

# Inteligencia de enjambres: sociedades para la solución de problemas (una revisión)

## Swarm intelligence: problem-solving societies (a review)

Mario A. Muñoz<sup>1</sup>, Jesús A. López<sup>2</sup> y Eduardo F. Caicedo<sup>3</sup>

### RESUMEN

En este artículo se presenta una revisión de los conceptos de inteligencia de enjambres, y algunas perspectivas en la investigación con estas técnicas, con el objetivo de establecer un punto de partida para trabajos futuros en diferentes áreas de la ingeniería. Para la construcción de esta revisión se llevó a cabo una búsqueda bibliográfica en las bases de datos más actualizadas de los artículos clásicos del tema y de las últimas aplicaciones y resultados publicados, en particular en las áreas de control automático, procesamiento de señales e imágenes, y robótica, extrayendo su concepto más relevante y organizándolo de manera cronológica. Como resultado se obtuvo taxonomía de la computación evolutiva, la diferencia entre la inteligencia de enjambres y otros algoritmos evolutivos, y una visión amplia de las diferentes técnicas y aplicaciones.

**Palabras clave:** algoritmos de optimización, computación evolutiva, inteligencia computacional, inteligencia de enjambres.

### ABSTRACT

This paper presents a review of the basic concepts of swarm intelligence and some views regarding the future of research in this area aimed at establishing a starting point for future work in different engineering fields. A bibliographic search of the most updated databases regarding classic articles on the subject and the most recent applications and results was used for constructing this review, especially in the areas of automatic control, signal and image processing and robotics. The main concepts were selected and organised in chronological order. A taxonomy was obtained for evolutionary computing techniques, a clear differentiation between swarm intelligence and other evolutionary algorithms and an overview of the different techniques and applications.

**Keywords:** computational intelligence, evolutionary computing, optimisation algorithm, swarm intelligence.

Recibido: abril 3 de 2008

Aceptado: junio 27 de 2008

### Introducción

Una de las aplicaciones más comunes de la inteligencia artificial (IA) es la búsqueda de la solución óptima en problemas de alta complejidad, tanto en espacios continuos como discretos. Un algoritmo de optimización es un método numérico que encuentra un valor  $\theta_i \in R^n$ , donde  $R^n$  es un espacio  $n$ -dimensional de búsqueda, que minimiza o maximiza una función  $J(\theta)$ , por medio de la selección sistemática de valores de la variable  $\theta_i$  posiblemente con algunas restricciones. La variable  $\theta_i$  puede ser un escalar o un vector de valores discretos o continuos llamados funciones factibles, mientras que  $J(\theta)$  es llamada función objetivo. Una solución factible que minimiza o maximiza la función objetivo es llamada una solución óptima. Un tipo de problemas de optimi-

zación son aquellos que requieren combinaciones de valores, y se le denomina de optimización combinatoria.

Según Hertz y Kobler (2000), las técnicas utilizadas para resolver problemas complejos de optimización combinatoria han evolucionado progresivamente de métodos constructivos a métodos de búsqueda local y finalmente a algoritmos basados en poblaciones. Estos últimos son muy populares actualmente puesto que proveen buenas soluciones al utilizar un método constructivo para la obtención de la población inicial, y una técnica de búsqueda local para mejorar la solución de la población. Además los métodos basados en poblaciones tienen la ventaja adicional de ser capaces de combinar buenas soluciones en orden de obtener unas mejores, ya que se considera que las buenas soluciones comparten componentes con las soluciones óptimas. A estos mé-

<sup>1</sup> Ingeniero electrónico. Candidato a Magíster, en Ingeniería Electrónica, Universidad del Valle, Colombia. Investigador, Grupo Percepción y Sistemas Inteligentes, Universidad del Valle, Colombia. andremun@univalle.edu.co.

<sup>2</sup> Ingeniero electricista. Magíster, en Automática y Doctor, en Ingeniería, Universidad del Valle, Colombia. Profesor Asistente, Universidad del Valle, Colombia. Investigador, Percepción y Sistemas Inteligentes, Universidad del Valle, Colombia. jesuslop@univalle.edu.co

<sup>3</sup> Ingeniero electricista. Master, en Tecnologías de la Información en Fabricación, Universidad Politécnica de Madrid, España. Doctor Ingeniero, en Informática Industrial, Universidad Politécnica de Madrid, España. Profesor Titular, Universidad del Valle, Colombia. Investigador, Grupo Percepción y Sistemas Inteligentes, Universidad del Valle, Colombia. ecaicedo@univalle.edu.co

todos se les conoce como algoritmos de computación evolutiva (*Evolutionary Computation – EC*).

Los algoritmos de EC comprenden un conjunto de técnicas iterativas que manejan una población de individuos que son evolucionados (modificados) mediante una serie de reglas que han sido claramente especificadas. En cada iteración hay periodos de autoadaptación, los cuales implican cambios en el individuo; son alternados con periodos de cooperación, lo que implica el intercambio de información entre individuos.

Debido a que varios algoritmos pueden ser descritos de esta manera, al considerar el diseño de un nuevo algoritmo de EC para la solución de un problema particular de optimización, se deben tener en cuenta las siguientes características principales:

- La existencia de *individuos* que pueden representar soluciones de un problema. Estas soluciones pueden ser factibles o no, parciales o completas, individuales o grupales.

- Un proceso de *evolución* que permite definir los cambios en la población a cada generación o de manera continua.

- Una definición de *vecindad* que permite conocer el modo como los individuos intercambian información. Puede ser estructurada si solo un grupo de la población puede conocer la información de un miembro, o por el contrario, todos los individuos pueden tener acceso a ella.

- Un mecanismo que permita identificar las *fuentes de información* de un individuo. Ya que la vecindad puede proveer una gran cantidad de información, se puede determinar cuál es más conveniente utilizar.

- Una medida de la *factibilidad* de la solución obtenida, lo cual permite determinar cuál es buena, óptima o inadecuada.

- Un mecanismo de *intensificación*, que corresponde al uso de métodos que puedan generar mejoras significativas durante la fase de autoadaptación. Este mecanismo realiza mejoras sobre un individuo sin tener en cuenta la información suministrada por otros individuos, permitiendo la intensificación de la búsqueda sobre algunas regiones del espacio.

- Un mecanismo de *diversificación*, el cual permite evitar convergencia hacia puntos óptimos locales. Este procedimiento modifica cada individuo independientemente, pero al contrario del mecanismo de intensificación, tiene resultados inesperados sobre un individuo. Un tipo de mecanismo de diversificación es el operador mutación en los GA.

Así, la EC representa un amplio grupo de técnicas, básicamente dividida en dos ramas, cuyas diferencias fundamentales se encuentran en la aplicación de los conceptos anteriores: los algoritmos evolutivos (*Evolutionary Algorithms – EA*) y la inteligencia de enjambres (*Swarm Intelligence – SI*).

- Los algoritmos evolutivos*: corresponden a un grupo de técnicas estocásticas que utilizan los conceptos de evolución biológica. Los EA actúan sobre una población de soluciones

potenciales aplicando los principios de diversidad de individuos y supervivencia del más fuerte para producir mejores aproximaciones a una solución. En cada generación es creado un nuevo grupo de aproximaciones por el proceso de selección de individuos de acuerdo a su nivel de “desempeño” en el dominio del problema, y se cruzan entre sí utilizando operadores que imitan los conceptos genéticos. Este proceso lleva a la evolución de poblaciones de individuos que están mejor adaptados a su ambiente. Sus principales técnicas son los algoritmos genéticos (*Genetic Algorithms – GA*), las estrategias evolutivas (*Evolutionsstrategie – ES*), la programación evolutiva (*Evolutionary Programming – EP*) y la programación genética (*Genetic Programming – GP*). Existen otros algoritmos que pueden ser catalogados dentro de esta rama, uno de ellos es la optimización por matrimonio de abejas (*Marriage in Honey Bees Optimization – MBO*), desarrollado por Abbass y Teo (2001), en donde se simula el proceso de apareamiento en una colonia de abejas.

- Inteligencia de enjambres*: corresponde a un grupo de técnicas que están basadas en el estudio del comportamiento colectivo en sistemas autoorganizados y descentralizados (distribuidos). Estos sistemas están conformados típicamente por una población de agentes computacionales simples capaces de percibir y modificar su ambiente de manera local. Tal capacidad hace posible la comunicación entre los individuos, que detectan los cambios en el ambiente generado por el comportamiento de sus semejantes. Aunque normalmente no hay una estructura centralizada de control que dictamina cómo los agentes deben comportarse, las interacciones locales entre los agentes usualmente llevan a la emergencia de un comportamiento global. Otra característica adicional es la inexistencia de un modelo explícito del ambiente. Dentro de esta rama las técnicas principales son la optimización por enjambre de partículas (*Particle Swarm Optimization – PSO*) y la optimización por colonia de hormigas (*Ant Colony Optimization – ACO*). Existen otros algoritmos que pueden ser catalogados dentro de esta rama, como lo son la optimización por enjambre de bacterias (*Bacteria Swarm Foraging Optimization – BSFO*), la búsqueda por difusión estocástica (*Stochastic Diffusion Search – SDS*) y el algoritmo de colmena de abejas artificiales (*Artificial Bee Hive Algorithm – ABHA*).

Al examinar las diferentes características de cada rama de la EC, se puede establecer una serie de diferencias entre las técnicas, siendo la más notoria la naturaleza centralizada de los EA frente al comportamiento distribuido en SI. Otra diferencia se encuentra en la interacción con la función objetivo, donde en los EA corresponde al nivel de “desempeño” y en la SI por lo general corresponde al ambiente a ser explorado. A su vez, los agentes en EA no pueden modificar su ambiente, mientras que en SI esta característica es fundamental. La Figura 1 muestra la taxonomía de EC propuesta.

En este artículo se presenta una revisión a los conceptos de las técnicas más importantes dentro de SI. Se enuncian los conceptos de las principales, junto con algunas de sus varia-

ciones y aplicaciones. Finalmente, se presentan algunas conclusiones sobre la teoría básica.

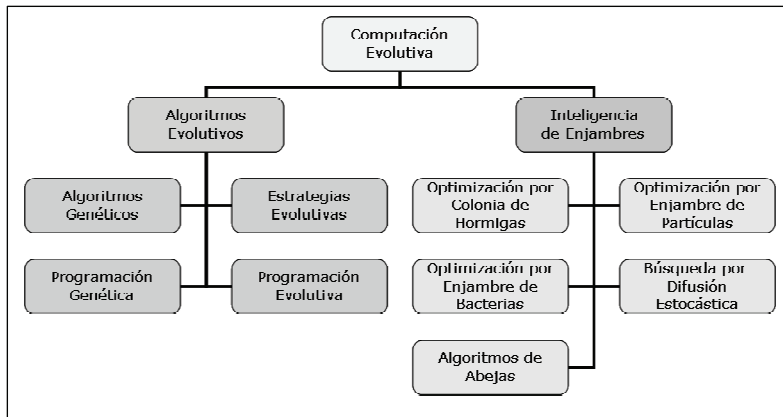


Figura 1: Taxonomía de la computación evolutiva

### Optimización por enjambre de partículas

La optimización por enjambre de partículas (PSO), desarrollada por Kennedy y Eberhart (1995), es un método de optimización para funciones no lineales en espacios continuos y discretos, basado en la simulación de un modelo social simple del desplazamiento de cardúmenes y bandadas (Kennedy y Eberhart, 1997). En un sistema PSO, la búsqueda se realiza utilizando una población de partículas que corresponden a los individuos, cada uno de los cuales representa una solución candidata al problema. Las partículas cambian su estado al “volar” a través del espacio de búsqueda hasta que se ha encontrado un estado relativamente estable. Un sistema PSO combina un modelo “únicamente social”, el cual sugiere que los individuos ignoran su propia experiencia y ajustan su conocimiento de acuerdo a las creencias exitosas de los individuos en la vecindad; y un modelo “únicamente cognitivo”, el cual trata a los individuos como seres aislados. Una partícula cambia de posición utilizando estos dos modelos (Ozcan y Mohan, 1999).

Dentro de PSO, la  $i$ -ésima partícula es tratada como un punto dentro de un espacio  $N$ -dimensional, representada por  $X_i = (x_{i1}, x_{i2}, \dots, x_{iN})$ . La mejor posición encontrada por la partícula anteriormente, o sea aquella donde se obtuvo el mejor valor en la función de costo, es representada como  $p_i = (p_{i1}, p_{i2}, \dots, p_{iN})$ . La mejor posición encontrada por el total de la población es representada por el símbolo  $g$ . La rata de cambio de posición (velocidad) para una partícula  $i$  es representada como  $v_i = (v_{i1}, v_{i2}, \dots, v_{iN})$ . Las partículas son manipuladas de acuerdo a las ecuaciones (1 y 2, donde  $c_1$  y  $c_2$  son dos constantes positivas,  $R_1$  y  $R_2$  son dos números aleatorios en el rango de  $[0, 1]$ , y  $w$  es el peso inercial. La ecuación (1 es utilizada para calcular la nueva velocidad de la partícula de acuerdo a su velocidad anterior y las distancias de su posición actual a su mejor posición y la mejor posición dentro del grupo. Luego la partícula se desplaza hacia una nueva posición de acuerdo a la ecuación 2. El peso inercial

$w$ , formulado por Shi y Eberhart (1998), es utilizado para controlar el impacto de las velocidades previas en la velocidad actual, influenciando el cambio entre las habilidades de exploración global (rango amplio) o local (rango corto) de las partículas. Un peso inercial mayor facilita la exploración global para buscar nuevas áreas, mientras que un peso inercial menor tiende a facilitar exploración local para sintonizar finamente el área de búsqueda actual. Una correcta selección del peso inercial puede proveer un balance entre las habilidades de búsqueda local y global, por lo tanto requiere menores iteraciones para encontrar el punto óptimo. En general el algoritmo realiza las operaciones mostradas en la Tabla 1.

$$v_{in}(t+1) = w \cdot v_{in}(t) + c_1 \cdot R_1 \cdot (p_{in} - x_{in}(t)) + c_2 \cdot R_2 \cdot (p_{gn} - x_{in}(t)) \tag{1}$$

$$x_{in}(t+1) = x_{in}(t) + v_{in}(t+1) \tag{2}$$

Tabla 1. Optimización por enjambre de partículas

```

Inicializar la población ubicándola
aleatoriamente
MIENTRAS no se cumpla el criterio de
terminación
  PARA i=1 HASTA n HACER
    Calcular el costo de la partícula  $J_i$ 
    SI  $J_i < p_{id}$ 
       $p_{id} = J_i$ 
    FIN SI
    SI  $J_i < p_{ig}$ 
       $p_{ig} = J_i$ 
    FIN SI
    Calcular la nueva velocidad de  $i$  con
    la ecuación (1)
    Calcular la nueva posición de  $i$  con
    la ecuación 2
  FIN PARA
FIN MIENTRAS
    
```

Conceptualmente el algoritmo de PSO es bastante simple, sin embargo existe interés en encontrar maneras de mejorar el desempeño del algoritmo por medio de varias técnicas. La primera modificación la realizaron Eberhart y Kennedy (1995) al cambiar el concepto vecindad, seleccionando en lugar de la mejor posición global  $p_g$ , la mejor posición de las  $m$  partículas más cercanas  $p_l$ . Clerc (1999), hace uso de un factor de constricción  $K$  que ayuda a la convergencia del algoritmo PSO al evitar que las partículas detengan su movimiento. Løvbjerg y Kink en sus trabajos de (2001, 2002) combinaron los conceptos de PSO con la reproducción encontrada en Algoritmos Genéticos, subpoblaciones y auto-organización crítica (*Self-Organized Criticality* - SOC), con el objetivo de aumentar la diversidad de la población. Miranda y Fonseca en sus trabajos (2002a, 2002b) presentan EPSO, el cual muta por medio de una distribución gaussiana los parámetros de inercia ( $w$ ), individual ( $c_1$ ) y social ( $c_2$ ) con el ob-

jetivo de autoadaptarlos durante la ejecución. Zhang y Xie (2003) incluyen un operador de evolución diferencial, que provee una mutación de tipo gaussiana sobre  $p_m$ , dando origen a la variación DEPSO. Monson y Seppi (2004) desarrollaron el KSwarm, que utiliza un filtro Kalman para actualizar las posiciones de las partículas. Poli et al., (2005) exploran la posibilidad de evolucionar la fuerza óptima generando ecuaciones para controlar las partículas por medio de programación genética. Grosan et al., (2005) presentan la variación con vecindades independientes (INPSO), donde se establecen subenjambres independientes, a diferencia de la definición anterior de vecindad. Tillett et al., (2005) introducen los conceptos de selección natural en la variación DPSO, donde se hace uso de sub-enjambres independientes, los cuales son reproducidos o eliminados dependiendo de su desempeño global. Liu y Abraham (2005) presentan las variaciones con turbulencia (TPSO) y con turbulencia adaptada difusamente (FATPSO), en las que la turbulencia evita la rápida convergencia al penalizar la reducción drástica de la velocidad de las partículas. Leontitsis et al., (2006) incluyen el concepto de repelencia del peor punto. Yisu et al., (2007) presentan las variaciones con vector de distribución (DVPSO), con habilidad de cruza (COPSO) y adaptable al entorno (LAPSO).

Otros campos de investigación sobre el algoritmo PSO corresponden al análisis matemático o estadístico para conocer mejor el comportamiento del sistema (e.g. Ozcan y Mohan, 1999; Clerc y Kennedy, 2002; Bartz-Beielstein et al., 2004). También se buscan nuevas aplicaciones del algoritmo donde se busca obtener mejores resultados que con otros métodos de optimización. Algunos ejemplos son el entrenamiento de redes neuronales reemplazando el algoritmo de entrenamiento por retropropagación (*backpropagation*) en redes tipo perceptrón multicapa (*Multilayer Perceptron* – MLP) (e.g. Salerno, 1997; El-Gallad et al., 2001) y de producto por unidad (*Product Unit Neural Network* – PUNN) (Ismail y Engelbrecht, 1999); la sintonización de controladores PID (Easter Selvan et al., 2003), robustos con ley  $H_2/H_\infty$  (Krohling et al., 2002), neuronales (E. Conradie et al., 2002) y predictivos (Yuan et al., 2006); la optimización de funciones de pertenencia en un conjunto difuso (Esmin et al., 2002); la identificación de modelos dinámicos (Voss y Feng, 2002); la identificación de grupos o *clustering* para la clasificación de imágenes (Omran et al., 2005) y otros problemas de optimización como el control de la potencia reactiva y voltaje (Yoshida et al., 2001).

## Optimización por colonia de hormigas

La optimización por colonia de hormigas (ACO) es una familia de algoritmos derivados del trabajo realizado por Dorigo et al., (1991a), basada en el comportamiento social de las hormigas, las cuales usan una forma de comunicación basada en sustancias químicas denominadas feromonas. Estas sustancias, depositadas por la hormiga al avanzar por un camino, ejercen una acción sobre la decisión de las hormigas precedentes, las cuales escogen el camino que posea una mayor concentración de sustancia, permitiendo que encuentren la

ubicación de las fuentes de alimento así como su nido. Se ha demostrado que los rastros de feromona permiten lentamente la optimización distribuida en la cual cada agente sencillo realiza una pequeña contribución en la búsqueda de la mejor solución.

En los algoritmos ACO cada agente construye una solución o parte de esta, comenzando de un estado inicial y desplazándose a través de una secuencia finita de estados vecinos, cuyo criterio de vecindad es dependiente del problema, haciendo uso de dos fuentes de información: la visibilidad y los rastros de feromona. La probabilidad de que la  $k$ -ésima hormiga se desplace del nodo  $i$  al nodo  $j$  está dada por la ecuación 3, siendo los nodos  $J_i^k$  los estados válidos. A su vez, los rastros son modificados para cambiar la representación del problema, que utilizarán las otras hormigas para tomar sus decisiones por medio de la ecuación 4, donde es el coeficiente de “evaporación” de la feromona usado para evitar la rápida convergencia de las hormigas hacia una región del espacio de búsqueda. Entonces la feromona se convierte en una memoria local compartida de largo término que influencia las decisiones subsecuentes de las hormigas. Las hormigas pueden actuar concurrente e independientemente, mostrando un comportamiento cooperativo que usa la *estigmergia*, una forma de comunicación indirecta por medio de la modificación del ambiente. Aunque cada una de las hormigas es capaz de encontrar una solución, probablemente inadecuada, las mejores soluciones son encontradas como el resultado de la cooperación global entre todos los agentes de la colonia, como una característica inesperada de la interacción cooperativa de los agentes.

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}(t)]^\beta}, & \text{si } j \in J_i^k \\ 0 & \text{si } j \notin J_i^k \end{cases} \quad (3)$$

$$\tau(t+1) = \rho \cdot \tau(t) + (1 - \rho) \cdot \Delta \tau(t, t+1) \quad (4)$$

Otras características usualmente incluidas dentro de un sistema ACO son el balance entre las características aleatorias y la severidad en la toma de decisiones (exploración/explotación del conocimiento); y estrategias de conocimiento global denominadas *daemon actions*. Las características de un algoritmo ACO pueden necesitar alguna clase de sincronización, usualmente obtenida por una programación secuencial, donde el conocimiento global es fácilmente accesible en cualquier instante. En general, un algoritmo de optimización por colonia de hormigas realiza las operaciones mostradas en la Tabla 2.

Las características de este grupo de algoritmos hacen que sean útiles en problemas estocásticos distribuidos donde la presencia de fuentes exógenas produce dinámicas en la presentación del problema. Por ejemplo, muchos problemas relacionados con redes de comunicación y transporte son intrínsecamente distribuidos y dinámicos, y a menudo es imposible obtener un modelo exacto de la variabilidad subyacente. Por el contrario, el uso de la *estigmergia* como único método

de comunicación entre las hormigas, los algoritmos ACO pueden presentar dificultades en problemas donde cada estado tiene un gran tamaño de vecindad, debido a sus características locales. De hecho, una hormiga que visita un estado con un gran tamaño de vecindad tiene un enorme número de posibles movimientos entre los cuales elegir, siendo la probabilidad de selección de cada estado muy pequeña, haciendo poco probable que varias hormigas visiten el mismo estado. Esta situación hace que la diferencia entre usar o no el rastro de feromona, en estos casos, sea muy pequeña o ninguna.

Tabla 2. Optimización por colonia de hormigas

```

PARA cada arco (i, j)
  Inicializar la feromona  $\tau_{ij}(0) = \tau_0$ 
FIN PARA
Ubicar las hormigas en nodos aleatorios
Obtener una solución inicial  $S_{min}$  con costo  $C_{min}$ 
PARA t=1 HASTA  $t_{max}$  HACER
  PARA k=1 HASTA m HACER
    PARA j=1 HASTA n•1 HACER
      Aplicar la regla de
      construcción/modificación en función
      de  $\tau$  y  $\eta$ 
    FIN PARA
    Calcular el costo  $C_k(t)$  de la solución
     $S_k(t)$ 
    SI  $C_k(t) < C_{min}$ 
       $C_{min} = C_k(t)$  y  $S_{min} = S_k(t)$ 
    FIN SI
  FIN PARA
  PARA cada arco (i, j)
    Actualizar los rastros de feromona de
    acuerdo a la regla de actualización
  FIN PARA
FIN PARA

```

Los algoritmos ACO tienen una naturaleza discreta, lo cual los hace útiles para la solución de problemas de optimización combinatoria. De hecho, el primer algoritmo básico de hormigas desarrollado, el *Ant System* (AS) (Dorigo et al., 1991a; Dorigo et al., 1991b; Colorni et al., 1991; Colorni et al., 1992), tuvo su primera aplicación en la solución del problema del viajero (*Traveling Salesman Problem* – TSP). En estos primeros trabajos ya se proponen variaciones al concepto de feromona, derivándose tres algoritmos del AS, el *Ant Quantity*, el *Ant Density* y el *Ant Cycle*. Otros problemas solucionados por medio del AS son la versión asimétrica del TSP, el *Quadratic Assignment Problem* (QAP) y el *Job-shop Scheduling Problem* (JSP) (Dorigo et al., 1996). Otros algoritmos ACO son: el ANT-Q, presentado por Gambardella y Dorigo en sus trabajos (1995, 1996), el cual se basa en el aprendizaje por refuerzo; el *Ant Colony System* (ACS), propuesto por Dorigo y Gambardella (1997), que difiere del AS al proveer una regla de transición de estados que balancea la exploración/explotación del conocimiento, la actualización global es realizada sobre el mejor recorrido y la actualización local es efectuada durante la exploración del espacio; el *min-max Ant System* (MMAS), propuesto por Stützle y Hoos (1997), donde la feromona está delimitada en un rango para

evitar el estancamiento; el *Rank Based Ant System* ( $AS_{rank}$ ), presentado por Bullnheimer et al., (1997), en el que la actualización de la feromona es llevada a cabo por las  $m$  mejores hormigas; el *Best-Worst Ant System* (BWAS), desarrollado por Cordon et al., (2000), el cual considera la actualización positiva para la mejor hormiga y negativa para la peor, la reinicialización de la búsqueda cuando ocurre estancamiento y la inclusión de mutaciones a la matriz de feromona para inducir variabilidad. Otras variaciones en el concepto de la actualización de la feromona se pueden encontrar en (Meuleau y Dorigo, 2000) donde se usa una regla derivada del descenso estocástico del gradiente (*Stochastic Gradient Descent* – SGD); en (Rubinstein, 2001) se utiliza el concepto de entropía cruzada, retomado en (Heegaard et al., 2004) de una manera distribuida.

Para su uso sobre espacios continuos se requiere un tratamiento diferente del problema donde la vecindad no sea un impedimento. El primer trabajo en este campo fue el de Bilchev y Parmee (1995), definiendo un nido del cual parten los agentes a explorar el espacio. Luego, Yan-jun y Tie-jun (2003) presentaron el *Adaptive Ant Colony Algorithm* (ACA), un algoritmo que utiliza un árbol binario para “mapear” un espacio continuo, obteniéndose una cadena binaria que representa la solución de manera similar a un algoritmo genético. Tsutsui (2004) propuso el sistema de agregación de feromona (*Aggregation Pheromone System* – APS), donde la feromona es una función de densidad continua.

## Optimización por enjambre de bacterias

La optimización por enjambre de bacterias representa una aproximación diferente a la búsqueda de valores óptimos en funciones no lineales desarrollado por Passino (2000), basado en el comportamiento quimiotáctico de la *E. Coli*. Si bien utilizar la quimiotaxis como modelo para optimización se propuso por primera vez en (Bremermann, 1974) y se ha utilizando en trabajos como (Leiviskä y Joensuu, 2006), el trabajo de Passino incluyó algunas modificaciones como la reproducción y la dispersión de los agentes.

La *E. Coli* es tal vez el microorganismo más comprendido, ya que su comportamiento y estructura genética están bien estudiados. Esta consta de una cápsula que incluye sus órganos, y flagelos que utiliza para su locomoción; posee capacidad de reproducirse por división y también es capaz de intercambiar información genética con sus congéneres. Además, puede detectar alimento y evitar sustancias nocivas, efectuando un tipo de búsqueda aleatoria, basado en dos estados de locomoción, el desplazamiento y el giro. La decisión de permanecer en uno de estos dos estados se debe a la concentración de nutrientes o sustancias nocivas en el medio. Este comportamiento se denomina quimiotaxis.

En este algoritmo, descrito en detalle en (Liu y Passino, 2002; Passino, 2002), se busca obtener el mínimo de una función  $J(\theta), \theta \in R^n$ , cuando se desconoce su gradiente. Suponiendo que  $\theta$  es la posición de la bacteria y  $J(\theta)$  representa la ubica-

ción de los nutrientes y sustancias nocivas, así  $\Delta J < 0$  un incremento en la concentración de nutrientes,  $\Delta J = 0$  representa un medio neutro y  $\Delta J > 0$  un incremento en la presencia de sustancias nocivas. Sea entonces  $P(j, k, l) = \theta_i(j, k, l) | i = 1, 2, \dots, S$  las posiciones de cada miembro en la población  $S$  de bacterias al  $j$ -ésimo paso quimiotáctico,  $k$ -ésimo paso reproductivo el  $l$ -ésimo evento de eliminación – dispersión. Sea  $J(i, j, k, l)$  el costo de la ubicación de la  $i$ -ésima bacteria  $\theta_i(j, k, l) \in R^p$ . Sea  $N_c$  el tiempo de vida de una bacteria medido en pasos quimiotácticos. Para representar un giro, se genera una unidad de longitud en una dirección aleatoria  $\psi(j)$ , por lo tanto la nueva posición de la bacteria se calcula por medio de la ecuación 5, donde  $C(i)$  es el tamaño del paso a tomar en la dirección especificada por el giro. Si en  $\theta_i(j+1, k, l)$  el valor de  $J(i, j+1, k, l)$  es menor que el obtenido en  $\theta_i(j, k, l)$ , entonces un nuevo paso quimiotáctico es tomado hasta un número máximo  $N_s$ .

$$\theta_i(j+1, k, l) = \theta_i(j, k, l) + C(i) \cdot \psi(j) \quad (5)$$

Adicionalmente, las bacterias mantienen comunicación haciendo uso de señales químicas que afectan el comportamiento quimiotáctico del enjambre, creando un ambiente cooperativo. A estas señales, denominadas atrayentes – repelentes, se las representa por  $J_{cc}(\theta, \theta_i(j, k, l))$  para la  $i$ -ésima bacteria. Sea  $d_{at}$  y  $w_{at}$  la profundidad y el ancho del atrayente,  $h_{re}$  y  $w_{re}$  la altitud y el ancho del repelente. La profundidad y el ancho de los atrayentes es relativamente pequeño a la concentración de nutrientes representados. Cada efecto atrayente y repelente es adicionado en la función de costo total de la bacteria como la suma de todos los efectos de una bacteria con todos los agentes de la población, de acuerdo a la ecuación 6.

$$J_{cc}(\theta, P(i, j, k)) = h_{re} \cdot \sum_{i=1}^S e^{-w_{re} \sum_{m=1}^p (\theta_m - \theta_m^i)^2} - d_{at} \cdot \sum_{i=1}^S e^{-w_{at} \sum_{m=1}^p (\theta_m - \theta_m^i)^2} \quad (6)$$

De acuerdo a la distribución de los agentes, se puede decir que entre más lejos se encuentre entre sí, el efecto de las señales será menor y por lo tanto no intentarán agruparse. Lo contrario ocurre cuando se encuentran muy cerca y por lo tanto los agentes se atraen para formar grupos cooperativos. Debido al movimiento de todas las bacterias, la función  $J_{cc}(\theta, P(j, k, l))$  es variante en el tiempo. Las bacterias, en efecto, no sólo intentarán buscar zonas con buenos nutrientes, evitando sustancias nocivas, sino que además intentarán moverse hacia otras bacterias, sin acercarse tanto entre sí. Por lo tanto, la función  $J_{cc}(\theta, P)$  deforma dinámicamente la superficie de búsqueda a medida que las bacterias se mueven.

Después de  $N_c$  pasos quimiotácticos se realiza un paso reproductivo, siendo  $N_{re}$  el número de pasos reproductivos. Durante la reproducción, las bacterias más saludables (las

que poseen un menor costo acumulado en su vida) se dividen y se eliminan el mismo número de bacterias no saludables, obteniendo así una población de tamaño constante. Luego se efectúa un evento de dispersión – eliminación  $N_{ed}$  en el que cada bacteria en la población es sujeta a la reubicación en una nueva posición aleatoria con una probabilidad  $p_{ed}$ . En general el algoritmo realiza las operaciones mostradas en la Tabla 3.

Tabla 3. Optimización por enjambre de bacterias

```

Inicializar la población aleatoriamente
Definir para cada individuo un  $C(i)$ 
PARA  $l=1$  HASTA  $N_{ed}$  HACER
  PARA  $k=1$  HASTA  $N_{re}$  HACER
    PARA  $j=1$  HASTA  $N_c$  HACER
      PARA  $i=1$  HASTA  $S$  HACER
        Calcular el  $J(i, j, k, l)$  incluyendo  $J_{cc}$ 
         $J_{last} = J(\bullet)$ 
        Calcular  $\bullet(j)$ 
        Realizar el desplazamiento (ecuación 5)
        Calcular el nuevo costo  $J(i, j+1, k, l)$ 
         $m=0$ 
        MIENTRAS  $m < N_s$ 
           $m=m+1$ 
          SI  $J(i, j+1, k, l) < J_{last}$ 
             $J_{last} = J(i, j+1, k, l)$ 
            Realizar el desplazamiento (ecuación 5)
            Calcular el nuevo costo  $J(i, j+1, k, l)$ 
          SINO
             $m=N_s$ 
          FIN SÍ
        FIN MIENTRAS
      FIN PARA
    FIN PARA
  FIN PARA
  Calcular la salud de cada bacteria
  Copiar las  $S_r$  bacterias más saludables
  Eliminar las bacterias inadecuadas
FIN PARA
Dispersar/Eliminar con una probabilidad  $p_{ed}$ 
FIN PARA

```

Este algoritmo se ha aplicado con éxito en la sintonización de controladores adaptativos (Passino, 2002), la sintonización de controladores PID (Kim y Cho, 2005) y la regulación de temperatura en superficies (Alfonso, 2007). Otros trabajos relacionados son el estudio de estabilidad realizados en (Gazi y Passino, 2003; Gazi y Passino, 2004) y la hibridización con algoritmos genéticos (Kim et al., 2007) para mejorar su desempeño. Otro algoritmo que utiliza el comportamiento quimiotáctico de las bacterias fue desarrollado por Müller et al., (2002).

## Búsqueda por difusión estocástica

La búsqueda por difusión estocástica es otro método de búsqueda genérico basado en poblaciones; creado por Bishop

en (1989a, 1989b), fue inspirado por el método de transformación Hinton (Hinton, 1981; Rumelhart y McClelland, 1986; Bishop y Torr, 1992), como una técnica de reconocimiento de patrones capaz de resolver el problema de equivalencia de estímulo: la habilidad de reconocer un patrón independiente de sus potenciales distorsiones o transformaciones en el espacio de búsqueda. En SDS, los agentes realizan evaluaciones parciales de una hipótesis, las cuales comparten a través de una comunicación directa uno a uno.

SDS está basada en el denominado “juego del restaurante” (De Meyer et al., 2003), donde un grupo de personas asisten a una conferencia en una ciudad desconocida y cada noche deben encontrar un lugar para comer. Puesto que hay una gran variedad de restaurantes y cada uno ofrece una variedad de platos, el grupo se encuentra ante la dificultad de encontrar el mejor restaurante, o sea aquel donde el mayor número de personas estén satisfechas comiendo. Puesto que una búsqueda paralela exhaustiva tomaría demasiado tiempo para obtener información válida, se instituye una política diferente. Cada individuo propone una hipótesis correspondiente al mejor restaurante en la ciudad. Cada noche los individuos prueban su hipótesis al comer en el sitio un plato al azar. A la mañana siguiente cada individuo que no obtuvo una buena experiencia le pregunta a un colega, al azar, cuál fue su experiencia. Si la información recibida es satisfactoria el individuo selecciona al restaurante como su favorito. De otra manera, selecciona un restaurante al azar. Usando esta estrategia, se ha encontrado que rápidamente un número significativo de individuos se congregan alrededor de la mejor solución. En términos generales, el algoritmo se comporta como se muestra en la Tabla 4.

Por medio de la iteración de fases de prueba y difusión, los agentes estocásticamente exploran todo el espacio de búsqueda. Sin embargo, puesto que las pruebas se realizan con mayor frecuencia sobre buenas soluciones que en regiones con información irrelevante, un agente tenderá a gastar una mayor cantidad de tiempo en regiones adecuadas y al mismo tiempo recluta más agentes, los cuales a su vez reclutan aún más. Las soluciones candidatas de alta calidad pueden ser identificadas por la concentración de agentes con la misma hipótesis.

La ventaja principal que presenta SDS es su capacidad de salir de mínimos locales. Esto se logra mediante la salida probabilística de la evaluación parcial de hipótesis en combinación con la reubicación de recursos (en este caso los agentes) por medio de mecanismos de reclutamiento estocástico. La evaluación parcial de la hipótesis permite al agente formarse rápidamente su opinión acerca de la calidad de la solución puesta a prueba sin una búsqueda exhaustiva (e.g. probando todos los platos disponibles en un restaurante).

A diferencia de muchas técnicas de búsqueda inspiradas en la naturaleza, existe un trabajo matemático muy profundo acerca del comportamiento y las propiedades básicas de SDS, de tal manera que es un algoritmo muy bien comprendido. Entre las características estudiadas están la convergen-

cia a la solución global óptima (Bishop y Torr, 1992; Nasuto et al., 1998), convergencia en el tiempo incrementando linealmente con el tamaño del espacio de búsqueda (Nasuto y Bishop, 1999), dinámicas de asignación de recursos (Bishop y Nasuto, 1999) y robustez (Myatt et al., 2004). En el contexto de clasificación de patrones, el algoritmo es robusto a distorsión por ruido y a múltiples instancias del objetivo (Nasuto, 1999).

Tabla 4. Búsqueda por difusión estocástica

```

Inicializar la población generando una hipótesis
aleatoria
REPITA
  Evaluar la evidencia de la hipótesis /*Etapa
  de prueba*/
  SI hipótesis comprobada
    Activar el agente
  SINO
    Desactivar el agente
  FIN SI
  SI agente inactivo /*Etapa de difusión*/
    Seleccionar un agente al azar
    SI el agente seleccionado es activo
      Recibir hipótesis apropiada
    SINO
      Generar una hipótesis aleatoria
  FIN SI
  FIN SI
HASTA cumplir el criterio de terminación

```

La técnica ha sido utilizada en una variedad de problemas de la vida real. La red híbrida de difusión estocástica (*Hybrid Stochastic Diffusion Network – HSDN*), una combinación de SDS con una red neuronal, fue usada para localizar características oculares en imágenes faciales humanas (Bishop y Torr, 1992) y para el seguimiento de labios en videos (Grech-Cini, 1995). En Beattie y Bishop (1998) SDS fue utilizado como método de autolocalización de una silla de ruedas autónomas, extendiendo el algoritmo original para proveer soluciones más rápidas en espacios de búsqueda mayores por medio de la Red enfocada de difusión estocástica (*Focused Stochastic Diffusion Network – FSDN*). Otros usos son la búsqueda de texto (Bishop, 1989a) y la selección de sitios para redes inalámbricas (Whitaker y Hurley, 2002).

## Colonias de abejas

A diferencia de otras técnicas de inteligencia de enjambres, donde se propuso una metaheurística y una serie de algoritmos se acogieron a ella, los algoritmos basados en abejas no cuentan con el mismo nivel de agrupamiento. Prueba de ello es la variedad de algoritmos que han surgido en los últimos años, desde la colonia artificial de abejas (*Artificial Bee Colony – ABC*) (Karaboga, 2005), pasando por la optimización por colonia de abejas (*Bee Colony Optimization – BCO*) (Teodorovic y Dell’Orco, 2005) y el algoritmo *Bees* (Pham, Ghanbarzadeh, Koç, Otri, Rahim y Zaidi, 2006), hasta la optimización por búsqueda de néctar por abejas (*Bee Nectar Search Optimization – BNSO*) (Alfonso et al., 2007) y el

algoritmo de colmena de abejas artificiales (*Artificial Bee Hive Algorithm* – ABHA) (Muñoz *et al.*, 2008). Por ahora el uso de estos algoritmos no es extensivo y la bibliografía es limitada.

Una primera metaheurística propuesta dentro de los algoritmos de abejas corresponde a BCO. Sus autores proponen un marco de referencia donde las abejas artificiales realizan movimientos de construcción y selección de soluciones. En el proceso de decisión, se asume que cada abeja puede obtener información acerca de la calidad de las soluciones generadas por la población. De esta manera, las abejas intercambian información acerca de la calidad de las soluciones parciales creadas. Basándose en la calidad, cada una decide si abandonar o no su solución parcial y convertirse en un seguidor no comprometido, si continuar con la expansión de su solución parcial sin reclutar a otras compañeras, o si danza y recluta a otras compañeras antes de regresar a su solución parcial. Dependiendo de la cualidad de las soluciones parciales generadas, cada abeja posee un nivel de lealtad hacia la solución parcial descubierta. El proceso iterativo permite que las soluciones parciales se completen hasta obtener una solución completa factible.

Como ejemplo de los algoritmos de abejas se presenta el algoritmo ABHA. Este algoritmo permite realizar búsquedas sobre espacios continuos, basados en el modelo orientado al individuo desarrollado en de Vries y Biesmeijer, (1998), donde se define una serie de reglas de comportamiento para cada agente que determinan las acciones a realizarse, incluyendo algunas adaptaciones no consideradas en el modelo biológico para incrementar el desempeño en la búsqueda de mejores soluciones. En términos generales, el algoritmo se comporta como se muestra en la Tabla 5.

Pese a ser relativamente recientes, los algoritmos de abejas ya presentan alguna variedad de aplicaciones, principalmente propuestas por sus creadores. Algunas de estas son la solución del problema del viajante (Lucic y Teodorovic, 2003), el entrenamiento de redes neuronales (Pham, Koç, Ghanbarzadeh y Otri, 2006), la sintonización de controladores difusos (Pham, Darwish, Eldukhr y Otri, 2007), la optimización de máquinas de soporte vectorial (Pham, Muhamad, Mahmuddin, Ghanbarzadeh, Koç y Otri, 2007), el diseño de celdas de manufactura (Pham, Afify y Koç, 2007), el diseño de uniones por soldadura (Pham y Ghanbarzadeha, 2007) y la regulación de temperatura sobre una superficie (Alfonso, 2007).

## Conclusiones

La SI comprende una amplia variedad de técnicas auto-organizadas y distribuidas que comparten características comunes como la existencia de una población de agentes computacionales simples capaces de percibir y modificar su ambiente de manera local, lo que hace posible la interacción entre los individuos; la inexistencia de una estructura centralizada de control que dictamina cómo los agentes deben comportarse, o de un modelo explícito del ambiente. El comportamiento global, por tanto, es producto de las inte-

racciones locales entre los agentes, al contrario de un conocimiento global.

Tabla 5. Algoritmo de colmena de abejas artificiales

```

Inicialice la población
PARA i=1 HASTA I HACER
  Calcule las probabilidades
  SI es experimentada y su información es válida
    Incremente la tendencia de abandono y deje el nido
  FIN
  SI no hay danzas disponibles
    SI puede iniciar una búsqueda aleatoria
      Ubique las abejas en posiciones aleatorias y deje el nido
    FIN
  SI abandonó la fuente
    Olvide la fuente y permanezca en el nido
  FIN
  SINO
    Calcule la duración de la danza
    SI sigue una danza
      Seleccione una danza por ruleta
      Agregue ruido a la información y deje el nido
    SINO
      Incremente la tendencia de abandono
    FIN
    SI puede iniciar una búsqueda aleatoria
      Ubique las abejas en posiciones aleatorias y deje el nido
    FIN
  FIN
  Calcule el costo actual
  SI puede seguir buscando
    SI la dirección es válida
      Incremente la posición
    SINO
      Cambie de dirección e incremente la posición
    FIN
  SINO
    Determine si la fuente es válida
  FIN
  Actualice el umbral y la memoria
FIN

```

Es notoria la diferencia entre la SI y la otra rama dentro de la EC, los EA, siendo las más importantes la naturaleza centralizada de los EA frente al comportamiento distribuido en SI; la interacción con la función objetivo, donde en los EA corresponde al nivel de “desempeño” y en la SI por lo general corresponde al ambiente a ser explorado; y la capacidad de modificar su ambiente de los algoritmos de SI.

Las aplicaciones de SI en la actualidad son muy diversas, desde las redes de comunicación, los sistemas de control automático, la solución de problemas combinatorios, el enrutamiento de vehículos, entre otros. En cada momento se encuentran nuevas aplicaciones y se desarrollan variaciones que mejoran el desempeño de los algoritmos como la hibri-



dización con los EA. Al mismo tiempo, se hace necesario el estudio profundo de las cualidades de estabilidad y convergencia de estos métodos. Además, el estudio detallado de algunos biólogos permite encontrar nuevos modelos que pueden llevar a proponer nuevos algoritmos de SI. En este último caso, el desarrollo de las técnicas de SI implica el trabajo conjunto de disciplinas tan diversas como las ciencias biológicas, sociales, computacionales, la ingeniería, la filosofía, entre otras, siendo un paradigma bastante rico para su exploración por su mezcla de naturaleza y computación (Nunez de Castro, 2007). Existen aún muchas sociedades naturales que pueden ser simuladas o emuladas en computadores, y los frutos de estas exploraciones se convierten en nuevas soluciones tecnológicas a problemas clásicos o recientes, y su completo potencial todavía está por alcanzarse.

Con el desarrollo de sistemas de cómputo de mayor poder, la aparición de nuevos problemas complejos y la colaboración entre científicos de diversas áreas, el futuro de las técnicas de IA, en particular de SI, es bastante prometedor.

## Agradecimientos

Este trabajo fue posible gracias al financiamiento de Colciencias y la Universidad del Valle, a través de una beca de investigación otorgada a Mario A. Muñoz, por medio del proyecto "Plataforma de procesamiento distribuido, con acceso remoto multiusuario y emulación de sistemas dinámicos, para la investigación y educación en ingeniería", Contrato No. 1106-11-17707.

## Bibliografía

- Abbass, H. A., Teo, J., A true annealing approach to the marriage in honey-bees optimization algorithm., *Memorias del Inaugural workshop on Artificial Life (AL'01)*, 2001.
- Alfonso, W., Regulación de temperatura en la plataforma UV-PTM01 basada en agentes cooperativos para la asignación dinámica de recursos., tesis presentada a la Universidad del Valle para optar al grado de Ingeniero Electrónico, 2007.
- Alfonso, W., Muñoz, M., López, J., Caicedo, E., Optimización de funciones inspirada en el comportamiento de búsqueda de néctar en abejas., *Memorias del Congreso Internacional de Inteligencia Computacional (CIIC2007)*, 2007.
- Bartz-Beielstein, T., Parsopoulos, K., Vrahatis, M., Analysis of particle swarm optimization using computational statistics., *Memorias del International Conference of Numerical Analysis and Applied Mathematics (ICNAAM)*, 2004, pp. 34–37.
- Beattie, P., Bishop, J., Self-localisation in the 'senario' autonomous wheelchair., *Journal of Intelligent and Robotic Systems*, Vol. 22, 1998, pp. 255–267.
- Bilchev, G., Parmee, I., The ant colony metaphor for searching continuous design spaces., *Evolutionary Computing, Selected Papers AISB Workshop*, Vol. 993 of Lecture Notes in Computer Science, T. Fogarty (ed.), Springer, 1995, pp. 25–39.
- Bishop, J., Anarchic Techniques for Pattern Classification, tesis presentada a la University of Reading para optar al grado de Doctor of Philosophy, 1989a.
- Bishop, J., Stochastic searching Networks., *Memorias del First IEE Conference on Artificial Neural Networks*, 1989b, pp. 329–331.
- Bishop, J., Nasuto, S., Communicating neurons – an alternative connectionism., *Memorias del Weightless Neural Networks Workshop*, 1999.
- Bishop, J., Torr, P., The stochastic search network., *Neural Networks for Images, Speech and Natural Language*, R. Linggard, D. Myers y C. Nightingale (eds.), Chapman & Hall, 1992, pp. 370–387.
- Bremermann, H., Chemotaxis and optimization., *Journal of the Franklin Institute*, Vol. 297, No. 5, 1974, pp. 397–404.
- Bullnheimer, B., Hartl, R. F., Strauss, C., A new rank-based version of the ant system: a computational study., *Technical Report POM-03/97*, Institute of Management Science, University of Vienna, 1997.
- Clerc, M., The swarm and the queen: towards a deterministic and adaptive particle swarm optimization., *Memorias del 1999 Congress on Evolutionary Computation (CEC 99)*, 1999.
- Clerc, M., Kennedy, J., The particle swarm—explosion, stability, and convergence in a multidimensional complex space., *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, 2002, pp. 58–73.
- Coloni, A., Dorigo, M., Maniezzo, V., Distributed optimization by ant colonies., *Memorias del European Conference on Artificial Life (ECAL91)*, Elsevier Publishing, 1991, pp. 134–142.
- Coloni, A., Dorigo, M., Maniezzo, V., An investigation of some properties of an "ant algorithm"., *Memorias del Parallel Problem Solving from Nature Conference*, Elsevier Publishing, 1992, pp. 509–520.
- Cordón, O., de Viana, I. F., Herrera, F., Moreno, L., A new aco model integrating evolutionary computation concepts: The best-worst ant system., *Memorias del ANTS 2000 - From Ant Colonies to Artificial Ants: Third International Workshop on Ant Algorithms*, 2000, pp. 22–29.
- De Meyer, K., Bishop, J., Nasuto, S., Stochastic diffusion: Using recruitment for search., *Memorias del Symposium on Evolvability & Interaction*, 2003.
- de Vries, H., Biesmeijer, J. C., Modelling collective foraging by means of individual behaviour rules in honey-bees., *Behav Ecol Sociobiol*, Vol. 44, 1998, pp. 109–124.
- Dorigo, M., Gambardella, L. M., A study of some properties of ANT-Q., *Memorias del Fourth International Conference on Parallel Problem Solving From Nature*, H. M. Voigt, W. Ebeling, I. Rechenberg y H. S. Schwefel (eds.), Springer Verlag, Berlin, Germany, 1996, pp. 656–665.
- Dorigo, M., Gambardella, L. M., Ant colony system: A cooperative learning approach to the traveling salesman

- problem., *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, 1997, pp. 53–66.
- Dorigo, M., Maniezzo, V., Colorni, A., Ant system: An autocatalytic optimizing process., Technical Report 91-016, Politecnico di Milano, Italy, 1991a.
- Dorigo, M., Maniezzo, V., Colorni, A., Positive feedback as a search strategy, Technical Report 91-016, Politecnico di Milano, Italy, 1991b.
- Dorigo, M., Maniezzo, V., Colorni, A., The ant system: Optimization by a colony of cooperating agents., *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, Vol 26, No. 1, 1996, pp. 1–13.
- Easter Selvan, S., Subramanian, S., Theban Solomon, S., Novel technique for PID tuning by particle swarm optimization., *Memorias del Seventh Annual Swarm Users/ Researchers Conference (SwarmFest 2003)*, 2003.
- Eberhart, R., Kennedy, J., A new optimizer using particle swarm theory., *Memorias del Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.
- El-Gallad, A. I., El-Hawary, M., Sallam, A. A., Kalas, A., Swarm-intelligently trained neural network for power transformer protection., *Memorias del Canadian Conference on Electrical and Computer Engineering*, 2001, pp. 265–269.
- Esmín, A. A. A., Aoki, A. R., Lambert-Torres, G., Particle swarm optimization for fuzzy membership functions optimization., *Memorias del IEEE International Conference on Systems, Man and Cybernetics*, 2002, pp. 108–113.
- Gambardella, L. M., Dorigo, M., Ant-Q: A reinforcement learning approach to the traveling salesman problem, *Memorias del Twelfth International Conference on Machine Learning*, Morgan Kaufmann, 1995, pp. 252–260.
- Gazi, V., Passino, K. M., Stability analysis of swarms., *IEEE Transactions on Automatic Control* Vol 48, No. 4, 2003, pp. 692–697.
- Gazi, V., Passino, K. M., Stability analysis of social foraging swarms., *IEEE Transactions of Systems, Man and Cybernetics - Part B*, Vol 34, No. 1, 2004, pp. 539–557.
- Grech-Cini, H., Locating Facial Features., tesis presentada a la University of Reading, para optar al grado de Doctor of Philosophy, 1995.
- Grosan, C., Abraham, A., Han, S., Gelbukh, A., Hybrid particle swarm – evolutionary algorithm for search and optimization., *Memorias del 4th Mexican International Conference on Artificial Intelligence (MICAI'05)*, Lecture Notes in Computer Science, 2005, pp. 623–632.
- Heegaard, P. E., Wittner, O., Incola, V. F., Helvik, B., Distributed asynchronous algorithm for cross-entropy-based combinatorial optimization., *Memorias del International Workshop on Rare Event Simulation & Combinatorial Optimization (RESIM 2004)*, 2004.
- Hertz, A., Kobler, D., A framework for the description of evolutionary algorithms., *European Journal of Operational Research*, Vol 126, No. 1, 2000, pp. 1–12.
- Hinton, G., A parallel computation that assigns canonical object-based frames of reference., *Memorias del 7th International Joint Conference on Artificial Intelligence*, 1981.
- Ismail, A., Engelbrecht, A. P., Training product units in feedforward neural networks using particle swarm optimization., *Memorias del International Conference on Artificial Intelligence*, 1999, pp. 36–40.
- Karaboga, D., An idea based on honey bee swarm for numerical optimization., Technical Report TR06, Erciyes University, 2005.
- Kennedy, J., Eberhart, R., Particle swarm optimization., *Memorias del IEEE International Conference on Neural Networks*, 1995, pp. 1942–1998.
- Kennedy, J., Eberhart, R., A discrete binary version of the particle swarm algorithm., *Memorias del World Multiconference on Systemics, Cybernetics and Informatics*, 1997, pp. 4104–4109.
- Kim, D. H., Abraham, A., Cho, J. H., A hybrid genetic algorithm and bacterial foraging approach for global optimization., *Information Sciences*, Vol. 177, 2007, pp. 3918–3937.
- Kim, D. H., Cho, J. H., Adaptive tuning of PID controller for multivariable system using bacterial foraging based optimization., *Advances in Web Intelligence*, Vol. 3528 of Lecture Notes in Computer Science, Springer, 2005, pp. 231–235.
- Krohling, R. A., d.S. Coelho, L., Shi, Y., Cooperative particle swarm optimization for robust control system design., *Memorias del 7th Online World Conference on Soft Computing in Industrial Applications*, 2002.
- Leiviskä, K., Joensuu, I., Chemotaxis for controller tuning, *Memorias del 2nd Annual Symposium of the Nature-inspired Smart Information Systems (NiSIS 2006)*, 2006.
- Leontitsis, A., Kontogiorgos, D., Pagge, J., Repel the swarm to the optimum!, *Applied Mathematics and Computation*, Vol, 173, 2006, pp. 265–272.
- Liu, H., Abraham, A., Fuzzy adaptive turbulent particle swarm optimization., *Memorias del Fifth International Conference on Hybrid Intelligent Systems*, 2005, pp. 6–9.
- Liu, Y., Passino, K. M., Biomimicry of social foraging bacteria for distributed optimization: Models, principles and emergent behaviors., *Journal of Optimization Theory and Applications*, Vol. 115, No. 3, 2002, pp. 603–628.
- Lucic, P., Teodorovic, D., Computing with bees: Attacking complex transportation engineering problems., *International Journal on Artificial Intelligence Tools*, Vol. 12, No. 3, 2003, pp. 375–394.
- Løvbjerg, M., Kink, T., Hybrid particle swarm optimiser with breeding and subpopulations., *Memorias del Genetic and Evolutionary Computation Conference (GECCO)*, 2001.
- Løvbjerg, M., Kink, T., Extending particle swarm optimizers with self-organized criticality., *Memorias del IEEE Congress on Evolutionary Computation (CEC2002)*, 2002.
- Meuleau, N., Dorigo, M., Ant colony optimization and stochastic gradient descent, Technical Report TR/IRIDIA/

- 2000-36, IRIDIA, Université Libre de Bruxelles, Belgium, 2000.
- Miranda, V., Fonseca, N., EPSO – evolutionary particle swarm optimization, a new algorithm with applications in power systems., *Memorias del Asia Pacific Transmission and Distribution Conference and Exhibition*, Vol. 2, 2002a, pp. 745–750.
- Miranda, V., Fonseca, N., EPSO – best-of-two-worlds meta-heuristic applied to power system problems., 2002 Congress on Evolutionary Computation (CEC02), Vol. 2, 2002b, pp. 1080–1085.
- Müller, S. D., Marchetto, J., Airaghi, S., Koumoutsakos, P., Optimization based on bacterial chemotaxis., *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, 2002, pp. 16–29.
- Monson, C. K., Seppi, K. D., The kalman swarm: A new approach to particle motion in swarm optimization, *Memorias del Genetic and Evolutionary Computation Conference 2004 (GECCO2004)*, 2004, pp. 140–150.
- Muñoz, M., López, J., Caicedo, E., An artificial bee hive for continuous optimization., *Aceptado en International Journal of Intelligent Systems*, 2008.
- Myatt, D., Bishop, J., Nasuto, S., Minimum stable convergence criteria for stochastic diffusion search., *Electronics Letters*, Vol. 40, No. 2, 2004, pp. 112–113.
- Nasuto, S., Resource Allocation Analysis of the Stochastic Diffusion Search., tesis presentada a la University of Reading para optar al grado de Doctor of Philosophy, 1999.
- Nasuto, S., Bishop, J., Convergence analysis of stochastic diffusion search., *Journal of Parallel Algorithms and Applications*, Vol. 14, No. 2, 1999, pp. 89–107.
- Nasuto, S., Bishop, J., Lauria, S., Time complexity of stochastic diffusion search., *Memorias del Neural Computation '98*, 1998.
- Nunez de Castro, L., Fundamentals of natural computing: an overview., *Physics of Life Reviews* Vol. 4, 2007, pp. 1–36.
- Omran, M. G. H., Engelbrecht, A. P., Salman, A., Dynamic clustering using particle swarm optimization with application in unsupervised image classification., *Transactions on Engineering, Computing and Technology*, Vol. 9, 2005, pp. 199–204.
- Ozcan, E., Mohan, C. K., Particle swarm optimization: Surfing the waves., *Memorias del IEEE Congress on Evolutionary Computation (CEC1999)*, 1999.
- Passino, K. M., Distributed optimization and control using only a germ of intelligence., *Memorias del 2000 IEEE International Symposium on Intelligent Control*, 2000, pp. 5–13.
- Passino, K. M., Biomimicry of bacterial foraging for distributed optimization and control., *IEEE Control Systems Magazine*, Vol. 22, No. 3, 2002, pp. 52–67.
- Pham, D., Afify, A., Koç, E., Manufacturing cell formation using the bees algorithm., *Memorias del Innovative Production Machines and Systems Virtual Conference*, 2007.
- Pham, D., Darwish, A. H., Eldukhr, E., Otri, S., Using the bees algorithm to tune a fuzzy logic controller for a robot gymnasta., *Memorias del Innovative Production Machines and Systems Virtual Conference*, 2007.
- Pham, D., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., Zaidi, M., The bees algorithm – a novel tool for complex optimisation problems., *Memorias del Innovative Production Machines and Systems Virtual Conference*, 2006.
- Pham, D., Ghanbarzadeh, A., Multi-objective optimisation using the bees algorithm., *Memorias del Innovative Production Machines and Systems Virtual Conference*, 2007.
- Pham, D., Koç, E., Ghanbarzadeh, A., Otri, S., Optimisation of the weights of multi-layered perceptrons using the bees algorithm, *Memorias del 5th International Symposium on Intelligent Manufacturing Systems*, 2006.
- Pham, D., Muhamad, Z., Mahmuddin, M., Ghanbarzadeh, A., Koç, E., Otri, S., Using the bees algorithm to optimise a support vector machine for wood defect classification., *Memorias del Innovative Production Machines and Systems Virtual Conference*, 2007.
- Poli, R., Langdon, W. B., Holland, O., Extending particle swarm optimization via genetic programming., *Memorias del 8th European Conference on Genetic Programming*, 2005, pp. 291–300.
- Rubinstein, R., Combinatorial optimization, cross-entropy, ants and rare events., *Stochastic Optimization: Algorithms and Applications*, S. Uryasev y P. M. Pardalos (eds.), Kluwer Academic Publishers, 2001.
- Rumelhart, D., McClelland, J., *Parallel Distributed Processing* Vol. 1, MIT Press, 1986, pp. 113–118.
- Salerno, J., Using the particle swarm optimization technique to train a recurrent neural model., *IEEE International Conference on Tools with Artificial Intelligence*, 1997, pp. 45–49.
- Shi, Y., Eberhart, R. C., Parameter selection in particle swarm optimization., *Memorias del Seventh Annual Conference on Evolutionary Programming*, 1998, pp. 591–600.
- Stützle, T., Hoos, H., The max–min ant system and local search for the traveling salesman problem., *Memorias del IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference (IEEE-ICEC-EPS'97)*, T. Baeck, Z. Michalewicz y X.ao (eds), IEEE Press, 1997, pp. 309–314.
- Teodorovic, D., Dell'Orco, M., Bee colony optimization – a cooperative learning approach to complex transportation problems., *Memorias del 10th EWGT Meeting and 16th Mini-EURO Conference*, 2005.
- Tillett, J., Rao, T. M., Sahin, F., Rao, R., Darwinian particle swarm optimization., *Memorias del 2nd Indian International Conference on Artificial Intelligence*, 2005, pp. 1474–1487.
- Tsutsui, S., Ant colony optimisation for continuous domains with aggregation pheromones metaphor., *Memorias del*

- 5th International Conference on Recent Advances in Soft Computing, 2004, pp. 207–212.
- v.E. Conradie, A., Miikkulainen, R., Aldrich, C., Adaptive control utilising neural swarming., *Memorias del Genetic and Evolutionary Computation Conference 2002 (GECCO2002)*, 2002.
- Voss, M. S., Feng, X., ARMA model selection using particle swarm optimization and AIC criteria., *Memorias del IFAC 15th Triennial World Congress*, 2002.
- Whitaker, R., Hurley, S., An agent based approach to site selection for wireless networks., *Memorias del ACM Symposium on Applied Computing*, 2002, pp. 574–577.
- Yan-jun, L., Tie-jun, W., An adaptative ant colony system algorithm for continuous-space optimization problems., *Journal of Zhejiang University SCIENCE*, Vol. 4, No. 1, 2003, pp. 40–46.
- Yisu, J., Knowles, J., Hongmei, L., Yizeng, L., Kell, D. B., The landscape adaptive particle swarm optimizer., *Applied Soft Computing*, Vol. 8, No. 1, 2007, pp. 295–304.
- Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S., Nakanishi, Y., A particle swarm optimization for reactive power and voltage control considering voltage security assessment., *IEEE Transactions on Power Systems*, Vol. 15, No. 4, 2001, pp. 1232–1239.
- Yuan, R., Guang-yi, C., Xin-jian, Z., Particle swarm optimization based predictive control of proton exchange membrane fuel cell (pemfc)., *Journal of Zhejiang University SCIENCE A*, Vol. 7, No. 3, 2006, pp. 458–462.
- Zhang, W., Xie, X., DEPSO: Hybrid particle swarm with differential evolution operator., *Memorias del IEEE International Conference on Systems, Man and Cybernetics (SMCC)*, 2003, pp. 3816–3821.