



resumen

En el artículo se describirá el trabajo realizado en el proyecto de grado “Bioescenarios, colonia de hormigas” que hace parte del proyecto “Bioescenarios”. En la primera parte del proyecto de grado se realizó una investigación centrada en las colonias de hormigas desde los puntos de vista biológico e informático. Teniendo en cuenta esta investigación, se desarrollaron los siguientes modelos de aprendizaje: conceptual, metodológico, virtual, físico y del programador.

## Escenarios para el aprendizaje de tendencias bioinspiradas

Daniel Felipe Jiménez B. • David Fernando Peñuela L.

La biología se ha convertido en un centro de investigación y en fuente de inspiración para la construcción de nuevas y mejores soluciones a problemas que se han tratado de resolver mediante el uso del computador y para los cuales, hasta el momento, no se tienen soluciones aceptables a través métodos convencionales, debido a la complejidad de los problemas y a lo difícil que resulta obtener una buena solución, en cuanto a espacio y tiempo se refiere. Es ahí donde los conceptos biológicos permiten, recurriendo a la abstracción de los mismos, desarrollar soluciones más apropiadas.

Una de las áreas en que los investigadores se han centrado es en el comportamiento de las colonias de hormigas, buscando el camino entre el hormiguero y su fuente de comida, abstraído a la informática para solucionar problemas de optimización y agrupando los cuerpos de hormigas muertas, abstraído a la informática para solucionar problemas de agrupamiento [9].

Bioescenarios es un proyecto que tiene como objetivo construir escenarios para el aprendizaje de conceptos y estrategias de tendencias informáticas inspiradas en biología. “Típicamente, un escenario estará compuesto por i) el concepto o estrategia de solución de problemas al cual se refiere el escenario; ii) material escrito (puede ser una guía de laboratorio); iii) material audiovisual que le permita al profesor

introducción



**seis**  
**2192**

redondear el trabajo que los estudiantes efectúan; y, iv) los modelos físicos o computacionales para que los estudiantes experimenten” [14]. Gracias a los bioescenarios se podrá enseñar de manera pedagógica los conceptos biológicos asociados a una o varias tendencias, para que luego sea posible implementarlos en la solución de problemas.

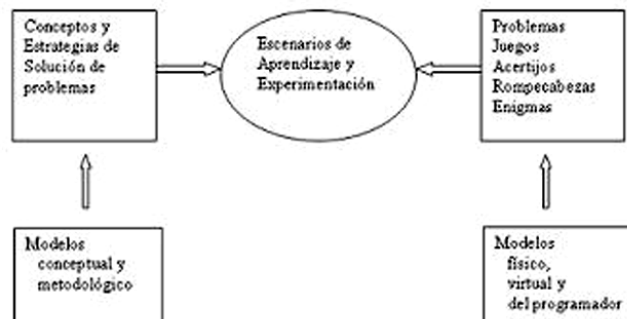
Este proyecto de grado, el segundo del proyecto Bioescenarios, tiene como propósito desarrollar modelos correspondientes a la tendencia de colonia de hormigas. En este artículo se presenta una síntesis del proceso de desarrollo, un resumen del marco teórico y la descripción de cada uno de los modelos desarrollados.

## Desarrollo

El trabajo se dividió en tres etapas: investigación general; definición de los modelos conceptual y metodológico; y, desarrollo de los modelos físico, virtual y del programador.

En la primera etapa, se seleccionaron tres tendencias de algoritmos bioinspirados: algoritmos inmunológicos, redes neuronales y colonia de hormigas, para hacer una investigación concienzuda de los conceptos de cada una de estas tendencias. La segunda etapa se desarrolló sobre redes neuronales y colonia de hormigas, dando como resultado un modelo conceptual abstraído de la biología y un modelo metodológico para la solución de problemas de cada una de estas tendencias. En la tercera etapa se crearon tres modelos basados en la teoría antes investigada sobre la tendencia de colonia de hormigas.

En la Figura 1 se puede observar claramente cada uno de los elementos necesarios para el desarrollo de los escenarios de aprendizaje y los modelos correspondientes.



**Figura 1.**  
**Modelo de escenarios.**



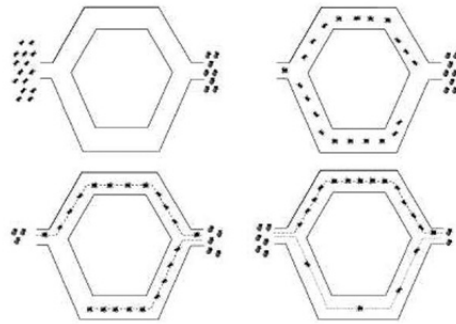


## Colonia de hormigas

Las hormigas son insectos sociales que, en conjunto, solucionan problemas complejos que no podrían resolver en forma individual. De esta manera, como colonia y mediante la comunicación existente entre ellas, pueden guiar su comportamiento para la solución de sus problemas, como por ejemplo, encontrar el camino más corto entre el hormiguero y la fuente de comida o agrupar las hormigas muertas.

Cuando las hormigas salen del hormiguero en busca de comida, se guían por el olor de feromonas<sup>1</sup> [13,11] que dejan sus compañeras o ellas mismas. Cuando existe rastro de feromona, las hormigas tienden a escoger el camino que este indica. Así, por ejemplo, en un cruce de caminos la hormiga tiene mayor probabilidad de seguir por la ruta cuyo rastro de feromona sea más fuerte. Cuando no existe ningún rastro, las hormigas se mueven aleatoriamente.

Cuando se tienen dos caminos que llevan a la comida, uno largo y otro corto, las hormigas que encontraron el camino corto van a demorarse menos en ir y volver desde el hormiguero hasta la fuente de comida, dejando feromonas por la ruta recorrida y haciendo el rastro de feromona más fuerte que el de las que recorrieron el camino más largo. Así, las que encontraron el sendero más largo, cuando vuelvan a salir se van a encontrar con un cruce de caminos, uno con un rastro de feromona más fuerte que el otro, y van a tener mayor probabilidad de recorrer el camino más corto esta vez (figura 2).



**Figura 2.**  
**Hormigas enfrentadas a caminos.**

El rastro de feromona se evapora con el tiempo, haciendo que cada vez se pierda más el rastro del camino más largo, ya que por la ruta más corta han pasado más hormigas, renovando el rastro en este camino y haciendo que el rastro sea tan intenso con respecto a los demás, que obliga a todas las hormigas a escoger esta vía.

## Modelo informático

Los algoritmos de ACO (*Ant Colony Optimization*) se basan directamente en el comportamiento de las colonias de hormigas reales, representando sus conceptos así:





<b>Biología</b>	<b>Informática</b>
Hormigas	Hormiga artificial (agente)
Feromona	Dato de feromona artificial
Comida	Nodo destino del grafo
Hormiguero	Nodo inicial del grafo
Obstáculo	Restricción
Vía	Aristas del grafo
Ecosistema de las hormigas	Medio (grafo)
Decisión de la hormiga	Heurística
Lugar de decisión	Nodo
Colonia	Colonia artificial

**Tabla 1.**  
**Comparación**  
**conceptos biología**  
**– informática.**

En la solución, cada agente resuelve el problema con un camino en el grafo. Cada arista del grafo representa cada paso que el agente puede dar y tiene asociados dos tipos de información que guían su movimiento:

- *Información heurística.* Mide la preferencia de que la arista sea seleccionada por los agentes, quienes no modifican esta información en la ejecución del algoritmo. Esta información la proporciona el programador o experto en el problema que se está solucionando, dando una función general calculada por la hormiga para cada nodo al cual puede pasar.
- *Rastro de feromona artificial.* Es la cantidad de feromona en una arista, modificada por las hormigas que ya pasaron por esa arista.

La probabilidad de que la hormiga escoja ir a un nodo se ve afectada entonces por la cantidad de feromona de la arista y el resultado de la función heurística que calcula en esta arista mediante una función de probabilidad, la cual tiene dos parámetros: el peso del valor de la heurística y el peso del valor de la feromona. Estos dan más o menos importancia a la heurística o a la feromona, afectando así la decisión final de la hormiga al escoger el nodo que va a pasar.

Existen dos formas de actualizar la feromona:

- Actualización, paso a paso. De esta manera las hormigas ponen una cantidad constante de feromona cada vez que van de un nodo a otro. Existen pequeñas variaciones que hacen que pongan más o menos feromona, según la cantidad de feromona presente antes de dar el paso.





- Actualización a posteriori. De este modo las hormigas ponen feromona después de haber encontrado una solución por todas las aristas por donde pasaron para encontrar la solución. Esto puede ser en una cantidad constante por todas las aristas, o dependiendo de la calidad de la solución encontrada, o de parámetros externos determinados por el programador, pueden poner más o menos feromona según el camino encontrado.

## Herramientas

Una parte clave del proceso de investigación es conocer las herramientas desarrolladas en torno a colonias de hormigas. Se presentarán a continuación tres grupos de herramientas: educativas, específicas y de propósito general.

**Herramientas educativas** Este grupo representa aplicaciones virtuales desarrolladas teniendo en cuenta los conceptos de la colonia de hormigas, que ayudan a entender e interactuar con dichos conceptos.

- *StarLogo (FreeWare)* [15]

Herramienta basada en la tortuga de Logo, que permite simular el comportamiento de una colonia de hormigas, permitiendo cambiar el número de hormigas, la fuente de comida y el porcentaje de feromona; utiliza su propio lenguaje procedimental (Logo).

- *Ants Viewer 1.0* [16]

*Applet* desarrollado bajo Java, que permite ver el comportamiento de la colonia de hormigas en su búsqueda de la fuente de comida.

- *Ant Farm Simulator (FreeWare)* [17]

Este juego, que simula la capacidad de las hormigas de coordinar en la tarea de reunión de alimentos, fue desarrollado en Visual Basic.

- *SimAnt* [27]

Juego que simula el comportamiento de una colonia de hormigas, que se enfrenta a otras colonias de hormigas por la supremacía de la colonia, al estilo *Age of Empires*.

## Herramientas específicas

Estas herramientas tienen como tarea fundamental resolver un problema específico por medio de la abstracción de los conceptos básicos de colonia de hormigas.

- *ACOTSP.V1.0 (GPL)* [18]

Está desarrollada en lenguaje Ansi-C para solucionar problemas del TSP. Los algoritmos utilizados son *Ant System*, *Elitist - Ant System*, *Max-Min*.





- *Antnet-1.1 (GPL)* [19]  
Permite solucionar el problema de encaminamiento de redes, el cual consiste en dirigir datos de una fuente a unos nodos destino, maximizando el funcionamiento de la red; está desarrollado en Omnet++, con el algoritmo Ant Net Routing.
- *Hc-mmas-ubqp (GPL)* [20]  
Herramienta exclusiva para la solución del hipercubo cuadrático (UBQP). Consiste en poner en práctica algoritmos de optimización de colonias de hormigas con intervalos de [0,1]. Fue desarrollado en C++ con el algoritmo de Min-Max.
- *GUIAnt-Miner (GPL)* [21]  
Herramienta para extraer reglas de clasificación de datos. Está desarrollada sobre Java con el algoritmo *Ant-Miner* (minería de datos).
- *Hamiltonian Cycle Problem (HCP)* [22]  
Prueba si un grafo contiene un ciclo hamiltoniano<sup>2</sup> o no.
- *Sequential Ordering Problem (SOP)* [23]  
Problema de TSP asimétrico, el cual consiste en dar una restricción de visitar un nodo determinado antes de pasar a otro.
- *Capacitated Vehicle Routing Problem (CVRP)* [24]  
Dado un conjunto de camiones, posibles elementos para cargar los camiones con sus pesos respectivos y las distancias hacia un destino, minimiza el número de viajes de los camiones, respetando la capacidad máxima de los mismos.
- *Atsp* [25]  
Se deben visitar todos los nodos desde un nodo inicial a un nodo final y viceversa, teniendo en cuenta que la distancia del nodo i al nodo j puede ser diferente del nodo j al nodo i, minimizando las distancias recorridas.

## Herramientas de propósito general

Estas herramientas se desarrollaron para tratar diferentes problemas con colonia de hormigas. En este caso sólo hay una herramienta debido a que no se encontraron más herramientas desarrolladas.

- *Operations Research 3.0* [26]  
Es necesario tener el software Matemática para poder usar esta herramienta, que utiliza el algoritmo “*Simulated annealing*”. Es una heurística para problemas de optimización global; es decir, encontrar una buena aproximación al óptimo global de una función en un espacio de búsqueda grande.





## Modelos desarrollados

Como ya se mencionó, se desarrollaron cinco modelos: conceptual, metodológico, físico, virtual y del programador para solución de problemas; a continuación se explicará cada uno de ellos.

## Modelo conceptual

Apoyándose en la investigación, se abstraieron los conceptos básicos del modelo biológico y así se desarrolló el modelo conceptual, para dar inicio y acercamiento a los algoritmos basados en colonias de hormigas para informática.

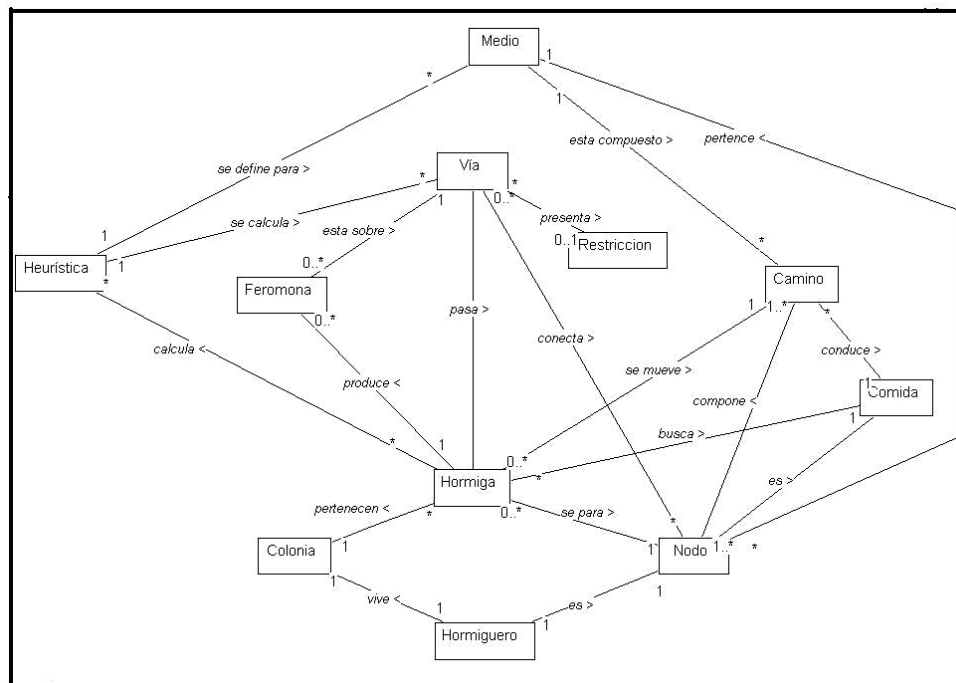


Figura 3. Modelo conceptual.

## Modelo metodológico

El modelo metodológico presenta los pasos para solucionar un problema, de manera fácil e intuitiva, usando la tendencia de las colonias de hormigas.





- **Represente el problema sobre un grafo**, permitiendo a las hormigas construir o modificar soluciones incrementalmente y definiendo:
  1. Punto inicial del problema (nodo inicial).
  2. Meta o solución (nodo final).
  3. Posibles pasos que se pueden dar del nodo inicial al nodo final.
- **Defina un método de restricción**, si es necesario, mediante el cual se obligue a las hormigas a la construcción o modificación de soluciones factibles (que cumplan con los requisitos del problema).
- **Defina una función de deseabilidad heurística**, que incida en la decisión de las hormigas para pasar de un nodo a otro en la construcción o modificación de la solución (en cada arista del grafo).
- **Defina una regla de actualización de feromona**, que especifique cómo actualizar el rastro de feromona en los diferentes pasos que pueden dar las hormigas (aristas del grafo).
- **Defina una regla de transición probabilística**, la cual dé la probabilidad afectada por la deseabilidad heurística y el rastro de feromona, para que la hormiga escoja el nodo a donde va a pasar.

## Modelo físico

El modelo físico muestra los conceptos básicos de colonia de hormigas en torno a un juego lúdico, donde la persona pueda aprenderlos de manera fácil y natural.

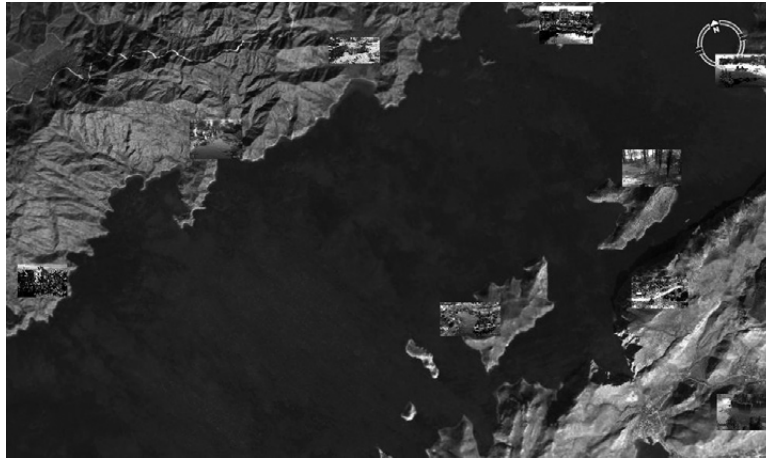
El modelo físico es el juego “El camino al tesoro”, basado en una historia con piratas: “Luego de haber robado el tesoro de la reina, los piratas pasaron por seis de las nueve ciudades que aparecen en el mapa, dejando pistas en cada una de estas ciudades relacionadas con el camino que siguieron y escondiendo el tesoro en la última ciudad. La reina ha contratado a expedicionarios para encontrar el camino por donde pasaron los piratas y, por consiguiente el tesoro” (ver Figura 4 página 62).

Los elementos del juego son un tablero, donde se realizará el mismo, las fichas que representan a cada persona dentro del juego, las tarjetas de las pistas y un dado (ver Figura 5 página 62).

Este juego permitirá aprender los conceptos claves de colonia de hormigas relacionados de la siguiente forma: el mapa es el grafo, las ciudades son los nodos, los expedicionarios







**Figura 4. Modelo físico (tablero).**



**Figura 5. Modelo físico (pista).**

son las hormigas, la información de las pistas sobre las vías son las feromonas y por último la experiencia de cada jugador y su inteligencia es la heurística.

Con todos estos elementos, lo interesante es ver cómo se aplican todos los conceptos de colonias de hormigas y utilizarlos en la solución de problemas asociados a esta tendencia.

## **Modelo virtual**

El modelo virtual se desarrolló con base en los conceptos de colonia de hormigas, tratando de mostrar los conceptos naturales en un entorno de ejecución informático. El





problema a resolver es precisamente el mismo de las hormigas: encontrar el camino más corto entre el hormiguero y la fuente de comida.

Con este modelo se pueden cambiar las variables de ejecución, como el número de hormigas, la heurística que se va a utilizar, si la hormiga tiene memoria o no, esto es, recordar el camino por donde llegó a la fuente de comida para devolverse por ese mismo, y la forma de actualización de feromona. Luego se pueden definir los nodos del grafo y las vías que los conectan.

Número de Hormigas

Heurística  Heurísticas:  
1. Menor Rotación  
2. Cambio de Dirección

Peso de la heurística

Hormiga con Memoria  1. Si  
2. No

Peso de la feromona

Cantidad de feromona a utilizar

Forma de actualización  1. Paso a Paso  
2. A posteriori

**SIGUIENTE!!!**  
**AYUDA!!!**

Figura 6. Configuración del modelo virtual.

Cuando comienza la ejecución, las hormigas salen del hormiguero y, dependiendo de la definición de las variables de ejecución, escogerán el camino siguiendo la función de probabilidad definida.

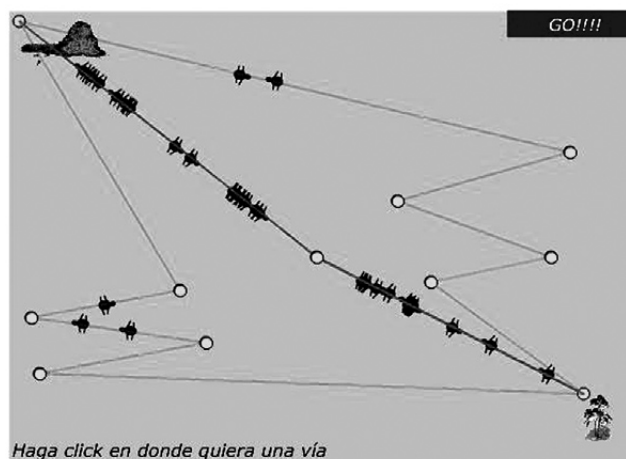


Figura 7. Ejecución del modelo virtual.





## Modelo del programador

El modelo de programador es un *framework*. En ingeniería de software, un *framework* es una estructura de soporte definida sobre la cual otro proyecto se puede organizar y desarrollar; típicamente, un *framework* puede incluir programas de soporte, bibliotecas y un lenguaje de *scripting* para ayudar a desarrollar y unir los diferentes componentes de un proyecto [28].

El modelo del programador es un paquete de software desarrollado en Java que modela los conceptos, las relaciones y las actividades generales que se pueden encontrar en las colonias de hormigas. Este componente se puede utilizar para construir la solución a problemas susceptibles de trabajarse usando la tendencia de colonia de hormigas.

El *framework* se desarrolló con base en el modelo conceptual de Colonia de Hormigas, lo cual permite a un estudiante, que ha asimilado los conceptos presentados en ese modelo, desarrollar fácilmente en un entorno informático soluciones a problemas utilizando esta tendencia.

Dentro del paquete del *framework* se encuentran disponibles las clases alusivas a cada uno de los conceptos del modelo conceptual con los atributos y métodos correspondientes a lo que el concepto conoce y hace, respectivamente. En el paquete concreto de solución estarán las clases que el programador crea para modelar y generar la solución a su problema específico usando y personalizando los elementos necesarios del *framework*.

En este momento, el *framework* está en su última parte de desarrollo.

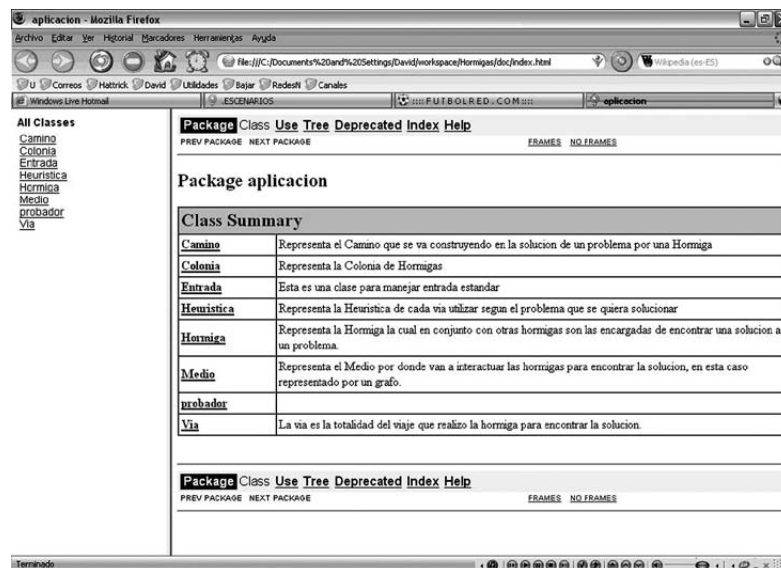


Figura 8.  
Modelo del programador.



## Conclusiones

Los algoritmos basados en la naturaleza sorprenden por su sencillez y su gran eficacia, demostrando que todavía hay muchas cosas por investigar para mejorar las soluciones que se tienen hoy en día.

Con el modelo físico se puede experimentar cambiando el significado de los conceptos y ejecutándolos de la misma manera para buscar una solución.

En el desarrollo del modelo virtual se comprobó que los algoritmos de colonias de hormigas, siguiendo la metodología propuesta, son sencillos para aplicarlos a los problemas. No se necesitan algoritmos sofisticados o difíciles de implementar para encontrar una solución factible.

El modelo del programador con el modelo metodológico permite solucionar problemas utilizando algoritmos de colonias de hormigas, siendo de gran utilidad si se entienden y se siguen los pasos que se deben dar para implementar la solución de un problema cuando se presente.

## Trabajos futuros

- Desarrollar escenarios para divulgar el modelo físico entre los interesados en el tema de bioescenarios para que aprendan de una manera divertida los conceptos de los algoritmos basados en colonias de hormigas.
- Formular problemas para resolver, utilizando los conceptos de las colonias de hormigas, con el modelo físico.
- Unir el modelo virtual con el modelo del programador (*framework*).
- Desarrollar mejores heurísticas y mejores reglas de actualización de feromona para el modelo virtual con el fin de ofrecer alternativas de exploración.
- Implementar otros problemas de optimización utilizando la metodología propuesta y el modelo del programador.
- Realizar la investigación y diseñar los modelos para el tema de agrupamiento de hormigas, que no se trató en este proyecto.
- Buscar más herramientas para las redes neuronales.
- Desarrollar los modelos correspondientes a los escenarios de redes neuronales.





## Referencias

- [1] Sergio, A. et. al. (2006) “La meta heurística de optimización basada en colonias de hormigas: modelos y nuevos enfoques”, en *Soft Computing and Intelligent Information Systems* [en línea] (6 de septiembre). [http://sci2s.ugr.es/publications/ficheros/OCH%20Modelos%20y%20Nuevos%20Enfoques%20\(Chapter\).pdf#search=%22algoritmo%20colonia%20de%20hormigas%22](http://sci2s.ugr.es/publications/ficheros/OCH%20Modelos%20y%20Nuevos%20Enfoques%20(Chapter).pdf#search=%22algoritmo%20colonia%20de%20hormigas%22).
- [2] Wikipedia, la enciclopedia libre [en línea] (2006), 6 de septiembre, disponible en <http://es.wikipedia.org/wiki/Feromona>.
- [3] Programa de Educación Continua (2006), “Algoritmos basados en colonias de hormigas”. Pontificia Universidad Javeriana [en línea] (6 de septiembre), disponible en: <http://fing.javeriana.edu.co/xi-stsiva-06/Hormigas.htm>.
- [4] Centro de Estudios de Posgrado (2006). “Master en ingeniería informática y de telecomunicación”. Universidad Autónoma de Madrid [en línea] (6 de septiembre 6), disponible en: [http://www.solocursos.net/master\\_en\\_ingenieria\\_informatica\\_y\\_de\\_telecomunicacion-slccurso2265685.htm](http://www.solocursos.net/master_en_ingenieria_informatica_y_de_telecomunicacion-slccurso2265685.htm).
- [5] Coello, A. (2006). “Introducción a la optimización evolutiva multiobjetivo”. Universidad de los Andes [en línea] (6 de septiembre), disponible en: <http://elavio2005.uniandes.edu.co/MiniCursosTutoriales/Coello%20-%20Coello%20syllabus-colombia.pdf#search=%22curso%20colonia%20de%20hormigas%22>.
- [6] Dorigo, M. y Gambardella, L.M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*.
- [7] Dorigo, M. y Gambardella, L.M. (2006). Ant colonies for the traveling salesman problem. Belgium. Université Libre de Bruxelles [en línea] (6 de septiembre), disponible en: <http://www.idsia.ch/~luca/acs-bio97.pdf#search=%22ant%20optimization%20results%20tsp%22>.
- [8] Dorigo, M. (2006). “Public Software” [en línea] (6 de septiembre), disponible en: <http://iridia.ulb.ac.be/~mdorigo/ACO/aco-code/public-software.html>.
- [9] Otero, J. y Guerrero, D. (2006). “Clasificación automática mediante colonia de hormigas”. Universidad de la Habana [en línea] (13 de septiembre), disponible en: [www.mfc.uclv.edu.cu/scmc/Boletin/N2/textos/Estad.%20y%20Probabilidades/JuanMOteroUH.doc](http://www.mfc.uclv.edu.cu/scmc/Boletin/N2/textos/Estad.%20y%20Probabilidades/JuanMOteroUH.doc).
- [10] Barán, B. y Almirón M. (2006) “Colonia de hormigas en un ambiente paralelo asíncrono”. Universidad Nacional de Asunción, Centro Nacional de Computación [en línea] (13 de septiembre), disponible en: [www.cnc.una.py/cms/invest/download.php?id=250344,48,1](http://www.cnc.una.py/cms/invest/download.php?id=250344,48,1).
- [11] Collin, A. (2006). “Ant Colony Algorithms”. *Dr. Dobb's* [en línea] (13 de septiembre), disponible en: <http://www.ddj.com/dept/ai/191800178?pgno=1>.
- [12] Anthes, G. (2006). “Ant Colony IT”. *Computer World* [en línea] (14 de septiembre), disponible en: <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=61394&pageNumber=2>.
- [13] Definición de palabras y sus raíces. (2006). [en línea] (3 de diciembre), disponible en: [clasicas.usal.es/dicciomed/fenol.htm](http://clasicas.usal.es/dicciomed/fenol.htm).
- [14] Chaparro Aguilar, R. (2006). “Construcción de escenarios para el aprendizaje de estrategias de solución de problemas y de conceptos fundamentales en informática”. Escuela Colombiana de Ingeniería.
- [15] StarLogo (15 de marzo de 2007), disponible en: <http://educationmit.edu/starlogo/>.
- [16] Ants Viewer 1.0, Jean-Philippe Rennard, Francia (15 de marzo de 2007), disponible en: <http://www.rennard.org/alife/english/antsgb.html>.





- [17] Ant Farm Simulator, Roberto Aguirre Maturana, Chile (16 de marzo de 2007), disponible en: <http://www.geocities.com/chamonate/hormigas/antfarm/indice.html>.
- [18] ACOTSPV1.0, Thomas Stütze (18 de marzo de 2007), disponible en: <http://www.aco-metaheuristic.org/>.
- [19] Antnet-1.1, Muddassar Farooq (18 de marzo de 2007), disponible en: <http://www.aco-metaheuristic.org/>.
- [20] hc-mmas-ubqp, Christian Blum (18 de marzo de 2007), disponible en: <http://www.aco-metaheuristic.org/>.
- [21] GUIAnt-Miner, Fernando Meyer y Rafael Stubs Parpinelli (21 de marzo de 2007), disponible en: <http://www.aco-metaheuristic.org/>.
- [22] Hamiltonian cycle problem. (21 de marzo de 2007), disponible en: [http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/hcp/ALL\\_hcp.tar](http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/hcp/ALL_hcp.tar).
- [23] Sequential ordering problem. (21 de marzo de 2007), disponible en: [http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/sop/ALL\\_sop.tar](http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/sop/ALL_sop.tar).
- [24] Capacitated vehicle routing problem. (21 de marzo de 2007), disponible en: [http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/vrp/ALL\\_vrp.tar](http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/vrp/ALL_vrp.tar).
- [25] Atsp (21 de marzo de 2007), disponible en: [http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/atsp/ALL\\_atsp.tar](http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/atsp/ALL_atsp.tar).
- [26] Operations Research 3.0, Addlink Software Científico. (21 de marzo de 2007), disponible en: <http://www.addlink.es/imprimirpag.asp?pid=31>.
- [27] SimAnt, Maxis Software Inc (Marzo 24 2007), disponible en: <http://www.abandonia.com/games/es/250/>.
- [28] Wikipedia enciclopedia (21 de abril de 2007) disponible en: <http://es.wikipedia.org/wiki/Portada>

## Notas de pie de página

<sup>1</sup>“Sustancia química producida por un organismo que sirve de estímulo para otros organismos de su misma especie o especies similares”.

<sup>2</sup>Es un ciclo simple que visita todos los vértices.

autores

**Daniel Felipe Jiménez B.** Estudiante de último año del Programa de Ingeniería de Sistemas de la Escuela Colombiana de Ingeniería Julio Garavito. Habilidades e interés en el campo de la ingeniería, en especial en ingeniería de software. Liderazgo y habilidades en manejo de relaciones sociales.

**David Fernando Peñuela L.** Estudiante de último año del Programa de Ingeniería de Sistemas de la Escuela Colombiana de Ingeniería Julio Garavito. Conocimiento e interés en bases de datos, gestión de tecnología y diseño de software. Habilidades para trabajo en equipo y excelentes relaciones interpersonales.

