

UNIVERSIDAD NACIONAL DE ASUNCIÓN

Maestría en Ingeniería de Sistemas  
Proyecto Final de Tesis

Bajo la orientación del Profesor D.Sc. Benjamín Barán

Razones del Éxito de Optimización por Colonia  
de Hormigas en el Problema del Cajero Viajante

Oswaldo Gómez

Septiembre de 2004

# Índice de Contenidos

<b>1. Introducción</b>	<b>1</b>
<b>2. El Problema del Cajero Viajante</b>	<b>3</b>
2.1. El Problema . . . . .	4
2.2. Estudio del Espacio de Soluciones del TSP . . . . .	4
2.2.1. Convexidad Global . . . . .	4
2.2.2. Estudio Experimental del Espacio de Soluciones del TSP	5
<b>3. Optimización por Colonia de Hormigas</b>	<b>11</b>
3.1. Comportamiento de Búsqueda de Alimentos de la Hormigas . . .	11
3.2. Optimización por Colonia de Hormigas . . . . .	12
3.2.1. Enfoque Estándar de ACO . . . . .	12
3.2.2. <i>MAX-MIN Ant System</i> . . . . .	13
3.2.3. ACOs Basados en Población . . . . .	14
3.2.4. Ómicron ACO . . . . .	15
<b>4. Resultados Experimentales</b>	<b>18</b>
4.1. Estudio Comparativo entre OA y <i>MMAS</i> . . . . .	18
<b>5. Razones del Éxito de ACO en el TSP</b>	<b>21</b>
5.1. Estudio del Comportamiento de Ómicron ACO . . . . .	21
5.2. Razones del Éxito de ACO en el TSP . . . . .	23
5.2.1. Geometría del Problema . . . . .	23
5.2.2. El OA en una Geometría Globalmente Convexa . . . . .	26
5.2.3. Interpretación en 2D del Espacio de Exploración del OA .	27
5.2.4. Razones Subyacentes del Éxito de ACO . . . . .	28
<b>6. Relación entre ACO y Algoritmos Genéticos</b>	<b>30</b>
6.1. Algoritmos Genéticos . . . . .	30
6.1.1. Algoritmo Genético Simple . . . . .	30
6.1.2. Algoritmo Genético Ómicron . . . . .	31
6.2. Nuevas Estrategias para OGA . . . . .	35
6.2.1. Cruzamiento . . . . .	35
6.2.2. Actualización de la Población . . . . .	35
6.2.3. Información Heurística . . . . .	35

6.3. Relación entre ACO Y GA . . . . .	36
6.3.1. Nueva Estrategia para el MOGA . . . . .	37
6.3.2. Nueva Estrategia para el EMOGA . . . . .	37
6.3.3. Razones del Éxito de AG en el TSP . . . . .	38
<b>7. Conclusiones y Trabajos Futuros</b>	<b>40</b>

# Índice de Figuras

2.1.	Interpretación intuitiva de un espacio de soluciones con estructura globalmente convexa . . . . .	5
2.2.	Sencillo ejemplo para explicar la conjetura de una estructura globalmente convexa del espacio de soluciones del TSP . . . . .	6
2.3.	$l(\Omega_x)$ en función de $\delta(\Omega_*, \Omega_x) \forall \Omega_x \in \Pi$ para el problema berlin52	7
2.4.	$l(\Omega_x)$ en función de $\delta(\Pi, \Omega_x) \forall \Omega_x \in \Pi$ para el problema berlin52	8
2.5.	$l(\Omega_x)$ en función de $\delta(\Omega_*, \Omega_x) \forall \Omega_x \in \Pi$ para el problema pcb442	9
2.6.	$l(\Omega_x)$ en función de $\delta(\Pi, \Omega_x) \forall \Omega_x \in \Pi$ para el problema pcb442 .	10
4.1.	Comparación entre el comportamiento medio del OA y el <i>MMAS</i> para el TSP eil51 . . . . .	19
4.2.	Comparación entre el comportamiento medio del OA y el <i>MMAS</i> en la fase de convergencia para el TSP eil51 . . . . .	19
4.3.	Comparación entre el comportamiento medio del OA y el <i>MMAS</i> para el TSP kroA100 . . . . .	20
4.4.	Comparación entre el comportamiento medio del OA y el <i>MMAS</i> en la fase de convergencia para el TSP kroA100 . . . . .	20
5.1.	Evolución de $\delta(Q, \Omega_*)_M, \delta(Q)_M, l(Q)_M, l(Q)_M$ y $\zeta(Q)_M$ en función de las iteraciones para 10 corridas. Izquierda, para el problema berlin52. Derecha, para el problema omi1 . . . . .	22
5.2.	$l(\Omega_x)$ en función a $\delta(\Omega_x, \Omega_*) \forall \Omega_x \in \Omega$ en el TSP omi1 . . . . .	24
5.3.	$l(\Omega_x)$ en función de $\delta(\Omega(e), \Omega_x) \forall \Omega_x \in \Omega$ para distintos valores de $e$ . . . . .	24
5.4.	$l(\Omega_x)$ en función de $\delta(Q, \Omega_x) \forall \Omega_x \in \Omega$ para $Q = \Omega(e, 25)$ y para distintos valores de $e$ . . . . .	25
5.5.	(a) Visión simplificada de la zona de búsqueda cercana a todas las soluciones de $Q$ ( $\Omega(Q)$ ) (b) Zona central geométrica de una población de 25 puntos elegidos al azar . . . . .	27
5.6.	Visión simplificada del comportamiento de OA . . . . .	28
5.7.	Sencillo ejemplo para explicar la pérdida de la propiedad de convexidad global en en el espacio de soluciones cercano a $\Omega_*$ . . . . .	29
6.1.	Cuadro resumido de las versiones de GA y ACO . . . . .	38

# Índice de Tablas

2.1.	$\Phi[L(\Pi), \delta^*(\Pi)]$ y $\Phi[L(\Pi), \delta^\Pi(\Pi)]$ para 13 instancias de la TSPLIB	7
2.2.	$\Phi[L(\Pi), \delta^*(\Pi)]$ y $\Phi[L(\Pi), \delta^\Pi(\Pi)]$ para muestras a todas las posibles distancias de $\Omega_*$ en 20 instancias de la TSPLIB . . . . .	9
5.1.	Correlación de las variables de comportamiento de OA estudiadas para el problema berlin52 . . . . .	23
5.2.	Correlación entre $l(\Omega_x)$ y $\delta(Q, \Omega_x)$ para distintos valores de $e$ . .	26
5.3.	Valores medios $\mu$ , $\lambda$ y $\frac{\mu}{\lambda}$ para distintos valores de $e$ . . . . .	27

# Abreviaturas

AS .....	Sistema de Hormigas o <i>Ant System</i>
CM .....	Cruzamiento-Mutación
DEMOGA .....	Algoritmo Genético Ómicron con Múltiples Padres, Super Elista y con Diversidad Genética Forzada o <i>Diversified Super-Elitist Multi-Parent Omicron Genetic Algorithm</i>
EMOGA .....	Algoritmo Genético Ómicron con Múltiples Padres y Super Elista o <i>Super-Elitist Multi-Parent Omicron Genetic Algorithm</i>
GA .....	Algoritmos Genéticos o <i>Genetic Algorithms</i>
GBAS .....	<i>Graph-based Ant System</i>
ACO .....	Optimización por Colonia de Hormigas o <i>Ant Colony Optimization</i>
MMAS .....	<i>MAX-MIN Ant System</i>
OA .....	Ómicron ACO
OGA .....	Algoritmo Genético Ómicron u <i>Omicron Genetic Algorithm</i>
P-ACO .....	ACOs basados en Población o <i>Population-based ACOs</i>
TSP .....	Problema del Cajero Viajante o <i>Traveling Salesman Problem</i>
MOGA .....	Algoritmo Genético Ómicron con Múltiples Padres o <i>Multi-Parent Omicron Genetic Algorithm</i>
PTS .....	<i>Pheromone Trail Smooth</i>
SGA .....	Algoritmo Genético Simple o <i>Simple Genetic Algorithm</i>
TSP LIB .....	Librería de instancias del TSP

# Símbolos

Símbolo	Significado
$G$	Gráfico completo del TSP
$N$	Conjunto de nodos o ciudades de $G$
$A$	Conjunto de arcos de $G$
$d(i, j)$	Distancia entre ciudades $i$ y $j$
$\Omega_*$	Recorrido óptimo del TSP
$n$	Número de ciudades del TSP. $n =  N $
$\Omega_x$	Recorrido válido del TSP
$l(\Omega_x)$	Longitud del recorrido $\Omega_x$
$\delta(\Omega_x, \Omega_y)$	Distancia entre $\Omega_x$ y $\Omega_y$ . $\delta(\Omega_x, \Omega_y) = n -  \text{Arcos comunes entre } \Omega_x \text{ y } \Omega_y $
$\Pi$	Población compuesta por recorridos válidos del TSP. $\Pi = \{\Pi_i\}$
$\pi$	Número de componentes de $\Pi$ . $\pi =  \Pi $
$ \cdot $	Cardinalidad
$\delta(\Pi, \Omega_x)$	Distancia media de $\Pi$ a $\Omega_x$ . $\delta(\Pi, \Omega_x) = \frac{1}{\pi} \sum_{i=1}^{\pi} \delta(\Pi_i, \Omega_x)$
$L(\Pi)$	Conjunto de longitudes de $\Pi_i$ . $L(\Pi) = \{l(\Pi_i)\}$
$\delta^*(\Pi)$	Conjunto de distancias de $\Pi_i$ a $\Omega_*$ . $\delta^*(\Pi) = \{\delta(\Pi_i, \Omega_*)\}$
$\delta^\Pi(\Pi)$	Conjunto de distancias medias de $\Pi_i$ a $\Pi$ . $\delta^\Pi(\Pi) = \{\delta(\Pi, \Pi_i)\}$
$\Phi[A, B]$	Correlación entre los conjuntos A y B
$\tau$	Matriz de feromonas en ACO. $\tau = \{\tau_{ij}\}$
$\tau_{init}$	Valor inicial para $\tau_{ij}$ en el enfoque estándar de ACO y P-ACO.
$\eta_{ij}$	Visibilidad. $\eta_{ij} = \frac{1}{d(i,j)}$
$\alpha$	Peso relativo del nivel de feromonas en ACO y del peso genético en MOGA
$\beta$	Peso relativo de la visibilidad en ACO y MOGA
$m$	Número de hormigas en ACO
$\mathcal{N}_i$	Conjunto de ciudades no visitadas y que permiten la construcción de una solución factible del TSP mientras es visitada una ciudad $i$
$\rho$	Tasa de evaporación en ACO

Símbolo	Significado
$k$	Hormiga en ACO
$M_k$	Recorrido construido por la hormiga $k$
$u$	Contador de iteraciones
$M$	Conjunto de soluciones construidas en una iteración de ACO
$M_{best}$	Mejor solución de $M$
$\Delta\tau$	Nivel de feromonas utilizado para la actualización de $\tau$
$\tau_{max}$	Nivel máximo de feromonas en $\mathcal{MMAS}$
$\tau_{min}$	Nivel mínimo de feromonas en $\mathcal{MMAS}$
$Q$	Población de un P-ACO. $Q = \{Q_x\}$
$q$	Número de componentes de $Q$ . $q =  Q $
$Q_{in}$	Individuo que ingresa a la población en P-ACO
$Q_{out}$	Individuo que abandona la población en P-ACO
$\Delta$	Nivel de feromonas con que colabora cada individuo en P-ACO $\Delta = (\tau_{max} - \tau_{init})/q$
$O$	Parámetro Ómicron
$\tau^0$	Matriz de feromonas inicial en OA
$Q_*$	Mejor individuo de la población $Q$
$Q_{worst}$	Peor individuo de la población $Q$
$Q_{new}$	Nuevo individuo generado en cada iteración de OA
$K$	Periodo de iteraciones para actualizar $\tau$ en OA
$\delta(\Pi)$	Distancia media de $\Pi$ . $\delta(\Pi) = \frac{2}{\pi(\pi-1)} \sum_{i=1}^{\pi-1} \sum_{j=i+1}^{\pi} \delta(\Pi_i, \Pi_j)$
$l(Q)$	Longitud media de una población. $l(Q) = \frac{1}{q} \sum_{i=1}^q l(Q_i)$
$\zeta(Q)$	Número de individuos que entraron a la población en OA
$\delta(Q, \Omega_x)_M$	Valor medio de $\delta(Q, \Omega_x)$ en varias corridas
$\delta(Q)_M$	Valor medio de $\delta(Q)$ en varias corridas
$l(Q_*)_M$	Valor medio de $l(Q_*)$ en varias corridas
$l(Q)_M$	Valor medio de $l(Q)$ en varias corridas
$\zeta(Q)_M$	Valor medio de $\zeta(Q)$ en varias corridas
$\Omega$	Todo el espacio de búsqueda discreto del TSP. $\Omega = \{\Omega_x\}$
$\Omega(\cdot)$	Subespacio de $\Omega$ . $\Omega(\cdot) \subseteq \Omega$
$\Omega(Q)$	Una zona central de $Q$ . $\Omega(Q) = \{\Omega_x \mid \delta(Q, \Omega_x) < \delta(Q)\}$
$\Omega(e)$	Subespacio conformado por las $e$ mejores soluciones de $\Omega$
$\Omega(e, \omega)$	Conjunto de $\omega$ elementos diferentes elegidos al azar de $\Omega(e)$
$\mu$	La proporción de $Q_{new}$ que cae dentro $\Omega(Q)$
$\lambda$	La cardinalidad media de $\Omega(Q)$
$P$	Población de un GA. $P = \{P_x\}$

Símbolo	Significado
$p$	Número de componentes de $P$ . $p =  P $
$mc$	Coefficiente de mutación en SGA
$g$	Contador de generaciones
$F$	Conjunto de padres en GA
$C$	Conjunto de hijos en GA
$V$	Conjunto de individuos alterados por la mutación en GA
$w_{ij}$	Peso genético asignado al arco $(i, j)$ por el operador CM de GA
$\mathcal{R}$	Ruleta en GA
$f_x$	<i>Fitness</i> del individuo $P_x$ en GA
$T$	Conjunto de hijos en MOGA
$t$	Número de hijos generados en MOGA. $t =  T $
$H$	Periodo de iteraciones para actualizar $\mathcal{R}$ en DEMOGA

# Capítulo 1

## Introducción

Optimización por Colonia de Hormigas o *Ant Colony Optimization* (ACO) es una metaheurística propuesta por Dorigo et al. que ha sido inspirada en el comportamiento de búsqueda de alimentos de las colonias de hormigas [Dorigo99]. En los últimos años ACO ha mostrado empíricamente su efectividad en la resolución de diferentes problemas de optimización combinatoria NP completos, como el Problema del Cajero Viajante o *Traveling Salesman Problem* (TSP) (el problema original resuelto con ACO), *Quadratic Assignment*, *Vehicle Routing*, *Sequential Ordering*, *Scheduling*, Ruteo en redes como Internet y varios más [Dorigo03]. Este trabajo se concentrará en el problema de optimización combinatoria más conocido, el TSP, debido a la simplicidad del mismo. Aunque ACO ha demostrado un rendimiento comparable al de otras heurísticas de clase mundial, sin embargo todavía poca teoría está disponible para explicar las razones subyacentes del éxito de ACO.

Birattari et al. desarrollaron un marco formal para *ant programming* con el objetivo de obtener una comprensión más profunda de ACO [Birattari00], mientras Meuleau y Dorigo estudiaron la relación entre ACO y la técnica *Stochastic Gradient Descent* [Meuleau02]. Gutjahr presentó una prueba de convergencia para una versión particular de ACO llamado *Graph-based Ant System* (GBAS) [Gutjahr00], que todavía tiene un rendimiento empírico desconocido. Él probó que GBAS converge, con una probabilidad que puede hacerse arbitrariamente cercana a 1, a la solución óptima de una instancia de un problema dado. Luego, Gutjahr demostró para una versión de GBAS dependiente del tiempo que sus soluciones convergen a la solución óptima con una probabilidad exactamente igual a 1 [Gutjahr02]. Stützle y Dorigo presentaron una pequeña prueba de convergencia para una clase de algoritmos ACO llamados  $ACO_{gb, \tau_{min}}$  [Stützle02], donde *gb* indica que es usada la política de actualización de feromonas considerando la mejor solución global encontrada (*global best*), mientras que  $\tau_{min}$  indica que un límite inferior en el nivel de rastros de feromonas es impuesto. Ellos probaron que la probabilidad de encontrar la solución óptima puede hacerse arbitrariamente cercana a 1 si el algoritmo es corrido por un número suficiente de iteraciones. Estudiando las características del espacio de búsqueda del TSP, Stützle y Hoos

[Stützle00] calcularon una correlación positiva entre la calidad de la solución y su distancia al óptimo global y comentan que parece razonable asumir que la concentración de la búsqueda alrededor de las mejores soluciones encontradas hasta el momento es el aspecto clave que posibilita el mejor rendimiento mostrado por los ACO elitistas. Sin embargo, no hay una clara comprensión de las razones reales del éxito de ACO, como es reconocido por Dorigo y Stützle [Stützle02, Dorigo01]. Ellos mencionan que aunque ha mostrado experimentalmente ser altamente efectiva, sólo limitado conocimiento está disponible para explicar por qué la metaheurística ACO es tan exitosa [Dorigo01].

En la búsqueda de herramientas de análisis de ACO, en esta investigación se diseñaron versiones muy simples de ACO, manteniendo características básicas del algoritmo. Así fue concebido el Ómicron ACO (OA), y éste fue evolucionando hasta convertirse en un algoritmo al mismo tiempo conceptualmente simple y competitivo [Gómez04b]. Esta simplicidad conceptual, sumada a un estudio sistemático del espacio de soluciones del TSP, permitió utilizarlo para analizar las razones del éxito de ACO en el TSP [Gómez04a, Gómez04c].

## Capítulo 2

# El Problema del Cajero Viajante

El miembro más prominente del rico conjunto de problemas de optimización combinatoria es indudablemente el Problema del Cajero Viajante o *Traveling Salesman Problem* (TSP). Este problema tiene una larga historia, con referencias desde el siglo 19 y es uno de los pocos problemas matemáticos que aparece frecuentemente en la prensa científica e inclusive en diarios [Reinelt94].

El estudio de este problema ha atraído a muchos investigadores de diferentes campos, e.g., Matemática, Investigación de Operaciones, Física, Biología o Inteligencia Artificial y existe una vasta literatura en el tema. Esto se debe a que, aunque es fácil de formularlo, exhibe todos los aspectos de un problema de optimización combinatoria. El TSP ha servido y continúa sirviendo como problema de prueba para distintos algoritmos, como Búsqueda local (*Local search*), Templado Simulado (*Simulated Annealing*), Búsqueda Tabu (*Tabu Search*), Redes Neuronales (*Neural Networks*) o Computación Evolutiva (*Evolutionary Computation*), por citar algunos de ellos.

Por otro lado, el TSP no es sólo interesante desde un punto de vista teórico, muchas aplicaciones prácticas pueden ser modeladas como un Problema del Cajero Viajante o como una variante del mismo y por lo tanto, existe una tremenda necesidad de algoritmos que lo resuelvan. Dadas las múltiples áreas de aplicación, también hay una amplia colección de algoritmos para tratar los casos especiales.

En las dos últimas décadas, un enorme progreso se ha alcanzado en la resolución del TSP. Este progreso es parcialmente debido al creciente avance del poder de cómputo de las computadoras, pero sobre todo fue posible por el desarrollo de teoría matemática y algoritmos eficientes. Aun considerando estos logros, el TSP está lejos de ser resuelto; muchos aspectos del problema todavía necesitan ser considerados y preguntas todavía necesitan ser satisfactoriamente respondidas.

Problemas con algunos cientos de ciudades son todavía difíciles de resolver

para los algoritmos de hoy en día y requieren horas de procesamiento en supercomputadores. Esto es fácil de comprender si consideramos que en un problema de tan sólo 100 ciudades existen del orden de  $1e155$  recorridos posibles. Además el crecimiento del número de recorridos en función al número de ciudades es exponencial.

El TSP simétrico es usado en este trabajo como problema de prueba para estudiar ACO, dadas las razones anteriormente citadas y dado el conocido éxito de ACO en resolverlo [Dorigo99, Stützle00].

## 2.1. El Problema

El TSP puede ser representado por un grafo completo  $G = (N, A)$  con  $N$  siendo el conjunto de nodos, también llamados ciudades, y  $A$  siendo el conjunto de arcos que conectan completamente a los nodos. A cada arco  $(i, j)$  se le asigna un valor  $d(i, j)$  que representa la distancia entre la ciudades  $i$  y  $j$ . El TSP es definido como el problema de encontrar el recorrido más corto  $\Omega_*$  visitando cada uno de los  $n = |N|$  nodos de  $G$  exactamente una vez.

Suponiendo que  $\Omega_x$  y  $\Omega_y$  son recorridos TSP o soluciones sobre el mismo conjunto de ciudades. Para este trabajo, la longitud del recorrido  $\Omega_x$  se denominará  $l(\Omega_x)$ . La distancia  $\delta(\Omega_x, \Omega_y)$  entre  $\Omega_x$  y  $\Omega_y$  es definida como  $n$  menos el número de arcos contenidos en ambos recorridos  $\Omega_x$  y  $\Omega_y$ , o en otras palabras, como el número de arcos diferentes entre los recorridos  $\Omega_x$  y  $\Omega_y$ .

## 2.2. Estudio del Espacio de Soluciones del TSP

Con el objetivo de estudiar las razones del éxito de ACO, se presenta un estudio del espacio de soluciones del TSP considerando 20 instancias extraídas de la conocida Librería de instancias del TSP (TSPLIB)<sup>1</sup>.

Note que en este trabajo las variables en mayúsculas (e.g.  $\Pi, Q, \dots$ ) representan conjuntos cuyos elementos son denotados con un subíndice (e.g.  $\Pi_i, Q_x, \dots$ ) y las minúsculas representan escalares (e.g.  $\pi, q, \dots$ ).

### 2.2.1. Convexidad Global

La convexidad global no es convexidad en el sentido estricto [Hu89]. Considerando un contexto de minimización, Boese sugirió una analogía con una estructura de gran valle, estructura que vista desde lejos parecería tener un sólo mínimo, pero que al verla desde cerca, se observa que tiene varios mínimos locales [Boese95]. Aunque no existe una definición formal, una interpretación intuitiva de una estructura globalmente convexa puede verse en la figura 2.1.

---

<sup>1</sup>Accesible en <http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/>

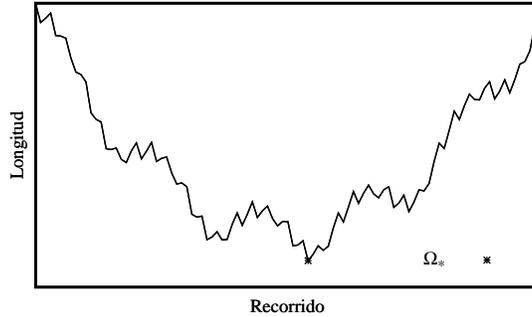


Figura 2.1: Interpretación intuitiva de un espacio de soluciones con estructura globalmente convexa

## 2.2.2. Estudio Experimental del Espacio de Soluciones del TSP

Con el objetivo de realizar este estudio, una definición de distancia media es presentada. Considérese una población  $\Pi = \{\Pi_i\}$  compuesta por  $\pi = |\Pi|$  recorridos válidos del TSP.

**Definición 1** *Distancia media de un recorrido  $\Omega_x$  a una población  $\Pi$ .*

$\delta(\Pi, \Omega_x) = \frac{1}{\pi} \sum_{i=1}^{\pi} \delta(\Pi_i, \Omega_x)$ . Si  $\Omega_x = \Omega_*$  da una noción de que tan cerca está una población a la solución óptima  $\Omega_*$ .

Boese estudió en [Boese95] el espacio de soluciones del TSP att532 de 532 ciudades. En este estudio fueron realizadas 2,500 corridas de distintas heurísticas de Búsqueda local. Para cada heurística,  $\pi$  recorridos diferentes fueron almacenados en un conjunto  $\Pi = \{\Pi_i\}$ . Cada recorrido  $\Pi_i$  almacenado tenía una longitud  $l(\Pi_i)$  (el conjunto de longitudes se denota  $L(\Pi) = \{l(\Pi_i)\}$ ), una distancia a  $\Omega_*$  denotada como  $\delta(\Pi_i, \Omega_*)$  (el conjunto de distancias al óptimo se denota  $\delta^*(\Pi) = \{\delta(\Pi_i, \Omega_*)\}$ ), y una distancia media a las otras soluciones de  $\Pi$  llamada  $\delta(\Pi, \Pi_i)$  (el conjunto de distancias medias a la población se denota  $\delta^\Pi(\Pi) = \{\delta(\Pi, \Pi_i)\}$ ). Boese calculó una correlación positiva entre estas tres variables. Dado que el conjunto  $L(\Pi)$  tiene una correlación positiva con el conjunto  $\delta^\Pi(\Pi)$ , Boese sugirió una estructura globalmente convexa del espacio de soluciones del TSP. En otras palabras, mientras más central es la posición de una solución  $\Pi_i$  dentro del conjunto de soluciones  $\Pi$ , menor su distancia media a las otras soluciones, por lo tanto menor es su longitud esperada  $l(\Pi_i)$ , esto es, mejor es la solución.

En la figura 2.2 se presenta un ejemplo sencillo para explicar este concepto de una manera más gráfica. La figura 2.2 (a) muestra una población de puntos en una curva globalmente convexa y es fácil concluir que aquellos puntos con menor distancia media a los demás (o sea con posición más central) tienen longitudes menores. En la figura 2.2 (b) se presenta un gráfico de la longitud de los puntos en función a la distancia media a los demás puntos y se puede observar con facilidad

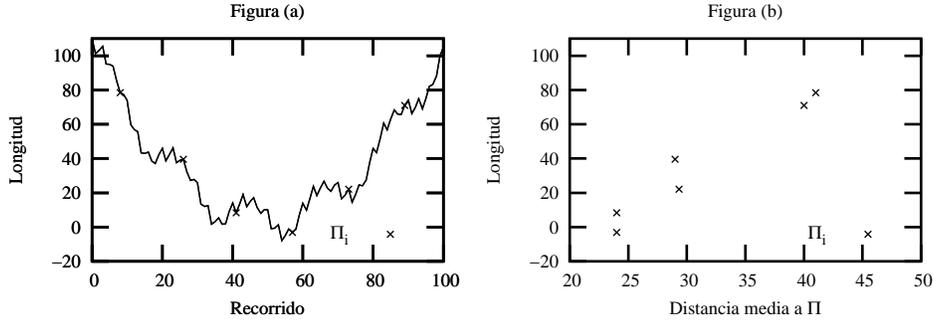


Figura 2.2: Sencillo ejemplo para explicar la conjetura de una estructura globalmente convexa del espacio de soluciones del TSP

la correlación positiva entre estas dos variables. Esta misma correlación es la calculada por Boese y por lo tanto de allí viene la sugerencia de una estructura globalmente convexa del espacio de soluciones del TSP.

Además Boese encontró resultados similares para dos instancias elegidas al azar con 100 y 500 ciudades [Boese94]. También Stützle y Hoos calcularon una correlación positiva entre la calidad de una solución y su distancia al óptimo global para los problemas rat783 y fl1577 [Stützle00].

Inspirados en estos estudios, en este trabajo se decidió hacer el mismo tipo de análisis empírico de la estructura del espacio del TSP, pero de manera más sistemática. Fueron estudiadas 13 instancias del TSP de distintos tamaños extraídas de la TSPLIB (en el nombre del problema se encuentra la información sobre el número de ciudades del mismo). La heurística de búsqueda local utilizada para hallar una población  $\Pi = \{\Pi_i\}$  de 1,000 mínimos locales fue 3-opt [Gardel04]. Los valores de correlación  $\Phi$  entre las variables  $L(\Pi)$  y  $\delta^*(\Pi)$ , o sea  $\Phi[L(\Pi), \delta^*(\Pi)]$ , así como  $\Phi[L(\Pi), \delta^\Pi(\Pi)]$ , son resumidos en la tabla 2.1.

Resulta interesante observar el gráfico de  $l(\Omega_x)$  en función de  $\delta(\Omega_*, \Omega_x)$  y de  $l(\Omega_x)$  en función de  $\delta(\Pi, \Omega_x)$  para estas poblaciones. En todos los problemas se observan gráficos muy similares, razón por la cual sólo se presenta una muestra de los mismos. Los gráficos para el problema berlin52 pueden observarse en la figuras 2.3 y 2.4 respectivamente.

No se ha realizado este estudio para instancias de gran tamaño considerando los siguientes puntos.

- Los ejemplos anteriormente citados de instancias de gran tamaño que ya fueron estudiadas por otros autores para heurísticas de búsqueda local.
- La ineficiente utilización de la capacidad de cómputo en la búsqueda de mínimos locales.
- Para los fines de este estudio, que es el análisis del espacio total de soluciones del TSP, no es la política más adecuada.

Tabla 2.1:  $\Phi[L(\Pi), \delta^*(\Pi)]$  y  $\Phi[L(\Pi), \delta^\Pi(\Pi)]$  para 13 instancias de la TSPLIB

Problema	$\Phi[L(\Pi), \delta^*(\Pi)]$	$\Phi[L(\Pi), \delta^\Pi(\Pi)]$
fri26	0.797	0.823
bayg29	0.757	0.875
bays29	0.696	0.868
att48	0.595	0.821
eil51	0.657	0.915
berlin52	0.715	0.865
st70	0.607	0.899
pr76	0.708	0.881
eil76	0.696	0.941
kroA100	0.692	0.904
kroC100	0.700	0.924
kroD100	0.699	0.912
eil101	0.708	0.944

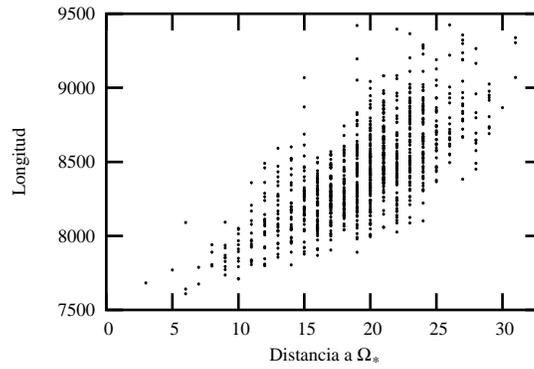


Figura 2.3:  $l(\Omega_x)$  en función de  $\delta(\Omega_*, \Omega_x) \forall \Omega_x \in \Pi$  para el problema berlin52

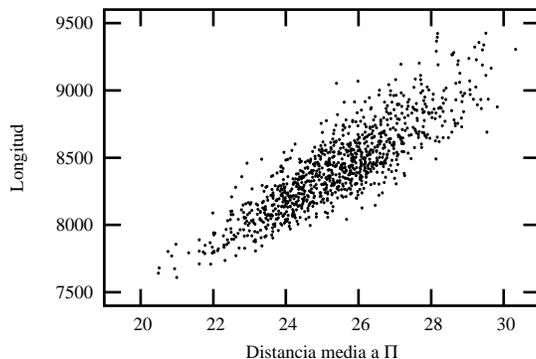


Figura 2.4:  $l(\Omega_x)$  en función de  $\delta(\Pi, \Omega_x) \forall \Omega_x \in \Pi$  para el problema berlin52

Para estudios en problemas grandes, se decidió que la población  $\Pi$  se encuentre compuesta por muestras del espacio de soluciones a todas las posibles distancias de la solución óptima  $\Omega_*$  [Gardel04]. Así, para un problema de  $n$  ciudades, existen soluciones a distancias desde 2 hasta  $n$ , o sea  $(n - 1)$  posibles distancias a  $\Omega_*$ . Para cada distancia posible se obtuvieron 24 muestras en todos los problemas, i.e.  $\pi = 24 \cdot (n - 1)$ , excepto en pcb442 donde se obtuvieron 9 muestras y en pr1002 donde se obtuvieron 6 muestras para cada distancia con el objetivo de mantener  $\pi$  en un valor manejable.

Esta nueva política fue tomada considerando los siguientes puntos.

- Resulta obvio que a excepción de instancias muy pequeñas, el número de recorridos posibles hace materialmente imposible un estudio del espacio completo de soluciones.
- Como en este estudio se busca hallar la correlación entre la longitud  $l$  de los recorridos y la distancia  $\delta$  al óptimo o a la población, resulta interesante hallar una población de muestras uniformemente distribuidas.
- Finalmente, debido a que los puntos de búsqueda para los algoritmos basados en población, como el estudiado en este trabajo, no necesariamente se encuentran en óptimos locales.

Los valores de correlación  $\Phi[L(\Pi), \delta^*(\Pi)]$  y  $\Phi[L(\Pi), \delta^\Pi(\Pi)]$  hallados para estas muestras fueron todos muy altos y son resumidos en la tabla 2.2.

Todas las figuras de  $l(\Omega_x)$  en función de  $\delta(\Omega_*, \Omega_x)$  y de  $l(\Omega_x)$  en función de  $\delta(\Pi, \Omega_x)$  para las poblaciones obtenidas con esta nueva política muestran características muy similares. En las figuras 2.5 y 2.6 se pueden observar una muestra de lo observado en las 20 instancias analizadas.

Todos estos resultados experimentales son consistentes con la conjetura de una estructura globalmente convexa del espacio de soluciones del TSP. Lógicamente no se puede afirmar que esta estructura globalmente convexa se da necesariamente en todos los casos, pero definitivamente, los resultados experimentales

Tabla 2.2:  $\Phi[L(\Pi), \delta^*(\Pi)]$  y  $\Phi[L(\Pi), \delta^\Pi(\Pi)]$  para muestras a todas las posibles distancias de  $\Omega_*$  en 20 instancias de la TSPLIB

Problema	$\Phi[L(\Pi), \delta^*(\Pi)]$	$\Phi[L(\Pi), \delta^\Pi(\Pi)]$
fri26	0.954	0.956
bayg29	0.961	0.960
bays29	0.961	0.958
att48	0.972	0.972
eil51	0.980	0.980
berlin52	0.978	0.978
st70	0.986	0.986
pr76	0.987	0.987
eil76	0.987	0.987
kroA100	0.988	0.988
kroC100	0.988	0.988
kroD100	0.989	0.989
eil101	0.990	0.990
lin105	0.988	0.988
ch130	0.993	0.993
ch150	0.994	0.994
tsp225	0.996	0.996
a280	0.996	0.996
pcb442	0.998	0.998
pr1002	0.999	0.999

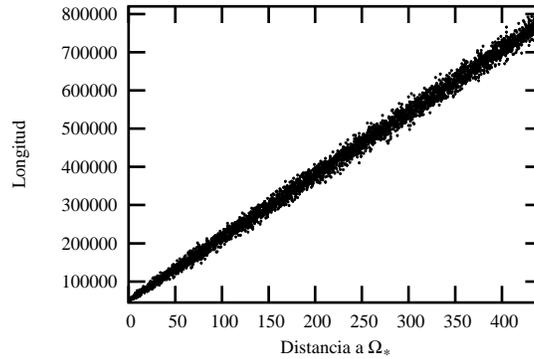


Figura 2.5:  $l(\Omega_x)$  en función de  $\delta(\Omega_*, \Omega_x) \forall \Omega_x \in \Pi$  para el problema pcb442

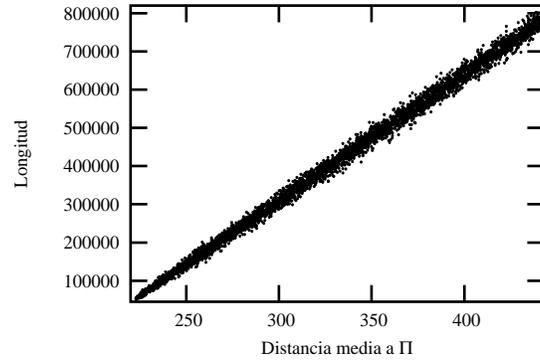


Figura 2.6:  $l(\Omega_x)$  en función de  $\delta(\Pi, \Omega_x) \forall \Omega_x \in \Pi$  para el problema pcb442

demuestran que sí se dan en la mayoría de los casos, y de hecho, no se logró encontrar siquiera un problema del TSP que no sea globalmente convexo.

## Capítulo 3

# Optimización por Colonia de Hormigas

Optimización por Colonia de Hormigas o *Ant Colony Optimization* (ACO) es una metaheurística propuesta por Dorigo et al. que ha sido inspirada en el comportamiento de búsqueda de alimentos de las colonias de hormigas [Dorigo99]. ACO es considerado un componente más del área de Inteligencia Artificial llamada Computación Evolutiva.

### 3.1. Comportamiento de Búsqueda de Alimentos de la Hormigas

Algunas especies de hormigas depositan feromonas (compuesto químico que las hormigas pueden oler) en el suelo cuando caminan. Por medio de las feromonas, las hormigas crean un rastro que puede ser usado, por ejemplo, para marcar el camino desde su nido a la comida. Biólogos utilizaron un puente de dos brazos para interconectar un nido de hormigas con una fuente de comida con el objetivo de estudiar a los rastros de feromonas y su comportamiento en condiciones experimentales controladas. Ellos realizaron experimentos donde variaban la longitud de los brazos. Los resultados más interesantes para el propósito de este trabajo se dan cuando un brazo es más largo que otro. Fue observado el porcentaje de hormigas que elegían uno u otro brazo a través del tiempo. Al inicio las hormigas eran libres de moverse con libertad y en esta etapa se observaron oscilaciones debidas al azar. El resultado final en la mayoría de los experimentos fue que la mayoría de las hormigas terminaron usando el brazo más corto.

Estos resultados pueden ser explicados de la siguiente manera. Cuando se inicia el experimento no hay feromonas en los brazos, por lo tanto las hormigas no tienen preferencia y eligen con la misma probabilidad cualquiera de los brazos. Entonces puede esperarse, en promedio, que la mitad de las hormigas elijan el

brazo corto y la otra mitad el brazo largo, aunque oscilaciones estocásticas pueden favorecer ocasionalmente a un brazo sobre otro. Sin embargo debido a que un brazo es más corto que el otro, las hormigas que eligieron este brazo llegan antes a la comida y por lo tanto inician antes su retorno al nido. En este momento deben decidir nuevamente entre el brazo corto y el largo, pero el mayor nivel de feromonas en el brazo corto influencia su decisión en su favor<sup>1</sup>. De esta manera, las feromonas comienzan a acumularse con mayor rapidez en el brazo corto, que eventualmente es utilizado por la gran mayoría de las hormigas

## 3.2. Optimización por Colonia de Hormigas

Inspirado en el comportamiento anteriormente explicado y considerando su semejanza con el TSP, el primer algoritmo ACO propuesto fue el Sistema de Hormigas o *Ant System* (AS) para resolver pequeñas instancias del Problema del Cajero Viajante [Dorigo91]. AS fue propuesto por Dorigo y es el primer ejemplo de un algoritmo ACO en la literatura. Aunque los primeros resultados fueron prometedores, AS no resultó competitivo con el estado del arte en algoritmos de optimización combinatoria diseñados para el TSP. Con la motivación de obtener algoritmos competitivos, ACO ha recibido una creciente atención de la comunidad científica. Esto puede comprobarse por la gran cantidad de publicaciones y sus diferentes campos de aplicación [Stützle00].

Con el objetivo de proveer un marco de referencia para el diseño de nuevas versiones de algoritmos, la metaheurística ACO fue propuesta posteriormente a las primeras versiones [Dorigo99]. Aun cuando existen varias versiones de ACO, se presenta a continuación lo que se considera un enfoque estándar.

### 3.2.1. Enfoque Estándar de ACO

ACO usa una matriz de feromonas  $\tau = \{\tau_{ij}\}$  para almacenar el nivel de feromonas de cada arco de un TSP. Esta matriz de feromonas es inicializada con los valores  $\tau_{ij} = \tau_{init} \forall (i, j)$ , donde  $\tau_{init} > 0$ . También explota información heurística usando  $\eta_{ij} = \frac{1}{d(i,j)}$ . Los parámetros  $\alpha$  y  $\beta$  definen la influencia relativa entre los rastros de feromonas y la información heurística.

En cada iteración del algoritmo, cada una de las  $m$  hormigas de la colonia construye un recorrido completo  $M_k$ , creándose por lo tanto un conjunto de recorridos  $M = \{M_k\}$ . Para la construcción de cada recorrido se realiza el procedimiento siguiente. Inicialmente es elegida una ciudad al azar, luego es seleccionada una ciudad iterativamente hasta completar un recorrido válido. Mientras es visitada una ciudad  $i$ ,  $\mathcal{N}_i$  representa el conjunto de ciudades todavía no visitadas y la probabilidad de elegir una ciudad  $j$  desde la ciudad  $i$  es definida como

---

<sup>1</sup>Un proceso como éste, en el cual una decisión tomada en un momento incrementa la probabilidad de tomar la misma decisión en un momento futuro es llamado un proceso autocatalítico. Un proceso autocatalítico explota la realimentación positiva

$$\mathcal{P}_{ij} = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{\forall g \in \mathcal{N}_i} \tau_{ig}^\alpha \cdot \eta_{ig}^\beta} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{en otro caso} \end{cases} \quad (3.1)$$

Luego se aplica evaporación de feromonas para todo arco  $(i, j) \in A$  de acuerdo a  $\tau_{ij} = (1 - \rho) \cdot \tau_{ij}$ , donde el parámetro de entrada  $\rho \in (0, 1]$  determina la tasa de evaporación. Finalmente cada hormiga  $k$  deposita feromonas en los arcos componentes del recorrido construido por ella  $M_k$ , de longitud  $l(M_k)$ , de acuerdo a  $\tau_{ij} = \tau_{ij} + 1/l(M_k)$ . Esto significa que mientras más corto es el recorrido de una hormiga, más cantidad de feromonas recibirán los arcos componentes de ese recorrido. En general, los arcos que son utilizados por muchas hormigas y los arcos componentes de recorridos de longitud pequeña recibirán más feromonas y por lo tanto serán más frecuentemente elegidos en las iteraciones futuras del algoritmo.

A continuación, el pseudocódigo del enfoque estándar de ACO es presentado, donde los comentarios están anteceditos por el símbolo %.

#### **Pseudocódigo de ACO (Enfoque Estándar)**

```

u = 0 % Inicialización del contador de iteraciones
τ = Inicializar la matriz de feromonas ( ) % τij = τinit ∀(i, j)
REPETIR HASTA condición de fin
    M = Construir soluciones (τ) % Generación de m soluciones usando τ
    τ = Actualizar la matriz de feromonas (M) % Actualización de τ usando M
    u = u + 1 % Incremento del contador de iteraciones

```

### **3.2.2. MAX-MIN Ant System**

Investigación referente a ACO ha demostrado que puede obtenerse un mejor rendimiento por medio de la explotación de las mejores soluciones encontradas durante la corrida, aun cuando este procedimiento agrava el problema del estancamiento prematuro de la búsqueda. Por lo tanto, la clave para lograr un mejor rendimiento de los algoritmos ACO es combinar la explotación de las mejores soluciones encontradas con un mecanismo para evitar el estancamiento temprano. El *MAX-MIN Ant System* (*MMAS*), diseñado por Stützle y Hoos [Stützle00], ha sido especialmente desarrollado para cumplir con estos requerimientos. Esto lo realiza utilizando la misma estructura que el enfoque estándar, pero considerando un estrategia elitista. En esta estrategia la mejor solución encontrada en la iteración o en la corrida  $M_{best}$  actualiza  $\tau$  de acuerdo  $\tau_{ij} = \tau_{ij} + \Delta\tau$ , donde  $\Delta\tau = 1/l(M_{best})$  si  $(i, j) \in M_{best}$  y  $\Delta\tau = 0$  si  $(i, j) \notin M_{best}$ . En *MMAS*, valores máximos y mínimos para  $\tau$  son impuestos ( $\tau_{min}$  y  $\tau_{max}$ ) para evitar el estancamiento temprano de la búsqueda. Además,  $\tau$  es inicializada con  $\tau_{max}$  para acentuar la exploración al inicio del algoritmo. Por medio de resultados experimentales se ha demostrado que *MMAS* mezclado con heurísticas de búsqueda local es uno de los mejores algoritmos para la resolución del TSP [Stützle00].

### 3.2.3. ACOs Basados en Población

Los ACOs basados en Población o *Population-based ACOs* (P-ACO) fueron inicialmente diseñados por Guntensch and Middendorf [Guntensch02a] para problemas de optimización combinatoria dinámicos. Así como en el enfoque estándar,  $\tau$  es inicializada con los valores  $\tau_{ij} = \tau_{init} \forall (i, j)$ , donde  $\tau_{init} > 0$ , aunque el enfoque P-ACO actualiza de una manera diferente la matriz de feromonas. P-ACO obtiene la matriz de feromonas a través de una población  $Q = \{Q_x\}$  de  $q$  soluciones o individuos como sigue. Primero, en cada iteración cada una de las  $m$  hormigas construye una solución usando las probabilidades dadas por la ecuación (3.1), la mejor solución hallada entra a la población  $Q$ . Cuando una solución  $Q_{in}$  entra a la población, entonces  $\tau_{ij}$  es actualizada de acuerdo con  $\tau_{ij} = \tau_{ij} + \Delta\tau$ , donde  $\Delta\tau = \Delta$  si  $(i, j) \in Q_{in}$  y  $\Delta\tau = 0$  si  $(i, j) \notin Q_{in}$ . Luego de que las primeras  $q$  soluciones entren en  $Q$ , i.e., se haya finalizado la inicialización de la población, una solución  $Q_{out}$  debe abandonar la población en cada iteración. Cual solución debe abandonar la población es decidido por la estrategia de actualización de la población. Cuando una solución  $Q_{out}$  abandona la población, entonces  $\tau_{ij} = \tau_{ij} - \Delta\tau$ , donde  $\Delta\tau = \Delta$  si  $(i, j) \in Q_{out}$  y  $\Delta\tau = 0$  si  $(i, j) \notin Q_{out}$ . P-ACO reemplaza la evaporación de los rastros de feromonas por medio del abandono de una solución por iteración. El valor  $\Delta$  es una constante determinada por los parámetros de entrada siguientes, tamaño de la población  $q$ , rastro de feromonas inicial o mínimo  $\tau_{init}$  y rastro de feromonas máximo  $\tau_{max}$ . Así,  $\Delta = (\tau_{max} - \tau_{init})/q$  [Guntensch02a].

#### Estrategia de Actualización de Población *FIFO-Queue*

La estrategia de actualización de población *FIFO-Queue* fue la primera estrategia diseñada [Guntensch02a], buscando simular el comportamiento del enfoque estándar de ACO. En esta estrategia,  $Q_{out}$  es el individuo más viejo de la población  $Q$ .

A continuación, el pseudocódigo de P-ACO *FIFO-Queue* es presentado.

#### Pseudocódigo de P-ACO *FIFO Queue*

```

u = 0 % Inicialización del contador de iteraciones
Q = Inicializar la población () % Generación de q individuos
REPETIR HASTA condición de fin
    τ = Actualizar la matriz de feromonas (Q) % Actualización de τ usando Q
    M = Construir soluciones (τ) % Generación de m soluciones usando τ
    Mbest = mejor solución de M
    Q = Actualizar la población (Mbest) % Mbest reemplaza al Qx más viejo
    u = u + 1 % Incremento del contador de iteraciones

```

#### Estrategia de Actualización de Población *Quality*

Una variedad de estrategias fueron estudiadas en [Guntensch02b], una de ellas que resulta relevante para este trabajo es la estrategia *Quality*. En esta estrategia la peor solución (considerando su calidad) del conjunto  $\{Q, Q_{in}\}$  abandona

este conjunto para componer nuevamente la población  $Q$ . Esto asegura que las mejores soluciones encontradas hasta el momento conformen la población  $Q$ .

A continuación, el pseudocódigo de P-ACO *Quality* es presentado.

#### Pseudocódigo de P-ACO *Quality*

```

u = 0                                     % Inicialización del contador de iteraciones
Q = Inicializar la población ()           % Generación de q individuos
REPETIR HASTA condición de fin
    τ = Actualizar la matriz de feromonas (Q) % Actualización de τ usando Q
    M = Construir soluciones (τ)           % Generación de m soluciones usando τ
    Mbest = mejor solución de M
    Q = Actualizar la población (Mbest) % Mbest reemplaza al peor Qx de {Q, Qin}
    u = u + 1                             % Incremento del contador de iteraciones

```

### 3.2.4. Ómicron ACO

En la búsqueda de herramientas de análisis de ACO se pensó en desarrollar un algoritmo muy simple que intentara mantener ciertas características de ACO, y más específicamente de *MMAS*. De esta manera fue concebido el Ómicron ACO (OA), su nombre proviene del principal parámetro de entrada utilizado, Ómicron ( $O$ ). Ómicron ACO (OA) fue evolucionando hasta convertirse en un algoritmo, al mismo tiempo, conceptualmente simple y competitivo [Gómez04b]. Esta simplicidad conceptual, sumada al estudio del espacio de soluciones del TSP, permitió utilizarlo para analizar las razones del éxito de ACO en el TSP [Gómez04a, Gómez04c].

La principal diferencia entre *MMAS* y OA se encuentra en la manera en que los algoritmos actualizan la matriz de feromonas. En OA, se define una matriz de feromonas constante  $\tau^0$  con  $\tau_{ij}^0 = 1, \forall i, j$ . OA mantiene una población  $Q = \{Q_x\}$  de  $q$  individuos, las mejores soluciones diferentes encontradas hasta el momento. El mejor individuo de  $Q$  en cualquier momento es llamado  $Q_*$ , mientras que el peor individuo  $Q_{worst}$ .

En OA la primera población es elegida usando  $\tau^0$ . A cada iteración un nuevo individuo  $Q_{new}$  es generado, reemplazando a  $Q_{worst} \in Q$  si  $Q_{new}$  es mejor que  $Q_{worst}$  y diferente de cualquier otro  $Q_x \in Q$ . Luego de  $K$  iteraciones,  $\tau$  es recalculada. Primero,  $\tau = \tau^0$ ; luego  $\frac{Q}{q}$  es sumado a cada elemento  $\tau_{ij}$  por cada vez que un arco  $(i, j)$  aparece en cualquiera de los  $q$  individuos presentes en  $Q$ . El anterior procedimiento es repetido cada  $K$  iteraciones hasta que la condición de fin sea satisfecha (vea el pseudocódigo para más detalles). Note que  $1 \leq \tau_{ij} \leq (1 + O)$ , donde  $\tau_{ij} = 1$  si el arco  $(i, j)$  no se encuentra presente en ningún  $Q_x$ , mientras  $\tau_{ij} = (1 + O)$  si el arco  $(i, j)$  se encuentra presente en cada  $Q_x$ .

A continuación es presentado el pseudocódigo de OA.

### Pseudocódigo principal de *Ómicron ACO*

Parámetros de entrada:  $n$ , matriz  $D = \{d_{ij}\}$ ,  $O$ ,  $K$ ,  $q$ ,  $\alpha$ ,  $\beta$   
Parámetros de salida:  $Q$  ( $q$  mejores soluciones encontradas)

$Q =$  Inicializar población ( $\tau^0$ )  
REPETIR HASTA condición de fin  
 $\tau =$  Calcular matriz de feromonas ( $Q$ )  
REPETIR  $K$  VECES  
 $Q_{new} =$  Construir una solución ( $\tau$ )  
SI  $l(Q_{new}) < l(Q_{worst})$  Y  $Q_{new} \notin Q$   
 $Q =$  Actualizar población ( $Q_{new}$ ,  $Q$ )

### Pseudocódigo de la función *Inicializar población* ( $\tau^0$ )

Inicializar conjunto  $Q$  como vacío  
MIENTRAS  $|Q| < q$   
 $Q_{new} =$  Construir una solución ( $\tau$ )  
SI  $Q_{new} \notin Q$  ENTONCES incluir  $Q_{new}$  en  $Q$   
Ordenar  $Q$  de peor a mejor considerando  $l(Q_x)$   
 $Q_{worst} = Q_0$

### Pseudocódigo de la función *Construir una solución* ( $\tau$ )

$Q_{new}[0] =$  Elegir una ciudad al azar  
REPETIR DESDE  $x = 1$  HASTA  $n - 1$   
 $Q_{new}[x] =$  Elegir una ciudad al azar según ecuación (3.1)

### Pseudocódigo de la función *Calcular matriz de feromonas* ( $Q$ )

$\tau = \tau^0$   
REPETIR por cada  $Q_x$  de  $Q$   
REPETIR por cada arco  $(i, j)$  de  $Q_x$   
 $\tau_{ij} = \tau_{ij} + \frac{O}{q}$

### Pseudocódigo de la función *Actualizar la población* ( $Q_{new}$ , $Q$ )

$Q_0 = Q_{new}$   
Ordenar  $Q$  eficientemente de peor a mejor considerando  $l(Q_x)$   
 $Q_{worst} = Q_0$

Aunque P-ACO y OA tuvieron motivaciones y orígenes diferentes, ambos algoritmos mantienen características similares, como que ambos se basan en una población. La principal diferencia entre OA y P-ACO *Quality* es que OA no permite individuos idénticos en su población. Además, OA actualiza  $\tau$  cada  $K$  iteraciones, mientras que P-ACO actualiza  $\tau$  en cada iteración.

Note que cualquier ACO elitista puede ser considerado de alguna manera como un ACO basado en población, con una población que crece a cada iteración y donde los individuos más viejos tienen menor influencia en  $\tau$  debido a

la evaporación. En otras palabras, un ACO elitista podría guardar en un vector los  $M_{best}$  de cada iteración (o sea una población). Con este vector podría reproducir el estado de  $\tau$  en cualquier momento (note que toda la información actualizada en  $\tau$  por medio de la evaporación y el depósito de feromonas puede ser completamente extraída de la población anteriormente citada). Obviamente las soluciones más viejas tendrán menos influencia en  $\tau$  debido a la reiterada evaporación en las mismas. Entonces se podrían eliminar las soluciones más viejas del vector (manteniendo una población de tamaño constante) sin distorsionar considerablemente el comportamiento del algoritmo. Por medio del anterior razonamiento se puede concluir que los ACOs basados en población y los ACOs elitistas fundamentan su éxito en las mismas razones.

## Capítulo 4

# Resultados Experimentales

Considerando que Ómicron ACO (OA) estuvo basado en *MAX-MIN Ant System* (*MMAS*), el siguiente estudio preliminar de rendimiento fue realizado comparando el OA con el *MMAS* para mostrar las ventajas obtenidas. Además hay que recordar que *MMAS* es uno de los mejores algoritmos ACO del momento. Para los siguientes resultados experimentales, se utilizó una computadora con un procesador de 2 GHz y con 256 MB de RAM. El lenguaje de programación usado fue lenguaje C y el sistema operativo Linux.

### 4.1. Estudio Comparativo entre OA y *MMAS*

Como referencia de rendimiento, los parámetros para el algoritmo *MMAS* fueron extraídos de [Stützle00], donde  $\alpha = 1$  y  $\beta = 2$  fueron siempre usados y los mismos problemas fueron resueltos. Para realizar una comparación justa, los mismos  $\alpha = 1$  y  $\beta = 2$  fueron siempre usados para el OA. El resto de los parámetros fueron hallados empíricamente, buscando un comportamiento equilibrado entre la velocidad de convergencia y la calidad de la solución final, eligiendo  $O = 600$ ,  $q = 25$  y  $K = 1,000$ . Debido a que no se hicieron intentos de optimizar los parámetros de OA o hacerlos dependientes del tiempo para buscar una mayor exploración, *MMAS* con *Pheromone Trail Smooth* (PTS) no fue utilizado para realizar la comparación. La idea básica de PTS es la de incentivar la exploración incrementando la probabilidad de seleccionar arcos con bajo nivel de feromonas. PTS es esencialmente interesante para corridas largas porque ayuda a lograr una exploración más eficiente del espacio de búsqueda. Consecuentemente la convergencia del algoritmo es más lenta.

Observaciones empíricas fueron hechas en función del tiempo dado que el concepto de iteración no es el mismo para ambos algoritmos. En general, OA produce mejores resultados que *MMAS* desde el principio y converge a una solución ligeramente mejor. Para el problema eil51 de 51 ciudades estudiado en [Stützle00], el comportamiento de la mejor solución encontrada para cada algoritmo fue estudiado. La media de la evolución de ambos algoritmos en 25

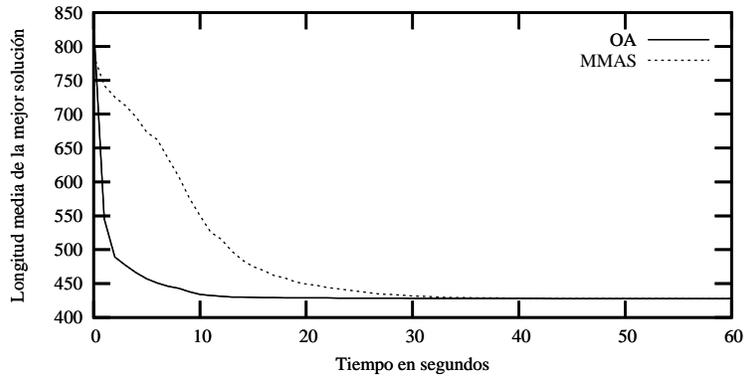


Figura 4.1: Comparación entre el comportamiento medio del OA y el *MMAS* para el TSP eil51

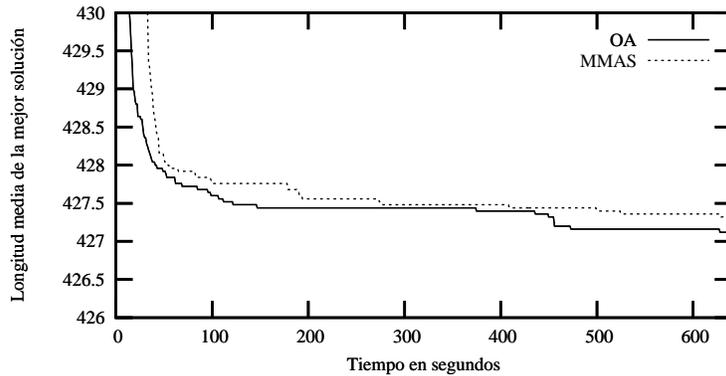


Figura 4.2: Comparación entre el comportamiento medio del OA y el *MMAS* en la fase de convergencia para el TSP eil51

corridas puede ser vista en la figura 4.1. La escalas fueron modificadas en la figura 4.2 para mostrar la clara ventaja de OA en la etapa de convergencia.

El OA resulta prometedor considerando dos razones. Primero, el mínimo trabajo de ajuste realizado para la selección de los parámetros y segundo, el hecho de que resultados parciales muestran una elevada robustez, considerando que usando exactamente los mismos parámetros, resultados similares son observados en el problema de 100 ciudades kroA100.

En la figura 4.3 se observa la media en 25 corridas de la evolución de la longitud del mejor individuo en ambos algoritmos en la resolución del conocido TSP kroA100 [Stützle00]. Las escalas fueron modificadas en la figura 4.4, así como en la figura 4.2, para mostrar la ventaja del OA sobre el *MMAS*, considerando la fase de convergencia. Una vez más, estudiando un comportamiento medio, OA supera al *MMAS*.

Note que mejores resultados fueron obtenidos e instancias de mayor tamaño fueron resueltas usando *MMAS* con búsqueda local [Stützle00]. En estas

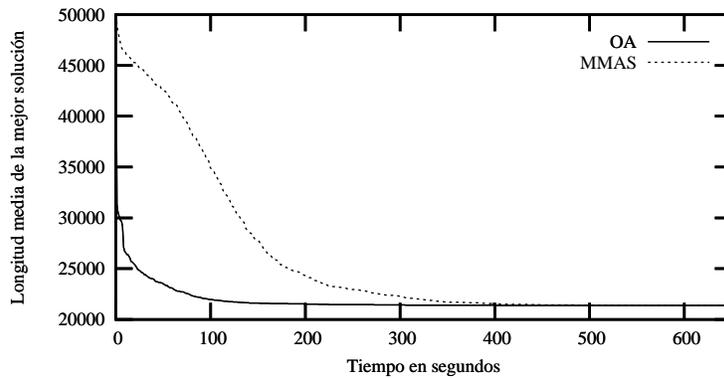


Figura 4.3: Comparación entre el comportamiento medio del OA y el *MMAS* para el TSP kroA100

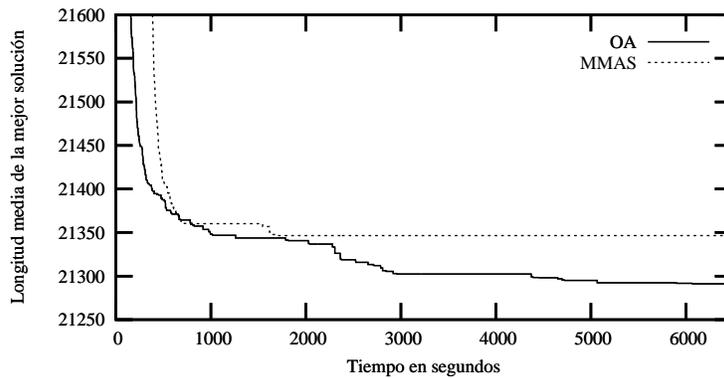


Figura 4.4: Comparación entre el comportamiento medio del OA y el *MMAS* en la fase de convergencia para el TSP kroA100

pruebas preliminares, sólo pequeñas instancias fueron resueltas (porque no fue implementado OA con búsqueda local) para hacer una comparación entre ambos algoritmos sin ningún tipo de interferencia.

El objetivo de este simple estudio experimental es mostrar que la sencillez conceptual de OA ha permitido comprender los aspectos importantes del éxito de ACO (detallados en el siguiente capítulo) y por medio esta comprensión se ha logrado un mejor rendimiento, además debido a su robustez, se ha conseguido una simplificación en la configuración de sus parámetros.

## Capítulo 5

# Razones del Éxito de ACO en el TSP

Basado en los estudios del espacio de soluciones del TSP presentados anteriormente, el presente trabajo usa el concepto de estructura globalmente convexa del problema como idea principal para explicar las razones del éxito de ACO. Antes de estudiar estas razones, resulta útil para este fin observar el comportamiento de Ómicron ACO y por lo tanto se presenta a continuación un estudio del mismo en dos instancias del TSP.

### 5.1. Estudio del Comportamiento de Ómicron ACO

Para esta sección otra definición de distancia media es presentada.

**Definición 2** *Distancia media de una población  $\Pi$ .*

$\delta(\Pi) = \frac{2}{\pi(\pi-1)} \sum_{i=1}^{\pi-1} \sum_{j=i+1}^{\pi} \delta(\Pi_i, \Pi_j)$ . Da una idea del grado de convergencia de la población.

En este estudio se observará principalmente el comportamiento de la distancia media de la población del OA al óptimo  $\delta(Q, \Omega_*)$  y la distancia media de la población del OA  $\delta(Q)$  en función al número de iteraciones. Además resulta interesante observar la longitud de  $Q_*$ ,  $l(Q_*)$ ; la longitud media de una población,  $l(Q) = \frac{1}{q} \sum_{i=1}^q l(Q_i)$ , y el número de individuos  $\zeta(Q)$  que entraron a la población. Sus valores medios para varias corridas de OA son denotados como  $l(Q_*)_M$ ,  $l(Q)_M$  y  $\zeta(Q)_M$  respectivamente. Por consiguiente  $\delta(Q, \Omega_*)_M$  y  $\delta(Q)_M$  representan el valor medio de  $\delta(Q, \Omega_*)$  y  $\delta(Q)$ .

Para mantener el número de recorridos posibles en un valor manejable, fue generado al azar un TSP de 8 ciudades con coordenadas (58,12), (2,73), (14,71), (29,8), (54,50), (0,7), (2,91) y (44,53). La figura 5.1 muestra la evolución de las variables medias definidas anteriormente en función al número de iteraciones en

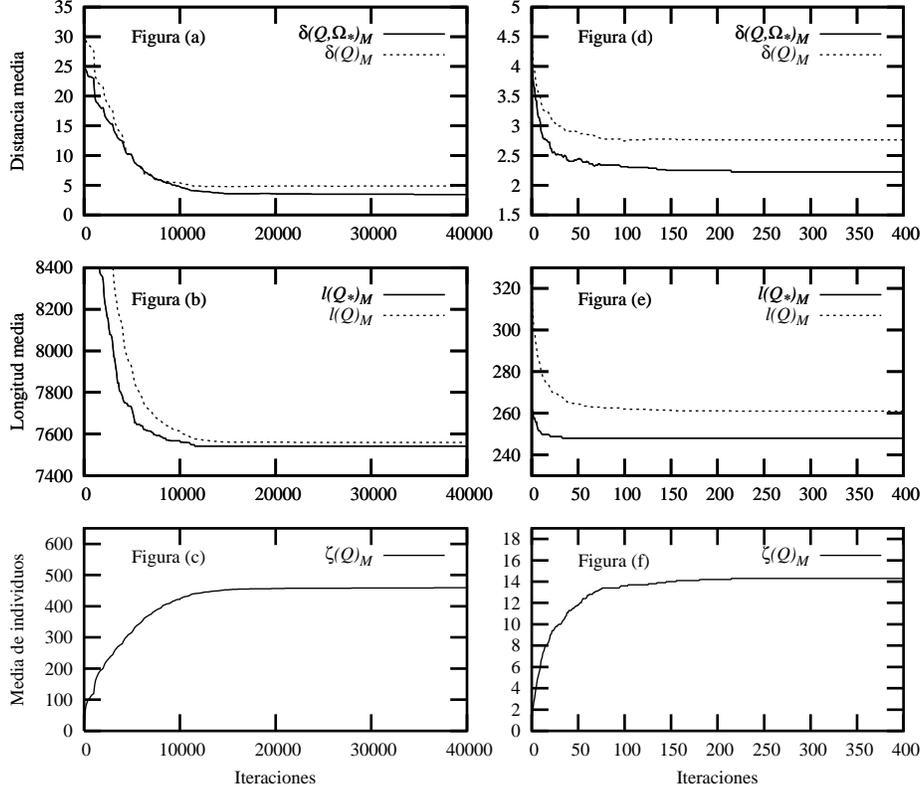


Figura 5.1: Evolución de  $\delta(Q, \Omega_*)_M$ ,  $\delta(Q)_M$ ,  $l(Q_*)_M$ ,  $l(Q)_M$  y  $\zeta(Q)_M$  en función de las iteraciones para 10 corridas. Izquierda, para el problema berlin52. Derecha, para el problema omi1

10 corridas del OA. El lado izquierdo de la figura 5.1 presenta los gráficos para el TSP berlin52 (usando los parámetros  $O = 600$ ,  $q = 25$ ,  $K = 1,000$ ,  $\alpha = 1$  y  $\beta = 2$ ), mientras el lado derecho presenta el gráfico para el TSP omi1 (usando los parámetros  $O = 30$ ,  $q = 8$ ,  $K = 10$ ,  $\alpha = 1$  y  $\beta = 2$ ).

Es interesante recalcar que el comportamiento típico de las variables para ambos problemas es similar al comportamiento medio mostrado en la figura 5.1. Además el comportamiento en ambos problemas (así como en varios otros estudiados) es casi idéntico y esto resulta importante porque los siguientes estudios se realizarán sobre el TSP omi1, que por su tamaño, tiene un espacio de soluciones manejable. De manera que, debido al comportamiento similar y considerando al estudio del espacio de soluciones, las conclusiones podrán ser extendidas del análisis de un problema pequeño a problemas de interés práctico.

Los valores de correlación entre  $\delta(Q, \Omega_*)_M$ ,  $\delta(Q)_M$ ,  $l(Q_*)_M$ ,  $l(Q)_M$  y  $\zeta(Q)_M$  para el problema berlin52 están resumidos en la tabla 5.1. Estos valores de corre-

Tabla 5.1: Correlación de las variables de comportamiento de OA estudiadas para el problema berlin52

	$\delta(Q, \Omega_*)_M$	$l(Q_*)_M$	$l(Q)_M$	$\zeta(Q)_M$
$\delta(Q)_M$	0.990	0.957	0.977	-0.972
$\delta(Q, \Omega_*)_M$		0.928	0.957	-0.995
$l(Q_*)_M$			0.996	-0.900
$l(Q)_M$				-0.934

lación son presentados para resaltar la relación entre la capacidad de búsqueda del algoritmo (la obtención de mejores soluciones), la disminución de  $\delta(Q)$ , o lo que es lo mismo el acercamiento entre los individuos de la población, y el aumento de  $\zeta(Q)$ .

Puede ser observado en la figura 5.1 que  $\delta(Q, \Omega_*)_M$ ,  $\delta(Q)_M$ ,  $l(Q_*)_M$  y  $l(Q)_M$  decrecen en la fase inicial, mientras  $\zeta(Q)_M$  crece. En otras palabras, nuevos individuos con menores longitudes entran a  $Q$  al comienzo de la corrida; estos individuos se acercan entre sí y al mismo tiempo se acercan a  $\Omega_*$ . En la fase final las variables se mantienen casi constantes. Podría decirse que casi no entran nuevos individuos a  $Q$  y que  $\delta(Q, \Omega_*)_M$  resulta menor que  $\delta(Q)_M$ , lo que significa que los individuos terminan más cerca de  $\Omega_*$  que de los otros individuos de  $Q$ . Estos resultados motivan el análisis de las razones del éxito de ACO de la siguiente sección.

## 5.2. Razones del Éxito de ACO en el TSP

El siguiente estudio exhaustivo es presentado usando el problema omi1 de 8 ciudades considerando las restricciones de procesamiento para realizar el mismo en problemas grandes. El mismo estudio exhaustivo fue realizado usando otros problemas de 7 y 9 ciudades y los resultados fueron muy similares, haciendo innecesaria la repetición de los mismos en este trabajo.

### 5.2.1. Geometría del Problema

Para estudiar la geometría del problema, tres definiciones son dadas.

**Definición 3**  $\Omega = \{\Omega_x\}$ , *i.e.* todo el espacio de búsqueda discreto del TSP.  $\Omega(\cdot)$  denotará un subespacio de  $\Omega$ , *i.e.*  $\Omega(\cdot) \subseteq \Omega$ .

**Definición 4**  $\Omega(Q) = \{\Omega_x \mid \delta(Q, \Omega_x) < \delta(Q)\}$ , *i.e.* el conjunto de recorridos  $\Omega_x$  con una distancia media a la población  $Q$  menor que la distancia media de  $Q$ .  $\Omega(Q)$  es una zona central de  $Q$ , como es ilustrado en la sección 5.2.3, figura 5.5 (b).

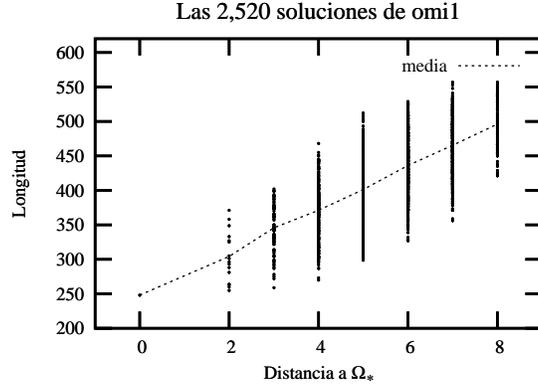


Figura 5.2:  $l(\Omega_x)$  en función a  $\delta(\Omega_x, \Omega_*) \forall \Omega_x \in \Omega$  en el TSP omi1

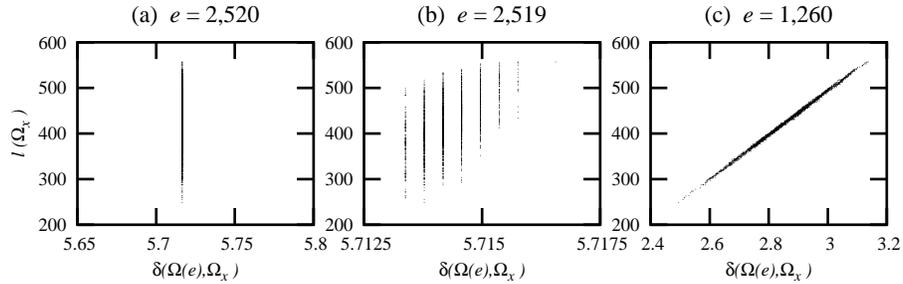


Figura 5.3:  $l(\Omega_x)$  en función de  $\delta(\Omega(e), \Omega_x) \forall \Omega_x \in \Omega$  para distintos valores de  $e$

**Definición 5**  $\Omega(e)$ . *Subespacio conformado por las  $e$  mejores soluciones de  $\Omega$ ; e.g.  $\Omega(100)$  denota el conjunto de los 100 recorridos más cortos.*

Note que los conceptos de  $\Omega(\cdot)$  son distintos dependiendo del parámetro. Cuando el parámetro es una población, se habla de una zona central de la población. Cuando el parámetro es un escalar, se habla de las mejores soluciones del problema. La figura 5.2 presenta  $l(\Omega_x)$  en función de  $\delta(\Omega_x, \Omega_*)$  para todo el espacio  $\Omega$  del problema de prueba omi1. Como en los estudios anteriores, una correlación positiva puede ser observada, en este caso una correlación de 0.7 fue calculada.

La figura 5.3 muestra  $l(\Omega_x)$  en función de  $\delta(\Omega(e), \Omega_x) \forall \Omega_x \in \Omega$ . Para  $e = 2,520$  ( $\Omega(2,520) = \Omega$ ), la figura 5.3 (a) muestra claramente que la correlación de las variables es 0 ya que la distancia media de cualquier solución a todas las demás es la misma. La figura 5.3 (b) muestra el mismo gráfico para  $e = 2,519$ , i.e. eliminando la peor solución de  $\Omega$ . Para este caso la correlación crece a 0.521. Finalmente, la figura 5.3 (c) muestra el gráfico para  $e = 1,260$  (las mejores  $\frac{|\Omega|}{2}$  soluciones) y la correlación entre las variables es 0.997.

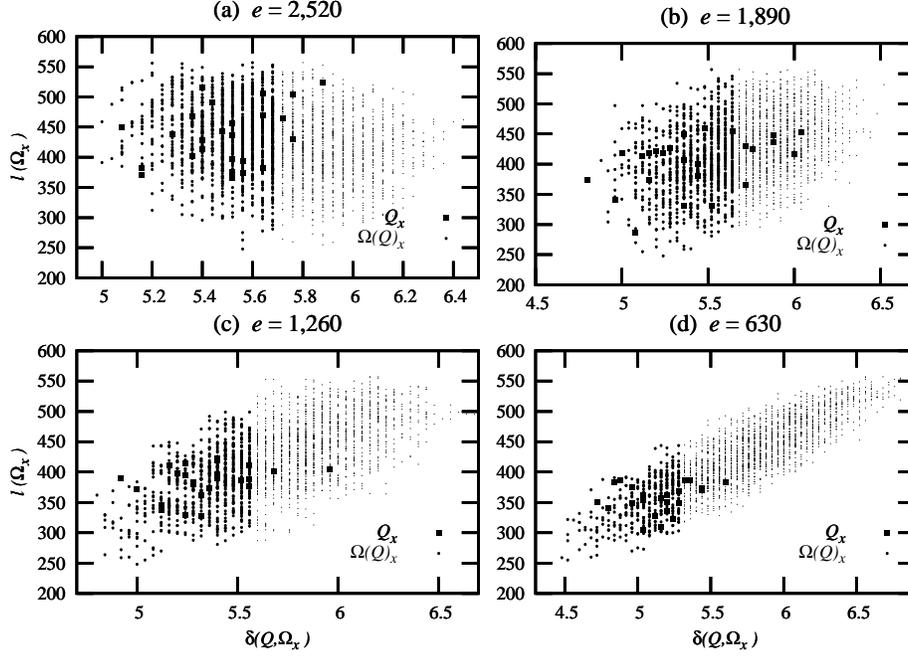


Figura 5.4:  $l(\Omega_x)$  en función de  $\delta(Q, \Omega_x) \forall \Omega_x \in \Omega$  para  $Q = \Omega(e, 25)$  y para distintos valores de  $e$

Estos resultados son nuevamente consistentes con la conjetura de una estructura globalmente convexa del espacio de soluciones del TSP, ya que mientras menor es la distancia de una solución  $\Omega_x$  a un conjunto de buenas soluciones (y por lo tanto, más central es su posición en  $\Omega(e) \subset \Omega$ ), menor es su longitud esperada.

**Definición 6**  $\Omega(e, \omega) = \{\Omega(e, \omega)_x\}$  es definido como un conjunto de elementos diferentes elegidos al azar de  $\Omega(e)$  con cardinalidad  $\omega$ .

Dada esta interesante característica geométrica de  $\Omega(e)$ , una buena pregunta es si esta propiedad de convexidad global se mantiene para  $\Omega(e, \omega)$ . Para entender la importancia de esta pregunta, es útil recalcar que una población  $Q$  de un OA puede ser considerada como  $\Omega(e, \omega)$ . La figura 5.4 muestra  $l(\Omega_x)$  en función de  $\delta(Q, \Omega_x) \forall \Omega_x \in \Omega$  para conjuntos  $Q = \Omega(e, \omega)$  elegidos al azar con  $\omega = 25$ . Gráficos para distintos valores de  $e$  son presentados en las figuras 5.4 (a) a (d).

La figura 5.4 muestra a los individuos de  $Q$ , denominados como  $Q_x$  y a los elementos de  $\Omega(Q)$  denominados como  $\Omega(Q)_x$ . Como puede ser visto en las figuras 5.4 (b) a (d), las mejores soluciones están en  $\Omega(Q)$ ; por lo tanto, es conveniente explorar  $\Omega(Q)$ . Para interpretar mejor la figura 5.4, la tabla 5.2 presenta la correlación  $\Phi$  entre  $l(\Omega_x)$  y  $\delta(Q, \Omega_x) \forall \Omega_x \in \Omega$  para las cuatro poblaciones diferentes de la figura 5.4. Para comparar los resultados experimentales de la

Tabla 5.2: Correlación entre  $l(\Omega_x)$  y  $\delta(Q, \Omega_x)$  para distintos valores de  $e$

	$e = 2,520$	$e = 1,890$	$e = 1,260$	$e = 630$
Correlación $\Phi$ (figura 5.4)	-0.201	0.387	0.641	0.862
Correlación $\Phi_M$	0.001	0.425	0.683	0.836

figura 5.4 con valores promedios de 1,000 conjuntos  $Q = \Omega(e, 25)$  elegidos al azar, la tabla 5.2 presenta también el promedio calculado  $\Phi_M$  para los mismos parámetros  $e$ , mostrando que la figura 5.4 representa bastante bien un caso promedio.

Como puede verse en la tabla 5.2, no hay una correlación significativa  $\Phi_M$  en  $\Omega([2, 520], 25)$ , como ocurrió con  $\Omega(2, 520) = \Omega$ . Cuando se decrece el valor de  $e$ ,  $\Phi_M$  crece como ocurrió con la correlación calculada para  $\Omega(e)$ . Así, con buena probabilidad,  $\Omega(e, 25)$  es también globalmente convexa (esta probabilidad crece con  $\omega$ ). Considerando a la propiedad de convexidad global, puede ser señalado que dada una población  $Q = \Omega(e, \omega)$  de buenas soluciones, es una buena idea buscar en una zona central de  $Q$ , y especialmente en  $\Omega(Q)$ , que contiene a las soluciones con las menores distancias a  $Q$ , dada la correlación positiva entre la calidad de una solución y su distancia a  $Q$ .

### 5.2.2. El OA en una Geometría Globalmente Convexa

El OA concentra una importante proporción de su búsqueda de nuevas soluciones en  $\Omega(Q)$ . Esto puede ser entendido ya que en la construcción de una nueva solución  $Q_{new}$ , una mayor probabilidad es asignada a los arcos de cada individuo de  $Q$ . Considerando el concepto de distancia  $\delta$ , esto puede ser visto como una búsqueda hecha cerca de cada individuo de  $Q$ . En consecuencia, es razonable esperar que  $Q_{new}$  tenga varios arcos de  $Q_x \in Q$ , lo que significa que la distancia esperada  $\delta(Q, Q_{new})$  debería ser menor que  $\delta(Q)$ , i.e. es de esperarse que  $Q_{new}$  se encuentre en  $\Omega(Q)$ .

Resultados experimentales ratifican esta teoría, 1,000 poblaciones  $Q = \Omega(e, \omega)$  fueron elegidas al azar y 1,000 nuevas soluciones  $Q_{new}$  fueron generadas con cada población, usando la ecuación (3.1) con  $O = 600$ ,  $\omega = 25$ ,  $\alpha = 1$ ,  $\beta = 2$  para distintos valores de  $e$ . La tabla 5.3 muestra la proporción  $\mu$  de  $Q_{new}$  que cae dentro  $\Omega(Q)$ , la cardinalidad media de  $\Omega(Q)$  (denotada  $\lambda$ ) y la relación  $\frac{\mu}{\lambda}$  (que puede ser entendida como la proporción de  $\Omega(Q)$  explorada en promedio con la generación de cada  $Q_{new}$ ).

Al comienzo de una corrida de OA,  $e$  es muy grande y existe una buena probabilidad de generar una solución en  $\Omega(Q)$  (vea  $\mu$  en la tabla 5.3 para  $e = 2,520$ ). Después de avanzar en su cálculo,  $e$  decrece y también lo hace  $|\Omega(Q)|$ ; por lo tanto, se vuelve más difícil encontrar una nueva solución en  $\Omega(Q)$  como se muestra en la tabla 5.3 (vea como  $\lambda$  y  $\mu$  decrecen con  $e$ ). Aun cuando  $\mu$

Tabla 5.3: Valores medios  $\mu$ ,  $\lambda$  y  $\frac{\mu}{\lambda}$  para distintos valores de  $e$

	$e = 2,520$	$e = 1,890$	$e = 1,260$	$e = 630$
$\mu$	0.758	0.718	0.636	0.516
$\lambda$	1,282.06	1,098.42	761.83	372.06
$\frac{\mu}{\lambda}$	5.91e-4	6.53e-4	8.34e-4	13.86e-4

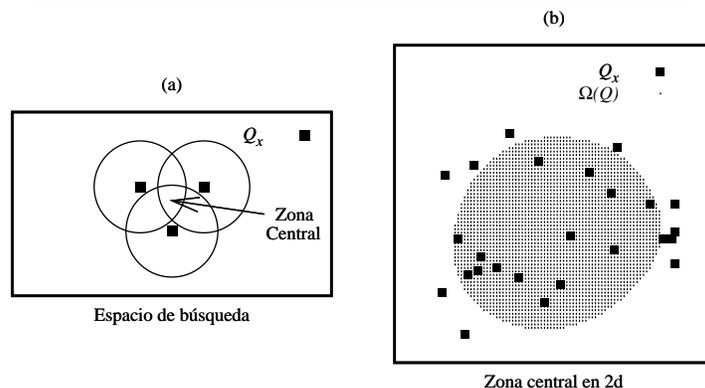


Figura 5.5: (a) Visión simplificada de la zona de búsqueda cercana a todas las soluciones de  $Q$  ( $\Omega(Q)$ ) (b) Zona central geométrica de una población de 25 puntos elegidos al azar

decrece, es interesante hacer notar que  $\frac{\mu}{\lambda}$ , que es la proporción de  $\Omega(Q)$  explorada con cada nuevo individuo, crece, i.e. OA busca más eficientemente mientras su cálculo continúa.

### 5.2.3. Interpretación en Dos Dimensiones del Espacio de Exploración del OA

Por razones didácticas, se presenta una analogía entre el espacio n-dimensional del TSP  $\Omega$  y una interpretación en dos dimensiones. Primero, la figura 5.5 (a) muestra que la búsqueda cerca de tres puntos implica una concentración de la exploración en el área central de los mismos. La intersección de las áreas cercanas a cada punto es su área central, i.e. la búsqueda cerca de cada  $Q_x \in Q$ , hecha por el OA, implica una exploración en una zona central de  $Q$ , una región recomendable para realizar la búsqueda según la sección 5.2.1

Una población  $Q$  de 25 puntos en un plano fueron seleccionados al azar y el área conformada por los puntos que poseen una distancia media a  $Q$  menor que la distancia media de la población es mostrada en la figura 5.5 (b). Esta es una interpretación en dos dimensiones de  $\Omega(Q)$  y es una zona central de la población. En consecuencia, la habilidad de OA de buscar principalmente en la zona central de  $Q$  significa que busca con buena probabilidad en  $\Omega(Q)$ .

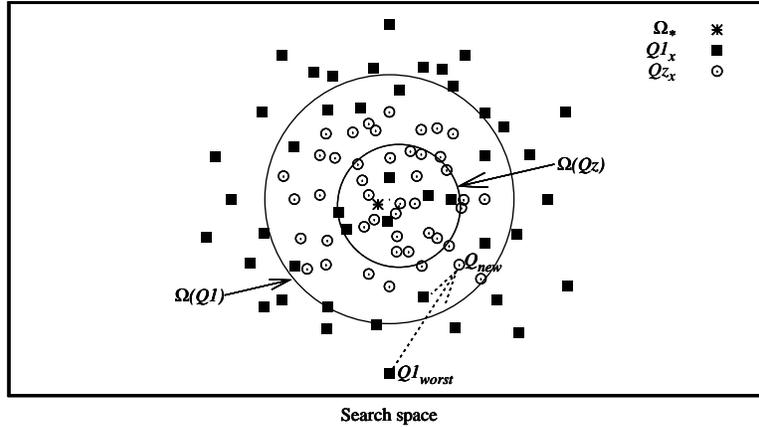


Figura 5.6: Visión simplificada del comportamiento de OA

#### 5.2.4. Razones Subyacentes del Éxito de ACO

En la figura 5.6, el espacio de búsqueda n-dimensional es simplificado a dos dimensiones para una visión geométrica del comportamiento de OA. Para entender el comportamiento típico de OA luego de su fase inicial, se asume en la figura 5.6 una población  $Q1 = \{Q1_x\} = \Omega(e, \omega)$  para  $e < |\Omega|$  de buenas soluciones uniformemente distribuidas. Como se mostró antes, OA concentra la búsqueda de nuevas soluciones en  $\Omega(Q1)$  y reemplaza a la peor solución de  $Q1$  ( $Q1_{worst}$ ) por una nueva solución  $Q_{new}$  con longitud menor  $l(Q_{new})$ . Una nueva población  $Q2$  es creada incluyendo a  $Q_{new}$ . Esto es mostrado en la figura 5.6 con una flecha en línea de puntos. Consecuentemente, es de esperarse que  $\delta(Q2, \Omega_*) < \delta(Q1, \Omega_*)$  porque existe una correlación positiva entre  $l(\Omega_x)$  y  $\delta(\Omega_x, \Omega_*)$ . De manera similar,  $\delta(Q, Q_{new}) < \delta(Q, Q_{worst})$  porque existe una correlación positiva entre  $l(\Omega_x)$  y  $\delta(Q, \Omega_x)$ , por lo tanto  $\delta(Q2) < \delta(Q1)$  y, recordando que  $\Omega(Q)$  se definió como  $\{\Omega_x \mid \delta(Q, \Omega_x) < \delta(Q)\}$ , es de esperarse que el subespacio donde se concentra la búsqueda de soluciones potenciales disminuya, como fue verificado experimentalmente en la sección 5.2.2. Otra propiedad fácilmente comprobable es que  $l(Q2) < l(Q1)$ .

OA realiza este procedimiento reiteradamente para disminuir la zona de búsqueda donde soluciones prometedoras están localizadas. Considerando la población  $Qz = \{Qz_x\}$  para  $z \gg 2$ , la figura 5.6 muestra como  $\Omega(Qz)$  ha disminuido considerablemente como consecuencia de la estructura globalmente convexa del espacio de soluciones del TSP. En este punto debería resultar claro que la razón principal del éxito de OA es su habilidad para buscar en la zona central de  $Q$ , donde usualmente mejores soluciones se sitúan. Este análisis es consistente con el comportamiento en las primeras fases de OA observado en la sección 5.1 para los problemas berlin52 y omi1. Luego de sobrepasada esta primera fase, la población está compuesta por individuos de excelente calidad y ya no existen muchos recorridos mejores. Además, el espacio de búsqueda del TSP pierde la

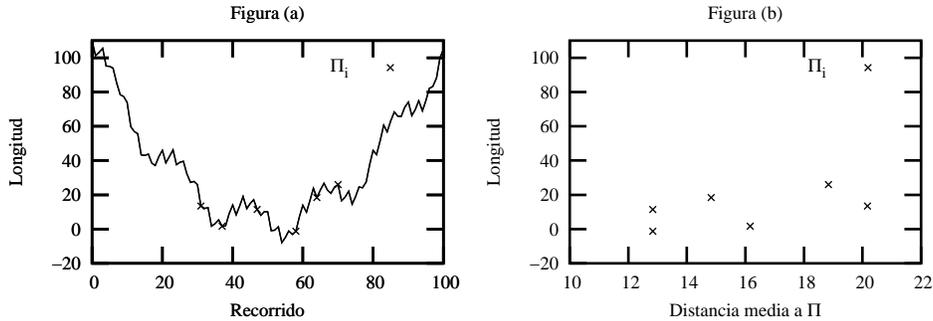


Figura 5.7: Sencillo ejemplo para explicar la pérdida de la propiedad de convexidad global en el espacio de soluciones cercano a  $\Omega_*$

característica de convexidad global como puede observarse en la figura 5.7, que presenta el mismo sencillo ejemplo que el de la figura 2.2, pero con una población  $\Pi = \{\Pi_i\}$  cercana a  $\Omega_*$ . La figura 5.7 (a) muestra una población de puntos en una curva globalmente convexa y la figura 5.7 (b) presenta un gráfico de la longitud de los puntos en función a la distancia media a los demás puntos. Se puede observar con facilidad que la correlación positiva significativa observada en la figura 2.2 (b) ya no existe para esta población. En conclusión, en la fase final de una corrida de OA casi no ingresan individuos a la población y la búsqueda se vuelve ineficiente debido al desconocimiento de la dirección hacia mejores soluciones que la característica de convexidad global facilitaba.

Dado que cualquier P-ACO mantiene una población de individuos  $Q$ , así como el OA aquí presentado, un análisis similar es aplicable para explicar las razones de su éxito. En otras palabras, la búsqueda es orientada principalmente hacia la zona central de  $Q$ , donde buenas soluciones son usualmente encontradas. Finalmente, como se mencionó en la sección 3.2.4, cualquier ACO elitista puede ser considerado como un P-ACO y por lo tanto la misma explicación es pertinente.

## Capítulo 6

# Relación entre ACO y Algoritmos Genéticos

### 6.1. Algoritmos Genéticos

Los Algoritmos Genéticos o *Genetic Algorithms* (GA) fueron introducidos por Holland como una analogía del proceso observado en la evolución de los seres vivos [Holland75]. Los GA son procedimientos de búsqueda basados en la mecánica de la selección natural y la genética. El Algoritmo Genético Simple o *Simple Genetic Algorithm* (SGA), el cual puede ser considerado un enfoque estándar [Goldberg89], es descrito a continuación.

#### 6.1.1. Algoritmo Genético Simple

El SGA tiene una población  $P$  de  $p$  individuos  $P_x$  que representan soluciones de un problema de optimización. El SGA usa una codificación binaria para sus individuos o soluciones. En un contexto de maximización, el valor de la función objetivo de cada individuo de  $P$  es considerado como su adaptabilidad o *fitness*. Una población inicial es elegida al azar, luego un conjunto de operaciones simples que utilizan y generan  $P$ s sucesivas es ejecutado iterativamente. Se espera que  $P$  mejore a través del tiempo hasta que una condición de fin sea satisfecha.

Los operadores del SGA son reproducción, cruzamiento y mutación. La reproducción es un proceso en el que  $p$  individuos son seleccionados con probabilidades proporcionales a sus *fitnesses* para ser padres. El cruzamiento es un proceso en el que 2 padres diferentes son seleccionados iterativamente del conjunto de  $p$  padres para intercambiar información entre ellos y generar 2 individuos nuevos (hijos). Esto es realizado eligiendo un punto de quiebre al azar en los padres e intercambiando partes entre ellos. La mutación es aplicada a cada hijo, y es la alteración de los bits de un individuo con una pequeña probabilidad predefinida, a veces conocida como coeficiente de mutación ( $mc$ ). Estos nuevos individuos alterados componen la nueva población.

A continuación, el pseudocódigo del SGA es presentado (los comentarios son precedidos por el símbolo %).

### Pseudocódigo del SGA

```

g = 0                                     % Inicialización del contador de generaciones
P = Inicializar la población ()          % Generación al azar de p individuos
REPETIR HASTA condición de fin
  F = Reproducción (P)                  % Selección de p padres a través de una ruleta
  C = Cruzamiento (F)                   % Intercambio de información entre diferentes pares de padres
  V = Mutación (C)                      % Alteración de bits de los individuos con probabilidad mc
  P = Actualizar la población (V)       % V es copiado a P (P = V)
g = g+1                                  % Incremento del contador de generaciones

```

## 6.1.2. Algoritmo Genético Ómicron

La literatura en Computación Evolutiva ha definido una gran variedad de GA que conservan la misma filosofía variando operadores y agregando diferentes principios como el elitismo [Goldberg89, Mühlenbein95]. Utilizando como referencia al Algoritmo Genético Simple, esta sección presenta una nueva versión, el Algoritmo Genético Ómicron u *Omicron Genetic Algorithm* (OGA), un algoritmo genético diseñado específicamente para el TSP [Gómez04d].

### Codificación

El OGA tiene una población  $P$  de  $p$  individuos o soluciones, así como el SGA. Cada individuo  $P_x$  de la población  $P$  de un OGA es un recorrido TSP válido y está determinado por los arcos  $(i, j)$  que componen el recorrido. A diferencia del SGA, donde la codificación es binaria, en OGA la misma es  $n$ -aria. Considerando un TSP con 5 ciudades  $c1, c2, c3, c4$  y  $c5$ , por ejemplo al recorrido definido por los arcos  $(c1, c4), (c4, c3), (c3, c2), (c2, c5)$  y  $(c5, c1)$  se lo codificará con el vector de la ciudades visitadas en orden, i.e.  $\{c1, c4, c3, c2, c5\}$ .

### Reproducción

Así como en la reproducción de un SGA, en OGA se seleccionan dos padres ( $F_1$  y  $F_2$ ) al azar con una probabilidad proporcional al *fitness* de cada individuo  $P_x$ , donde  $fitness(P_x) \propto 1/l(P_x)$ . A diferencia del SGA, donde dos padres generan dos hijos, en OGA ambos padres generan un solo hijo ( $C_1$ ). En SGA, primero se obtienen  $p$  hijos para que reemplacen completamente a la antigua generación. En OGA, una vez generado un hijo, este reemplaza al elemento más viejo de  $P$ . Así, en  $p$  iteraciones la población será una totalmente nueva y se podría hablar de una nueva generación. En conclusión, en OGA se realiza el mismo recambio de la población que en SGA, pero de manera paulatina.

## Cruzamiento y Mutación

El principio del cruzamiento en el SGA es que los hijos compartan información de ambos padres. En la mutación, el objetivo es que nueva información ingrese a los hijos, y por lo tanto a la población. En el SGA los operadores cruzamiento y mutación se realizan por separado. Con el objetivo de facilitar la obtención de hijos que representen recorridos válidos, en OGA el cruzamiento y la mutación se realizan en una sola operación llamada Cruzamiento-Mutación (CM). Aun así, los objetivos de ambos operadores anteriormente citados se mantendrán intactos.

Para realizar CM, los arcos del problema son representados en una ruleta  $\mathcal{R}$ , en la cual cada arco tiene un peso  $w$  o probabilidad de ser elegido. CM asigna un peso  $w$  de 1 a cada arco  $(i, j)$  de  $A$  o sea  $w_{ij} = 1 \forall (i, j) \in A$ . Luego a cada arco  $(i, j)$  de  $F_1$  se le agrega un peso de  $\frac{O}{2}$  o sea  $w_{ij} = w_{ij} + \frac{O}{2} \forall (i, j) \in F_1$ , donde Ómicron ( $O$ ) es un parámetro de entrada del OGA. Análogamente a cada arco  $(i, j)$  de  $F_2$  se le agrega un peso de  $\frac{O}{2}$ . Finalmente se toman iterativamente arcos al azar por medio de la ruleta para la formación de un nuevo hijo  $C_1$ . Partiendo de la ciudad  $i$ , considérese  $\mathcal{N}_i$  como el conjunto de ciudades no visitadas, o sea que permiten la creación de un recorrido válido. Por lo tanto, en la ruleta sólo participan los arcos  $(i, j) \forall j \in \mathcal{N}_i$ , con sus respectivos pesos  $w_{ij}$ . Aun cuando el cruzamiento en SGA se realiza simplemente cortando en un punto a los padres e intercambiando las partes, en vez de ir tomando iterativamente arcos con alta probabilidad de uno de los padres como en OGA, la filosofía es la misma.

Con alta probabilidad se seleccionará un arco de uno de los padres para el hijo  $C_1$  (similar a cruzamiento). Pero también es posible incluir nueva información, ya que todos los arcos que permiten la creación de un recorrido válido participan de la ruleta con probabilidad mayor a 0 (similar a mutación). Se usa el valor  $\frac{O}{2}$  porque se tienen dos padres y entonces  $O$  se puede interpretar como la mayor cantidad de peso que puede ser agregada por los padres y  $(O + 1)$  es el mayor peso que puede tener un arco en la ruleta (cuando el arco pertenece a los dos padres). Cuando el arco no pertenece a ninguno de los dos padres, este obtiene el peso mínimo en la ruleta que es 1. Por lo tanto,  $O$  determina el peso relativo entre el cruzamiento y la mutación.

Formalmente, partiendo de la ciudad  $i$ , la probabilidad de que el arco  $(i, j)$  sea elegido para componer al hijo  $C_1$  está dada por la ecuación (6.1).

$$\mathcal{P}_{ij} = \begin{cases} \frac{w_{ij}}{\sum_{\forall h \in \mathcal{N}_i} w_{ih}} & \text{si } j \in \mathcal{N}_i. \\ 0 & \text{en otro caso.} \end{cases} \quad (6.1)$$

Luego, el pseudocódigo del OGA es presentado.

### Pseudocódigo del OGA (versión 1)

```
g = 0 % Inicialización del contador de generaciones
P = Inicializar la población () % Generación al azar de p individuos
REPETIR HASTA condición de fin
  REPETIR p VECES
    {F1, F2} = Reproducción (P) % Selección de 2 padres a través de una ruleta
    C1 = CM({F1, F2}, 1) % Generación de 1 hijo usando {F1, F2}
    P = Actualizar la población (C1) % C1 reemplaza al Px más viejo de P
  g = g + 1 % Incremento del contador de generaciones
```

El pseudocódigo anterior fue diseñado para enfatizar la similitud con el SGA. Sin embargo, el siguiente pseudocódigo es más simple y completamente equivalente.

### Pseudocódigo del OGA (versión 2)

```
u = 0 % Inicialización del contador de iteraciones
P = Inicializar la población () % Generación al azar de p individuos
REPETIR HASTA condición de fin
  {F1, F2} = Reproducción (P) % Selección de 2 padres a través de una ruleta
  C1 = CM({F1, F2}, 1) % Generación de 1 hijo usando {F1, F2}
  P = Actualizar la población (C1) % C1 reemplaza al Px más viejo de P
  u = u + 1 % Incremento del contador de iteraciones
```

### Ejemplo de una Iteración de OGA

Con el objetivo de graficar claramente el procedimiento anterior se presenta a continuación un ejemplo de una iteración del OGA con el TSP con 5 ciudades citado anteriormente. Los parámetros  $O = 4$  y  $p = 4$  son considerados para este caso.

#### ■ Reproducción

Este ejemplo asume una población inicial  $P = \{P_x\}$  compuesta por 4 individuos elegidos al azar con sus respectivos *fitnesses*  $f_x$ . A continuación se presenta esta población inicial.

Primer individuo elegido al azar:  $P_1 = \{c1, c4, c3, c2, c5\}$  con  $f_1 = 10$

Segundo individuo elegido al azar:  $P_2 = \{c1, c3, c2, c5, c4\}$  con  $f_2 = 8$

Tercer individuo elegido al azar:  $P_3 = \{c3, c5, c1, c2, c4\}$  con  $f_3 = 1$

Cuarto individuo elegido al azar:  $P_4 = \{c2, c5, c4, c1, c3\}$  con  $f_4 = 5$

Por medio de una ruleta se eligen al azar dos padres, donde el peso de cada  $P_x$  es proporcional a su *fitness*. Se suponen elegidos los individuos  $P_1$  y  $P_4$ , entonces se puede considerar.

$F_1 = \{c1, c4, c3, c2, c5\} = \{(c1, c4), (c4, c3), (c3, c2), (c2, c5), (c5, c1)\}$

$F_2 = \{c2, c5, c4, c1, c3\} = \{(c2, c5), (c5, c4), (c4, c1), (c1, c3), (c3, c2)\}$

- **Cruzamiento-Mutación. Iteración 1:** Para realizar CM se selecciona una ciudad inicial al azar, y se supone elegida como ciudad inicial a  $c4$ . Entonces  $\mathcal{N}_{c4}$  está compuesto por  $\{c1, c2, c3, c5\}$ , i.e. las ciudades todavía no visitadas. Como el arco  $(c4, c2)$  no se encuentra en ninguno de los padres, tiene un peso de 1 en  $\mathcal{R}$ . Como los arcos  $\{(c4, c3), (c4, c5)\}$  se encuentran una vez en los padres, tienen un peso de  $1 + \frac{O}{2} = 3$  en  $\mathcal{R}$ . Finalmente, como el arco  $(c4, c1)$  se encuentra en ambos padres, tiene un peso de  $1 + O = 5$  en  $\mathcal{R}$ . Se lanza  $\mathcal{R}$  y se supone elegido el arco  $(c4, c3)$ .
- **Cruzamiento-Mutación. Iteración 2:**  $\mathcal{N}_{c3}$  está compuesto por  $\{c1, c2, c5\}$ . Como el arco  $(c3, c5)$  no se encuentra en los padres, tiene un peso de 1 en  $\mathcal{R}$ . Como el arco  $(c3, c1)$  se encuentra una vez en los padres, tiene un peso de  $1 + \frac{O}{2} = 3$  en  $\mathcal{R}$ . Finalmente, como el arco  $(c3, c2)$  se encuentra en ambos padres, tiene un peso de  $1 + O = 5$  en  $\mathcal{R}$ . Se supone elegido el arco  $(c3, c2)$ .
- **Cruzamiento-Mutación. Iteración 3:**  $\mathcal{N}_{c2}$  está compuesto por  $\{c1, c5\}$ . Como el arco  $(c2, c1)$  no se encuentra en los padres, tiene un peso de 1 en  $\mathcal{R}$ . Finalmente, como el arco  $(c2, c5)$  se encuentra en ambos padres, tiene un peso de  $1 + O = 5$  en  $\mathcal{R}$ . Se supone elegido el arco  $(c2, c1)$ .
- **Cruzamiento-Mutación. Iteración 4:**  $\mathcal{N}_{c1}$  está compuesto por  $\{c5\}$ . Como el arco  $(c1, c5)$  se encuentra en uno de los padres, tiene un peso de  $1 + \frac{O}{2} = 3$  en  $\mathcal{R}$ . Obviamente el arco elegido es el único arco representado en  $\mathcal{R}$ , el arco  $(c1, c5)$ .

Por lo tanto el nuevo individuo es  $C_1 = \{c4, c3, c2, c1, c5\} = \{(c4, c3), (c3, c2), (c2, c1), (c1, c5), (c5, c4)\}$ . Es importante hacer notar que  $C_1$  tiene tres arcos de  $F_1$ , i.e.  $\{(c4, c3), (c3, c2), (c1, c5)\}$  y 2 arcos de  $F_2$ , i.e.  $\{(c3, c2), (c1, c5)\}$ . Además  $C_1$  tiene un arco  $\{(c2, c1)\}$  que no se encuentra en ninguno de los padres. Esto permite observar que los objetivos de los operadores, cruzamiento y mutación, del SGA no han sido alterados en el OGA.

- **Actualización de la población:** El nuevo individuo  $C_1$  reemplaza al individuo más viejo  $P_1$  y la nueva población se muestra a continuación.
 
$$P_1 = \{c4, c3, c2, c1, c5\} \text{ con } f_1 = 7$$

$$P_2 = \{c1, c3, c2, c5, c4\} \text{ con } f_2 = 8$$

$$P_3 = \{c3, c5, c1, c2, c4\} \text{ con } f_3 = 1$$

$$P_4 = \{c2, c5, c4, c1, c3\} \text{ con } f_4 = 5$$

Todo este procedimiento se realiza iterativamente hasta que una condición de parada sea satisfecha. Note que OGA es una nueva versión de AG como cualquiera otra de las tantas publicadas.

## 6.2. Nuevas Estrategias para OGA

Nuevas estrategias son ideadas para el OGA, éstas se refieren al cruzamiento, a la actualización de la población y al uso de información heurística. Ya que el aspecto más relevante de estas nuevas estrategias es el cruzamiento de múltiples padres, esta nueva versión de OGA se denominará Algoritmo Genético Ómicron con Múltiples Padres o *Multi-Parent Omicron Genetic Algorithm* (MOGA).

### 6.2.1. Cruzamiento

Otra opción de cruzamiento es considerada en esta sección. Consiste en generar los hijos de más de dos padres. Esta no es una nueva idea y ha sido ya planteada anteriormente [Mühlenbein95]. Más específicamente se concibe la posibilidad de que los  $p$  individuos de  $P$  contribuyan para la creación de cada uno de los hijos. Esto es, lograr que los hijos hereden características de todos los individuos de la población usualmente, en vez de heredar características de dos de ellos únicamente. Obviamente esto elimina la competencia entre los  $P_x$ , debido a que los individuos no participan de una ruleta que les da probabilidades proporcionales a sus *fitnesses* de ser padres. En conclusión, en esta estrategia todos los  $P_x$  son padres, sin competencia alguna. Sin embargo, este inconveniente será resuelto en la estrategia de actualización de la población presentada en la próxima sección, donde la responsabilidad de la competencia es relegada a los hijos. Finalmente, como ahora se tienen  $p$  padres en lugar de 2, a cada arco  $(i, j)$  perteneciente a cada  $P_x$  se le suma  $\frac{Q}{p}$  en lugar de  $\frac{Q}{2}$ , o sea  $w_{ij} = w_{ij} + \frac{Q}{p} \forall (i, j) \in P_x$ . De esta manera, se mantiene la propiedad de que  $(O + 1)$  es el mayor peso que puede tener un arco en  $\mathcal{R}$  (cuando el arco pertenece a los  $p$  padres) y 1 es el menor peso (cuando el arco no pertenece a ningún padre).

### 6.2.2. Actualización de la Población

Con el objetivo regular el ingreso de malos individuos a  $P$  se considera la estrategia de una competencia entre hijos antes de actualizar la población. Esta política se aplica en reemplazo de la estrategia más tradicional donde los padres compiten para reproducirse. Así, se decide generar  $t$  hijos  $\{T_1, \dots, T_t\}$  en una iteración. Solamente el hijo con el mejor *fitness* ( $T_{best} \in \{T_1, \dots, T_t\}$ ) será el elegido para ingresar a la población reemplazando al individuo más viejo. Nótese que se obtiene el mismo efecto (competencia entre individuos) con una estrategia diferente.

### 6.2.3. Información Heurística

Como buenos recorridos en el TSP están formados con gran probabilidad por arcos que unen ciudades que se encuentran a una corta distancia, parece una buena idea darles mayor peso en  $\mathcal{R}$ . Por lo tanto, considerando a la visibilidad  $\eta_{ij} = \frac{1}{d(i,j)}$ , la probabilidad de que el arco  $(i, j)$  sea elegido para componer al hijo  $T_x$  es definida ahora por la ecuación (6.2).

$$\mathcal{P}_{ij} = \begin{cases} \frac{w_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{\forall h \in \mathcal{N}_i} w_{ih}^\alpha \cdot \eta_{ih}^\beta} & \text{si } j \in \mathcal{N}_i. \\ 0 & \text{en otro caso.} \end{cases} \quad (6.2)$$

Donde los parámetros de entrada  $\alpha$  y  $\beta$  se definen como la influencia relativa entre peso de la información genética  $w$  y la visibilidad  $\eta$ .

A continuación, el pseudocódigo del MOGA es presentado.

#### Pseudocódigo del MOGA

```

u = 0                                     % Inicialización del contador de iteraciones
P = Inicializar la población ()           % Generación al azar de p individuos
REPETIR HASTA condición de fin
  T = CM (P, t)                           % Generación de t hijos usando P como padres
  Tbest = mejor elemento de T
  P = Actualizar la población (Tbest)     % Tbest reemplaza al Px más viejo de P
  u = u + 1                               % Incremento del contador de iteraciones

```

### 6.3. Relación entre ACO Y GA

Considerando un P-ACO *FIFO-Queue* y un MOGA con el mismo tamaño de población (i.e.  $q = p$ ), el mismo número de hormigas o hijos (i.e.  $m = t$ ) y los mismos límites  $\tau_{ij}$ ,  $w_{ij}$  (i.e.  $\tau_{max} = O + 1$  y  $\tau_{init} = 1$ ), la cantidad de feromonas que una hormiga deposita y el peso genético de un individuo son también los mismos (i.e.  $(\tau_{max} - \tau_{init})/q = O/p$ ). Prestando atención a la actualización de la matriz de feromonas y a la generación de la ruleta explicados anteriormente, es fácil ver que ambos procedimientos son idénticos. Además, los procedimientos de construcción de soluciones y las probabilidades mostradas en las ecuaciones (6.2) y (3.1) para ambos algoritmos son los mismos.

Aunque la inicialización de la población del P-ACO *FIFO-Queue* no es enteramente al azar como en el MOGA, este aspecto es irrelevante para el funcionamiento básico de estos algoritmos. Consecuentemente, ambos algoritmos son los mismos como es resaltado en los siguientes pseudocódigos, donde el Cruzamiento-Mutación es dividido en dos etapas.

#### Pseudocódigo del MOGA

```

u = 0                                     % Inicialización del contador de iteraciones
P = Inicializar la población ()           % Generación al azar de p individuos
REPETIR HASTA condición de fin
  R = Generar la ruleta (P)               % Generación de la ruleta usando P como padres
  T = CM (R, t)                           % Generación de t hijos a través de la ruleta
  Tbest = mejor elemento de T
  P = Actualizar la población (Tbest)     % Tbest reemplaza al Px más viejo de P
  u = u + 1                               % Incremento del contador de iteraciones

```

### Pseudocódigo de P-ACO FIFO Queue

```
u = 0 % Inicialización del contador de iteraciones
Q = Inicializar la población () % Generación de p individuos
REPETIR HASTA condición de fin
    τ = Actualizar la matriz de feromonas (Q) % Actualización de τ usando Q
    M = Construir soluciones (τ) % Generación de m soluciones usando τ
    Mbest = mejor solución de M
    Q = Actualizar la población (Mbest) % Mbest reemplaza al Qx más viejo
    u = u + 1 % Incremento del contador de iteraciones
```

#### 6.3.1. Nueva Estrategia para el MOGA

En esta sección se propone una nueva estrategia para MOGA. Considérese que  $T_{best}$  reemplace al peor individuo del conjunto  $\{P, T_{best}\}$  en vez de al individuo más viejo de  $P$ , i.e. si el *fitness* de  $T_{best}$  es mayor que el *fitness* de  $P_{worst}$  entonces  $T_{best}$  reemplaza a  $P_{worst}$  en  $P$ , donde  $P_{worst}$  es el peor individuo de  $P$ . Esta nueva estrategia puede verse como un super elitismo, ya que  $P$  estará compuesta por lo mejores individuos encontrados hasta el momento. Por lo tanto a esta nueva versión de GA se denomina Algoritmo Genético Ómicron con Múltiples Padres y Super Elitista o *Super-Elitist Multi-Parent Omicron Genetic Algorithm* (EMOGA). En este punto es fácil concluir que el EMOGA es equivalente a la versión *Quality* de P-ACO.

#### 6.3.2. Nueva Estrategia para el EMOGA

Una característica indeseada en EMOGA es que la población puede ser dominada por un buen individuo. Así, la diversidad genética, una propiedad deseable en los GA, es fuertemente disminuida. Una manera sencilla de evitar este problema es modificar la estrategia de actualización de la población. En esta nueva estrategia  $T_{best}$  reemplaza  $P_{worst}$  solamente si además de ser mejor que  $P_{worst}$ , también es diferente a todos los elementos de la población. Debido a la competencia entre los individuos impuesta por esta estrategia de actualización de la población, la competencia entre hijos es inútil y por lo tanto es eliminada. Además por causa del pequeño dinamismo causado por esta estrategia de actualización de la población, la ruleta es generada cada  $H$  iteraciones sin consecuencias significativas.

Esta es una estrategia que fuerza la diversidad genética de  $P$ , por lo tanto se denominará a esta versión Algoritmo Genético Ómicron con Múltiples Padres, Super Elitista y con Diversidad Genética Forzada o *Diversified Super-Elitist Multi-Parent Omicron Genetic Algorithm* (DEMOGA).

Los pseudocódigos del DEMOGA y del OA son presentados a continuación para mostrar que ambos algoritmos son similares.

### Pseudocódigo del DEMOGA

```

u = 0                                     % Inicialización del contador de iteraciones
P = Inicializar la población ()           % Generación al azar de p individuos
REPETIR HASTA condición de fin
  R = Generar la ruleta (P)               % Generación de la ruleta usando P como padres
  REPETIR H VECES
    T1 = CM (R, 1)                       % Generación de 1 hijo a través de la ruleta
    SI T1 ∉ P
      P = Actualizar la población (T1)    % T1 reemplaza al peor individuo
                                          % de {P, T1}
  u = u + 1                               % Incremento del contador de iteraciones

```

### Pseudocódigo del OA

```

u = 0                                     % Inicialización del contador de iteraciones
Q = Inicializar la población ()           % Generación de q individuos
REPETIR HASTA condición de fin
  τ = Actualizar la matriz de feromonas (Q) % Actualización de τ usando Q
  REPETIR K VECES
    Qnew = Construir una solución (τ)      % Generación de 1 solución usando τ
    SI Qnew ∉ Q
      Q = Actualizar la población (Qnew)  % Qnew reemplaza al peor individuo
                                          % de {Q, Qnew}
  u = u + 1                               % Incremento del contador de iteraciones

```

Para aclarar el origen y las equivalencias de tantas versiones, un cuadro resumido es presentado en la figura 6.1.

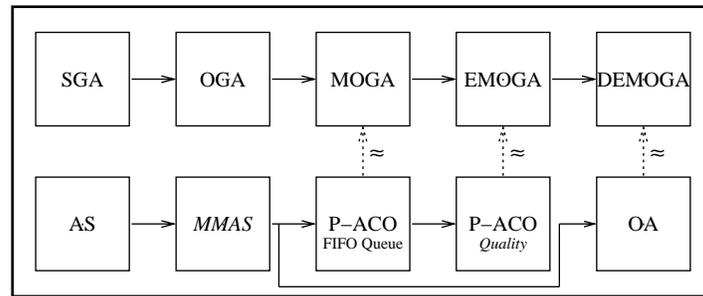


Figura 6.1: Cuadro resumido de las versiones de GA y ACO

### 6.3.3. Razones del Éxito de AG en el TSP

Como OA y DEMOGA son algoritmos idénticos, las mismas razones de éxito en el TSP pueden ser concluidas para ambos algoritmos. Por lo tanto puede ser

inferido que los algoritmos ACO y los Algoritmos Genéticos usan principios similares para lograr el buen rendimiento mostrado en problemas de optimización combinatoria.

OA concentra la búsqueda en la zona central de toda su población, donde están localizadas soluciones prometedoras en estructuras globalmente convexas, como fue explicado en la sección 5.2.4. MOGA y sus versiones posteriores también concentran la búsqueda en la zona central de su población a través del cruzamiento de múltiples padres. Sin embargo OGA, un AG más tradicional, concentra la búsqueda en la zona central de dos individuos de su población a través del cruzamiento de dos padres. Aun así, la búsqueda de OGA es dirigida hacia la zona central de la población porque los padres son iterativamente elegidos al azar. En conclusión el principio es el mismo y explica el importante rol del cruzamiento en estructuras globalmente convexas.

Si un buen individuo  $P_{good}$  domina la población  $P$  en OGA, el cruzamiento pierde su habilidad de buscar en la zona central de dos individuos, ya que en muchos de los casos los padres elegidos son idénticos y consecuentemente los hijos también son idénticos. De esta manera, la mutación concentra la búsqueda en una zona cercana a  $P_{good}$  únicamente, y este algoritmo basado en población se comporta como un algoritmo de búsqueda de un punto, como las heurísticas de Búsqueda local. En conclusión, cuando pocos individuos dominan la población, la diversidad genética es reducida y consecuentemente esto elimina los beneficios del cruzamiento y la búsqueda es dirigida principalmente a través de la mutación.

## Capítulo 7

# Conclusiones y Trabajos Futuros

En la búsqueda de las razones del éxito de Optimización por Colonia de Hormigas o *Ant Colony Optimization* (ACO) en el Problema del Cajero Viajante o *Traveling Salesman Problem* (TSP), el concepto de convexidad global se vuelve trascendente. Por lo tanto en este trabajo se presentan resultados experimentales sistemáticos que soportan completamente la conjetura de un espacio de soluciones globalmente convexo del TSP, como había sido sugerido anteriormente [Boese95].

Como herramienta de estudio se diseñaron versiones de ACO basadas en el *MAX-MIN Ant System* (MMAS), pero conceptualmente más simples, que permitieron una mayor comprensión de sus razones de éxito. Mediante la comprensión de las razones del éxito de ACO, estas versiones fueron evolucionando hasta obtenerse un algoritmo que además demostró ser muy competitivo. Así, este trabajo presenta a Ómicron ACO (OA), un nuevo algoritmo inspirado en uno de mejores algoritmos ACO, el MMAS [Stützle00]. OA demostró en el estudio experimental preliminar presentado en este trabajo que supera en rendimiento al MMAS.

OA concentra la búsqueda de buenas soluciones en una zona central  $\Omega(P)$  de su población  $P$ . En problemas globalmente convexos, buenas soluciones son usualmente encontradas en esta región; por lo tanto, OA concentra su búsqueda en un subespacio prometedor. Cada vez que es encontrada una solución, ésta entra en la población reduciendo la zona de búsqueda iterativamente. Así, este trabajo explica las razones principales del éxito de OA, y de cualquier ACO elitista en general (e.g. MMAS).

OA no usa realimentación positiva y por lo tanto los ACO elitistas no comparten necesariamente todas las mismas razones de éxito con las hormigas reales, aun cuando ACO fue *inspirado* por el comportamiento de las hormigas reales. Esto sugiere no basar el estudio de propiedades útiles de ACO fundamentándose en el comportamiento de la hormigas reales.

Los ACOs basados en Población o *Population-based* ACOs (P-ACO) y entre ellos el OA, debido a su simplicidad conceptual, permiten un análisis más profundo de ACO. Así por ejemplo, por medio de ellos se demostró que P-ACO *FIFO Queue* es un Algoritmo Genético (GA) con múltiples padres, competencia entre hijos y un operador específico de cruzamiento y mutación. De manera que ACO y GA usan los mismos principios para su éxito en el TSP, o en problemas de optimización combinatoria con estructuras globalmente convexas de su espacio de soluciones. Finalmente, a través de esta extensión, puede ser explicada la importancia del cruzamiento y de la diversidad genética, i.e. que dirigen la búsqueda hacia la zona central de una población de individuos, donde se localizan soluciones prometedoras.

Trabajos futuros deberían concentrarse en el estudio de otros problemas de optimización combinatoria para identificar aquellos con estructuras globalmente convexas, donde estos algoritmos mostraron ser muy exitosos. También en la aplicación de OA a otros problemas de optimización combinatoria y en la comparación de su rendimiento con otras metaheurísticas de clase mundial. Además se puede considerar el estudio de una definición formal de convexidad global en la cual podrían analizar grados de convexidad.

# Bibliografía

- [Birattari00] Mauro Birattari, Gianni Di Caro, y Marco Dorigo. For a Formal Foundation of the Ant Programming Approach to Combinatorial Optimization. Part 1: The problem, the representation, and the general solution strategy. Reporte Técnico TR-H-301, ATR-Human Information Processing Labs, Kyoto, Japan, 2000.
- [Boese94] Kenneth Boese, Andrew Kahng, y Sudhakar Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16:101–113, 1994.
- [Boese95] Kenneth Boese. Cost Versus Distance in the Traveling Salesman Problem. Reporte Técnico 950018, University of California, Los Angeles, Computer Science Department, mayo 19, 1995.
- [Dorigo91] Marco Dorigo, Vittorio Maniezzo, y Alberto Colomi. The Ant System: An Autocatalytic Optimizing Process. Reporte Técnico 91-016, Politecnico di Milano, Italy., 1991.
- [Dorigo99] Marco Dorigo y Gianni Di Caro. The Ant Colony Optimization Meta-Heuristic. En David Corne, Marco Dorigo, y Fred Glover, editores, *New Ideas in Optimization*, páginas 11–32. McGraw-Hill, London, 1999.
- [Dorigo01] Marco Dorigo y Thomas Stützle. An Experimental Study of the Simple Ant Colony Optimization Algorithm. En *2001 WSES International Conference on Evolutionary Computation (EC'01)*. WSES-Press International, 2001.
- [Dorigo03] Marco Dorigo y Thomas Stützle. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. En F. Glover y G. Kochenberger, editores, *Handbook of Metaheuristics*, tomo 57 de *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, 2003.
- [Gardel04] Pedro Gardel, Osvaldo Gómez, y Benjamín Barán. Estudio del Espacio de Soluciones del Problema del Cajero Viajante. En

*Proceedings de la Conferencia Latinoamericana en Informática (CLEI'04)*. septiembre 2004.

- [Gómez04a] Osvaldo Gómez y Benjamín Barán. Arguments for ACO's Success. En E. Cantú-Paz, J. A. Foster, K. Deb, D. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. Dowsland, N. Jonoska, y J. Miller, editores, *Genetic and Evolutionary Computation – GECCO-2004*, tomo 3102 de *LNCS*. Springer-Verlag, Seattle, junio 2004.
- [Gómez04b] Osvaldo Gómez y Benjamín Barán. Omicron ACO. En *Proceedings de la Conferencia Latinoamericana en Informática (CLEI'04)*. septiembre 2004.
- [Gómez04c] Osvaldo Gómez y Benjamín Barán. Reasons of ACO's Success in TSP. En M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, y T. Stützle, editores, *Proceedings of ANTS 2004 - Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence*, tomo 3172 de *LNCS*. Springer-Verlag, Brussels, septiembre 2004.
- [Gómez04d] Osvaldo Gómez y Benjamín Barán. Relationship between Genetic Algorithms and Ant Colony Optimization. En *Proceedings de la Conferencia Latinoamericana en Informática (CLEI'04)*. septiembre 2004.
- [Goldberg89] David Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [Guntsch02a] Michael Guntsch y Martin Middendorf. A Population Based Approach for ACO. En Stefano Cagnoni, Jens Gottlieb, Emma Hart, Martin Middendorf, y Günther Raidl, editores, *Applications of Evolutionary Computing, Proceedings of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim*, tomo 2279, páginas 71–80. Springer-Verlag, Kinsale, Ireland, 3-4 2002.
- [Guntsch02b] Michael Guntsch y Martin Middendorf. Applying Population Based ACO to Dynamic Optimization Problems. En *Ant Algorithms, Proceedings of Third International Workshop ANTS 2002*, tomo 2463 de *LNCS*, páginas 111–122. 2002.
- [Gutjahr00] Walter Gutjahr. A graph-based Ant System and its convergence. *Future Generation Computer Systems*, 16(8):873–888, junio 2000.
- [Gutjahr02] Walter Gutjahr. ACO Algorithms with Guaranteed Convergence to the Optimal Solution. *Information Processing Letters*, 82(3):145–153, mayo 2002.

- [Holland75] John Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- [Hu89] T. C. Hu, Victor Klee, y David Larman. Optimization of globally convex functions. *SIAM Journal on Control and Optimization*, 27(5):1026–1047, septiembre 1989.
- [Meuleau02] Nicolas Meuleau y Marco Dorigo. Ant colony optimization and stochastic gradient descent. *Artificial Life*, 8(2):103–121, 2002.
- [Mühlenbein95] Heinz Mühlenbein y Hans-Michael Voigt. Gene Pool Recombination in Genetic Algorithms. En Ibrahim H. Osman y James P. Kelly, editores, *Proceedings of the Meta-heuristics Conference*, páginas 53–62. Kluwer Academic Publishers, Norwell, USA, 1995.
- [Reinelt94] Gerhard Reinelt. The Traveling Salesman: Computational Solutions for TSP Applications. *Lecture Notes in Computer Science*, 840:VIII + 223, 1994.
- [Stützle00] Thomas Stützle y Holger Hoos. *MAX-MIN* Ant System. *Future Generation Computer Systems*, 16(8):889–914, junio 2000.
- [Stützle02] Thomas Stützle y Marco Dorigo. A Short Convergence Proof for a Class of Ant Colony Optimization Algorithms. *IEEE Transactions on Evolutionary Computation*, 6:358–365, agosto 2002.