

# Incorporación de memoria explícita e hibridación en algoritmos ACO

Franco Arito      Guillermo Leguizamón

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional  
Departamento de Informática  
Facultad de Ciencias Físico Matemáticas y Naturales  
Universidad Nacional de San Luis  
e-mail: {farito, legui}@unsl.edu.ar

## Resumen

Desde la invención de la metaheurística ACO se han propuesto numerosas variantes, la mayoría de éstas con el objetivo de mejorar su rendimiento. Estas variantes incluyen tanto incorporación de algoritmos de búsqueda local para mejorar soluciones (versiones híbridas) como así también mecanismos que modifican la manera de construir soluciones. Dentro de estas variantes, se encuentra la que supone el uso de un mecanismo auxiliar para la toma de decisiones durante el proceso de construcción de soluciones por las hormigas a través de memoria explícita.

## Contexto

Este artículo describe parte de los trabajos de investigación y desarrollo que se están llevando a cabo en la línea “Metaheurísticas” del Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC) de la UNSL.

Esta línea tiene como principal objetivo el diseño, implementación y aplicación de diferentes metaheurísticas y sus posibles variantes híbridas derivadas de heurísticas convencionales, inteligencia artificial, evolución biológica, sistemas neuronales, sistemas inmunes y otros bioinspirados.

## 1. Introducción

Los algoritmos ACO hacen uso de agentes simples llamados *hormigas* que construyen soluciones candidatas iterativamente para un problema de optimización combinatoria. La construcción de soluciones realizada por las hormigas está guiada por rastros de feromona e información heurística dependiente del problema. En principio, los algoritmos ACO pueden ser aplicados a cualquier problema de optimización combinatoria definiendo las componentes de solución que las hormigas usan para construir soluciones candidatas y sobre las cuales se podrá depositar feromona [5, 6, 8]. Cada hormiga construye soluciones candidatas comenzando con una solución vacía y luego va agregando componentes incrementalmente hasta que se genera una solución candidata completa. Después de que se ha completado la construcción de una solución, las hormigas depositan feromona en las componentes que han utilizado en su solución teniendo en cuenta su calidad. Generalmente, los componentes de solución que forman parte de las mejores soluciones o que son utilizadas por muchas hormigas son las que recibirán mayor cantidad de feromona y, por lo tanto, tendrán mayor probabilidad de ser usadas por las hormigas en futuras iteraciones del algoritmo. Para evitar que la búsqueda sufra estancamiento<sup>1</sup>, por lo general antes de que se refuerzen los rastros de feromona,

<sup>1</sup>Se define en [7] como la situación en donde todas las hormigas siguen la misma ruta y construyen la misma solución.

todos los rastros de feromona son disminuidos por un factor  $\rho$  [8].

La línea de investigación presentada está relacionada a una de las tendencias actuales en algoritmos ACO, en los cuales se incorpora una memoria externa como alternativa al mecanismo de elección de componentes de soluciones [1, 12, 13], y está inspirada en la metaheurística Tabu Search.

Tabu Search (TS) es una metaheurística cuya característica distintiva es el uso de una memoria adaptativa y de estrategias especiales de resolución de problemas. Su filosofía se basa en la explotación de diversas estrategias inteligentes para la resolución de problemas, basadas en procedimientos de aprendizaje. El marco de memoria adaptativa de TS explota la historia del proceso de resolución del problema haciendo referencia a cuatro dimensiones principales: ser reciente, en frecuencia, en calidad, y en influencia.

## 2. Líneas de Investigación y Desarrollo

En esta sección, se describen las ideas generales de los temas que se están investigando en nuestra línea.

### 2.1. Memoria externa

El uso de la memoria externa se introduce en los algoritmos ACO durante la construcción de soluciones realizada por las hormigas. Esto supone dos perspectivas:

- utilizar (y aprovechar) información **relevante** recolectada durante la búsqueda que en algoritmos ACO estandar no se realizaría. Esta información es un resumen del historial del proceso de búsqueda, y permite registrar en que medida han sido visitadas las zonas del espacio de búsqueda del problema en particular; para permitir volver a visitar regiones prometedoras del espacio de búsqueda, o por el contrario, promover la búsqueda hacia regiones menos exploradas.
- introducir un cierto grado de determinismo en la construcción de soluciones, debido a que no

todas las decisiones respecto de la construcción de soluciones se toman en forma probabilística.

Es importante mencionar que el uso de la memoria complementa al modelo de feromona de algoritmos ACO sin perder la filosofía propia de la metaheurística. Además, si bien la metaheurística TS considera cuatro dimensiones (i.e., ser reciente, frecuencia, calidad e influencia), para el caso de los algoritmos ACO en proceso de investigación se consideran dos alternativas:

- **Memoria basada en lo reciente:** La forma habitual de utilizar este tipo de memoria es etiquetando las componentes seleccionadas de soluciones visitadas recientemente.
- **Memoria basada en frecuencia:** La forma habitual de explotar este tipo de memoria es etiquetando las componentes seleccionadas de soluciones más frecuentemente escogidas.

### 2.2. Hibridación

En muchas aplicaciones a problemas de optimización combinatoria, los algoritmos ACO obtienen mejores resultados cuando se los hibridiza con algoritmos de búsqueda local (que es un tipo particular de acciones auxiliares de la metaheurística ACO). Los algoritmos de búsqueda local optimizan localmente las soluciones construidas por las hormigas y luego estas soluciones son usadas en la actualización de feromona.

Por otro lado, es necesario tener en cuenta el hecho de que generar soluciones iniciales para algoritmos de búsqueda local no es tarea sencilla. Por ejemplo, se ha demostrado que, para muchos problemas, repetir búsqueda local a partir de soluciones iniciales generadas aleatoriamente no es eficiente [9]. En la práctica, las hormigas combinan probabilísticamente componentes de soluciones que son parte de las mejores soluciones localmente óptimas encontradas hasta el momento y generan soluciones nuevas (más prometedoras) para la búsqueda local. Experimentalmente, se ha mostrado que la combinación de una heurística constructiva (adaptativa y probabilística) con búsqueda local puede obtener excelentes resultados [4, 7, 11].

### 2.3. Balance entre exploración y explotación

Cualquier algoritmo metaheurístico efectivo tiene que lograr un equilibrio adecuado entre la explotación de la experiencia de búsqueda obtenida hasta el momento y la exploración de regiones del espacio de búsqueda no visitadas o relativamente poco exploradas. En los algoritmos ACO existen varias maneras de lograr ese equilibrio, normalmente a través del manejo de los rastros de feromona. De hecho, los rastros de feromona inducen una distribución de probabilidad sobre el espacio de búsqueda y determinan qué partes del espacio de búsqueda son muestreadas efectivamente, es decir, en qué parte del espacio de búsqueda se encuentran con mayor frecuencia las soluciones construidas. Se debe tener en cuenta que, dependiendo de la distribución de los rastros de feromona, la distribución muestral puede variar de una distribución uniforme a una distribución sesgada que asigne probabilidad uno a una única solución y probabilidad cero a todos los demás. De hecho, esta última situación corresponde al estancamiento de la búsqueda anteriormente mencionado.

La forma más sencilla de explotar la experiencia de búsqueda de las hormigas es hacer la actualización de feromona en función de la calidad de la solución realizada por cada hormiga en particular. Sin embargo, este sesgo por sí solo es a menudo demasiado débil como para obtener un buen rendimiento, como se ha demostrado experimentalmente en [7]. Por lo tanto, en muchos variantes de algoritmos ACO se introduce una estrategia elitista en la que las mejores soluciones encontradas durante la búsqueda contribuyan en gran medida en actualización de los rastros de feromona.

También se puede lograr una explotación más marcada de los rastros de feromona “aprendidos” durante la construcción de soluciones. Por ejemplo mediante la aplicación de la regla proporcional pseudo-aleatoria de Ant Colony System [7]; mientras que la exploración del espacio de búsqueda se realiza en ACO principalmente a través de la construcción aleatoria de soluciones por las hormigas. Estos dos mecanismos son complementados con el uso de la memoria externa tanto en el proceso de intensificación como en el proceso de diversificación. Otra de las características que introduce el uso de la memoria es la de garantizar (aunque en forma

mínima) una exploración controlada del espacio de búsqueda, con lo cual se contribuye a tratar de evitar estancamiento prematuro de la búsqueda.

## 3. Resultados obtenidos/esperados

Los resultados obtenidos en relación al uso de memoria externa e hibridación han sido presentados en las siguientes publicaciones científicas [2, 3, 10]. Estos trabajos reflejan la evolución que esta teniendo la original propuesta.

Los resultados obtenidos hasta el momento son:

- Diseño e implementación de algoritmos ACO que incluyen uso de memoria externa para problemas de optimización combinatoria.
- Se ha logrado implementar algoritmos ACO que alcanzaron estado-del-arte en la resolución de problemas  $\mathcal{NP}$  que utilizan las variantes de memoria externa descritas en secciones anteriores
- Se ha logrado hibridar algoritmos ACO alcanzando estado-del-arte para clase de problemas  $\mathcal{NP}$  con algoritmos de búsqueda local: clásica y Tabu Search, entre otras.

Los resultados esperados en esta línea de trabajo, se orientan a lograr:

- Extender la aplicación de los algoritmos ACO con uso de memoria externa a otros problemas y dominios (tanto discretos como continuos).
- Determinar el efecto de los mecanismos de memoria externa en cuanto a la diversificación/intensificación del algoritmo.
- Establecer bajo que condiciones el uso de memoria externa en algoritmos ACO supera a sus pares sin uso de memoria, en aspectos tales como: velocidad de convergencia, control de diversidad poblacional, explotación de información acumulada en las soluciones, entre otras.

## 4. Formación de recursos humanos

La propuesta particular planteada aquí junto con los estudios derivados de la misma han dado lugar a la formación de becarios de grado y posgrado. En particular se pretende continuar en este y otros temas relacionados a través del contacto realizado por los autores con grupos de la Universidad de Málaga (España).

## Referencias

- [1] A. Acan. An external memory implementation in ant colony optimization. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stützle, editors, *ANTS 2004*, volume 3172 of *LNCS*, pages 73–84. Springer, Heidelberg, 2004.
- [2] F. Arito and G. Leguizamón. Extensión de algoritmos ACO usando conceptos de Tabu Search. Anales del XI Workshop de Investigadores en Ciencias de la Computación, 2009.
- [3] F. Arito and G. Leguizamón. Incorporating Tabu Search principles into ACO algorithms. In M. J. Blesa, C. Blum, L. Di Gaspero, A. Roli, M. Sampels, and A. Schaerf, editors, *Hybrid Metaheuristics*, volume 5818 of *Lecture Notes in Computer Science*, pages 130–140. Springer, 2009.
- [4] K. D. Boese, A. B. Kahng, and S. Muddu. A new adaptive multi-start technique for combinatorial global optimization. *Operations Research Letters*, 16:101–113, 1994.
- [5] M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, chapter 2, pages 11–32. McGraw-Hill, London, UK, 1999.
- [6] M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.
- [7] M. Dorigo and L. M. Gambardella. Ant Colony System: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
- [8] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
- [9] D. S. Johnson and L. A. Mcgeoch. The traveling salesman problem: A case study in local optimization. In E. H. L. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 215–310. John Wiley & Sons, 1997.
- [10] G. Leguizamón, F. Arito, and C. A. Coello Coello. A hybrid memory-based ACO algorithm for the QAP. Aceptado para publicación en el IEEE World Congress on Computational Intelligence. July 18–23, 2010.
- [11] T. Stützle and H. Hoos. *MAX – MIN* ant system. *Future Generation Computer Systems*, 16(8):889–914, 2000.
- [12] S. Tsutsui. *cAS*: Ant colony optimization with cunning ants. In T. P. Runarsson *et al.*, editor, *Proc. of the 9th Int. Conf. on Parallel Problem Solving from Nature (PPSN IX)*, volume 4193 of *LNCS*, pages 162–171. Springer, Heidelberg, 2006.
- [13] W. Wiesemann and T. Stützle. Iterated ants: An experimental study for the quadratic assignment problem. In M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, editors, *ANTS 2006*, volume 4150 of *LNCS*, pages 179–190. Springer, Heidelberg, 2006.