x_i}^{s_a} \quad \forall \tau_{i,x_i} \in T$$

$$\text{siendo} \quad \Delta\tau_{i,x_i}^{s_a} = \begin{cases} F(s_a) & \text{si } c_{i,x_i} \in s_a \\ 0 & \text{en otro caso} \end{cases}$$

(Ecuación 4). (Maniezzo et al., 2007)**.**

El objetivo es aumentar la cantidad de feromona para los componentes de la solución que han sido utilizadas por las soluciones de alta calidad. (Maniezzo et al., 2007). Respecto al tamaño de la población se recomienda que sea igual a la cantidad de ciudades (n). En la **Tabla 6**, se muestran los parámetros utilizados por esta variante dentro del Ant System.

Tabla 6. Los parámetros empleados por el algoritmo para el AS. Parámetros para AS (Maniezzo et al., 2007)

Parámetro	Descripción	Valor Recomendado
m	Tamaño de la población de hormigas	n
τ_{init}	Cantidad inicial de feromona en las componentes de las soluciones	$\frac{e+m}{\rho * Costo(SolucionAuxiliar)}$
α	Influencia relativa de la componente de feromona	1
β	Influencia relativa del componente heurístico	$2 \leq \beta \leq 5$
ρ	Tasa de evaporación de la feromona	0,5

Es muy importante lograr identificar que existen múltiples variantes que se derivan de la estructura algorítmica de Ant System, que es el esquema general de programación del ACO. Existen más de catorce estructuras metaheurística de resolución de problemas derivadas de la misma, para la solución en la optimización combinatoria, dependiendo de las características generales, existen variaciones en su código en la función objetivo y en los parámetros que usa.

En este caso elegimos ejemplificar un problema de ruteo de vehículos el cual se solucionó con la metaheurística de optimización de Colonia de hormigas (ACO) por medio del algoritmo Elitist Ant System (EAS). En el cual los utilizados por esta variante y sus valores se determinan a partir de las pruebas realizadas para el TSP, que es el problema del agente viajero.

5.4.1 Ilustración de Traveling Salesman Problem para ACO (Ant Colony Optimization)

Se utilizó el algoritmo de optimización por colonia de hormigas (ACO) además del sistema de colonia de hormigas para simular el problema del TSP (*travel salesman problem*), respondiendo a que existen un número de ciudades con unas distancias respectivas entre ellas, el objetivo es asegurar la búsqueda óptima de la ruta menos costosa, éste problema, hace parte de los NP-Hard, que se han mencionado para muchos casos como la planificación, la logística y el ruteo de vehículos, entre otros, se simulará el comportamiento de forrajeo que usan las hormigas, asociando los valores de la feromona directamente a la solución del problema del TSP. Los nodos representan ciudades y los arcos representan las distancias entre ellas. El objetivo es encontrar una ruta cerrada que contenga cada nodo una vez y que pase por todos (realizando un tour). El valor de la función objetivo f(s) es calculado por la suma de todos los valores cuantitativos de los arcos contenidos en el espacio de solución s. Ante todo, el problema del TSP se puede resolver de muchas maneras utilizando la programación discreta, utilizando el valor de los arcos (0,1) y escogiendo si éste es válido dentro de la ruta (Xe =1). (Maniezzo et al., 2007).

Para ilustrar este problema, se tomó de la biblioteca de programación de MatLab ® un código fuente que sirve para entender el desarrollo del algoritmo, y que genera una solución con sus respectivas variables de salidas, indicadas para el sistema de hormigas.

Según la forma canónica y la vía por la cual el algoritmo de ACO trabaja para encontrar la solución al problema, se encontraron 6 clases y el main (principal) dentro del código obtenido de MatLab ®, que es software que funciona como herramienta matemática y contiene un entorno de desarrollo integrado con un lenguaje de programación propio, en este caso se realiza la implementación de un algoritmo que ha sido simulado a través de Simulink obteniendo gráficos 2D.

Funciones:

- ant_information
- ant_primaryplacing
- ant_cycle
- ant_cost
- ant_traceupdating

Main

La clase ‘main’ es el seudo código del algoritmo para la búsqueda local de soluciones (local search), éste contiene en su inicialización, que se observa en la línea 1, el lugar donde se inician todos los parámetros de la información de las hormigas y además de todos los las entradas para las feromonas que son inicializados a un valor τ0, seguido por un ‘for’ (línea 2) que da el inicio para la construcción de las soluciones, esta función ‘for’ nombra una matriz llamada ‘app’ que obtiene y llama a la clase ‘ant_primaryplacing’ con el número (m) de hormigas y el número (n) de nodos que empieza a realizar la construcción de las soluciones inicializando a la solución (s) en 0; luego se declara una matriz ‘at’ que contiene la clase ‘ant_cycle’; después se declara ‘at’ que opera de manera horizontal para el mismo ‘at’ en todas las filas en la primera columna; contiguo a esto se declara la matriz [‘cost,f] que relaciona y llama la clase ‘ant_cost’ con el hecho de encontrar el costo menor y que cumpla la función objetivo de minimizar el mismo; la actualización de la feromona se realiza en el momento que se declara la matriz [t] que llama la clase ‘ant_traceupdating’; así mismo se muestra el costo para las itereaciones a través de ‘costa’ que referencia al costo medio a través de ‘mean(cost)’, esto se explicará en la clase referente al costo; luego se declara una matriz ‘[mincost (i), number] que llama al valor mínimo para el costo en cada iteración; en la siguiente línea (10) se realiza el mejor recorrido ‘besttour’ para el ciclo de las hormigas, que tiene en cuenta el menor costo y el ciclo de ellas, terminando con el hecho de obtener el número de la iteración. Se termina cuando se halle la mejor solución desde el inicio del algoritmo.

Análogamente, se muestran en gráficos el promedio del costo (distancia) vs el número de ciclos, con el fin de mostrar el mejor tour con el mejor costo, en el eje X, está la iteración y en el eje Y se muestre la distancia. También se programa un gráfico donde se muestra el tour cerrado con la solución óptima local y con eso se muestra

ants_information

En la clase ant_information se declara toda la información de los parámetros iniciales y que usaran para la construcción de la solución y la actualización de la feromona. En esta clase se observan los siguientes parámetros.

La función ant_information contiene:

- Las coordenadas en eje X y en el eje Y que van a ser parte del espacio solución;
- El número de iteraciones (i) = 100;
- El número de hormigas (m);
- El número de nodos (n) que es una longitud para el eje X;

Ahora bien, para generar la longitud de la matriz de enlace se hace un ‘for’ que evalúa cada componente en la coordenada X y en la coordenada Y, y se pude establecer el parámetro d(i,j) que mide la hipotenusa entre dos puntos para crear un arco entre un punto y otro.

Se declara el parámetro ‘e’ que es el coeficiente de evaporación de la feromona para este caso, también los parámetros alfa (α) y beta (β) que son respectivamente, la influencia de la feromona y la influencia del componente heurístico.

Para generar el lugar de interés de inicio dentro de la matriz de ubicaciones (X,Y), esto con el fin de declarar la cantidad inicial de feromona en los componentes de la solución. Finalmente el parámetro ‘t’ dará el trazado primario respecto al camino inicial de cada hormiga y el parámetro ‘el’ que muestra el coeficiente de eliminación de costos comunes o iguales, con el fin de buscar el objetivo de la solución.

ant_primaryplacing

Esta clase decalara una funcion [app] que designa el punto de inicio para empezar el tour, esta función en Matlab determina aleatoriedad, en este caso esta aleatoriedad se ve reflejada en el nodo de inicio de las hormigas.

Ant_cycle

Esta clase está declarada [at] en realizar el ciclo cerrado de las hormigas en los nodos, con el contenido heurístico y de feromona, adiciona también el punto inicial

del trazado de la hormona, todo en referencia a ‘c’ que es el ciclo o tour, este declara al atributo app es cada punto en el que empiezan el tour las 200 hormigas.

Es en esta clase que se aplica la

$$p(c_{i,x_i}|s^p) = \begin{cases} \frac{[\tau_{i,x_i}]^\alpha [\eta_{i,x_i}]^\beta}{\sum_{c_{j,x_j} \in J(s^p)} [\tau_{j,x_j}]^\alpha [\eta_{j,x_j}]^\beta} & \text{si } c_{i,x_i} \in J(s^p) \\ 0 & \text{en otro caso} \end{cases}$$

(Ecuación 2) y se evidencia que se abre un operador ‘for’ que inicia en la columna 1, aplicando el tour para las posiciones iniciales de las hormigas, cuando pasa por esas posiciones iniciales, a la matriz ‘h’ que define el inverso de la distancia toma el valor nominal de ‘mh’ que son los valores de la feromona, con esto y el valor de ‘t’ es posible calcular el numerador de la ecuación 3 que representa la regla de transición de estados, dentro de la construcción de soluciones para las hormigas.

El numerador de la ecuación es declarado por ‘temp’ y el denominador es ‘s’ que es la suma de todas aquellas soluciones parciales respecto a la relación de los valores de la feromona y la influencia relativa heurística, el inverso de este (1/s) es el denominador de la función. Teniendo en cuenta que: “dentro de la construcción de soluciones una hormiga construye incrementalmente una solución a partir de agregar componentes en una solución parcial (s), inicialmente vacía”, es por esto que s=0. Luego se evalúa si la solución es consideradamente buena para ser agregada al tour. Y así se genera un recorrido cerrado de hormigas durante un ciclo en la etapa de construcción de soluciones.

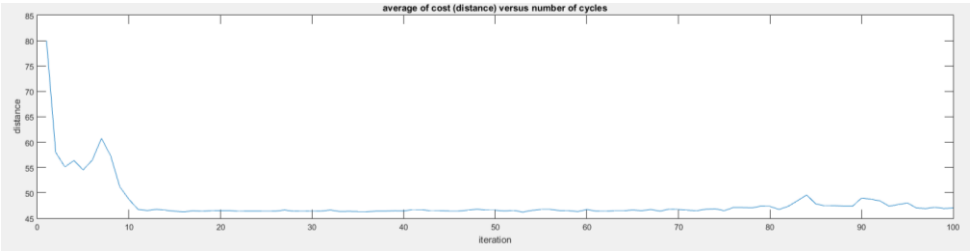
Ant_cost

La función costo está estrechamente ligada a la función objetivo, una función de calidad de la solución. Aquí se calcula se arregla en una matriz [cost,f] a todos los atributos que van encaminados a minimizar el costos del ciclo que es la función (f) objetivo, se incluye tanto los nodos como la cantidad de hormigas, también la matriz del ciclo cerrado y el coeficiente de eliminación de los costos comunes (el). Para una iteración donde todas las hormigas trabajan en busca de la solución, que está inicialmente vacía, se deben ajusta esa soluciones aprciales teniendo en cuenta la distancia entre nodos y así mismo el nodo de inicio de las hormigas que es el mismo donde cierra el tour; finalmente al evaluar cada solución (s) y darle su costos, se elimina el costo común.

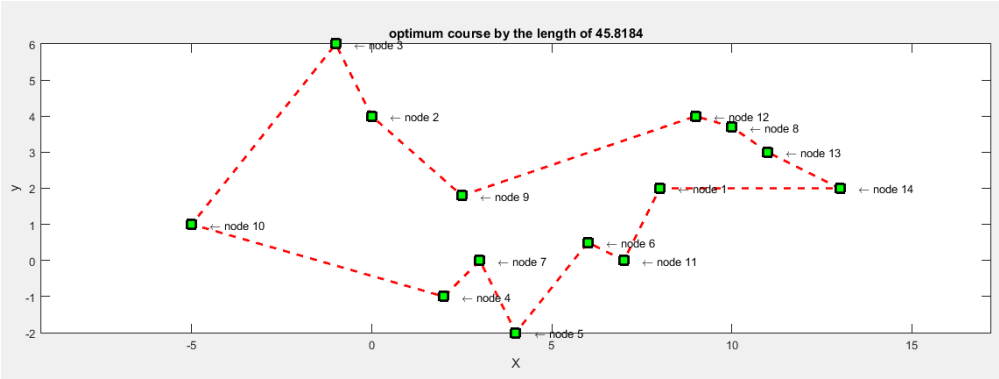
Ant_traceupdating

La clase de la actualización de la feromona, muestra que es necesario tener, el costo que es la función f, las hormigas, los nodos, el vector inicial de partida (at), el tour, y la tasa de evaporación de la feromona (e) entre (0,1). En este caso, para cada iteración del algoritmo respecto a las hormigas y a los nodos, como muestra la ecuación (HHH) ‘dt’ es el cambio en la feromona, que indicca que una vez que todas las homigas han construido sus soluciones, se realiza la actualización de la feromona mediante la aplicación de la regla de actualización (HHH), el objetivo es aumentar la cantidad de feromona en los componentes de la solución que han sido utilizados por las soluciones de calidad alta.

PLOTS



Esta gráfica es el resultado de las 100 iteraciones y es comparada con la distancia que es el costo promedio, se evidencia que al iniciar las iteraciones las distancias son muy grandes, que resulta más costoso a comparación con el aumento de los ciclos que hace que el costo disminuye. Puntualmente cerca de la iteración 10, el costo aun es de 60, pero al pasar las 10 iteraciones, este se estabiliza, es decir el 90% de las iteraciones son a un costo bajo.



El mejor tour encontrado fue en orden para los nodos: 1,11,6,5,7,4,10,3,2,9,12,8,13,14,1. Es el menos costoso y cumple con la función objetivo.

5.4.2 Ejemplificación de ruteo de vehículos.

Para resolver el problema del agente viajero se aplicaron diferentes tipos de heurísticos tales como Barrido, Ahorros, Vecino más próximo y Convex Hull-Nearest neighbor. **(Goetschalckx, 2011)**. El otro método aplicado fue la programación matemática a través de dos medios computacionales tales como OpenSolver y Solver Minto de AMPL® y finalmente la aplicación de metaheurísticos de optimización de colonia de hormigas con diferentes configuraciones.

El problema TSP que se analizó cuenta con 21 nodos, lo cual no es tan crítico para un problema de programación dinámica. La distancia aplicada es una distancia euclidiana, en la

$$d_E(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$ **Ecuación 5** encontramos a continuación, con un factor de desviación que es el k=1,25.

$$d_E(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$ **Ecuación 5**.

El modelo (*Asymmetric Travelig Salesman Problem- Variante 2*) de programación matemática del problema del agente viajero que se utilizó para correr el problema es (Goetschalckx, 2011):

Min $\sum_{i=1}^{N}\sum_{j=1}^{N} Cij\ Xij$

$$\sum_{i=1}^{N} Xij = 1\ \forall j$$

$$\sum_{i=1}^{N} Xij = 1\ \forall \text{i}$$

$$\sum_{K} q\ 1k = N$$

$$\sum q\ k1 = 1$$

Según los niveles de complejidad de los problemas varían dependiendo de los diferentes tipos de algoritmos y del tiempo de corrida que difiriere en cada uno, tal como se muestra en la tabla que se muestra a continuación:

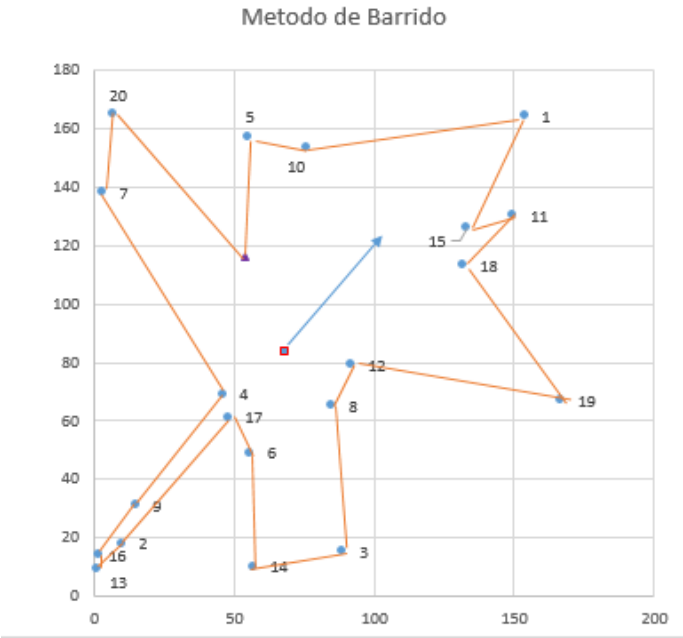
Tabla 7. Algorithm running times. (Goetschalckx, 2011)

Tiempo de Ejecución	Tamaño del Problema			
	10	20	40	80
n	0,001 seg	0,02 seg	0,04 seg	0,008 seg
n^3	0,001 seg	0,08 seg	0,064 seg	0,512 seg
2^n	0,001 seg	1,024 seg	12,43 días	37,43 Milenios de siglos

En la **Tabla 7** se observa el tamaño del problema y los tiempos de corrida de los algoritmos 'n' y n^3 2^n para problemas exponenciales, observando que la n es el número de nodos. El inter es analizar la calidad de los resultados, por tal motivo se corrió un problema de tamaño reducido a comparación con problemas de gran complejidad de rutas más larga. El tiempo de corrida no tiene relevancia en la investigación y análisis.

Para la solución de este problema de agente viajero se encuentra un archivo computacional adjunto de Excel donde están los cálculos usados para el desarrollo de cada uno de los diferentes algoritmos aplicados. Los Diagramas del tour de cada uno de los diferentes tipos de heurísticos y programación matemática utilizada se muestra a continuación.

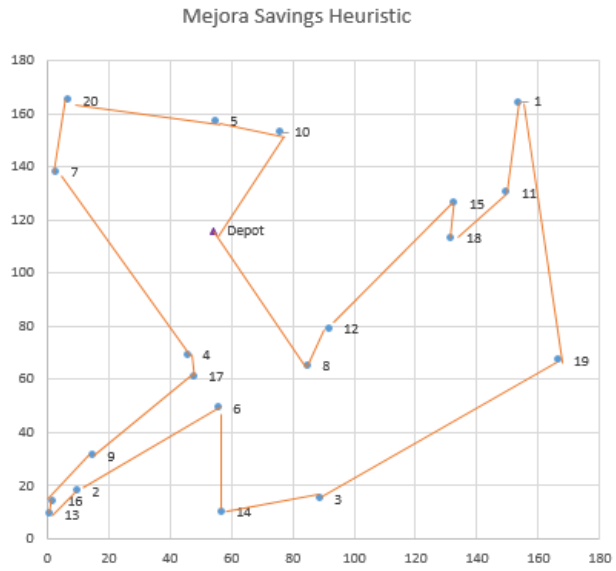
a) Barrido



Valor de la solución: 1049

Figura 23. Solución obtenida por el método de barrido

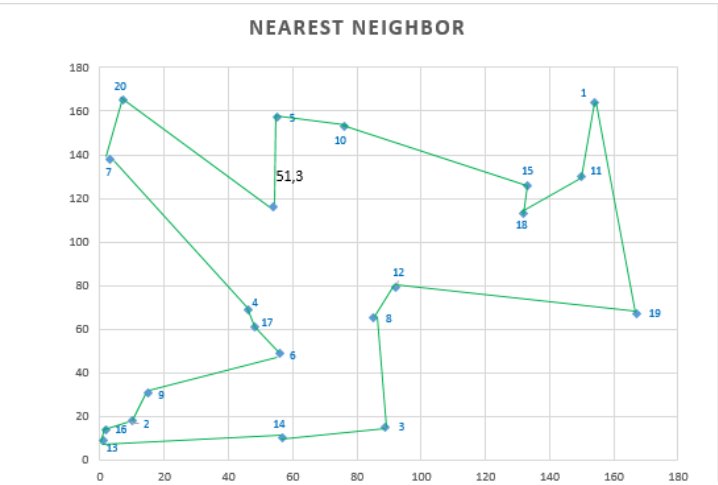
b) Método Ahorros



Valor de la solución: 978.98

Figura 24. Solución obtenida por Ahorros

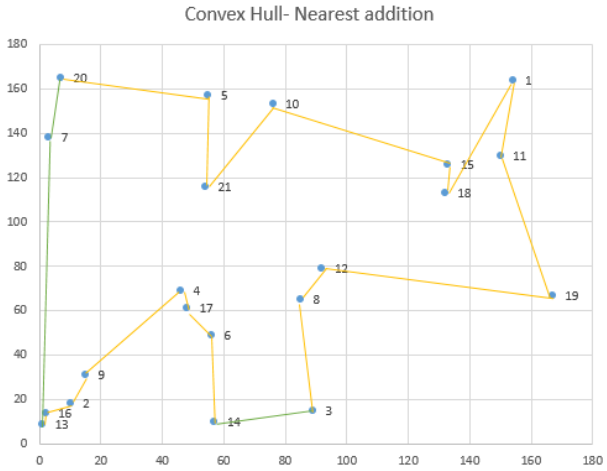
C) Vecino más Proximo



Valor de la solución: 1066.6

Figura 25. Solución obtenida por Vecino más Proximo

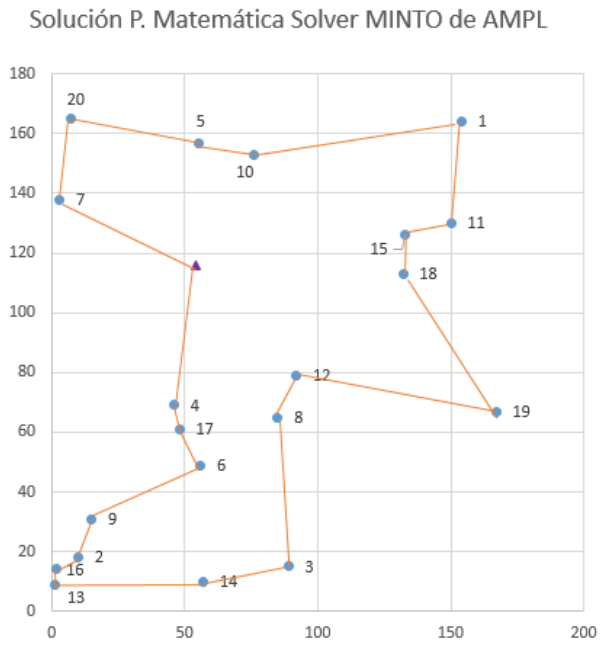
D) Convex Hull- Nearest Insertion



Valor de la solución: 1041.27

Figura 26. Solución obtenida por Convex Hull-Nearest Insertion

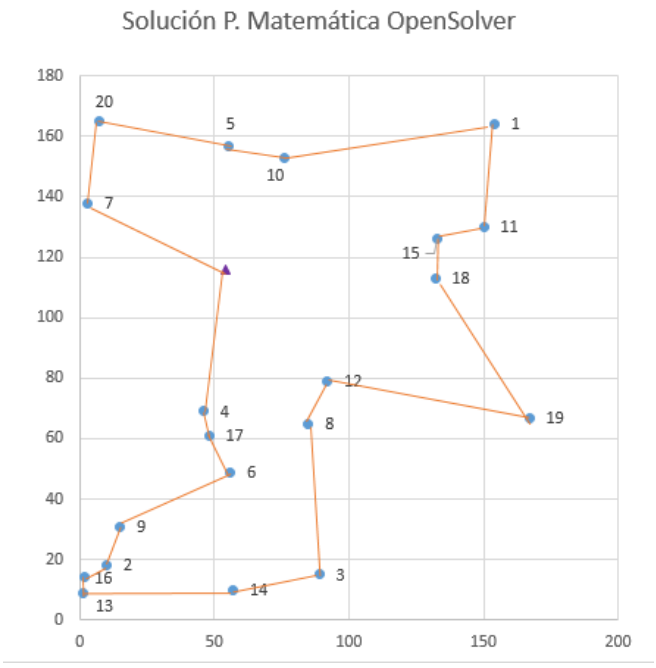
E) Programación matemática MINTO AMPL



Valor de la solución óptima: 915,405

Figura 27. Solución obtenida por MINTO AMPL

F) Programación Matemática OpenSolver



Valor de la solución óptima: 915,405

Figura 28. Solución obtenida por OpenSolver

Los resultados de metaheuristica de ACO se corrieron en el programa Matlab ® y los resultados obtenidos se consolidaron en la tabla comparativa.

Tabla 8. Tabla de comparación para los diferentes métodos usados (Fuente: Los autores)

	Comparativo	*Tiempo*	*F.O.*	*Suboptimalidad*
Heurísticos	Barrido	N.A.	1049	15%
	Ahorros	N.A.	978,98	7%
	Vecino más Proximo	N.A.	1066,6	17%
	Convex Hull + Nearest Addition Insertion	N.A.	1041,27	14%
Metaheuristicos (ACO)	ACO m= 300 iter = 150	20 seg	919,50	0,4%
	ACO m=500 iter = 950	1,51 min	915,31	0%
	ACO m=400 iter = 200	45 seg	949,23	4%
	ACO m=650 iter=450	60 seg	928,21	1,4%
	ACO m=800 iter=750	142 seg	922,28	1%
	ACO m=1000 iter=950	3 min	936,76	2%
	ACO m=2000 iter=950	6 min	922,28	0,8%
Programación Matemática	MINTO de AMPL	15 min	915,4	
	OPenSolver (Excel)	2 min	915,4	

En la **Tabla 8** se comparan los tiempos, resultados y suboptimalidad respecto a la mejor solución de corrida de los diferentes caminos que se usaron para solucionar el problema.

Se encontró la solución óptima a través de la optimización de colonia de hormigas con una combinación de m: 500 y 950 iteraciones. Lo cual se muestra a continuación.

Esta solución logra ser igual a la obtenida por los métodos exactos. Lo cual confirma que la metaheurística de optimización de Colonia de hormigas es un procedimiento eficiente por el cual se obtienen soluciones cuya diferencia con el óptimo es mínima como se observa en los porcentajes de suboptimalidad de la tabla 8.

Optimun course by the lenght of 925.31

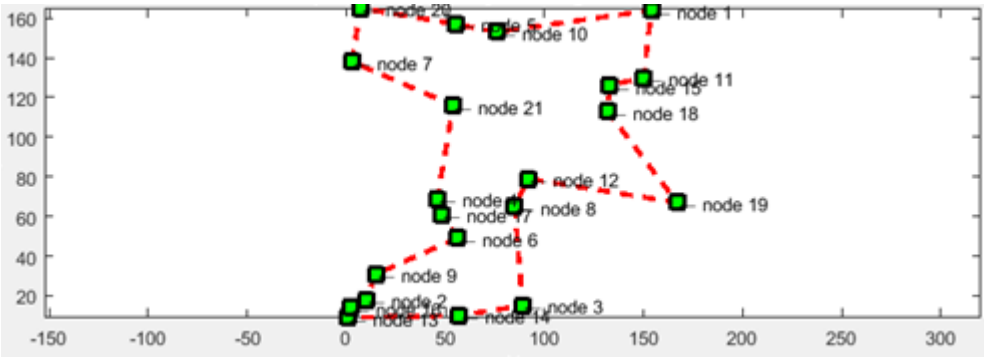


Figura 29. El tour óptimo de ACO obtenido de MatLab

Conclusiones de las aplicaciones algorítmicas.

El diseño de técnicas para resolver problemas en ingeniería es necesario en la estructuración del diseño de los procesos de calidad respecto a la cadena de suministro. La efectiva selección de un modelo, la fidelidad en la solución y la validación del mismo son necesarios para definir el impacto de la técnica. La resolución del modelo y los intentos de aportes a la solución constituyen el objetivo de muchas representaciones para la toma de decisión en la constitución de la solución, muchos de los problemas en logística son muy extensos y robustos para resolverlos por técnicas manuales y ad-hoc.

6 CONCLUSIONES

1. Se presentaron y se describieron os fundamentos que son punto de partida para la construcción de la línea de investigación basada en la ciencia biomímesis, específicamente en el subgrupo de la computación.

2. la Biomímesis se ha constituido en una fuente enorme de inspiración para la creación de nuevos diseños en productos y procesos con necesidades similares a las que organismos y sistemas naturales han enfrentado en su adaptación a un entorno.

3. La organizada lógica taxonómica desde la función de adaptación del organismo o sistema natural, permite obtener claridad y rápido acceso, a la identificación del sistema análogo para cualquier diseñador, ofreciendo un catálogo de estrategias que determinan el cómo se logra el fin o solución.

4. Durante la investigación y búsqueda bibliográfica se logró enfatizar en los algoritmos inspirados en los sistemas biológicos, como el algoritmo genético, la optimización por la colonia de hormigas, y los algoritmos inspirados en partículas, todos en el hecho que son usados para solucionar problemas de gran complejidad dentro de la logística, la cadena de abastecimiento y la manufactura.

5. Las metaheurísticas asociadas y sus aplicaciones en la industria y la búsqueda de soluciones son un campo en crecimiento debido a la variabilidad de problemas, su cantidad de parámetros y de combinaciones entre ellos permiten generar alternativas para los algoritmos en el proceso de la optimización de las soluciones o de la interpretación de los problemas generados, el número de metaheurísticas tiene un efecto directo en la complejidad en el uso y aplicación del algoritmo

6. Se evidenció el uso de la metaheurística de la colonia de hormigas en la optimización para el problema del agente viajero (TSP, Traveling Salesman Problem) como una herramienta esencial en la solución de problemas NP-Hard.

7. Con respecto al análisis teórico que se realizó para la programación del algoritmo, se observó que la selección de los parámetros iniciales, como

coordenadas, cantidades de hormigas, componente heurístico, entre otros, son determinantes en el rendimiento de la optimización.

8. En los algoritmos, ACO, se evidenció que poseen un componente estocástico, otro probabilístico, y otro referenciado al comportamiento de las hormigas y su relación química con las feromonas.

7 RECOMENDACIONES:

1. Se está por probar el nivel de impacto de la taxonomía en los usuarios, sin embargo su popularidad marca un éxito en el esfuerzo esquemático por hallar las líneas funcionales comunes y sus niveles jerárquicos inferiores que estructuren la taxonomía, se pretende entonces seguir en la búsqueda de literatura referente a las ramas enfocadas al área ingenieril, que no se desarrollaron en el documento y encontrar otro camino de bioinspiración para las soluciones algorítmicas.

2. Hoy día existen paquetes específicamente construidos, con la función de facilitar el uso y la aplicación de meta heurísticas. Se recomiendo continuar utilizando estas herramientas como “GLOBAL OPTIMIZATION TOOLBOX” de Matlab ® para mostrar desarrollo desde estas meta heurísticas.

3. Se busca ir desarrollando nuevas áreas de investigación donde existirá una oportunidad para explorar nuevos alcances de los algoritmos y las metaheurísticas, y esto es posible gracias a una investigación colaborativa desde muchas áreas inmersas desde la ciencia de la computación para la computación natural.

GLOSARIO

Para entender el desarrollo del principio de selección natural y el Algoritmo Evolutivo dentro de la Computación Evolutiva es relevante mostrar un glosario de términos claves.

Adaptación: proceso y resultado de la evolución natural, resulta en una mejor adaptación de una población a su hábitat. Ayuda a los organismos a sobrevivir en su nicho ecológico.

Codificación: el mapeo desde el fenotipo hasta el espacio genotípico.

Clave dicotómica: Herramienta que permite identificar a los organismos. Existen claves para cada una de las organizaciones de organismos –animales, plantas, hongos, procariotas-.

Decodificación: el mapeo inverso, desde el genotipo hasta el fenotipo.

Espacio fenotípico: espacio que contiene las soluciones candidatas. Para la computación Evolutiva.

Espacio genotípico: espacio que contiene las soluciones candidatas, estos puntos de solución están en el espacio donde la búsqueda evolutiva toma lugar. Para el algoritmo evolutivo.

Fenotipo: son las propiedades, características y rasgos observables de un organismo.

Feromona: Sustancia química utilizada en la comunicación entre organismos de la misma especie.

***Fitness*:** es la propensión a sobrevivir y reproducirse en un ambiente particular. Medida de la aptitud.

Genotipo: es el contenido genético de un organismo que se establece dentro del estado de factores hereditarios internos de un organismo, sus genes y genoma.

Nicho ecológico: describe la posición de una especie o población en un ecosistema

Sinapsis: Es la unión intercelular entre neuronas, en este punto de contacto se lleva a cabo la transmisión, ya sea química o eléctrica, del impulso nervioso.

Solución candidata: denota un punto en el espacio de posibles soluciones. Computación evolutiva y/o Algoritmo Evolutivo.

Población: cualquier grupo de organismos de la misma especie que ocupan un espacio particular. Se cruzan entre sí y habitan en un área particular en un tiempo determinado.

Auto replicación: Cualquier proceso por el cual una cosa puede hacer una copia de sí misma. Por ejemplo durante la división celular el ADN se replica y puede transmitirse a la descendencia durante la reproducción.

BIBLIOGRAFÍA

Alfonso, H., Salto, C., Minetti, G., Stark, N., Bermúdez, C., Orellana, A., … Lisi, I. (2010). Metaheurísticas aplicadas a problemas de optimización, 72–76.

Atehortúa, A. (2012). Optimización basada en Colonia de Hormigas: Generalidades y estudio del algoritmo Sistema Hormiga y aplicación a un Job Shop. *Tópicos Avanzados En Producción Y Logística*, *16*(09), 1–12.

Banzhaf, W. (2012). Evolutionary Computation and Genetic Programming, (July 2012), 1–49.

Bar-Cohen, Y. (2006). *BIOMIMETIC, Biologically Inspired Technologies.* (Y. Bar-cohen, Ed.). Taylor & Francis Group.

Bar-cohen, Y. (2012). Nature as a Model for Mimicking and Inspiration of New Technologies, *13*(1), 1–13. http://doi.org/10.5139/IJASS.2012.13.1.1

Bck, T., Kok, J., & Rozenberg, G. (2010). Computer Science and Natural Computing (pp. 6–11).

Binitha, S., & Sathya, S. S. (2012). A Survey of Bio inspired Optimization Algorithms. *International Journal of Soft Computing and Engineering (IJSCE)*, *2*(2), 137–151.

Brabazon, A., O`Neill, M., Editors, S., Th, R., Spaink, E. J. N. K. H. P., Board, A., … Winfree, D. W. E. (2006). *Biologically Inspired Algorithms for Financial Modelling*. (R. Th, E. J. N. K. H. P. Spaink, A. Board, A. G. Brassard, K. A. D. J. C. C. A. M. Gielen, T. H. L. Kari, … D. W. E. Winfree, Eds.). Springer.

Darwin, C. (1859). On the Origin of Species by Means of Natural Selection , or the Preservation of Favoured Races in the Struggle for Life, (February 2009), 204–208.

Deldin, J., & Schuknecht, M. (2014). Biologically Inspired Design, 17–28. http://doi.org/10.1007/978-1-4471-5248-4

Eiben, a. E., & Smith, J. (2003). Introduction to evolutionary computing. http://doi.org/10.1162/evco.2004.12.2.269

Eiben, A. E., & Smith, J. E. (2003). What is an Evolutionary Algorithm? *Introduction*

to Evolutionary Computing, 15–35. http://doi.org/10.1007/978-3-662-05094-1_2

Fogel, G. B., & Corne, D. W. (2002). An Introduction to Evolutionary Computation for Biologists (pp. 22–28). Morgan-Kaufmann.

Gendreau, M., & Potvin, J.-Y. (2010). *Handbook of Metaheuristics*. *Handbook of Metaheuristics* (Vol. 146). http://doi.org/10.1007/978-1-4419-1665-5

Goetschalckx, M. (2011). Supply Chain Engineering. In *Supply Chain Engineering* (pp. 15–58 y 213–268). Springer.

Griffis, S. E., Bell, J. E., & Closs, D. J. (2012). Metaheuristics in Logistics and Supply Chain Management. *Journal of Business Logistics*, *33*(2), 90–106. http://doi.org/10.1111/j.0000-0000.2012.01042.x

Hillier, F. S. (2010). *International Series in Operations Research & Management Science Series Editor :* (Vol. 157). http://doi.org/10.1007/978-1-4614-1900-6

Liao, C.-J., Gen, M., Tiwari, M. K., & Chang, P.-C. (2013). Meta-heuristics for manufacturing scheduling and logistics problems. *International Journal of Production Economics*, *141*(1), 1–3. http://doi.org/10.1016/j.ijpe.2012.09.004

Maniezzo, V., Gambardella, L. M., & Luigi, F. De. (2007). Ant colony optimization. *Swarm Intelligence*, *133*(1), 87–151. http://doi.org/10.1007/s11721-007-0005-x

Melanie, M. (1996). An introduction to genetic algorithms. *Cambridge, Massachusetts London, England, …*, 162. http://doi.org/10.1016/S0898-1221(96)90227-8

Siang, T. C. (2013). BIOMIMICRY ENGINEERING : NEW AREA OF TRANFORMATION. *IEEE Business Engineering and Industrial Aplications Colloquium (BEIAC)*, 477–482.

Solomon, E., Berg, L., & Martin, D. (2008). *BIOLOGÍA* (8th ed.). McGraw-Hill.

Suárez, O. D. A. (2011). Una aproximación a la heuristica y metaheuristicas. *IngeUan*, *1*(1), 44–51.

Tinsley, a, Midha, P., Nagel, R., & McAdams, D. (2007). Exploring the Use of Functional Models As a Foundation for Biomimetic Conceptual Design. *ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 1–15. http://doi.org/10.1115/DETC2007-35604

Xinjie, Y. X. Y. (2010). *Introduction to evolutionary algorithms*. *Computers and Industrial Engineering (CIE), 2010 40th International Conference on*. http://doi.org/10.1109/ICCIE.2010.5668407

8 ANEXOS

8.1 Taxonomía Biomimética.

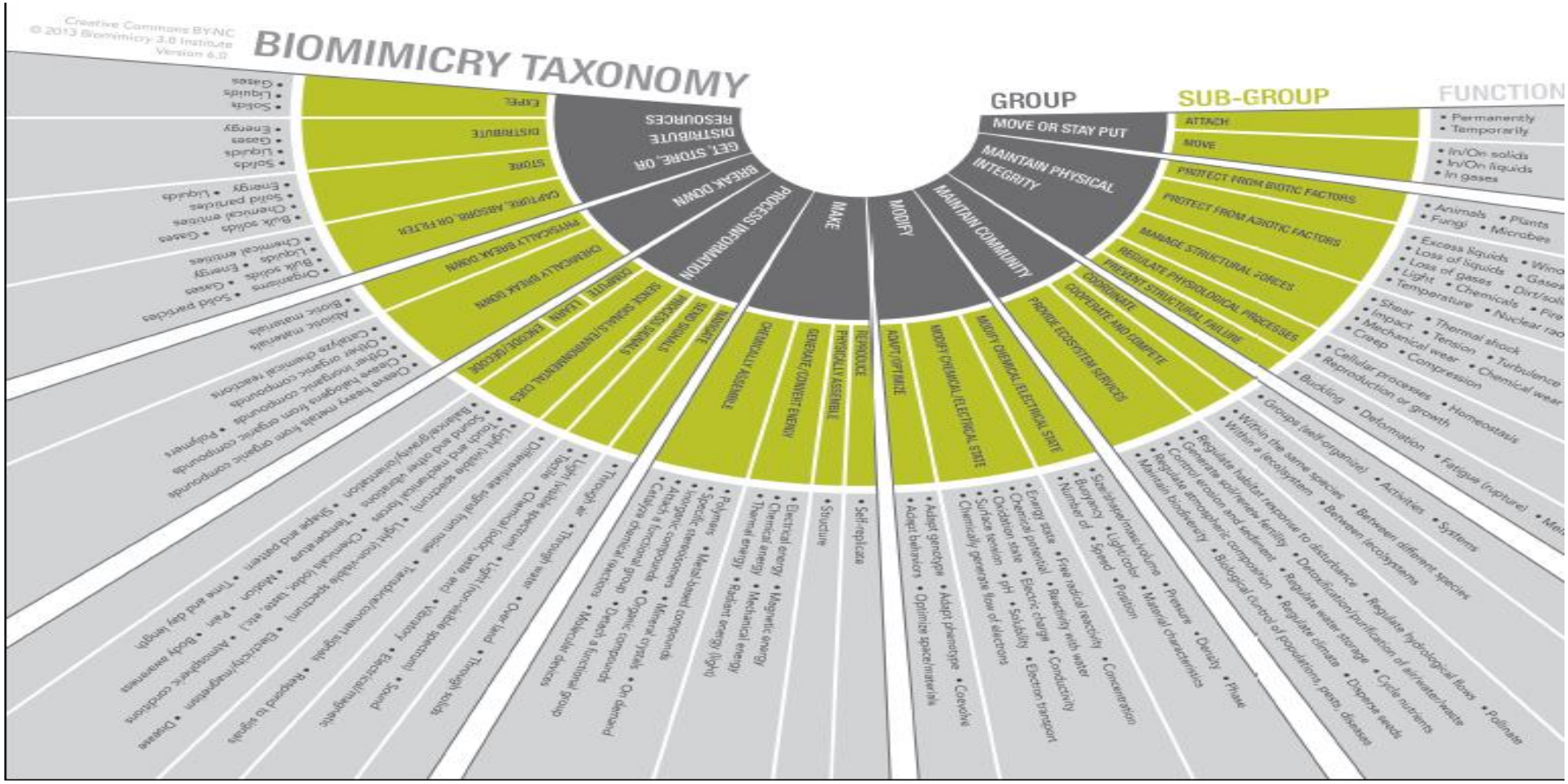


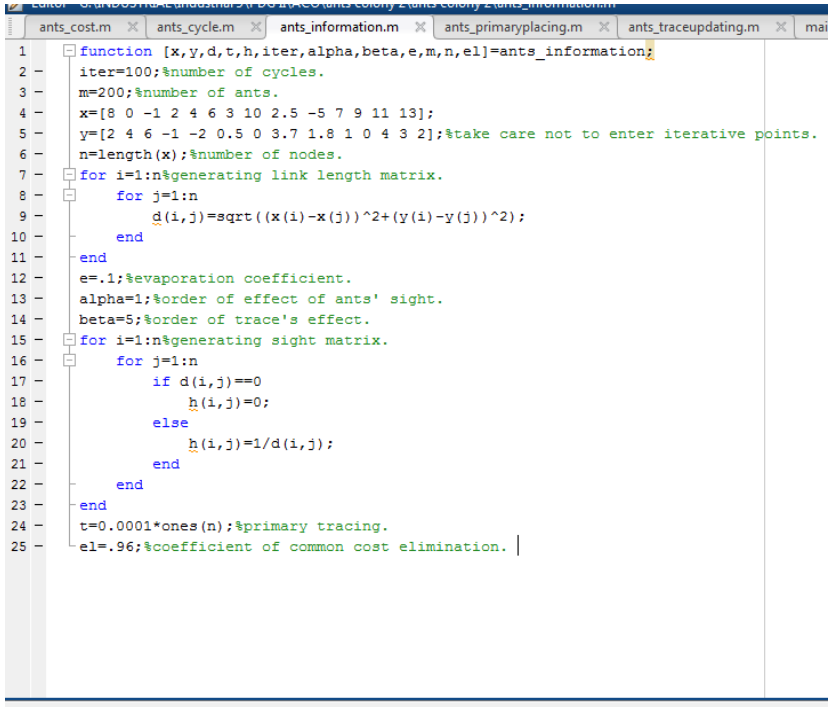
Figura 30. Diagrama de la taxonomía dentro de la Biomimética. Sistema de clasificación. (Tomado de: http://www.asknature.org/article/view/biomimicry_taxonomy)

8.2 Clases de la programación del algoritmo ACO.

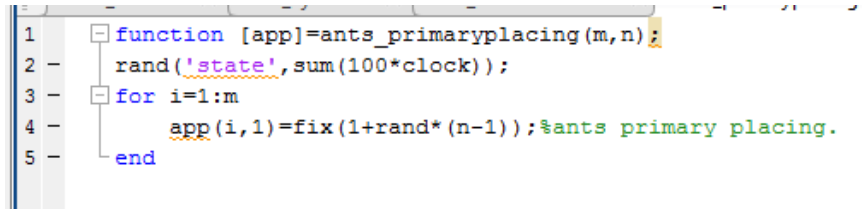
MAIN

```
[x,y,d,t,h,iter,alpha,beta,e,m,n,el]=ants_information;
for i=1:iter
    [app]=ants_primaryplacing(m,n);
    [at]=ants_cycle(app,m,n,h,t,alpha,beta);
    at=horzcat(at,at(:,1));
    [cost,f]=ants_cost(m,n,d,at,el);
    [t]=ants_traceupdating(m,n,t,at,f,e);
    costoa(i)=mean(cost);
    [mincost(i),number]=min(cost);
    besttour(i,:)=at(number,:);
    iteration(i)=i;
end
subplot(2,1,1);
plot(iteration,costoa);
title('average of cost (distance) versus number of cycles');
xlabel('iteration');
ylabel('distance');
[k,l]=min(mincost);
for i=1:n+1
    X(i)=x(besttour(l,i));
    Y(i)=y(besttour(l,i));
end
subplot(2,1,2);plot(X,Y,'--rs','LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',10)
xlabel('X');ylabel('y');axis('equal');
for i=1:n
    text(X(i)+.5,Y(i),['\leftarrow node ',num2str(besttour(l,i))]);
end
title(['optimum course by the length of ',num2str(k)]);
```

ANT_INFORMATION.



ANT_PRIMARYPLACING



ANT_CYCLE

```
function [at]=ants_cycle(app,m,n,h,t,alpha,beta);
for i=1:m
    mh=h;
    for j=1:n-1
        c=app(i,j);
        mh(:,c)=0;
        temp=(t(c,:).^beta).*(mh(c,:).^alpha);
        s=(sum(temp));
        p=(1/s).*temp;
        r=rand;
        s=0;
        for k=1:n
            s=s+p(k);
            if r<=s
                app(i,j+1)=k;
                break
            end
        end
    end
end
at=app;% generation of ants tour matrix during a cycle.
```

ANT_COST

```
function [cost,f]=ants_cost(m,n,d,at,el);
for i=1:m
    s=0;
    for j=1:n
        s=s+d(at(i,j),at(i,j+1));
    end
    f(i)=s;
end
cost=f;
f=f-el*min(f);%elimination of common cost.

```

ANT_TRACEUPDATING

```
function [t]=ants_traceupdating(m,n,t,at,f,e);
for i=1:m
    for j=1:n
        dt=1/f(i);
        t(at(i,j),at(i,j+1))=(1-e)*t(at(i,j),at(i,j+1))+dt;%updating traces.
    end
end

```