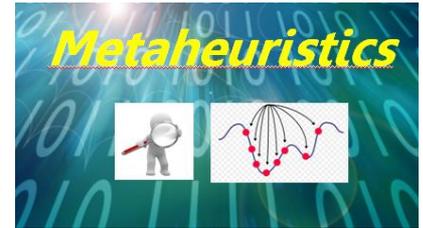


METAHEURÍSTICAS

2020 - 2021



- Tema 1. Introducción a las Metaheurísticas
- Tema 2. Modelos de Búsqueda: Entornos y Trayectorias vs Poblaciones
- Tema 3. Metaheurísticas Basadas en Poblaciones
- Tema 4: Algoritmos Meméticos
- Tema 5. Metaheurísticas Basadas en Trayectorias
- Tema 6. Metaheurísticas Basadas en Adaptación Social
- Tema 7. Aspectos Avanzados en Metaheurísticas
- Tema 8. Metaheurísticas Paralelas

METAHEURÍSTICAS

TEMA 6. METAHEURÍSTICAS BASADAS EN ADAPTACIÓN SOCIAL

- 1. INTRODUCCIÓN A LOS MODELOS BASADOS EN ADAPTACIÓN SOCIAL: SWARM INTELLIGENCE**
- 2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS**
- 3. OPTIMIZACIÓN BASADA EN NUBES DE PARTÍCULOS (PARTICLE SWARM OPTIMIZATION)**
- 4. CONCLUSIONES**

Bibliografía

E. Bonabeau, M. Dorigo, G. Theraulaz. *Swarm Intelligence. From Nature to Artificial Systems*. Oxford University Press, 1999.

Kennedy, J., Eberhart, R. C., and Shi, Y.. *Swarm intelligence*. San Francisco: Morgan Kaufmann Publishers, 2001.

METAHEURÍSTICAS

TEMA 6. METAHEURÍSTICAS BASADAS EN ADAPTACIÓN SOCIAL

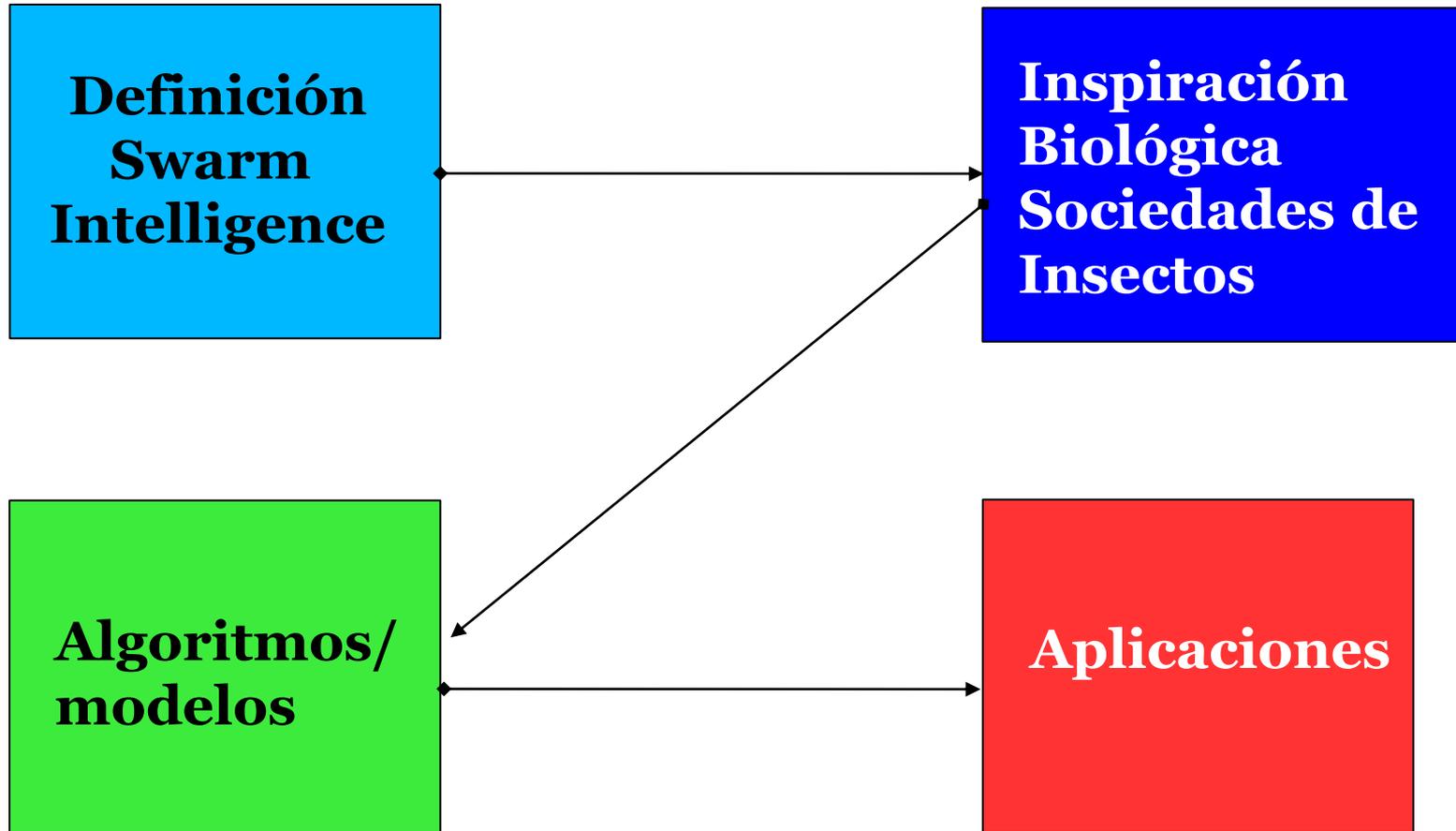
1. **INTRODUCCIÓN A LOS MODELOS BASADOS EN ADAPTACION SOCIAL: SWARM INTELLIGENCE**
2. **ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS**
3. **OPTIMIZACIÓN BASADA EN NUBES DE PARTÍCULOS (PARTICLE SWARM OPTIMIZATION)**
4. **CONCLUSIONES**

Bibliografía

E. Bonabeau, M. Dorigo, G. Theraulaz. *Swarm Intelligence. From Nature to Artificial Systems*. Oxford University Press, 1999.

Kennedy, J., Eberhart, R. C., and Shi, Y.. *Swarm intelligence*. San Francisco: Morgan Kaufmann Publishers, 2001.

INTRODUCCIÓN A LOS MODELOS BASADOS EN ADAPTACIÓN SOCIAL: SWARM INTELLIGENCE



Introducción: Swarm Intelligence

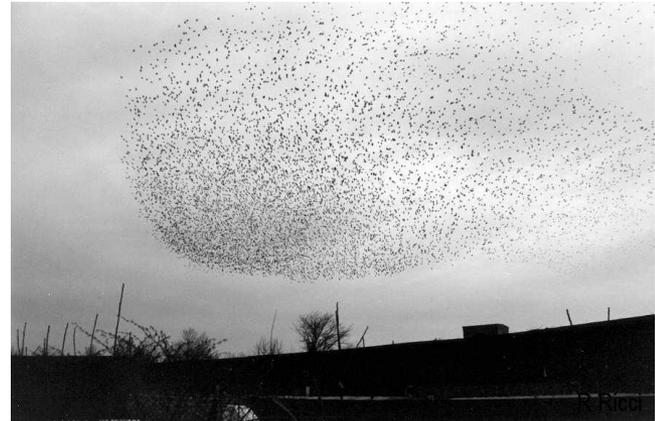
“La inteligencia colectiva emergente de un grupo de agentes simples”

“The emergent collective intelligence of groups of simple agents”

“Algoritmos o mecanismos distribuidos de resolución de problemas inspirados en el comportamiento colectivo de colonias de insectos sociales u otras sociedades de animales”.

(Bonabeau, Dorigo, Theraulaz, 1999)

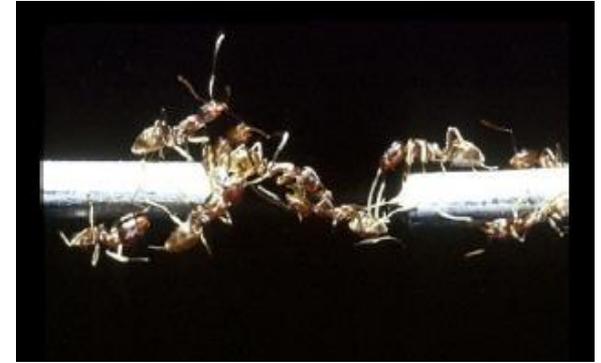
**E. Bonabeau, M. Dorigo, G. Theraulaz
Swarm Intelligence. From Nature to
Artificial Systems.
Oxford University Press, 1999.**



Introduction: Swarm Intelligence

Inspiración Biológica

“Dumb parts, properly connected into a swarm, yield to smart results”



**sociedades de insectos
(bees, wasps, ants, termites)
(abejas, avispas, hormigas, termitas)**



**flocks of birds
(bandadas de aves)**

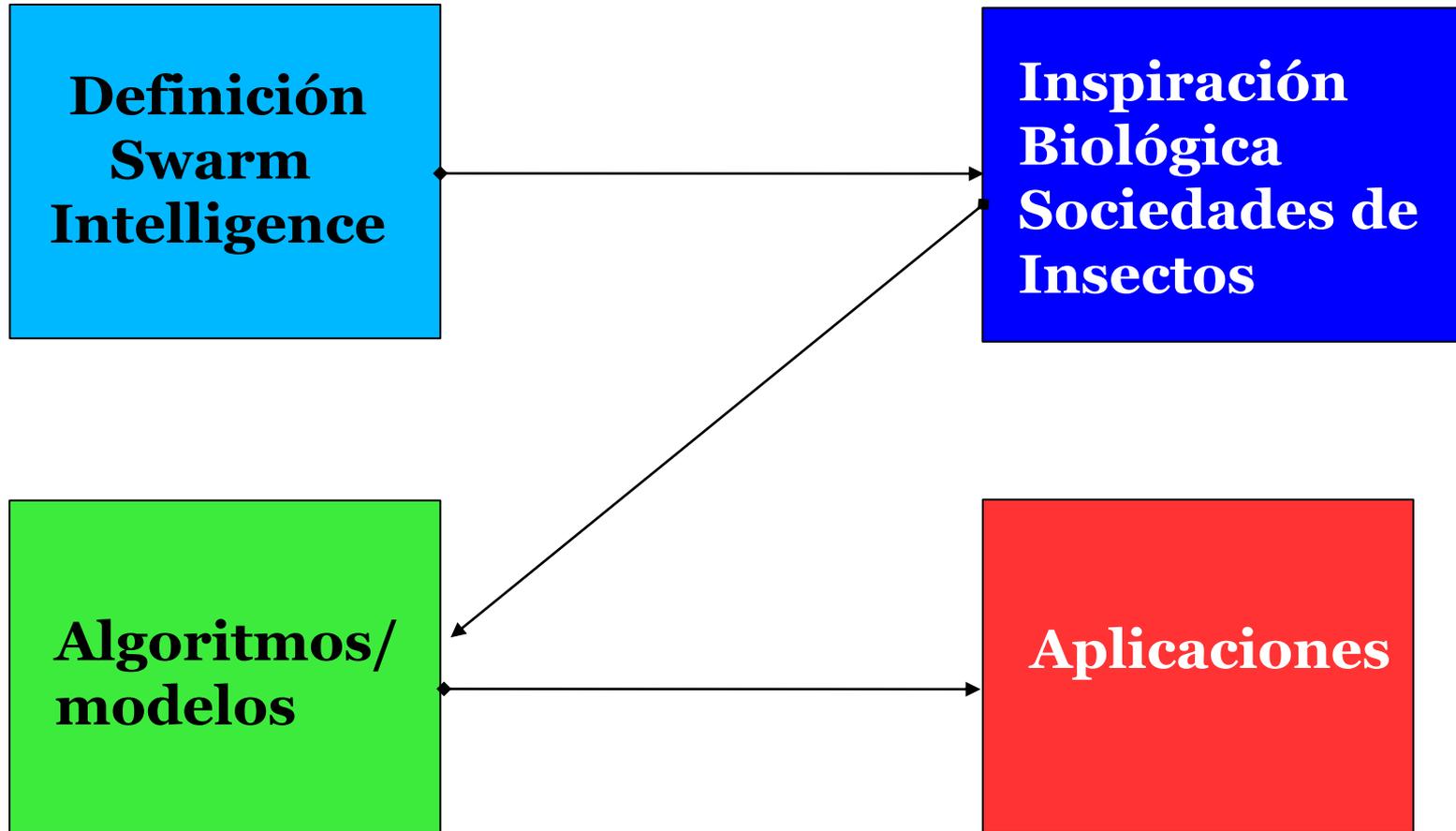


**schools of fish
(bancos de peces)**



**herds of mammals
(manadas de mamíferos)**

INTRODUCCIÓN A LOS MODELOS BASADOS EN ADAPTACIÓN SOCIAL: SWARM INTELLIGENCE



De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Insectos sociales: La complejidad y la sofisticación de auto-organización se lleva a cabo sin un líder claro

Los modelos de las colonias/sociedades de insectos por medio de sistemas auto-organizativos puede ayudar al diseño de sistemas artificiales distribuidos para la resolución de problemas.

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Comportamiento emergente

- Las colonias de insectos llevan a cabo actuaciones de nivel complejo de forma inteligente, flexible y fiable, actuaciones que no serían factibles si tuviesen que ser realizadas por un insecto de forma individual (éstos son no inteligentes, no fiables, simples).
- **Los insectos siguen reglas simples, y utilizan comunicación local simple**
- La estructura global (nido) emerge desde las acciones de los insectos (las cuales son no fiables atendidas individualmente)

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Abejas



- **Cooperación de la colmena**
- **Regulan la temperatura de la colmena**
- **Eficiencia vía especialización: división de la labor en la colonia**
- **Comunicación: Las fuentes de comida son explotadas de acuerdo a la calidad y distancia desde la colmena**

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

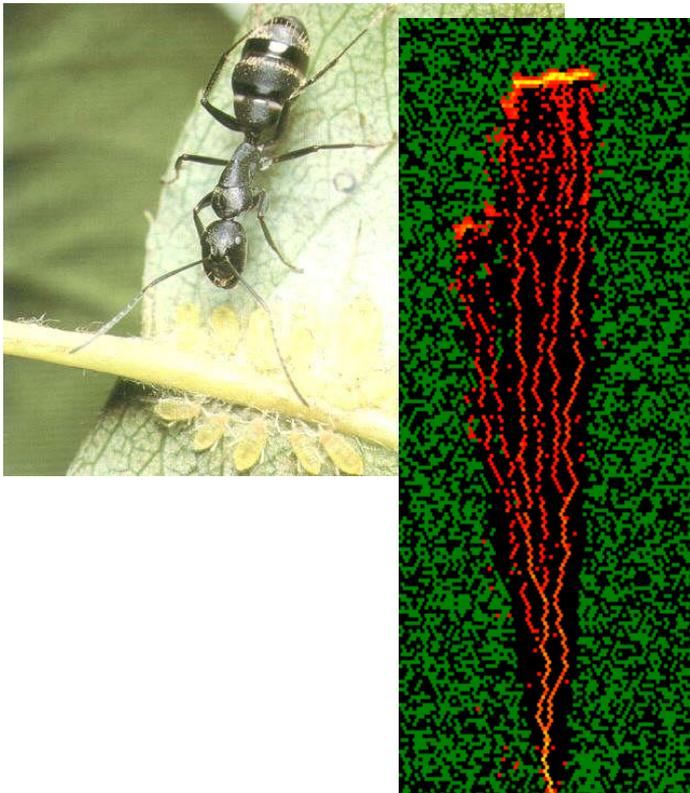
Termitas



- Nido con forma de cono con paredes externas y conductos de ventilación
- Camaras de camadas en el centro de la colmena
- Rejillas del ventilación en espirales
- Columnas de soporte

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Hormigas



- Organizan autopistas “hacia y desde” la comida por medio de rastros de feromona (pheromone)

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Sociedades de Insectos/Tipos de interacción entre Insectos Sociales

- Sistemas de toma de decisión colectiva
- Comunicación directa/interacción directa
 - Comida/intercambio de líquidos, contacto visual, contacto químico (pheromones)
- Comunicación indirecta/interacción indirecta (Stigmergy)
 - El comportamiento individual modifica el entorno, el cual a su vez modifica el comportamiento de otros individuos

⇒ **sociedades auto-organizadas**

De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Características de un Enjambre

- Compuesto de agentes *simples* y auto-organizados (Self-Organized)
- Descentralizado
 - No hay un único supervisor
- No hay un plan global (emergente)
- Robusto
 - Las actuaciones se completan aunque un individuo falle
- Flexible
 - Puede responder a cambios externos
 - Percepción del entorno (sentidos)
 - No existe un modelo explícito de entorno/abilidad para cambiarlo

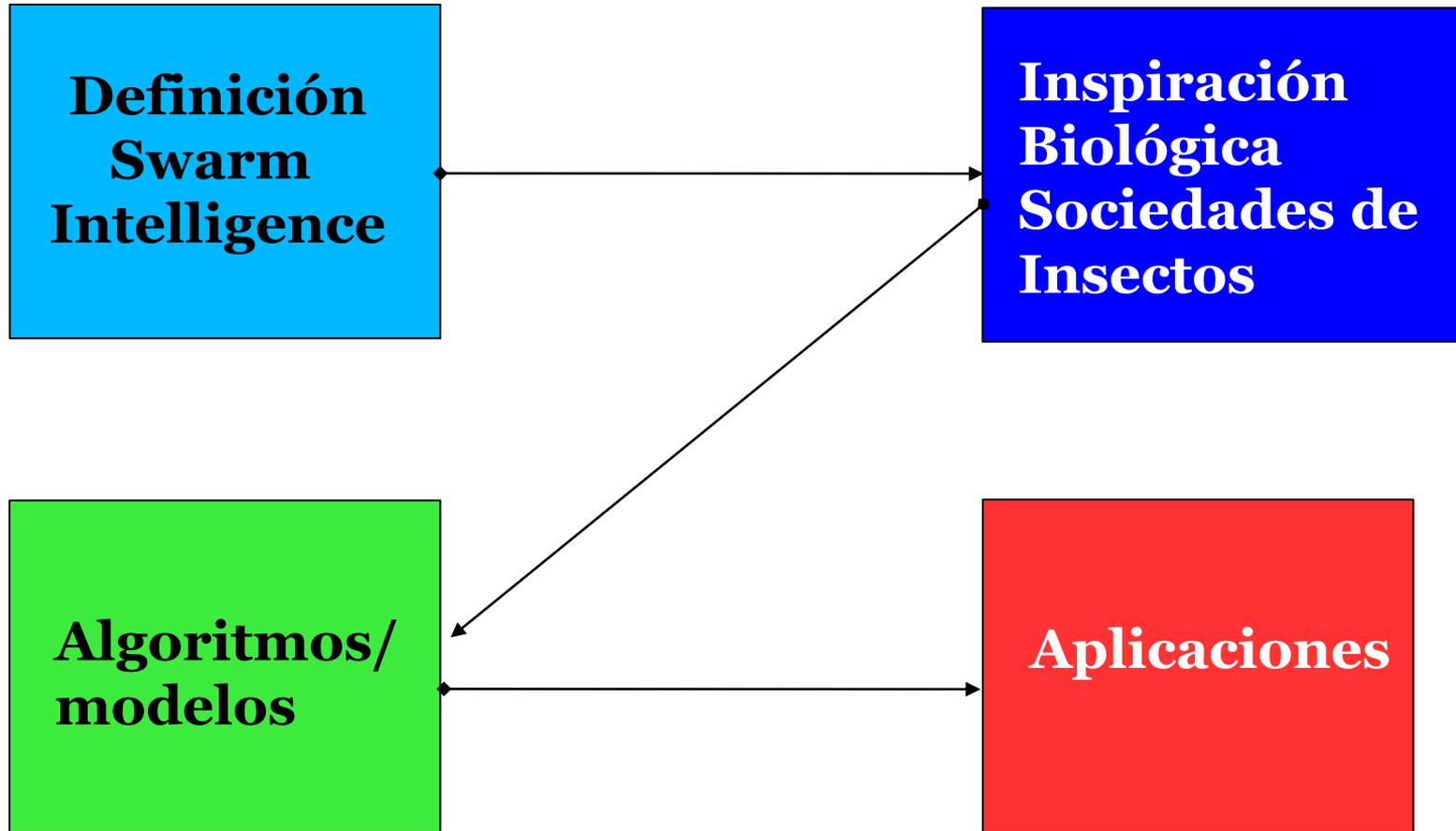
De las Sociedades de Insectos a los Sistemas Inteligentes de Enjambre

Resumen

- La complejidad y sofisticación de la auto-organización se lleva a cabo sin un líder/jefe de la sociedad
- Lo que podemos aprender de los insectos sociales lo podemos aplicar al campo del diseño de Sistemas Inteligentes
- La modelización de los insectos sociales por medio de la auto-organización puede ser de ayuda para el diseño de modelos artificiales distribuidos de resolución de problemas. Esto es conocido como:

Swarm Intelligent Systems.

INTRODUCCIÓN A LOS MODELOS BASADOS EN ADAPTACIÓN SOCIAL: SWARM INTELLIGENCE



Swarm Intelligence: Técnicas

Swarm Inspired Methods

■ Particle swarm optimization – PSO

Optimización basada en nubes de partículas

- Conjunto de técnicas inspiradas en el comportamiento de las bandadas de aves o bancos de peces

■ Ant colony optimization – ACO

Optimización basada en colonias de hormigas

- Conjunto de técnicas inspiradas por las actividades de una colonia de hormigas

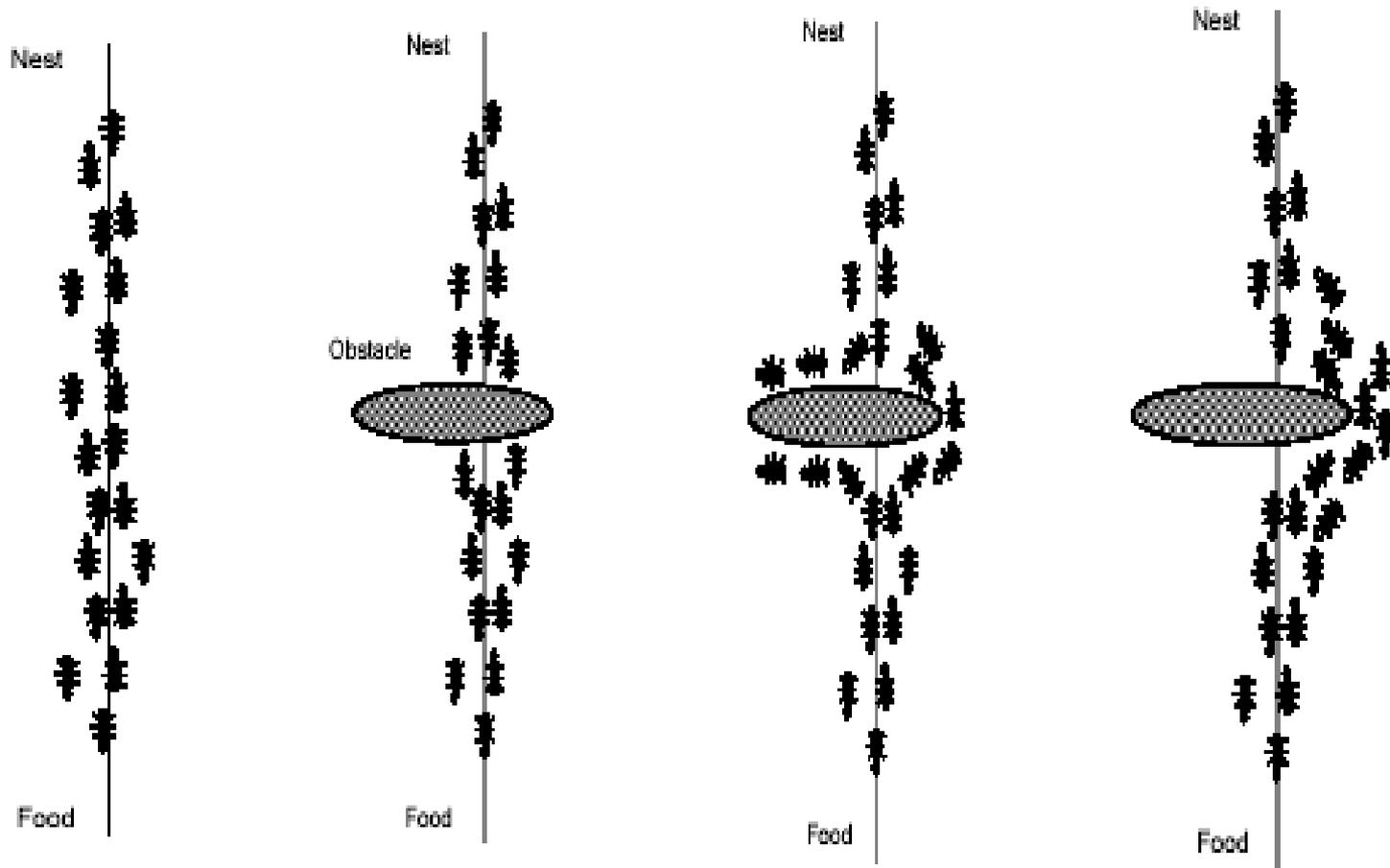
■ Bee colony optimization – BCO

Optimización basada en colonias de abejas

- Conjunto de técnicas inspiradas por las actividades de una colonia de abejas

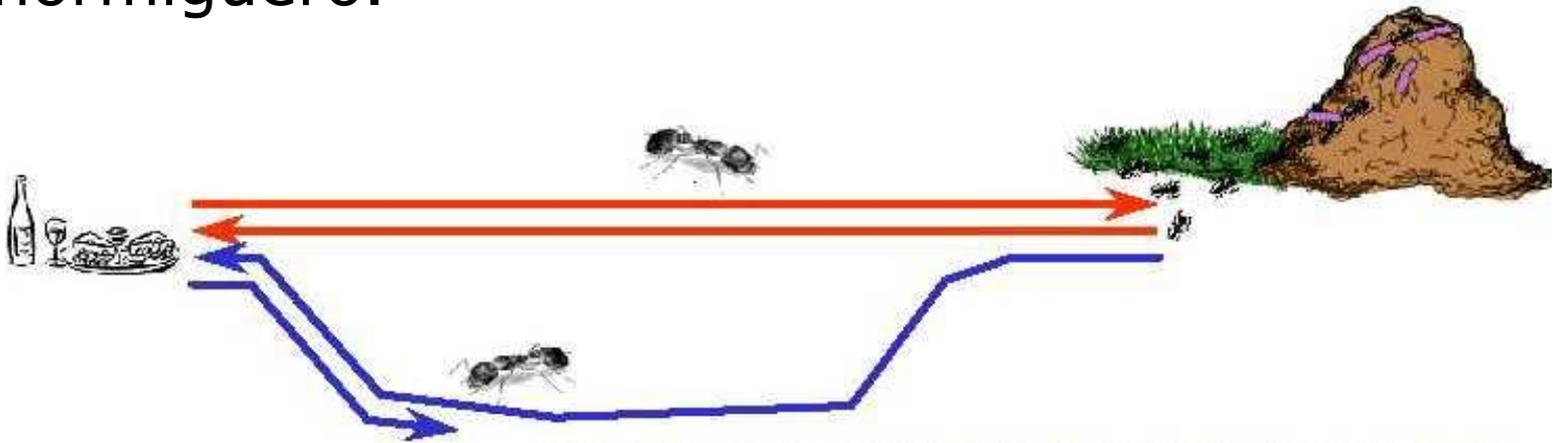
Ant Colony Optimization

Comportamiento natural

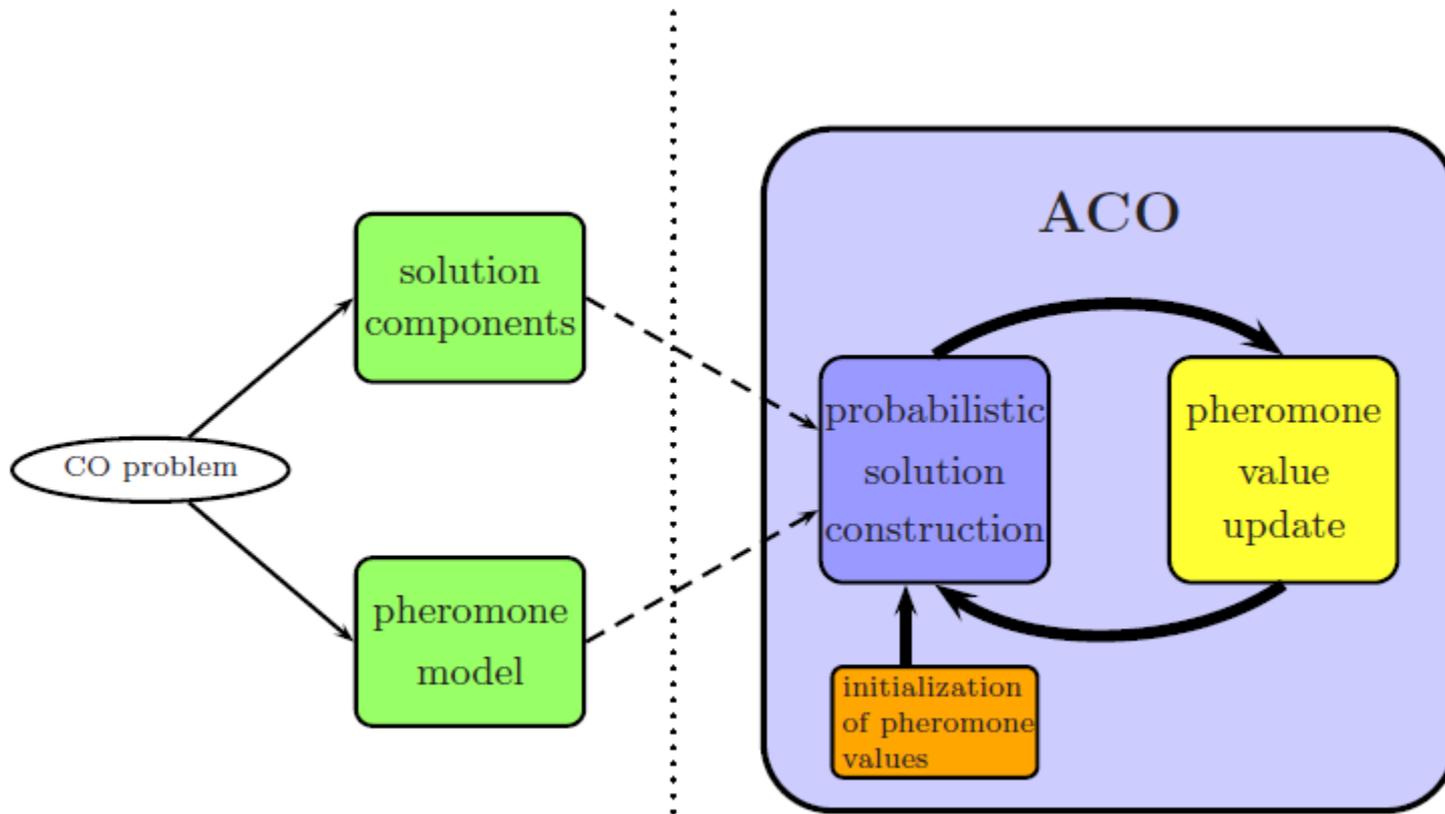


Ant Colony Optimization

- La analogía más cercana a ACO son los problemas de rutas en grafos
- Mientras las hormigas buscan comida, depositan rastros de feromona que atraen a otras hormigas. Desarrollan caminos mínimos entre la comida y el hormiguero.



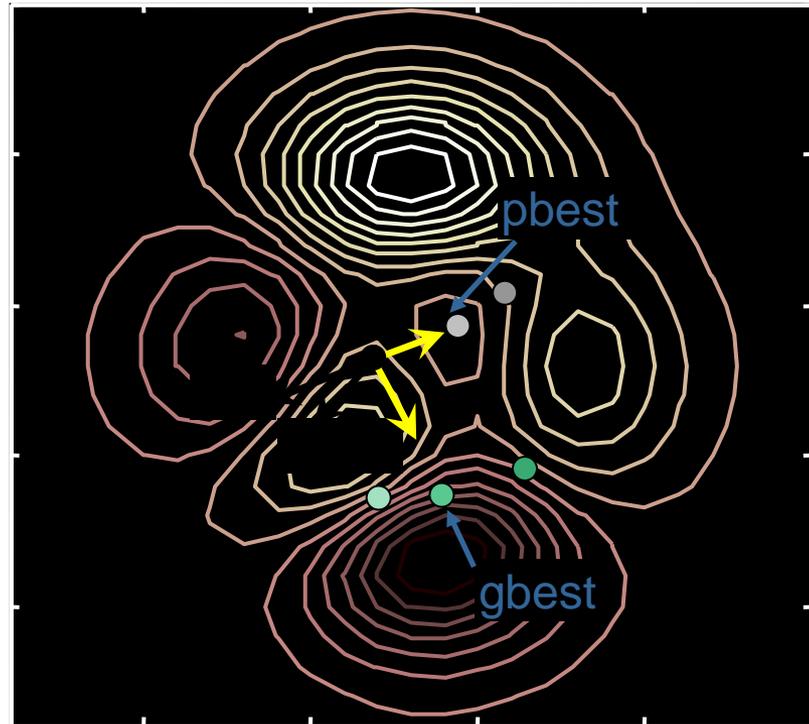
Ant Colony Optimization



Particle Swarm Optimization

Particle Swarm Optimization (PSO) aplica conceptos de interacción social a la resolución de problemas de búsqueda/optimización.

En PSO, un enjambre de n individuos se comunica directa o indirectamente con otros vía las direcciones de búsqueda.

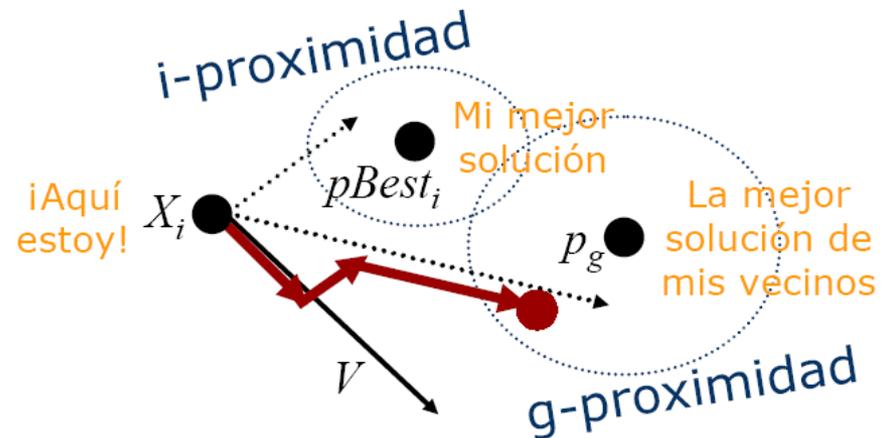


Particle Swarm Optimization

Las Partículas pueden ser simples agentes que vuelan a través del espacio de búsqueda y almacenan (y posiblemente comunican) la mejor solución que han descubierto.

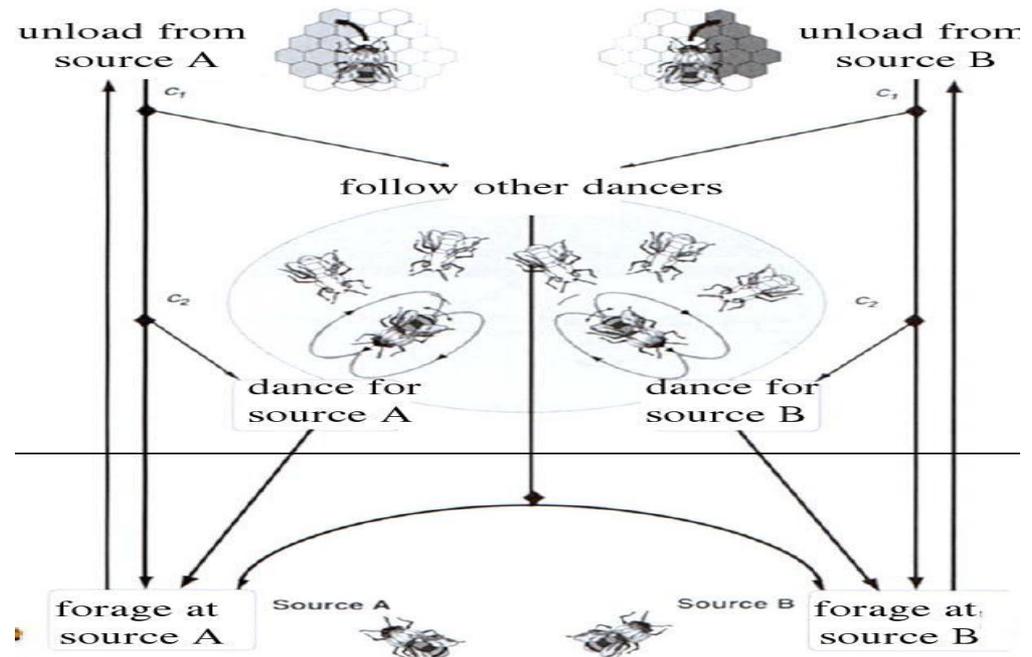
En PSO, las partículas nunca mueren.

La pregunta es, "¿Cómo se mueve una partícula desde una localización a otro del espacio de búsqueda?"



Bee Colony Optimization

Los algoritmos de optimización basados en abejas están basados en la recogida de comida (nectar) de las abejas.



Swarm Intelligence: Técnicas

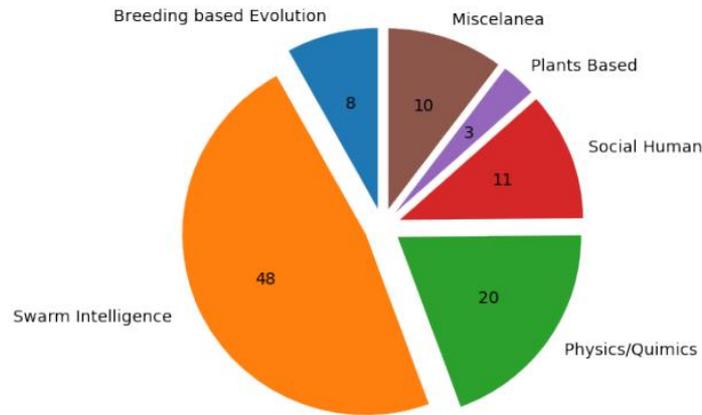


Figure 3: Ratio of reviewed algorithms by its category (first taxonomy).

[Comprehensive Taxonomies of Nature-and Bio-inspired Optimization: Inspiration versus Algorithmic Behavior, Critical Analysis and Recommendations](#)
Authors

Daniel Molina, Javier Poyatos, Javier Del Ser, Salvador García, Amir Hussain, Francisco Herrera
 Publication date 2020/2/19, arXiv preprint arXiv:2002.08136 (Versión 1)

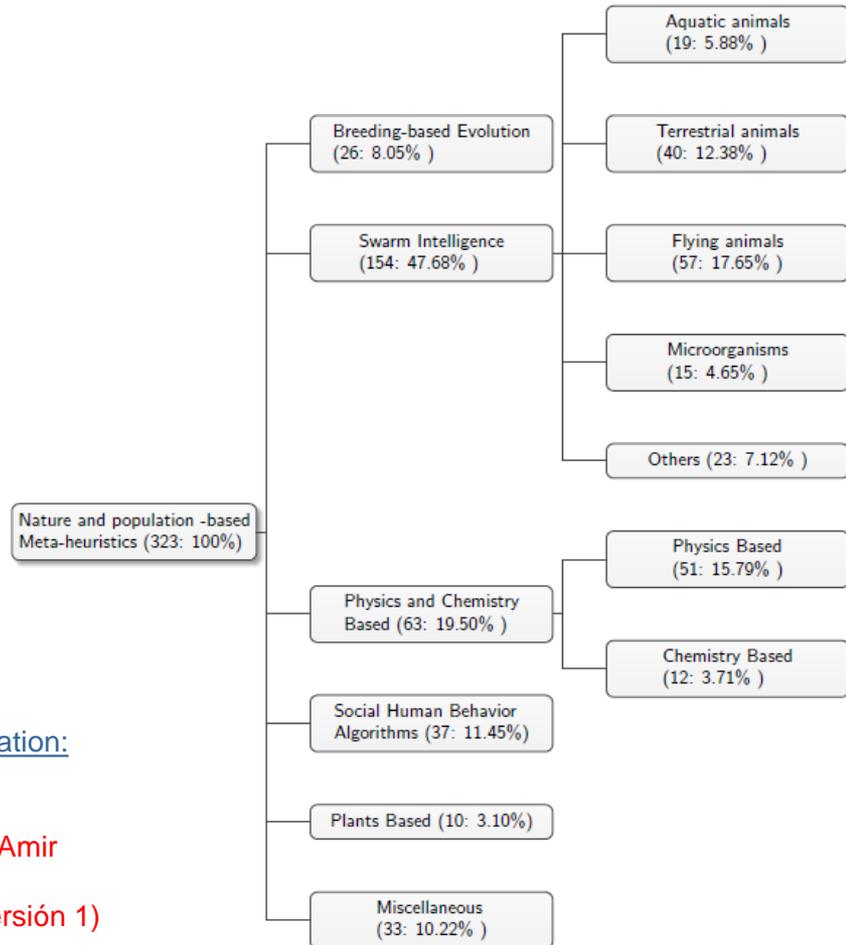


Figure 2: Classification of the reviewed papers using the *inspiration source* based taxonomy.

Swarm Intelligence: Técnicas

Table 2: Nature- and bio-inspired meta-heuristics within the *Swarm Intelligence* category (I).

Swarm Intelligence (I)					
Algorithm Name	Acronym	Subcategory	Type	Year	Reference
Artificial Algae Algorithm	AAA	Micro	Movement	2015	[101]
Artificial Beehive Algorithm	ABA	Flying	Foraging	2009	[102]
Artificial Bee Colony	ABC	Flying	Foraging	2007	[8]
Animal Behavior Hunting	ABH	Other	Foraging	2014	[103]
African Buffalo Optimization	ABO	Terrestrial	Foraging	2016	[104]
Andean Condor Algorithm	ACA	Flying	Foraging	2019	[105]
Ant Colony Optimization	ACO	Terrestrial	Foraging	1996	[6]
Ant Lion Optimizer	ALO	Terrestrial	Foraging	2015	[106]
Artificial Searching Swarm Algorithm	ASSA	Other	Movement	2009	[107]
Artificial Tribe Algorithm	ATA	Other	Movement	2009	[108]
African Wild Dog Algorithm	AWDA	Terrestrial	Foraging	2013	[109]
Bald Eagle Search	BES	Flying	Foraging	2019	[110]
Bees Algorithm	BA	Flying	Foraging	2006	[111]
Bumblebees	BB	Flying	Foraging	2009	[112]
Bison Behavior Algorithm	BBA	Terrestrial	Movement	2019	[113]
Bee Colony-Inspired Algorithm	BCIA	Flying	Foraging	2009	[114]
Bee Colony Optimization	BCO	Flying	Foraging	2005	[115]
Bacterial Colony Optimization	BCO.1	Micro	Foraging	2012	[116]
Bacterial Chemotaxis Optimization	BCO.2	Micro	Foraging	2002	[117]
Biomimicry Of Social Foraging	BFOA	Micro	Foraging	2002	[14]
Distributed Optimization					
Bacterial Foraging Optimization	BFOA.1	Micro	Foraging	2009	[56]
Bacterial-GA Foraging	BGAF	Micro	Foraging	2007	[118]
BeeHive Algorithm	BHA	Flying	Foraging	2004	[119]
Bees Life Algorithm	BLA	Flying	Foraging	2018	[120]
Bat Intelligence	BI	Flying	Foraging	2012	[121]
Bat Inspired Algorithm	BIA	Flying	Foraging	2010	[3]
Biology Migration Algorithm	BMA	Other	Movement	2019	[122]
Blind, Naked Mole-Rats Algorithm	BNMR	Terrestrial	Foraging	2013	[123]
Butterfly Optimizer	BO	Flying	Movement	2015	[124]
Bee System	BS	Flying	Foraging	1997	[125]
Bee System	BS.1	Flying	Foraging	2002	[126]
Bird Swarm Algorithm	BSA	Flying	Movement	2016	[127]
Bee Swarm Optimization	BSO	Flying	Foraging	2010	[128]
Bioluminescent Swarm Optimization Algorithm	BSO.1	Flying	Foraging	2011	[129]
Bees Swarm Optimization Algorithm	BSOA	Flying	Foraging	2005	[130]
Binary Whale Optimization Algorithm	BWOA	Aquatic	Foraging	2019	[131]

Table 3: Nature- and bio-inspired meta-heuristics within the *Swarm Intelligence* category (II).

Swarm Intelligence (II)					
Algorithm Name	Acronym	Subcategory	Type	Year	Reference
Collective Animal Behavior	CAB	Other	Foraging	2012	[132]
Cheetah Based Algorithm	CBA	Terrestrial	Movement	2018	[133]
Catfish Optimization Algorithm	CAO	Aquatic	Movement	2011	[134]
Cricket Behavior-Based Algorithm	CBBE	Terrestrial	Movement	2016	[135]
Cultural Coyote Optimization Algorithm	CCOA	Terrestrial	Movement	2019	[136]
Chaotic Dragonfly Algorithm	CDA	Flying	Foraging	2018	[137]
Cuttlefish Algorithm	CFA	Aquatic	Movement	2013	[138]
Consultant Guide Search	CGS	Other	Movement	2010	[139]
Cuckoo Optimization Algorithm	COA	Flying	Foraging	2011	[140]
Camel Travelling Behavior	COA.1	Terrestrial	Movement	2016	[141]
Coyote Optimization Algorithm	COA.2	Terrestrial	Movement	2018	[142]
Cuckoo Search	CS	Flying	Foraging	2009	[143]
Crow Search Algorithm	CSA	Flying	Movement	2016	[144]
Cat Swarm Optimization	CSO	Terrestrial	Movement	2006	[145]
Chicken Swarm Optimization	CSO.1	Terrestrial	Movement	2014	[146]
Dragonfly Algorithm	DA	Flying	Foraging	2016	[9]
Dolphin Echolocation	DE.1	Aquatic	Foraging	2013	[147]
Dolphin Partner Optimization	DPO	Aquatic	Movement	2009	[148]
Elephant Herding Optimization	EHO	Terrestrial	Movement	2016	[149]
Eagle Strategy	ES.1	Flying	Foraging	2010	[150]
Elephant Search Algorithm	ESA	Terrestrial	Foraging	2015	[151]
Egyptian Vulture Optimization Algorithm	EV	Flying	Foraging	2013	[152]
Firefly Algorithm	FA	Flying	Foraging	2009	[4]
Flocking Base Algorithms	FBA	Flying	Movement	2006	[153]
Fast Bacterial Swarming Algorithm	FBSA	Micro	Foraging	2008	[154]
Frog Call Inspired Algorithm	FCA	Terrestrial	Movement	2009	[155]
Flock by Leader	FL	Flying	Movement	2012	[156]
Fruit Fly Optimization Algorithm	FOA	Flying	Foraging	2012	[157]
Fish Swarm Algorithm	FSA	Aquatic	Foraging	2011	[158]
Fish School Search	FSS	Aquatic	Foraging	2008	[159]
Group Escape Behavior	GEB	Aquatic	Movement	2011	[160]
Good Lattice Swarm Optimization	GLSO	Other	Movement	2007	[161]
Grasshopper Optimisation Algorithm	GOA	Terrestrial	Foraging	2017	[5]
Glowworm Swarm Optimization	GSO	Micro	Movement	2013	[20]
Group Search Optimizer	GSO.1	Other	Movement	2009	[162]
Goose Team Optimization	GTO	Flying	Movement	2008	[163]
Grey Wolf Optimizer	GWO	Terrestrial	Foraging	2014	[164]

Swarm Intelligence: Técnicas

Table 4: Nature- and bio-inspired meta-heuristics within the *Swarm Intelligence* category (III).

Swarm Intelligence (III)

Algorithm Name	Acronym	Subcategory	Type	Year	Reference
Harry's Hawk Optimization Algorithm	HHO	Flying	Foraging	2019	[74]
Hoopoe Heuristic Optimization	HHO.1	Flying	Foraging	2012	[165]
Hunting Search	HuS	Other	Foraging	2010	[166]
Honeybee Social Foraging	HSF	Flying	Foraging	2007	[167]
Hierarchical Swarm Model	HSM	Other	Movement	2010	[168]
Hypercube Natural Aggregation Algorithm	HYNAA	Other	Movement	2019	[169]
Improved Raven Roosting Algorithm	IRRO	Flying	Movement	2018	[170]
Invasive Tumor Optimization Algorithm	ITGO	Micro	Movement	2015	[171]
Jaguar Algorithm	JA	Terrestrial	Foraging	2015	[172]
Krill Herd	KH	Aquatic	Foraging	2012	[13]
Killer Whale Algorithm	KWA	Aquatic	Foraging	2017	[173]
Lion Algorithm	LA	Terrestrial	Foraging	2012	[174]
Seven-Spot Labybird Optimization	LBO	Flying	Foraging	2013	[175]
Laying Chicken Algorithm	LCA	Terrestrial	Movement	2017	[176]
Lion Optimization Algorithm	LOA	Terrestrial	Foraging	2016	[177]
Locust Swarms Optimization	LSO	Aquatic	Foraging	2009	[178]
Magnetotactic Bacteria Optimization Algorithm	MBO	Micro	Movement	2013	[179]
Monarch Butterfly Optimization	MBO.1	Flying	Movement	2017	[180]
Migrating Birds Optimization	MBO.2	Flying	Movement	2012	[181]
Mouth Breeding Fish Algorithm	MBF	Aquatic	Foraging	2018	[182]
Modified Cuckoo Search	MCS	Flying	Foraging	2009	[183]
Modified Cockroach Swarm Optimization	MCSO	Terrestrial	Foraging	2011	[184]
Moth Flame Optimization Algorithm	MFO	Flying	Movement	2015	[185]
Mosquito Flying Optimization	MFO.1	Flying	Foraging	2016	[186]
Meerkats Inspired Algorithm	MIA	Terrestrial	Movement	2018	[187]
Mox Optimization Algorithm	MOX	Flying	Movement	2011	[188]
Monkey Search	MS	Terrestrial	Foraging	2007	[189]
Natural Aggregation Algorithm	NAA	Other	Movement	2016	[190]
Naked Mole Rat	NMR	Terrestrial	Movement	2019	[191]
Nomadic People Optimizer	NPO	Other	Movement	2019	[192]
OptBees	OB	Flying	Foraging	2012	[193]
Optimal Foraging Algorithm	OFA	Other	Foraging	2017	[194]
Pity Beetle Algorithm	PBA	Terrestrial	Foraging	2018	[195]
Pigeon Inspired Optimization	PIO	Flying	Movement	2014	[196]
Population Migration Algorithm	PMA	Other	Movement	2009	[197]
Prey Predator Algorithm	PPA	Other	Foraging	2015	[198]
Particle Swarm Optimization	PSO	Flying	Movement	1995	[2]
Penguins Search Optimization Algorithm	PSOA	Aquatic	Foraging	2013	[199]
Regular Butterfly Optimization Algorithm	RBOA	Flying	Foraging	2019	[200]

Table 5: Nature- and bio-inspired meta-heuristics within the *Swarm Intelligence* category (IV).

Swarm Intelligence (IV)

Algorithm Name	Acronym	Subcategory	Type	Year	Reference
Red Deer Algorithm	RDA	Terrestrial	Movement	2016	[201]
Rhino Herd Behavior	RHB	Terrestrial	Movement	2018	[202]
Roach Infestation Problem	RIO	Terrestrial	Foraging	2008	[203]
Reincarnation Algorithm	ROA	Other	Movement	2010	[204]
Shark Search Algorithm	SA	Aquatic	Foraging	1998	[205]
Simulated Bee Colony	SBC	Flying	Foraging	2009	[206]
Satin Bowerbird Optimizer	SBO	Flying	Movement	2017	[207]
Sine Cosine Algorithm	SCA.2	Other	Movement	2016	[208]
Snap-Drift Cuckoo Search	SDCS	Flying	Foraging	2016	[209]
Shuffled Frog-Leaping Algorithm	SFLA	Terrestrial	Movement	2006	[210]
Spotted Hyena Optimizer	SHO	Terrestrial	Foraging	2017	[211]
Swarm Inspired Projection Algorithm	SIP	Flying	Foraging	2009	[212]
Slime Mould Algorithm	SMA	Micro	Foraging	2008	[213]
Spider Monkey Optimization	SMO	Other	Foraging	2014	[214]
Seeker Optimization Algorithm	SOA	Other	Movement	2007	[215]
Symbiosis Organisms Search	SOS	Other	Movement	2014	[216]
Social Spider Algorithm	SSA	Terrestrial	Foraging	2015	[217]
Squirrel Search Algorithm	SSA.1	Flying	Movement	2019	[218]
Salp Swarm Algorithm	SSA.2	Aquatic	Foraging	2017	[219]
Shark Smell Optimization	SSO	Aquatic	Foraging	2016	[220]
Swallow Swarm Optimization	SSO.1	Flying	Foraging	2013	[221]
Social Spider Optimization	SSO.2	Terrestrial	Foraging	2013	[222]
See-See Partridge Chicks Optimization	SSPCO	Flying	Movement	2015	[223]
Surface-Simplex Swarm Evolution Algorithm	SSSE	Other	Movement	2017	[224]
Sperm Whale Algorithm	SWA	Aquatic	Movement	2016	[225]
Termite Hill Algorithm	TA	Terrestrial	Foraging	2012	[226]
Termite Colony Optimization	TCO	Terrestrial	Foraging	2010	[227]
The Great Salmon Run Algorithm	TGSR	Aquatic	Movement	2013	[228]
Virtual Ants Algorithm	VAA	Flying	Foraging	2006	[229]
Virtual Bees Algorithm	VBA	Flying	Foraging	2005	[230]
Virus Colony Search	VCS	Micro	Movement	2016	[231]
Virus Optimization Algorithm	VOA.1	Micro	Movement	2009	[232]
Viral Systems Optimization	VSO	Micro	Movement	2008	[233]
Wasp Colonies Algorithm	WCA	Flying	Foraging	1991	[10]
Wolf Colony Algorithm	WCA.1	Terrestrial	Foraging	2011	[234]
Worm Optimization	WO	Micro	Foraging	2014	[235]
Whale Optimization Algorithm	WOA	Aquatic	Foraging	2016	[11]
Wolf Pack Search	WPS	Terrestrial	Foraging	2007	[236]
Weightless Swarm Algorithm	WSA	Other	Movement	2012	[237]
Wolf Search Algorithm	WSA.1	Terrestrial	Foraging	2012	[238]
Wasp Swarm Optimization	WSO	Flying	Foraging	2005	[239]
Zombie Survival Optimization	ZSO	Other	Foraging	2012	[240]

Swarm Intelligence: Técnicas

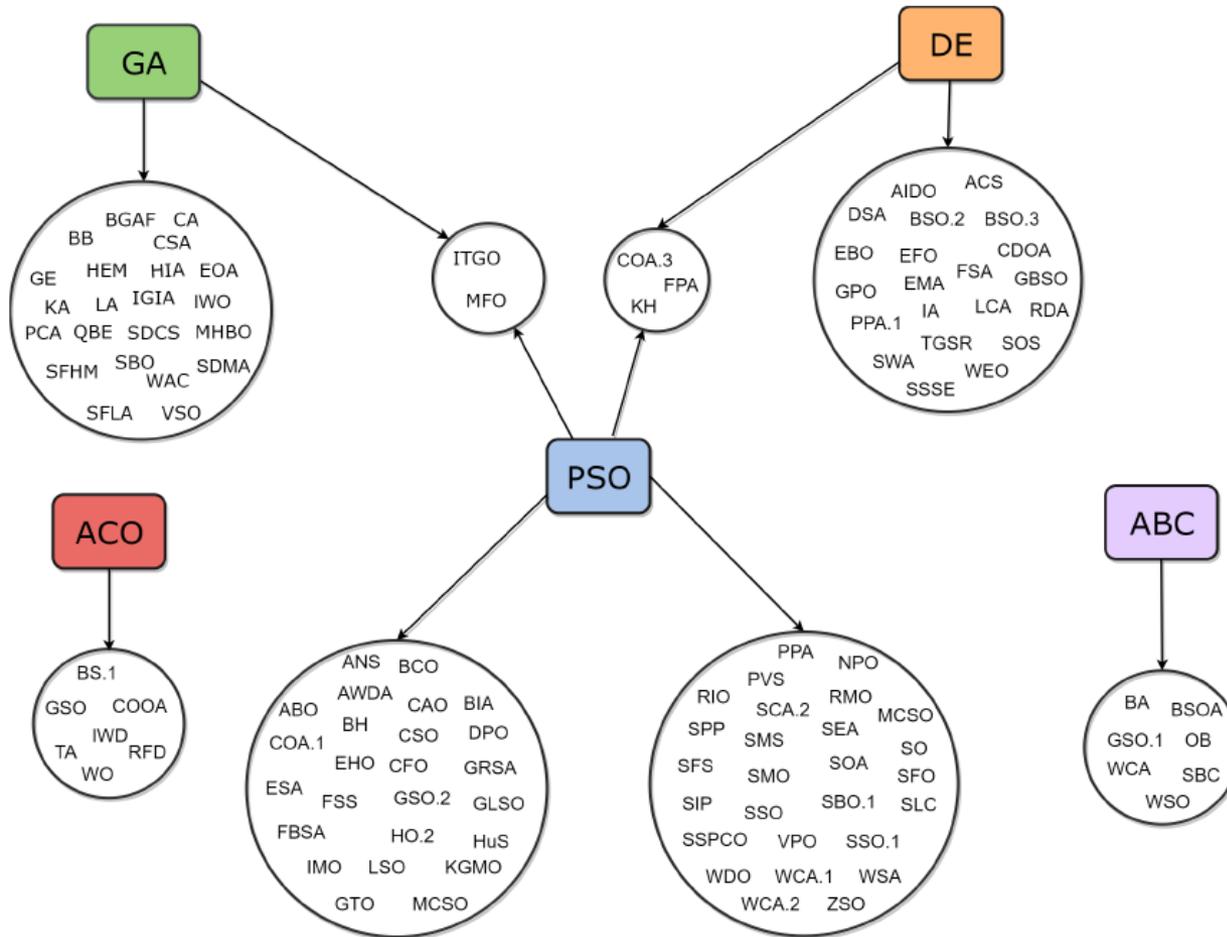


Figure 6: Classification of proposals by its original algorithm.

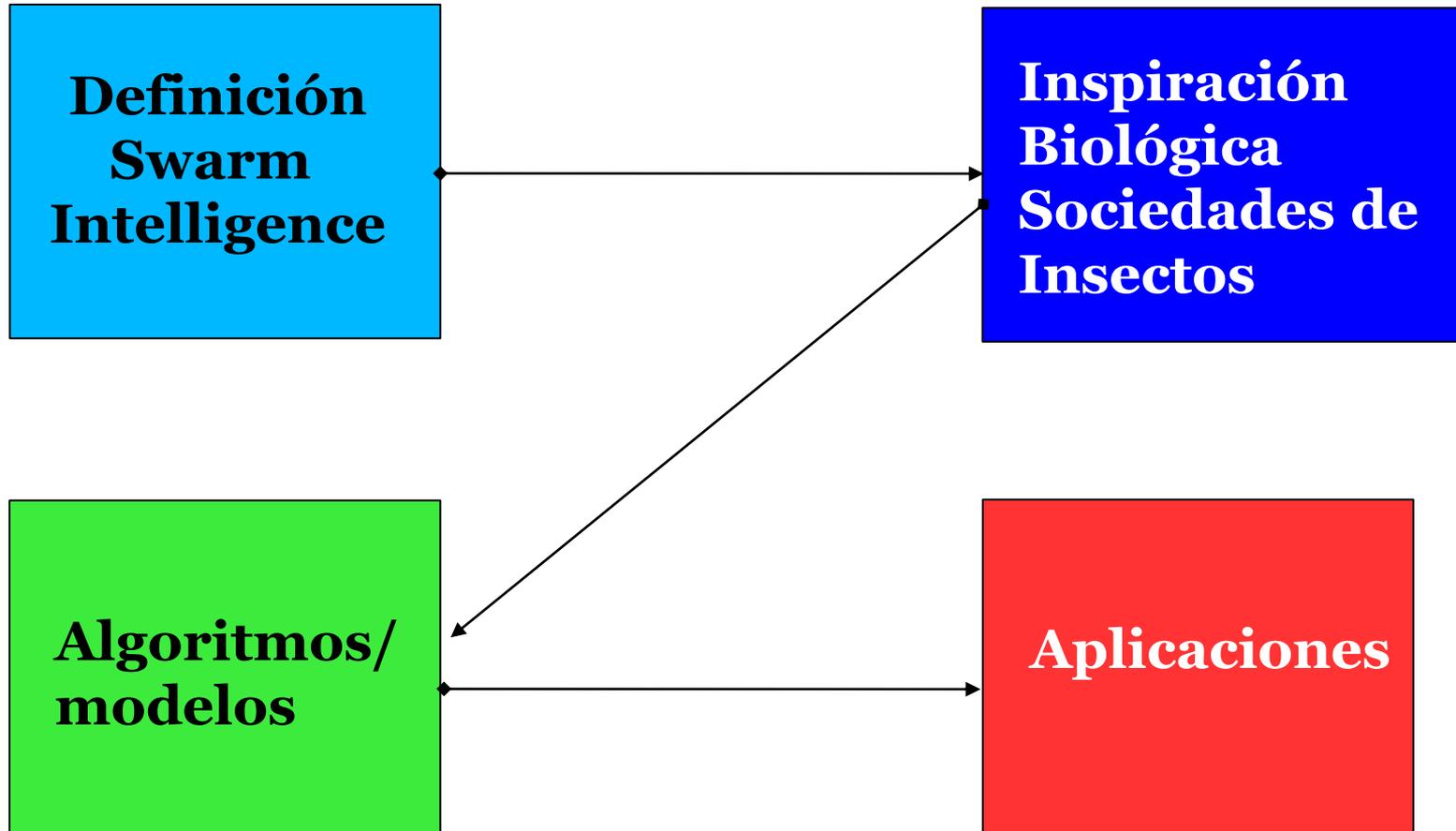
Swarm Intelligence: Técnicas

Subcategorías de algoritmos basados en el SI por el comportamiento inspirador

Movimiento (Movement): Se considera que un algoritmo pertenece a la subcategoría de inspiración de movimiento si la inspiración reside principalmente en la forma en que el animal que inspira el algoritmo se mueve regularmente por su entorno.

Buscando comida (Foraging): Más que la estrategia de movimiento, en algunas otras variantes del algoritmo es el mecanismo utilizado para obtener su alimento lo que impulsa el comportamiento del animal y, en última instancia, el diseño del algoritmo metaheurístico.

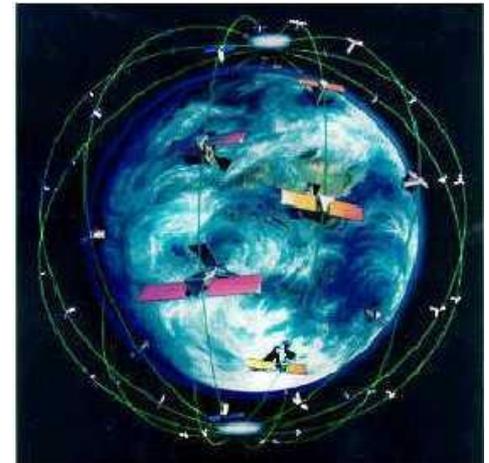
INTRODUCCIÓN A LOS MODELOS BASADOS EN ADAPTACIÓN SOCIAL: SWARM INTELLIGENCE



SI: Algunas Aplicaciones

ACO Búsqueda de Rutas para una Red de Satélites

- di Caro, Dorigo, y otros autores mostraron que ACO da buenos resultados en la búsqueda de rutas en grandes sistemas de telecomunicaciones y redes de ordenadores.

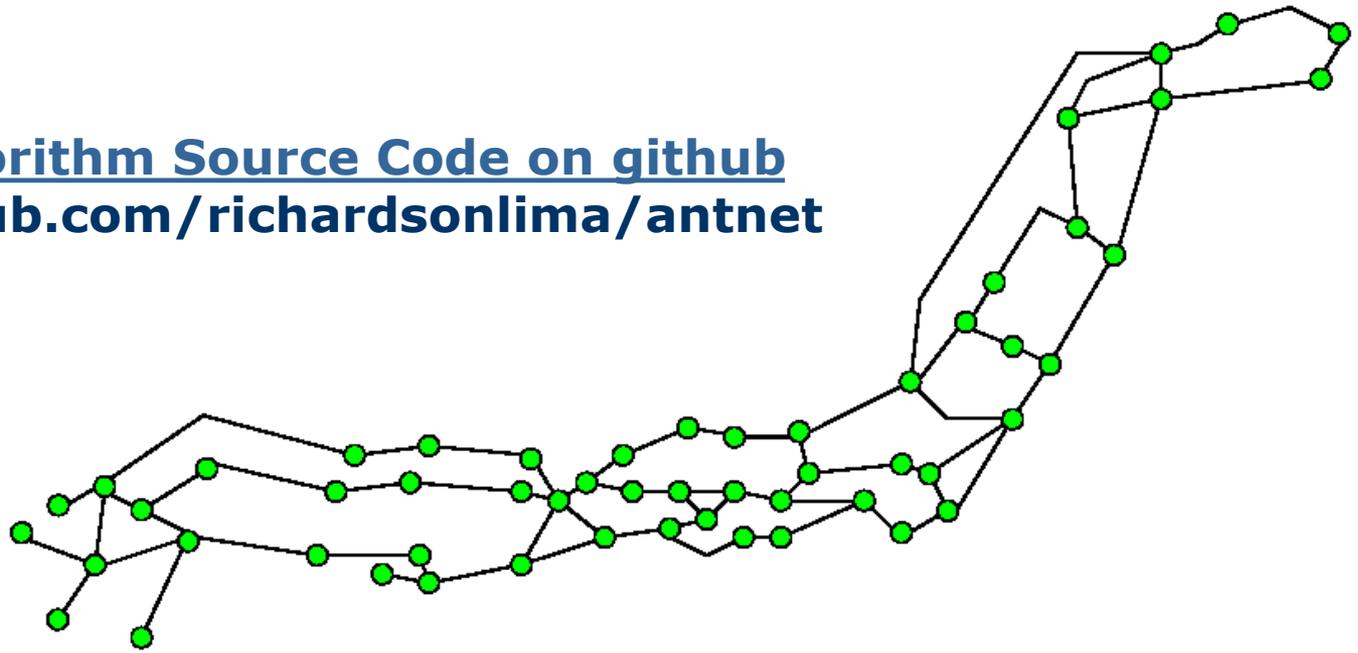


AntNet es un algoritmo para la adaptación de rutas en redes y ha sido diseñado basado en Ant Colony Optimization (ACO).

SI: Algunas Aplicaciones

- Una red utilizada en el estudio. La red de datos de Japón: NNTnet

[AntNet Algorithm Source Code on github](https://github.com/richardsonlima/antnet)
<https://github.com/richardsonlima/antnet>



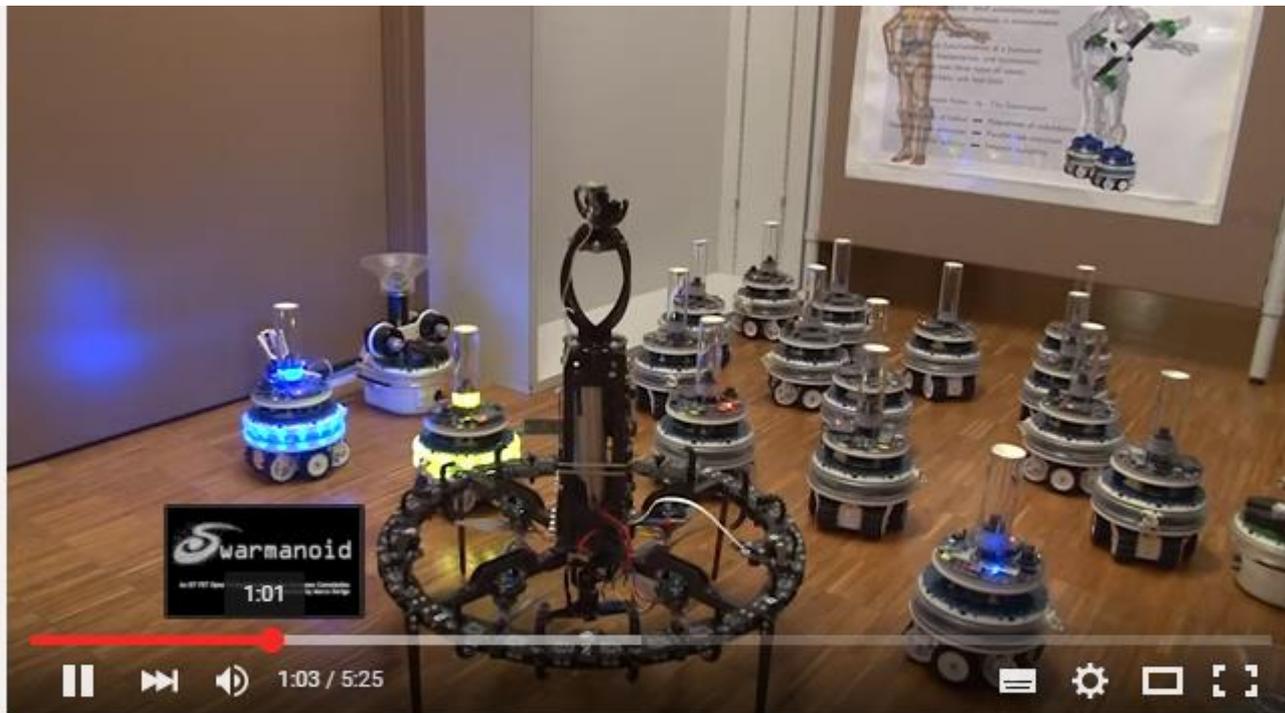
SI: Algunas Aplicaciones

ACO: Resuelven problemas que se pueden representar como rutas/caminos entre nodos de un grafo.

PSO: Resuelven problemas de optimización de variables continuas.



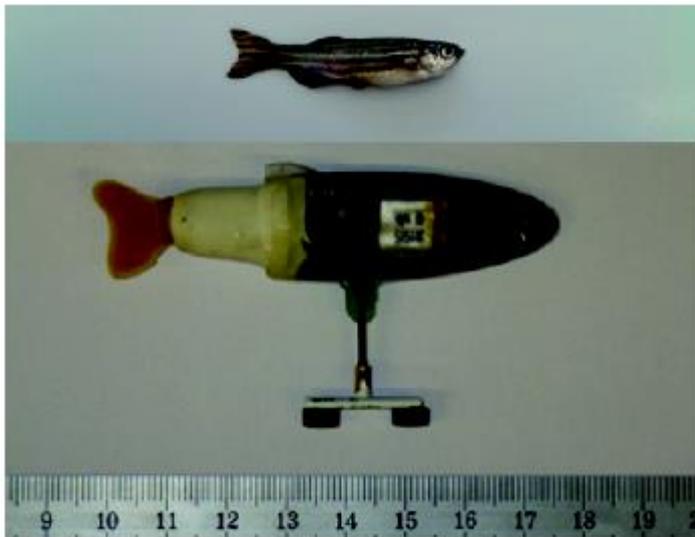
SI: Interacción Animal-robot (swarm robotics)



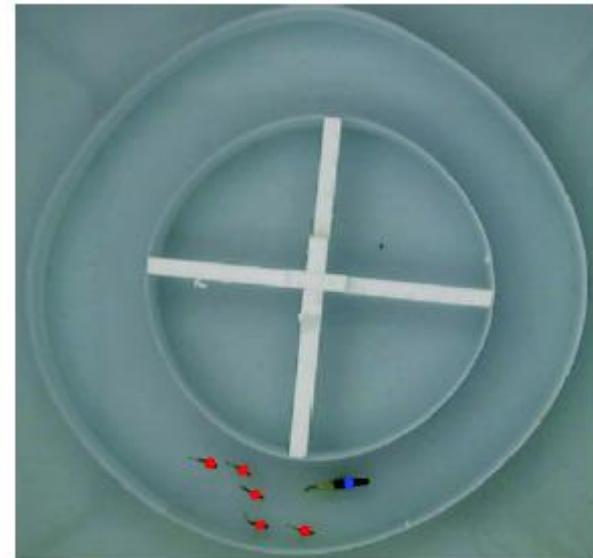
Swarmanoid, the movie

SI: Interacción Animal-robot (swarm robotics)

Robot fish



Influencing the fish school



- ▶ F. Bonnet, Y. Kato, J. Halloy, F. Mondada. Infiltrating the Zebrafish Swarm: Design, Implementation and Experimental Tests of a Miniature Robotic Fish Lure for Fish-Robot Interaction Studies. *SWARM 2015: The First International Symposium on Swarm Behavior and Bio-Inspired Robotics*, 2015.

SI: Interacción Animal-robot



- ▶ S. Gade, A. A. Paranjape, and S.-J. Chung. a Flock of Birds Approaching an Airport Using an Unmanned Aerial Vehicle. AIAA Guidance, Navigation, and Control Conference, 2015.

METAHEURÍSTICAS

TEMA 6. METAHEURÍSTICAS BASADAS EN ADAPTACIÓN SOCIAL

1. **INTRODUCCIÓN A LOS MODELOS BASADOS EN ADAPTACIÓN SOCIAL: SWARM INTELLIGENCE**
2. **ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS**
3. **OPTIMIZACIÓN BASADA EN NUBES DE PARTÍCULOS (PARTICLE SWARM OPTIMIZATION)**
4. **CONCLUSIONES**

Bibliografía

E. Bonabeau, M. Dorigo, G. Theraulaz. *Swarm Intelligence. From Nature to Artificial Systems*. Oxford University Press, 1999.

Kennedy, J., Eberhart, R. C., and Shi, Y.. *Swarm intelligence*. San Francisco: Morgan Kaufmann Publishers, 2001.

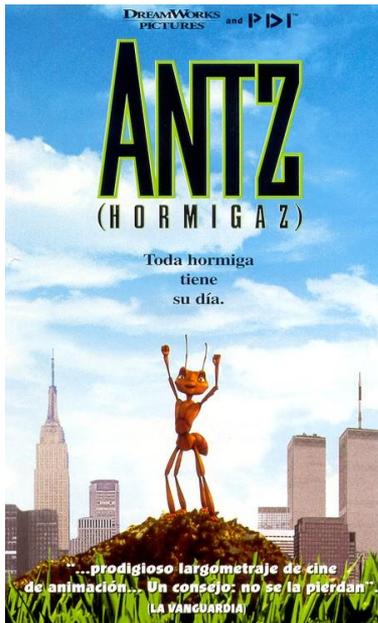
METAHEURÍSTICAS

ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- 1. LAS COLONIAS DE HORMIGAS NATURALES**
- 2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS**
- 3. APLICACIONES**
- 4. COMENTARIOS FINALES**
- 5. BIBLIOGRAFÍA Y RECURSOS**

1. LAS COLONIAS DE HORMIGAS NATURALES

- Las hormigas son insectos sociales que viven en colonias y que tienen un comportamiento dirigido al desarrollo de la colonia como un todo mas que a un desarrollo individual



“Antz (Hormiga Z)”

© DreamWorks Pictures. 1998

Recordad...

¡SED LA BOLA!

1. LAS COLONIAS DE HORMIGAS NATURALES (2)

- Una característica interesante del comportamiento de las colonias de hormigas es cómo pueden encontrar los caminos más cortos entre el hormiguero y la comida
- Sobre todo porque... ¡¡LAS HORMIGAS SON CIEGAS!!

Entonces...

¿Cómo lo hacen?



1. LAS COLONIAS DE HORMIGAS NATURALES (3)

- En su recorrido, depositan una sustancia llamada feromona que todas pueden oler (*estimergia*)
- Este rastro permite a las hormigas volver a su hormiguero desde la comida

“Bichos. Una aventura en miniatura”

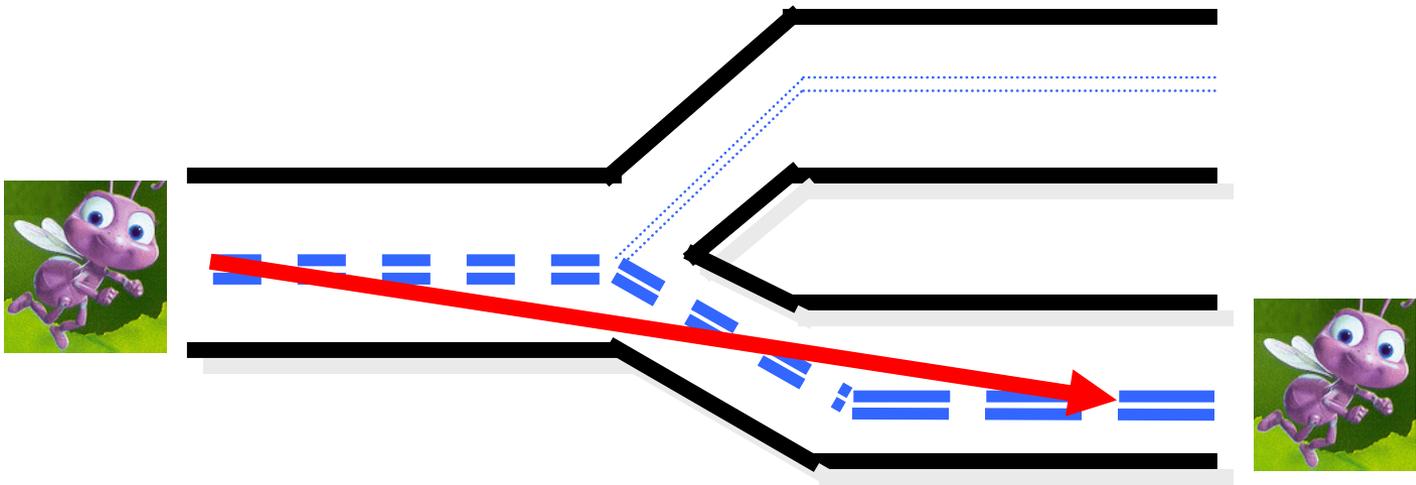
© Disney-Pixar. 1999

¡He perdido el rastro,
he perdido el rastro!



1. LAS COLONIAS DE HORMIGAS NATURALES (4)

- Cada vez que una hormiga llega a una intersección, decide el camino a seguir de un modo probabilístico

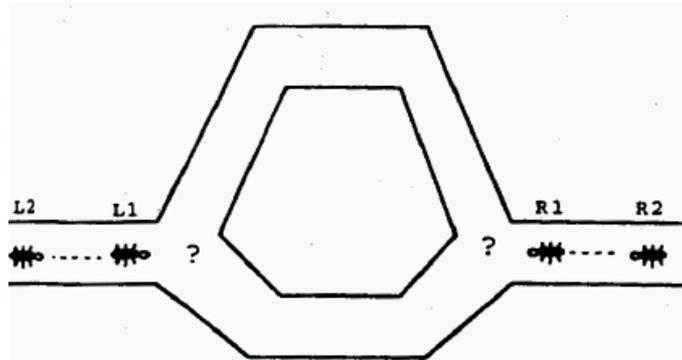


- Las hormigas eligen con mayor probabilidad los caminos con un alto rastro de feromona

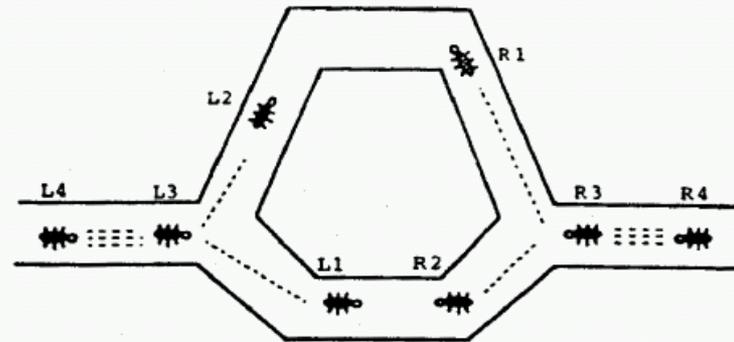
1. LAS COLONIAS DE HORMIGAS NATURALES (5)

- Las bifurcaciones más prometedoras (más cercanas a la comida) van acumulando feromona al ser recorridas por más hormigas (*reclutamiento de masas*)
- Las menos prometedoras pierden feromona por *evaporación* al ser visitadas por menos hormigas cada vez. Aún así, *la gran perduración de los rastros hace que la evaporación influya poco*
- La acción continuada de la colonia da lugar a un rastro de feromona que permite a las hormigas encontrar un camino cada vez más corto desde el hormiguero a la comida

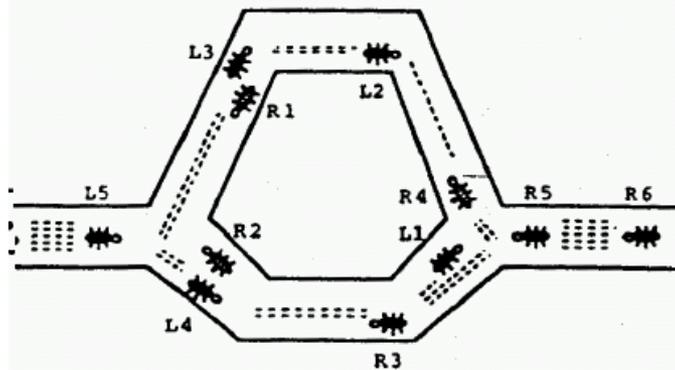
1. LAS COLONIAS DE HORMIGAS NATURALES (6)



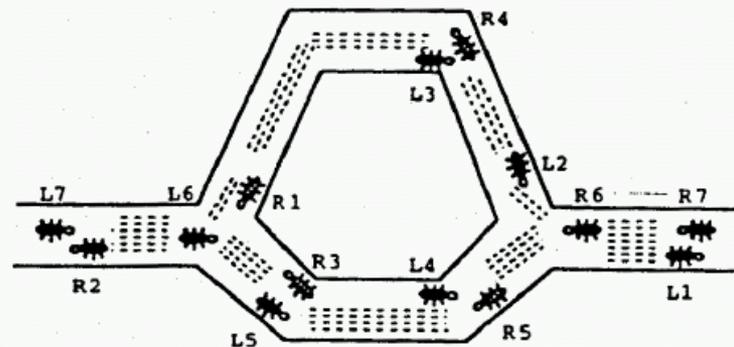
(a)



(b)



(c)



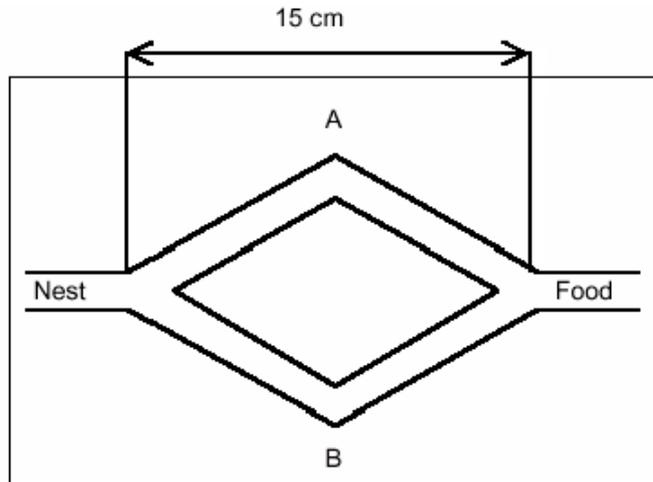
(d)

1. LAS COLONIAS DE HORMIGAS NATURALES (7)

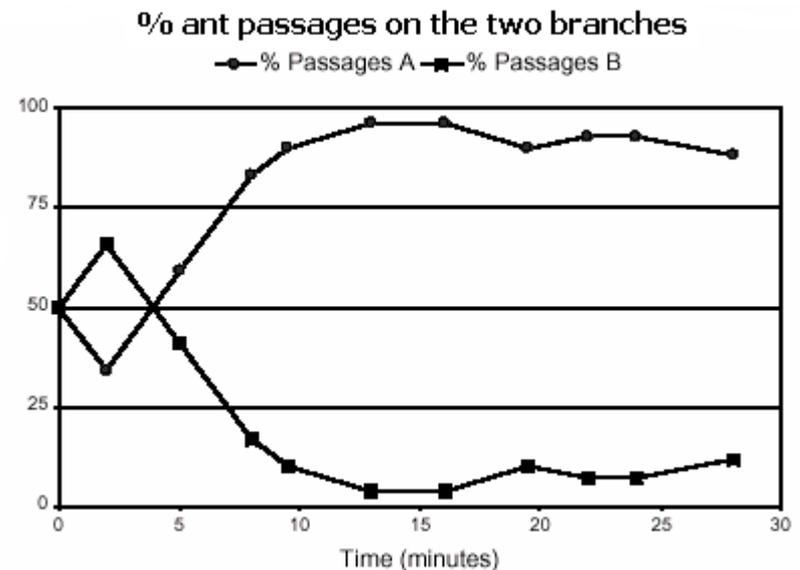
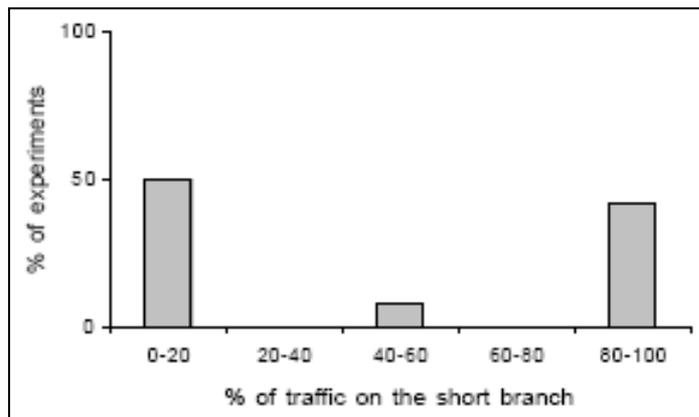
EXPERIMENTOS DEL DOBLE PUENTE

- Deneubourg realizó un experimento de laboratorio con un tipo concreto de hormigas que depositan feromona al ir del hormiguero a la comida y al volver
- Usaron dos tipos de circuitos (**puentes**). En el primero, las dos ramas del puente tenían la misma longitud. En el segundo, una rama era el doble de larga que la otra
- Después, unieron dos puentes cruzados del segundo tipo

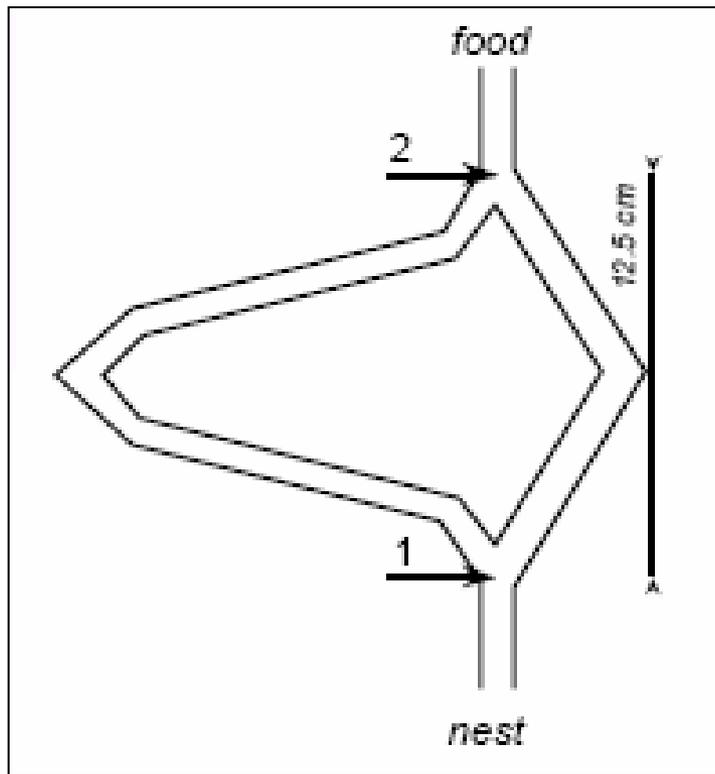
1. LAS COLONIAS DE HORMIGAS NATURALES (8)



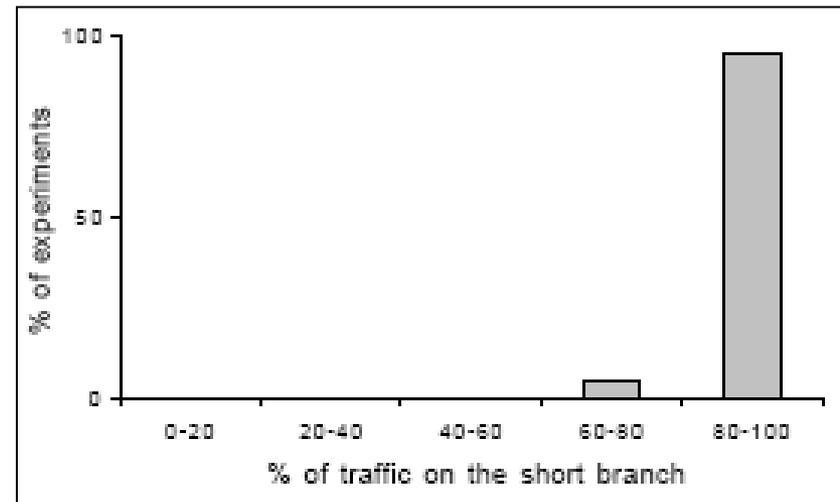
- En el primer puente, las hormigas terminaban por converger a una sola rama (cualquiera de las dos)



1. LAS COLONIAS DE HORMIGAS NATURALES (9)

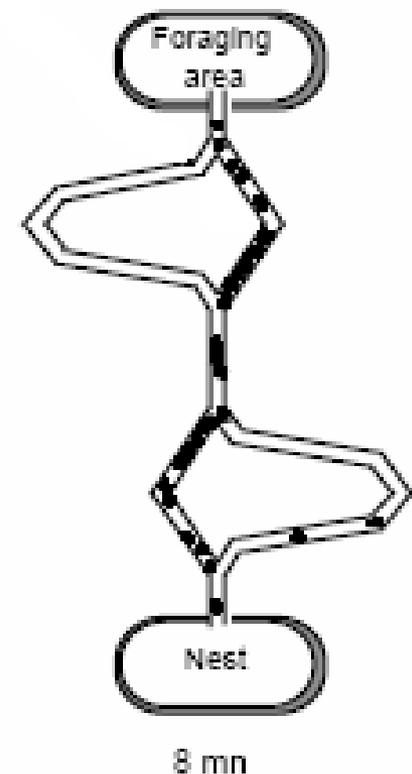
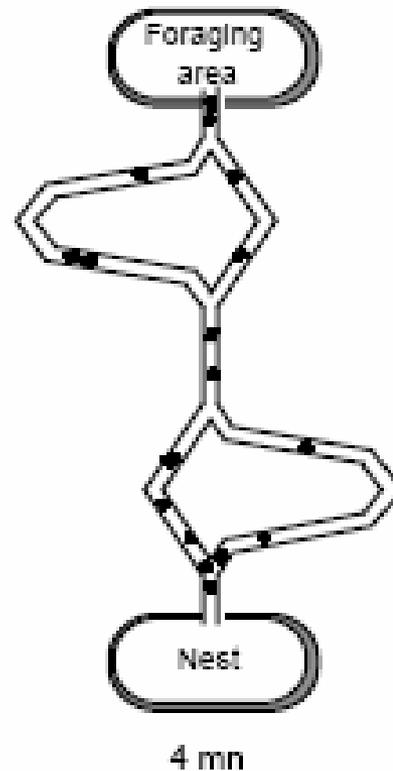
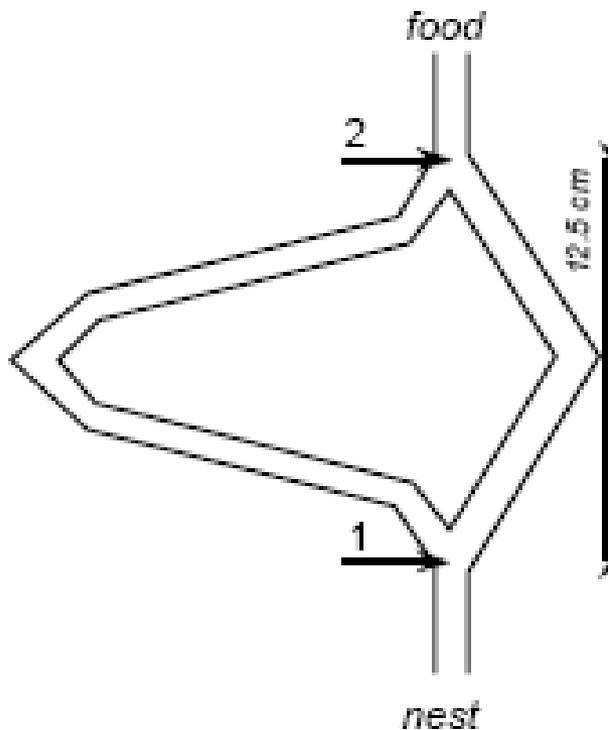


- En el segundo, las hormigas convergían a la rama más corta



1. LAS COLONIAS DE HORMIGAS NATURALES (10)

- En el circuito con dos puentes dobles cruzados, las hormigas consiguen encontrar el camino más corto



1. LAS COLONIAS DE HORMIGAS NATURALES (11)

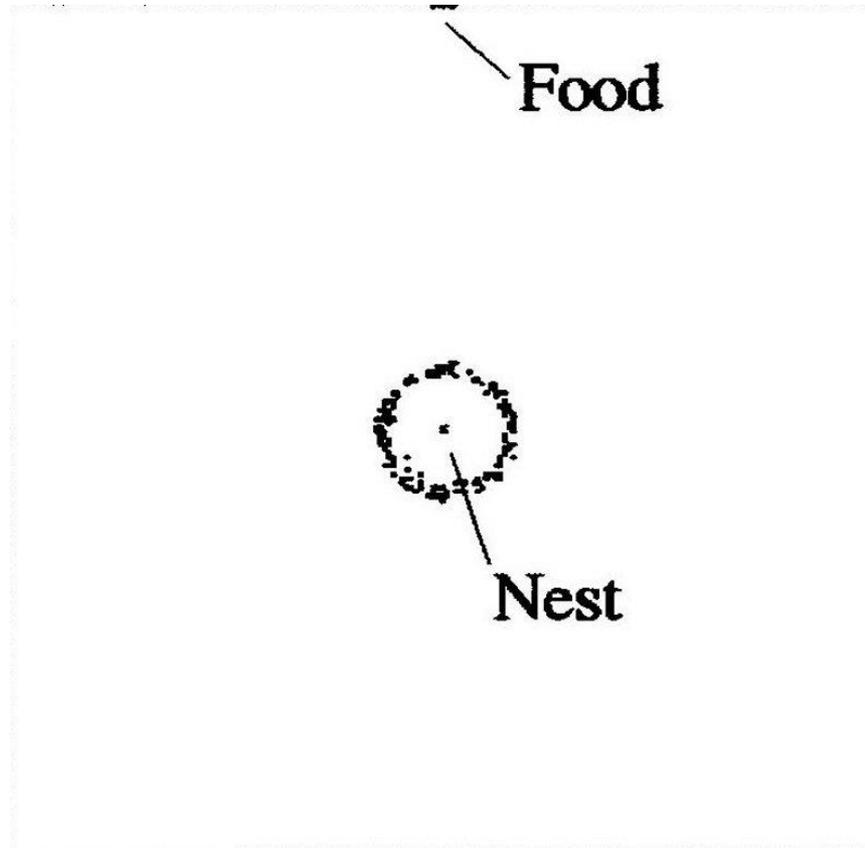
- Como resultado de estos experimentos, Deneubourg y su equipo diseñaron un **modelo estocástico** del proceso de decisión de las hormigas naturales:

$$p_{i,a} = \frac{[k + \tau_{i,a}]^\alpha}{[k + \tau_{i,a}]^\alpha + [k + \tau_{i,a'}]^\alpha}$$

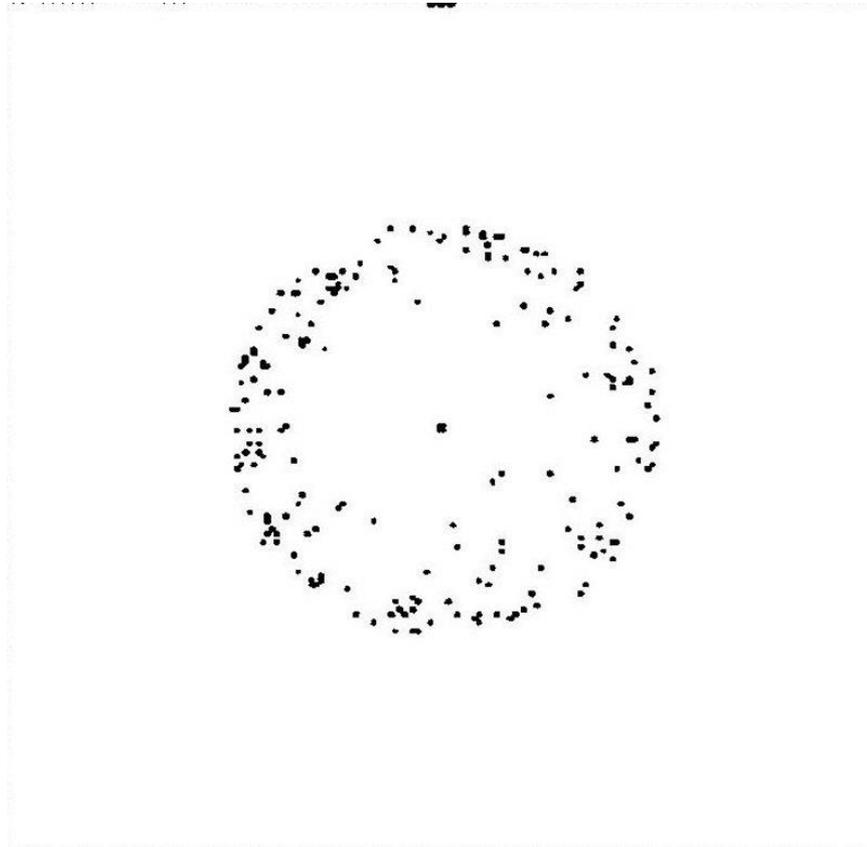
donde:

- $p_{i,a}$ es la probabilidad de escoger la rama a estando en el punto de decisión i , y
- $\tau_{i,a}$ es la concentración de feromona en la rama a

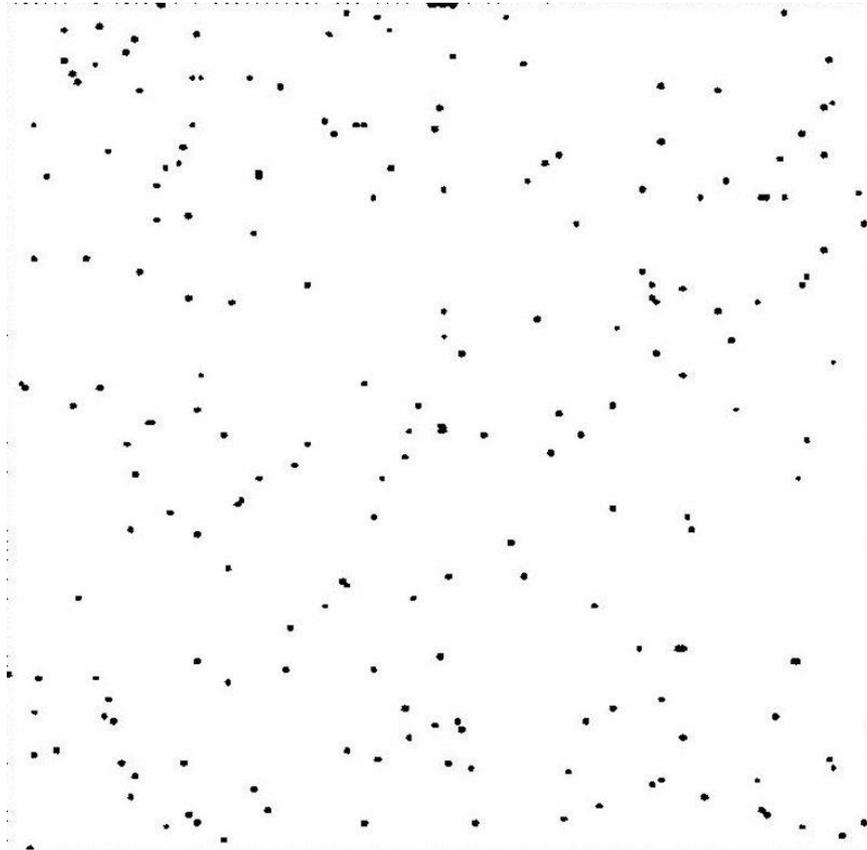
1. LAS COLONIAS DE HORMIGAS NATURALES (12)



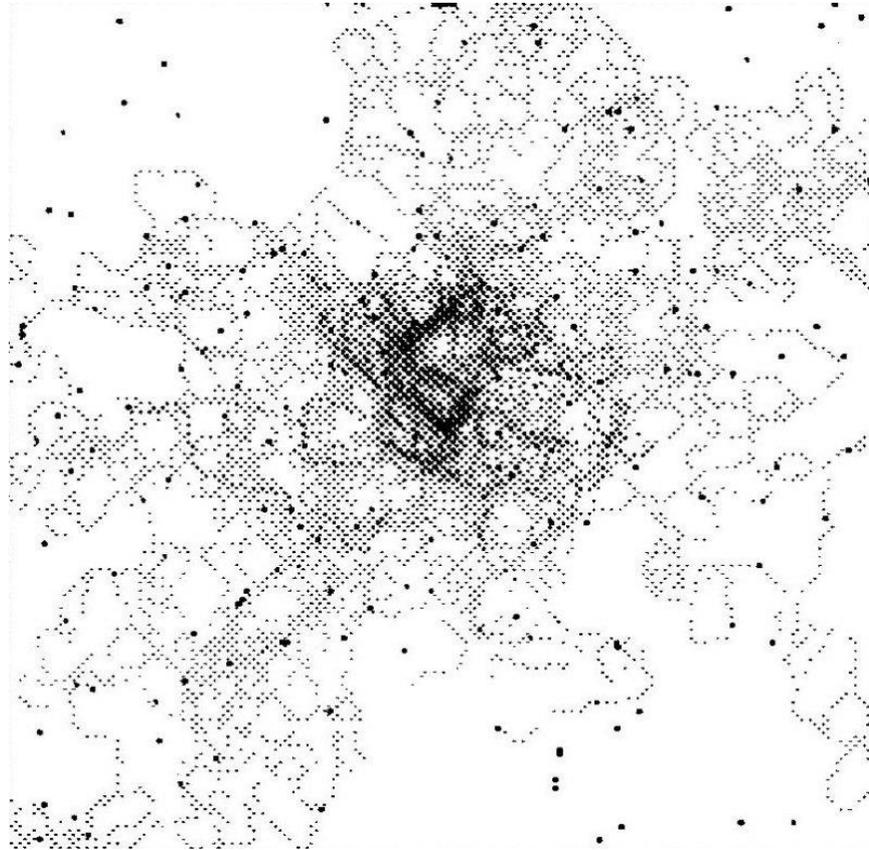
1. LAS COLONIAS DE HORMIGAS NATURALES (12)



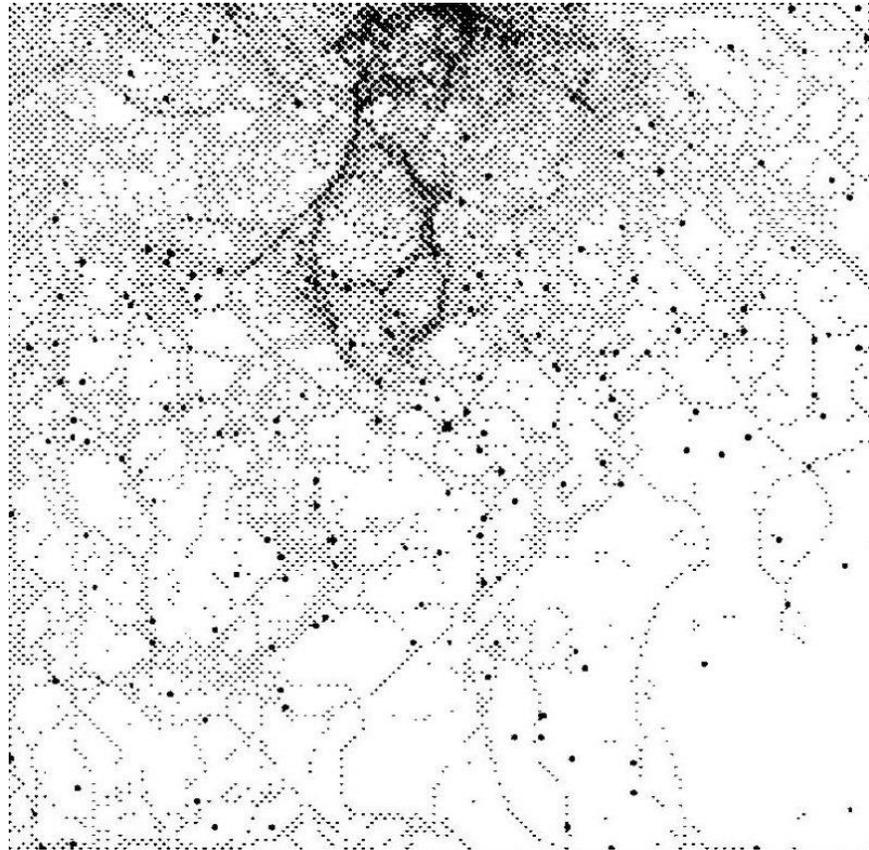
1. LAS COLONIAS DE HORMIGAS NATURALES (12)



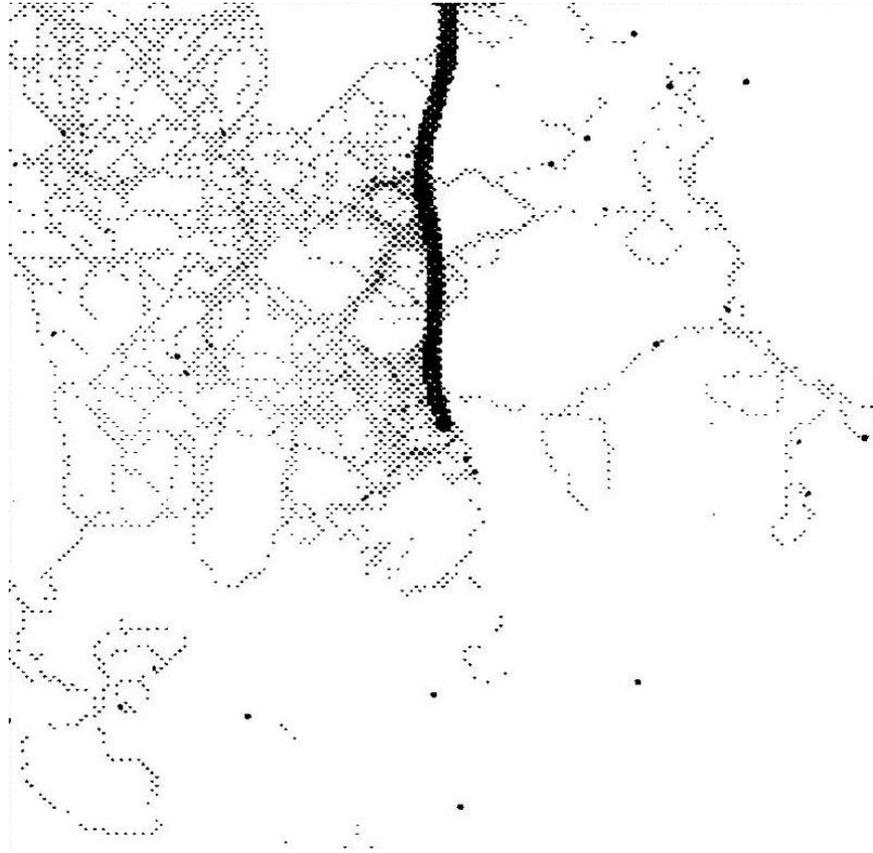
1. LAS COLONIAS DE HORMIGAS NATURALES (12)



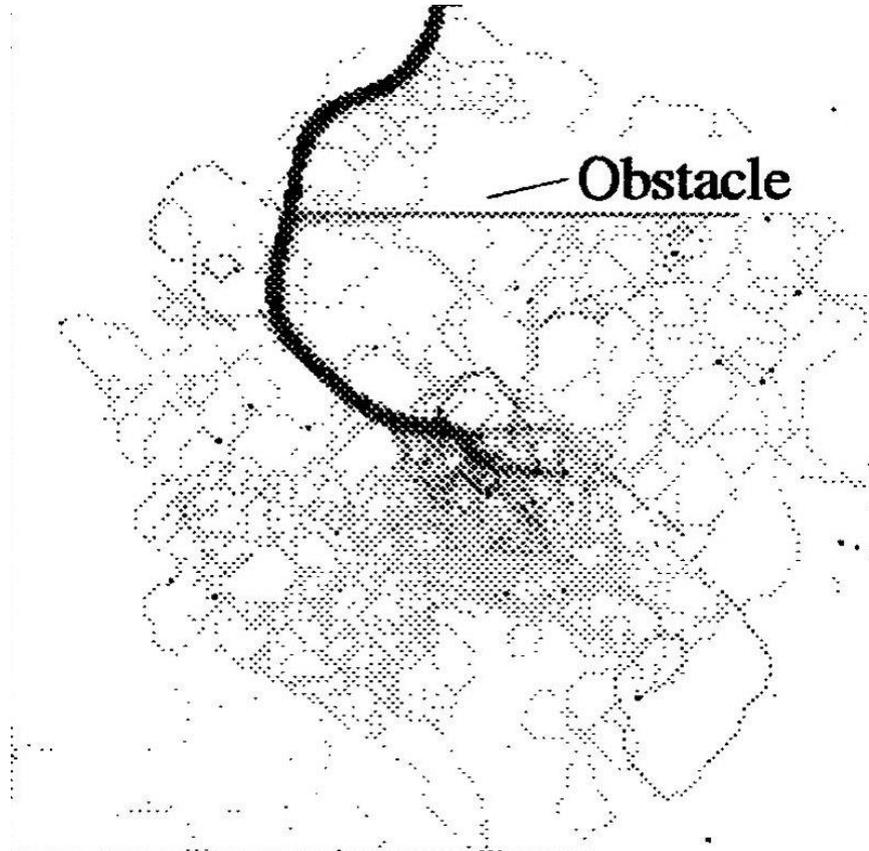
1. LAS COLONIAS DE HORMIGAS NATURALES (12)



1. LAS COLONIAS DE HORMIGAS NATURALES (12)

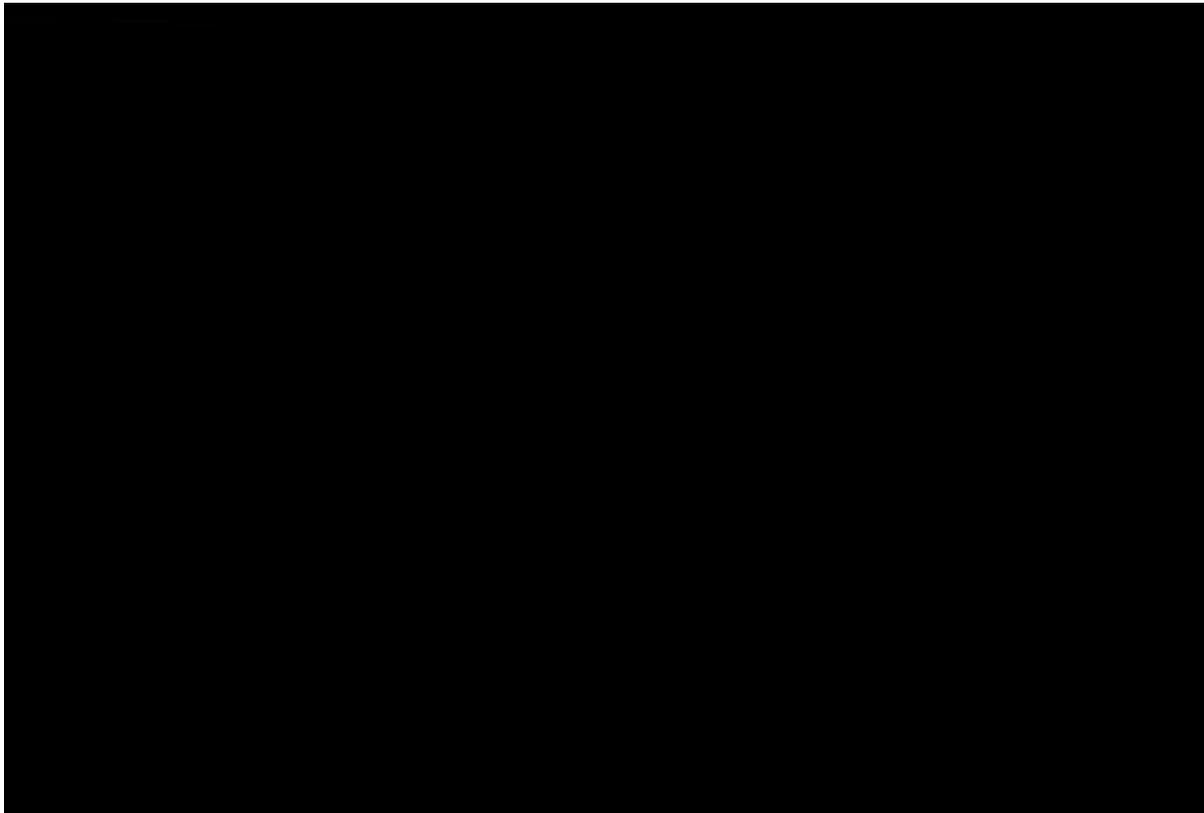


1. LAS COLONIAS DE HORMIGAS NATURALES (12)



1. LAS COLONIAS DE HORMIGAS NATURALES (13)

Experimento con Hormigas reales. Como encuentran el camino mínimo (159 segundos)

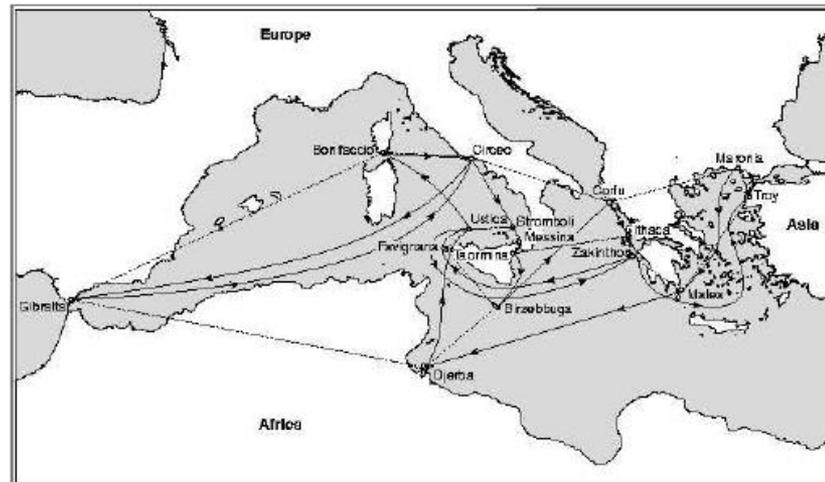


2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- **De las hormigas naturales a la OCH**
- **La hormiga artificial**
- **Actualización de feromona**
- **El Sistema de Hormigas**
- **Ejemplo de aplicación del SH al TSP**
- **Otros algoritmos de OCH**
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

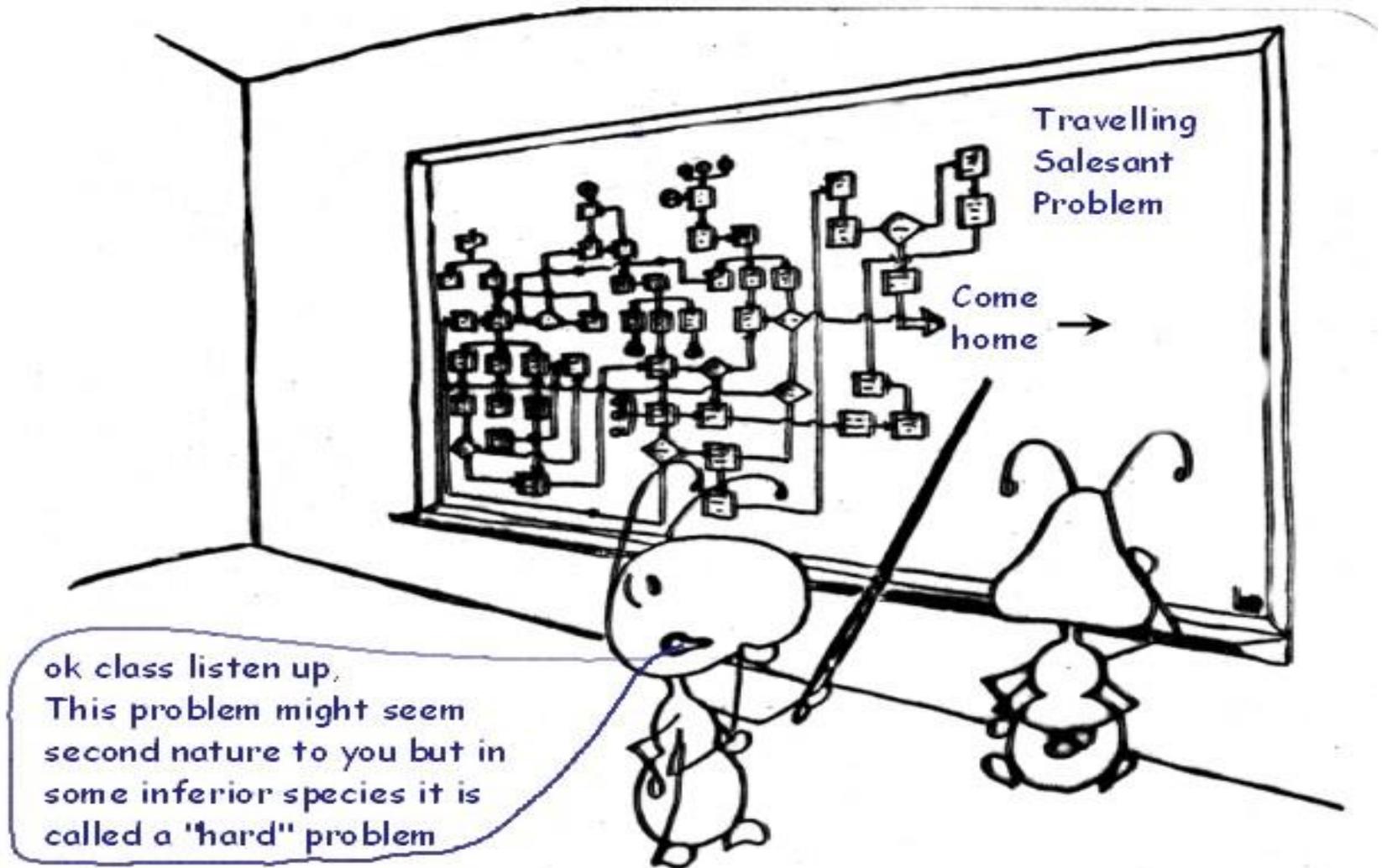
DE LAS HORMIGAS NATURALES A LA OCH

- El Problema del Viajante de Comercio (TSP) es uno de los problemas de optimización combinatoria más conocido
- Se dispone de un conjunto $N = \{1, \dots, n\}$ de ciudades, que han de ser visitadas una sola vez, volviendo a la ciudad de origen, y recorriendo la menor distancia posible



- **Las hormigas naturales son capaces de resolver problemas de camino mínimo como el TSP**

DE LAS HORMIGAS NATURALES A LA OCH

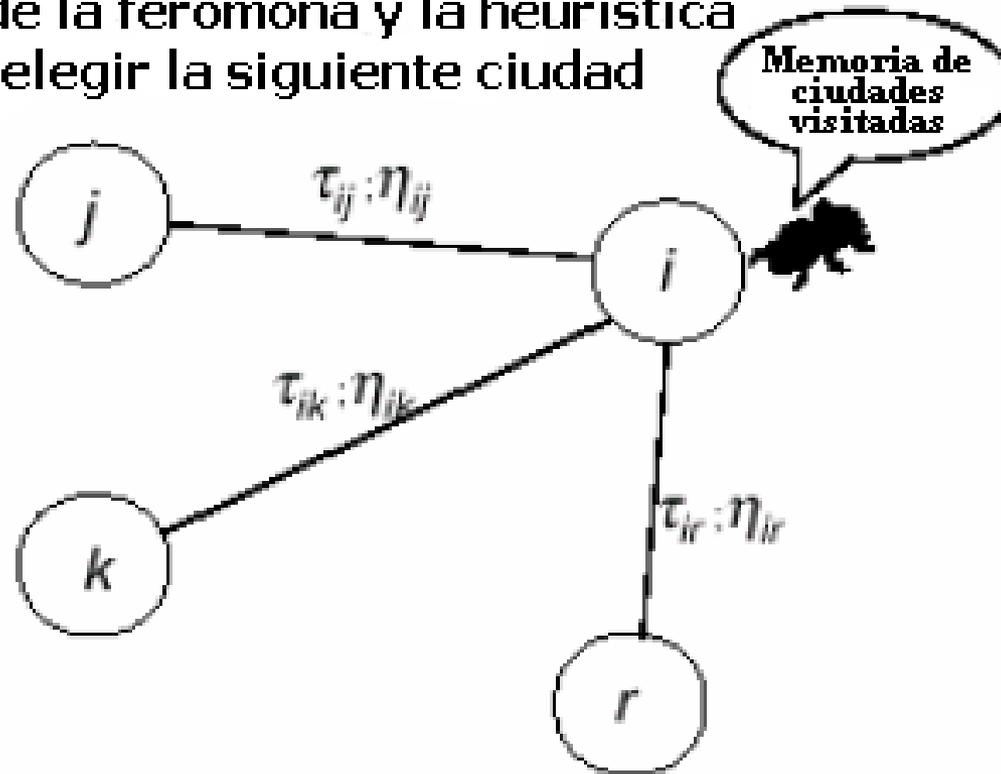


DE LAS HORMIGAS NATURALES A LA OCH (2)

- Los algoritmos de OCH reproducen el comportamiento de las hormigas reales en una colonia artificial de hormigas para resolver problemas complejos de camino mínimo
- Cada hormiga artificial es un mecanismo probabilístico de construcción de soluciones al problema (un agente que imita a la hormiga natural) que usa:
 - Unos rastros de feromona (artificiales) τ que cambian con el tiempo para reflejar la experiencia adquirida por los agentes en la resolución del problema
 - Información heurística η sobre la instancia concreta del problema

DE LAS HORMIGAS NATURALES A LA OCH (3)

Uso de la feromona y la heurística
para elegir la siguiente ciudad



2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- El Sistema de Hormigas
- Ejemplo de aplicación del SH al TSP
- Otros algoritmos de OCH
- Estudio comparativo de la OCH en el TSP
- Algoritmos de OCH con búsqueda local
- La metaheurística OCH

LA HORMIGA ARTIFICIAL

La hormiga artificial es un agente que:

- Recuerda los nodos que ha recorrido, utilizando para ello una lista de nodos visitados (L). Al finalizar, esta lista contiene la solución construida por la hormiga
- En cada paso, estando en la ciudad r elige hacia qué ciudad s moverse de entre las vecinas de r que no hayan sido visitados aún ($J(r) = \{s \mid \exists a_{rs} \text{ y } s \notin L\}$), según una regla probabilística de transición

La decisión tomada es función de la preferencia heurística $\eta_{rs}=1/d_{rs}$ y la feromona τ_{rs}

LA HORMIGA ARTIFICIAL (2)

- La regla probabilística de transición más habitual define la probabilidad con la que la hormiga k , situada en la ciudad r , decide moverse hacia la ciudad s :

$$p_k(r, s) = \begin{cases} \frac{[\tau_{rs}]^\alpha \cdot [\eta_{rs}]^\beta}{\sum_{u \in J_k(r)} [\tau_{ru}]^\alpha \cdot [\eta_{ru}]^\beta}, & \text{si } s \in J_k(r) \\ 0, & \text{en otro caso} \end{cases}$$

- τ_{rs} es la feromona del arco a_{rs}
- η_{rs} es la inf. heurística del arco a_{rs}
- $J_k(r)$ es el conjunto de nodos alcanzables desde r no visitados aún por la hormiga k
- α y β son pesos que establecen un equilibrio entre la importancia de la información memorística y heurística

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- **Actualización de feromona**
- **El Sistema de Hormigas**
- **Ejemplo de aplicación del SH al TSP**
- **Otros algoritmos de OCH**
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

ACTUALIZACIÓN DE FEROMONA

1. Se usa una **retroalimentación positiva** para reforzar en el futuro los componentes de las buenas soluciones mediante un aporte adicional de feromona

Cuanto mejor sea la solución, más feromona se aporta

2. Se usa la **evaporación de feromona** para evitar un incremento ilimitado de los rastros de feromona y para permitir olvidar las malas decisiones tomadas

La **evaporación** es la misma para todos los rastros, eliminándose un porcentaje de su valor actual: $0 \leq \rho \leq 1$

Es un **mecanismo de evaporación más activo** que el natural, lo que evita la perduración de los rastros de feromona y, por tanto, el estancamiento en óptimos locales

ACTUALIZACIÓN DE FEROMONA (2)

- Un ejemplo de regla de actualización de feromona sería:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \sum_{k=1}^m \Delta \tau_{rs}^k$$

$$- \Delta \tau_{rs}^k = \begin{cases} \frac{1}{C(S_k)}, & \text{si la hormiga } k \text{ ha visitado el arco } a_{rs} \\ 0, & \text{en otro caso} \end{cases}$$

- $C(S_k)$ es el coste de la solución generada por la hormiga k , es decir, la longitud del circuito S_k
 - El valor del aporte anterior es para problemas de minimización como el TSP. Si el problema es de maximización, $\Delta \tau_{rs}^k = C(S_k)$
 - m es el número de hormigas
- Los arcos visitados por hormigas en la iteración actual (**arcos prometedores**) reciben un aporte extra de feromona y los no visitados por ninguna hormiga (**poco prometedores**) la pierden

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- **El Sistema de Hormigas**
- **Ejemplo de aplicación del SH al TSP**
- **Otros algoritmos de OCH**
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

EL SISTEMA DE HORMIGAS

- El algoritmo anterior fue el primero de OCH que se propuso, denominado Sistema de Hormigas (*Ant System*) (SH)

M. Dorigo, V. Maniezzo, A. Coloni, *The Ant System: Optimization by a Colony of Cooperating Agents*. IEEE Trans. On Systems, Man and Cybernetics-Part B, Vol. 26, 1996, 1-13

- Obtenía buenos resultados en el TSP pero no eran lo suficientemente competitivos con respecto a los de los mejores algoritmos (state-of-the-art)
- Fue la base para el desarrollo posterior de la OCH, al proponerse muchas mejoras

EL SISTEMA DE HORMIGAS (2)

ALGORITMO SISTEMA DE HORMIGAS

- Inicialización de parámetros (p.e., asignación de la cantidad inicial a los rastros de feromona $\tau[i][j] \leftarrow \tau_0$)
- *Para It=1 hasta Número_de_Iteraciones hacer*
 1. *Para k=1 hasta m (Número_de_Hormigas) hacer*
 $L[k][1] \leftarrow$ nodo inicial
 - /* Construcción de soluciones por las hormigas */*
 2. *Para i=2 hasta Número_de_Nodos hacer*
 Para k=1 hasta Número_de_Hormigas hacer
 $L[k][i] \leftarrow$ Regla_transición ($L[k], \tau, \eta$)

EL SISTEMA DE HORMIGAS (3)

/ Actualización de feromona */*

3. Para $k=1$ hasta Número_de_Hormigas hacer

Coste[k] \leftarrow C(L[k])

Mejor_Actual \leftarrow Mejor (L[k])

4. Para $i=1$ hasta Número_de_Nodos hacer

Para $j=1$ hasta Número_de_Nodos hacer

Actualización_feromona($\tau[i][j]$, L, C(L))

5. Si C(Mejor_Actual) es mejor que C(Mejor_Global) entonces Mejor_Global \leftarrow Mejor_Actual

▪ *DEVOLVER* Mejor_Global

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- El Sistema de Hormigas
- **Ejemplo de aplicación del SH al TSP**
- **Otros algoritmos de OCH**
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

EJEMPLO DE APLICACIÓN DEL SH AL TSP

A modo de ejemplo, vamos a describir una iteración del SH sobre un caso sencillo del TSP de tamaño $n=6$ con las siguientes matrices de distancias D y heurística η :

$D =$

∞	1	$\sqrt{5}$	$\sqrt{5}$	2	$\sqrt{2}$
1	∞	$\sqrt{2}$	2	$\sqrt{5}$	$\sqrt{5}$
$\sqrt{5}$	$\sqrt{2}$	∞	$\sqrt{2}$	$\sqrt{5}$	3
$\sqrt{5}$	2	$\sqrt{2}$	∞	1	$\sqrt{5}$
2	$\sqrt{5}$	$\sqrt{5}$	1	∞	$\sqrt{2}$
$\sqrt{2}$	$\sqrt{5}$	3	$\sqrt{5}$	$\sqrt{2}$	∞

$\eta =$

-	1.000	0.447	0.447	.500	0.707
1.000	-	0.707	0.500	0.447	0.447
0.447	0.707	-	0.707	0.447	0.333
0.447	0.500	0.707	-	1.000	0.447
0.500	0.447	0.447	1.000	-	0.707
0.707	0.447	0.333	0.447	0.707	-

Fuente: J.M. Moreno-Vega, J.A. Moreno-Pérez, *Heurísticas en Optimización*, Consejería de Educación, Cultura y Deportes. Gobierno de Canarias, 1999.

EJEMPLO DE APLICACIÓN DEL SH AL TSP (2)

Suponiendo una población de $m=n=6$ hormigas, cada una partiendo de una ciudad distinta, y la matriz de feromona inicializada a $\tau_0=10$, el proceso constructivo de las tres primeras podría ser el siguiente:

Probabilidades de transición								
	1	2	3	4	5	6	Uniforme	Solución
1	-	0.322	0.144	0.144	0.161	0.227	0.000	(1 2 - - - -)
2	-	-	0.336	0.237	0.212	0.212	0.031	(1 2 3 - - -)
3	-	-	-	0.475	0.300	0.225	0.673	(1 2 3 5 - -)
5	-	-	-	0.585	-	0.415	0.842	(1 2 3 5 6 -)
6	-	-	-	1.000			-	(1 2 3 5 6 4)

EJEMPLO DE APLICACIÓN DEL SH AL TSP (3)

Probabilidades de transición								
	1	2	3	4	5	6	Uniforme	Solución
2	0.322	-	0.227	0.161	0.144	0.144	0.279	(2 3 - - - -)
3	0.231	-	-	0.365	0.231	0.172	0.671	(2 3 5 - - -)
5	0.227	-	-	0.453	-	0.320	0.931	(2 3 5 6 - -)
6	0.612	-	-	0.388	-	-	0.873	(2 3 5 6 4 -)
4	1.000	-	-	-	-	-	-	(2 3 5 6 4 1)

Probabilidades de transición								
	1	2	3	4	5	6	Uniforme	Solución
3	0.169	0.267	-	0.267	0.169	0.126	0.372	(3 2 - - - -)
2	0.417	-	-	0.208	0.186	0.186	0.415	(3 2 1 - - -)
1	-	-	-	0.267	0.309	0.434	0.321	(3 2 1 5 - -)
5	-	-	-	0.585	-	0.415	0.474	(3 2 1 5 4 -)
4	-	-	-	-	-	1.000	-	(3 2 1 5 4 6)

EJEMPLO DE APLICACIÓN DEL SH AL TSP (4)

Una vez construidas las 6 soluciones, se aplica la actualización de feromona. Considerando que cada hormiga k aporta $100/C(S_k)$ en los arcos visitados, tendríamos los siguientes aportes:

Hormiga	$C(S_k)$	Aporte	Solución	Arcos afectados
1	10.53	9.49	(1 2 3 5 6 4)	{ a_{12} , a_{23} , a_{35} , a_{56} , a_{64} , a_{41} }
2	10.53	9.49	(2 3 5 6 4 1)	{ a_{23} , a_{35} , a_{56} , a_{64} , a_{41} , a_{12} }
3	9.05	11.04	(3 2 1 5 4 6)	{ a_{32} , a_{21} , a_{15} , a_{54} , a_{46} , a_{63} }
4	11.12	8.99	(4 2 6 1 3 5)	{ a_{42} , a_{26} , a_{61} , a_{13} , a_{35} , a_{54} }
5	10.88	9.19	(5 1 6 2 4 3)	{ a_{51} , a_{16} , a_{62} , a_{24} , a_{43} , a_{35} }
6	9.47	10.55	(6 5 3 4 2 1)	{ a_{65} , a_{53} , a_{34} , a_{42} , a_{21} , a_{16} }

EJEMPLO DE APLICACIÓN DEL SH AL TSP (5)

Aplicando el mecanismo de actualización de feromona, se evapora la feromona y se realizan los aportes comentados:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \sum_{k=1}^m \Delta \tau_{rs}^k$$

Considerando $\rho=0.2$, la matriz de feromona quedaría:

$\tau =$

0.00	45.58	13.99	23.99	25.23	33.73
-	0.00	35.03	33.73	8.00	23.18
-	-	0.00	24.74	52.72	16.04
-	-	-	0.00	35.03	35.03
-	-	-	-	0.00	79.54
-	-	-	-	-	0.00

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- El Sistema de Hormigas
- Ejemplo de aplicación del SH al TSP
- **Otros algoritmos de OCH**
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS ELITISTA

- En el mismo trabajo en el que propusieron el SH, Dorigo y sus colaboradores introdujeron el **Sistema de Hormigas Elitista** (Elitist Ant System) (SHE) para solucionar el problema de la lentitud de convergencia del SH
- La única diferencia entre ambos es la **regla de actualización** que aplica un **refuerzo adicional de los buenos arcos**:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \sum_{k=1}^m \Delta \tau_{rs}^k + e \cdot \Delta \tau_{rs}^{mejor_global}$$

donde e es el número de hormigas elitistas consideradas y *mejor_global* es la mejor solución obtenida hasta el momento.

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS

M. Dorigo, L.M. Gambardella, *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*. IEEE Trans. on Evolutionary Computation, Vol. 1, n° 1, 1997, 53-66

- El Sistema de Colonia de Hormigas (Ant Colony System) (SCH) extiende a su predecesor, el SH, en tres aspectos:
 - La **regla de transición** establece un equilibrio entre la exploración de nuevos arcos y la explotación de la información acumulada
 - Para la **actualización (global) de feromona** sólo se considera la hormiga que generó la mejor solución hasta ahora. Sólo se evapora feromona en los arcos que componen ésta
 - Se añade una nueva **actualización (local) de feromona** basada en que cada hormiga modifica automáticamente la feromona de cada arco que visita para diversificar la búsqueda

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS

- La regla de transición de SCH (regla proporcional pseudo-aleatoria) es:

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{ [\tau_{ru}]^\alpha \cdot [\eta_{ru}]^\beta \}, & \text{si } q \leq q_0 \\ S, & \text{en otro caso} \end{cases}$$

- s es la ciudad escogida por la hormiga k en su siguiente movimiento
- q es un uniforme en $[0,1]$
- $q_0 \in [0,1]$ es la probabilidad con la que se escoge determinísticamente el arco más prometedor (**explotación**)
- S es una ciudad aleatoria seleccionada según la regla de transición del SH (**exploración dirigida**)

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS

- La actualización global de feromona se realiza con:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \rho \cdot \Delta\tau_{rs}^{mejor_global}$$

que sólo se aplica sobre los arcos $\tau_{rs} \in S_{mejor_global}$

- Sólo la hormiga que generó la mejor solución hasta el momento modifica los niveles de feromona
- El aporte de feromona es función de la calidad de esta solución ($C(S_{mejor_global})$). En problemas de minimización como el TSP:

$$\Delta\tau_{rs}^{mejor_global} = \frac{1}{C(S_{mejor_global})}$$

En maximización:

$$\Delta\tau_{rs}^{mejor_global} = C(S_{mejor_global})$$

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS

- Cada vez que una hormiga recorre un arco, aplica la actualización local de feromona:

$$\tau_{rs}(t) = (1 - \varphi) \cdot \tau_{rs}(t-1) + \varphi \cdot \tau_0$$

- Con esta operación, la feromona asociada a un arco disminuye cada vez que lo visita una hormiga
- Los arcos ya visitados van siendo menos prometedores según los recorren más hormigas en la iteración actual, lo que favorece la exploración de arcos no visitados
- Así, las hormigas tienden a no converger a soluciones parecidas en la iteración actual

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS

ALGORITMO SISTEMA DE COLONIA DE HORMIGAS

- Inicialización de parámetros (p.e., $\tau_{rs} \leftarrow \tau_0$)
- *Para It=1 hasta Número_de_Iteraciones hacer*
 1. *Para k=1 hasta Número_de_Hormigas hacer*
 - L[k][1] \leftarrow nodo inicial
 - /* Construcción de soluciones por las hormigas */*
 2. *Para i=2 hasta Número_de_Nodos hacer*
 - Para k=1 hasta Número_de_Hormigas hacer*
 - L[k][i] \leftarrow Regla_transición (L[k], τ , η)
 - Actualización_local_feromona* (L[k][i-1],L[k][i])

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS

/ Actualización global de feromona */*

3. Para $k=1$ hasta Número_de_Hormigas hacer

Coste[k] \leftarrow C(L[k])

Mejor_Actual \leftarrow Mejor (L[k])

4. Si C(Mejor_Actual) es mejor que C(Mejor_Global) entonces Mejor_Global \leftarrow Mejor_Actual

5. Para $i=1$ hasta Número_de_Nodos hacer

*Actualización_global_feromona (Mejor_Global[k][i],
Mejor_Global [k][i+1], C(Mejor_Solución))*

▪ *DEVOLVER* Mejor_Global

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MAX-MIN

T. Stützle, H.H. Hoos, MAX-MIN Ant System. Future Generation Computer Systems, Vol. 16, nº 8, 2000, 889-914

- El Sistema de Hormigas Max-Min (Max-Min Ant System) (SHMM) es una nueva extensión del SH con una mayor explotación de las mejores soluciones y un mecanismo adicional para evitar el estancamiento de la búsqueda
- Mantiene la regla de transición del SH y cambia:
 - El mecanismo de actualización es más agresivo al evaporar todos los rastros y aportar sólo en los de la mejor solución
 - Define unos topes mínimo y máximo para los rastros de feromona
 - Reinicializa la búsqueda cuando se estanca

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MAX-MIN (2)

- En la actualización de feromona sólo se considera la mejor solución, ya sea de la iteración actual o la global:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \Delta \tau_{rs}^{mejor}$$

donde *mejor* es $S_{mejor-global}$ o $S_{mejor-actual}$

- Se evaporan todos los rastros de feromona

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MAX-MIN (3)

- Se establecen unos límites en los valores posibles de feromona en los rastros:

$$\tau_{\min} \leq \tau_{rs}(t) \leq \tau_{\max}$$

Los límites se calculan de forma heurística

- Todos los rastros de feromona se inicializan al máximo valor τ_{\max} , en lugar de a un valor pequeño τ_0
- Así, al aplicar la regla de actualización los arcos de las buenas soluciones mantienen valores altos mientras que los de las malas reducen el valor de sus rastros
- También da lugar a una mayor exploración al comienzo de la ejecución del algoritmo

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MAX-MIN (4)

- Además, cuando se detecta que la búsqueda está estancada, se aplica una **reinicialización**, volviendo a poner todos los rastros de feromona a τ_{\max}
- La combinación de todo lo anterior:
 - establece un buen balance exploración-explotación y
 - reduce la posibilidad de estancamiento de la búsqueda
- El SHMM es hoy en día el algoritmo de OCH más competitivo

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MEJOR-PEOR

O. Cordón, F. Herrera, Ll. Moreno, *Integración de Conceptos de Computación Evolutiva en un Nuevo Modelo de Colonia de Hormigas*. Actas de la Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA'99), 1999, pp. 98-104.

O. Cordón, I. Fernández de Viana, F. Herrera, Ll. Moreno, *A New ACO Model Integrating Evolutionary Computation Concepts: The Best-Worst Ant System*. Actas de ANTS'2000 - From Ant Colonies to Artificial Ants: Second International Workshop on Ant Algorithms, 2000, pp. 22-29.

- El Sistema de Hormigas Mejor-Peor (Best-Worst Ant System) (SHMP) es otra extensión del SH basada en la incorporación de componentes de Computación Evolutiva para mejorar el equilibrio intensificación-diversificación
- Mantiene la regla de transición del SH y cambia:
 - El mecanismo de actualización es más explotativo al evaporar todos los rastros, reforzar positivamente sólo los de la mejor solución global y negativamente los de la peor solución actual
 - Aplica una mutación de los rastros de feromona para diversificar
 - Reinicializa la búsqueda cuando se estanca

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MEJOR-PEOR (2)

- La actualización de feromona de la mejor y la peor hormiga se realiza en dos pasos:

1. Se evaporan todos los rastros de feromona y se aporta en los de la mejor solución global:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \Delta \tau_{rs}^{mejor_global}$$

2. Se realiza una evaporación adicional de los rastros de feromona de la peor solución de la iteración actual que no estén contenidos en la mejor global:

$$\tau_{rs}(t) \leftarrow (1 - \rho) \cdot \tau_{rs}(t), \quad \forall a_{rs} \in S_{peor-actual} \quad y \quad a_{rs} \notin S_{mejor-global}$$

- El refuerzo negativo de $S_{peor-actual}$ hace que la regla de actualización tenga un comportamiento más intensificativo

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MEJOR-PEOR (3)

- El SHMP considera la búsqueda estancada si durante un número consecutivo de iteraciones (un porcentaje del total) no se consigue mejorar la mejor solución global obtenida
- En ese caso, se aplica la **reinicialización**, volviendo a poner todos los rastros de feromona a τ_0

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MEJOR-PEOR (4)

- Para conseguir diversidad en el proceso de búsqueda se mutan los valores de los rastros de feromona
- La **mutación** se aplica en cada rastro de feromona con probabilidad P_m :

$$\tau'_{rs}(t) = \tau_{rs}(t) + N(0, \tau_{umbral}) \quad ; \quad \tau_{umbral} = \frac{\sum_{a_{rs} \in S_{mejor-global}} \tau_{rs}(t)}{n}$$

A cada rastro mutado se le añade un valor normal de media 0 en $[-\tau_{umbral}, \tau_{umbral}]$. τ_{umbral} corresponde a la media de los rastros de feromona de $S_{mejor-global}$

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MEJOR-PEOR (5)

- Esta función de mutación se caracteriza por:
 1. La fuerza de la mutación aumenta con las iteraciones: primero, τ_{umbral} es cercano a τ_0 y la mutación es pequeña. Luego, según crecen los rastros de $S_{mejor-global}$ va siendo más grande
 2. Al reinicializar, vuelve a su rango inicial

- El SMPH consigue un buen balance diversificación-intensificación combinando:
 - La intensificación que introduce la regla de actualización de feromona con la mejor y la peor hormiga
 - La diversificación de la mutación de rastros de feromona y la reinicialización

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- El Sistema de Hormigas
- Ejemplo de aplicación del SH al TSP
- Otros algoritmos de OCH
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

ESTUDIO COMPARATIVO DE LA OCH EN EL TSP

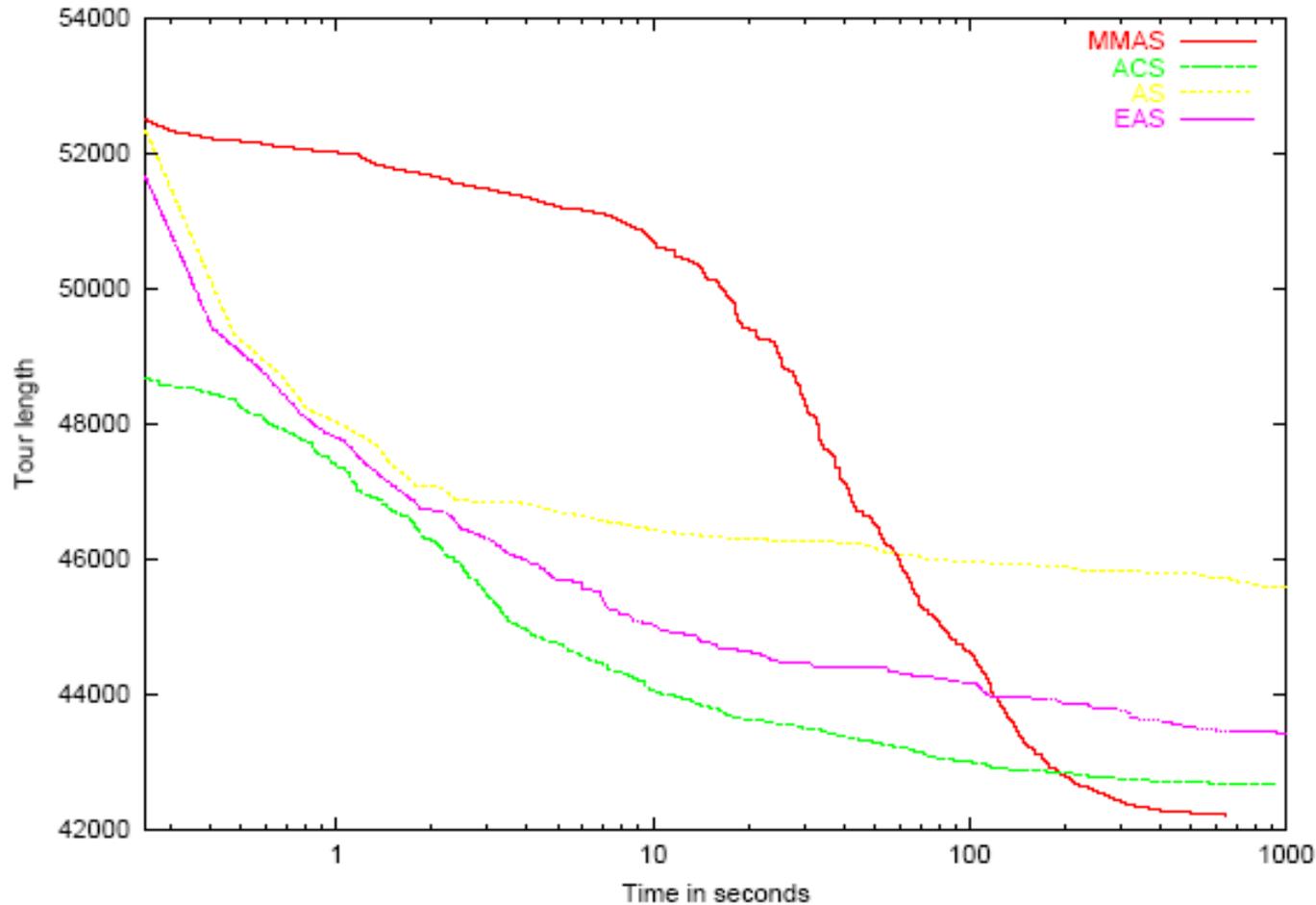
Ejemplo de valores de parámetros para la OCH en el TSP:

- Número de hormigas: $m = 10$
- Nivel inicial de feromona: $\tau_0 = 1/(n \cdot C(S_{\text{greedy}}))$
(S_{greedy} = solución obtenida por un algoritmo greedy)
- Regla de transición: $\alpha=1, \beta=2, q_0=0.98$ (SCH)
- Regla de actualización de feromona: $\rho=0.1$
- Regla de actualización local (SCH): $\varphi=0.1$
- Criterio de parada: 10000·n iteraciones para TSP simétrico y 20000·n para el TSP asimétrico

ESTUDIO COMPARATIVO DE LA OCH EN EL TSP (2)

Instance	<i>opt</i>	<i>MMAS</i>	ACS	<i>AS_e</i>	AS
eil51.tsp	426	427.6	428.1	428.3	437.3
kroA100.tsp	21282	21320.3	21420.0	21522.83	22471.4
d198.tsp	15780	15972.5	16054.0	16205.0	16705.6
lin318.tsp	42029	42220.2	42570.0	43422.8	45535.2
ry48p.atsp	14422	14553.2	14565.4	14685.2	15296.4
ft70.atsp	38673	39040.2	39099.0	39261.8	39596.3
kro124p.atsp	36230	36773.5	36857.0	37510.2	38733.1
ftv170.atsp	2755	2828.8	2826.5	2952.4	3154.5

ESTUDIO COMPARATIVO DE LA OCH EN EL TSP (3)



2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- El Sistema de Hormigas
- Ejemplo de aplicación del SH al TSP
- Otros algoritmos de OCH
- Estudio comparativo de la OCH en el TSP
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL

- Existe la posibilidad de hibridar los algoritmos de OCH con técnicas de búsqueda local para mejorar su eficacia
- La hibridación consiste en aplicar una búsqueda local sobre las soluciones construidas por todas las hormigas en cada iteración antes de actualizar la feromona
- El aumento en la eficacia reduce la eficiencia. Es habitual emplear la búsqueda local junto con las llamadas Listas de Candidatos, que consisten en estudiar sólo las ciudades candidatas más prometedoras en cada paso de la hormiga
- Al usar búsqueda local se obtienen muy buenos resultados con muchas menos hormigas que en la OCH básica (lo que implica mayor rapidez al hacer menos búsquedas locales)

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (2)

- Además, se suelen considerar optimizaciones en el cálculo de la función objetivo dentro de la búsqueda local para acelerar la ejecución
- Por ejemplo, las búsquedas locales habitualmente usadas en el TSP son la 2-opt y la 3-opt (que intercambian el orden de 2 y 3 dos ciudades, es decir, de 4 y 8 arcos, respectivamente)
- Así, en la 2-opt, la diferencia de costes entre la solución actual S y la vecina S' se puede calcular a partir de los cuatro arcos cambiados como:

$$\begin{aligned} F(S') = & F(S) - D(S[i-1],S[i]) - D(S[i],S[i+1]) \\ & - D(S[j-1],S[j]) - D(S[j],S[j+1]) \\ & + D(S[i-1],S[j]) + D(S[j],S[i+1]) \\ & + D(S[j-1],S[i]) + D(S[i],S[j+1]) \end{aligned}$$

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (3)

- El algoritmo básico de OCH con búsqueda local es:

Mientras (no se dé la condición de parada) *hacer*

- Construcción probabilística de las soluciones preliminares mediante la colonia de hormigas
 - Refinamiento de dichas soluciones mediante la búsqueda local
 - Actualización global de feromona
- De este modo, Los algoritmos de OCH con búsqueda local son algoritmos híbridos con una generación probabilística de soluciones iniciales para una técnica clásica de búsqueda local

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (4)

- Así, pueden encuadrarse en los algoritmos de **Búsqueda Local Múltiarranque**, aunque éstos generan aleatoriamente las soluciones iniciales sin usar información heurística
- También son similares a los algoritmos **GRASP**, aunque con otra filosofía de generación probabilística de soluciones
- La principal diferencia con los dos algoritmos anteriores es el mecanismo de cooperación global de las soluciones generadas que hace que las ejecuciones no sean independientes entre sí

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (5)

EJEMPLO: SCH CON BÚSQUEDA LOCAL PARA EL TSP

Caso TSP	SCH (media)	SCH (mejor)	Otros (media)	Otros (mejor)
d198	15781.7	15780	15780	15780
lin318	42029	42029	42029	42029
att532	27718.2	27693	27693.7	27686
rat783	8837.9	8818	8807.3	8806

Fuente: Bonabeau, Dorigo, Theraulaz, *Swarm Intelligence*, Oxford, 1999

- Nuevos valores de parámetros: $q_0 = 0.9$ y $cl=20$
- Algoritmo de búsqueda local: 3-opt (búsqueda local del mejor)
- Otros = mejores resultados obtenidos en el Primer Concurso Internacional de Optimización Evolutiva (ICEO)
- Resultados medios de 10 ejecuciones

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (6)

EJEMPLO 2: COMPARATIVA DE OCH CON BL PARA EL TSP

<i>Parámetro</i>	<i>Valor</i>
Número de hormigas	$m = 15$
Número de ejecuciones	10
Tiempo máximo	$N_{it} = 900 \text{ a } 3600$
Regla de transición	$\alpha = 1 ; \beta = 2$
Cte. evaporación global	$\rho = 0.2$
<i>SHE</i>	
Hormigas elitistas	$m (15)$
<i>SCH</i>	
Cte. evaporación local	$q_0 = 0.8 \quad \varphi = 0.2$
<i>SMPH</i>	
Prob. de mutación	$P_m = 0.3, 0.25, 0.2$
Iteraciones sin mejora	0.3, 0.2, 0.3
<i>Búsqueda local</i>	
Número de vecinos	40
Regla de selección	primero mejor
Algoritmo	2-opt

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (7)

Problema	Eil51 ($C_{opt} = 426$, $n = 51$)					Berlín52 ($C_{opt} = 7542$), $n = 52$)				
Modelo	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	#R	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	#R
SHE	426	426.2	0.44	0.04	-	7542	7542	0	0	-
SCH	426	426.7	0.48	0.48	-	7542	7542	0	0	-
SHE + Re	426	426	426.6	0.26	0	7542	7542	0	0	0
SCH + Re	426	426.5	0.53	0.12	2.4	7542	7542	0	0	0
SMPH	426	426	0	0	0	7542	7542	0	0	0
Problema	Brazil58 ($C_{opt} = 25395$, $n = 58$)					Kroal100 ($C_{opt} = 21282$), $n = 100$)				
Modelo	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	#R	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	#R
SHE	25395	25395	0	0	-	21282	21282	0	0	-
SCH	25395	25395	0	0	-	21282	21282	0	0	-
SHE + Re	25395	25395	0	0	0	21282	21282	0	0	0
SCH + Re	25395	25395	0	0	0	21282	21282	0	0	0
SMPH	25395	25395	0	0	0	21282	21282	0	0	0
Problema	Gr120 ($C_{opt} = 6942$, $n = 120$)					D198 ($C_{opt} = 15780$), $n = 198$)				
Modelo	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	#R	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	#R
SHE	6942	6942	0	0	-	15780	15781	0.70	$\cong 0$	-
SCH	6942	6946.1	5.49	0.06	-	15780	15784.9	5.67	0.03	-
SHE + Re	6942	6942	0	0	0	15780	15781	0.70	$\cong 0$	0
SCH + Re	6942	6943.8	3.79	0.03	1.1	15780	15782.9	4.31	0.02	2.3
SMPH	6942	6942	0	0	0.5	15780	15780.4	0.51	$\cong 0$	1.7

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (8)

Problema	Lin318 ($C_{opt} = 42029$, $n = 318$)					Pcb442 ($C_{opt} = 50788$), $n = 442$)				
Modelo	Mejor	Media	Dev.	Error	#R	Mejor	Media	Dev.	Error	#R
SHE	42029	42123.8	67.28	0.22	-	51015	51104.6	82.44	0.63	-
SCH	42029	42230	148.48	0.48	-	50919	51048	75.29	0.53	-
SHE + Re	42029	42105	53.79	0.18	3	50980	51066.8	71.52	0.56	3.2
SCH + Re	42029	42182.4	118.12	0.36	5	50860	51147.5	173.11	0.72	8
SMPH	42029	42084.1	98.60	0.13	2	50788	50888.8	87.83	0.21	6.9
Problema	Att532 ($C_{opt} = 27686$, $n = 532$)					Rat783 ($C_{opt} = 8806$), $n = 783$)				
Modelo	Mejor	Media	Dev.	Error	#R	Mejor	Media	Dev.	Error	#R
SHE	27745	27823	70.67	0.49	-	8860	8878.6	17.06	0.81	-
SCH	27705	27810.3	64.44	0.45	-	8857	8892.7	20.93	0.97	-
SHE + Re	27793	27825.6	41.59	0.50	3.6	8843	8878.4	32.25	0.81	3.8
SCH + Re	27745	27835	57.56	0.54	7	8875	8899.5	22.33	1.05	7.6
SMPH	27686	27711	12.16	0.09	7.7	8816	8833.4	15.37	0.31	9.5
Problema	U1060 ($C_{opt} = 224094$, $n = 1060$)					D1291 ($C_{opt} = 50801$), $n = 1291$)				
Modelo	Mejor	Media	Dev.	Error	#R	Mejor	Media	Dev.	Error	#R
SHE	231644	232145.8	297.91	3.46	-	51210	51347.2	138.15	1.06	-
SCH	225675	226387.8	668.9	1.01	-	51901	51953.6	60.01	2.21	-
SHE + Re	230360	230806.4	390.8	2.90	2.9	51073	51236.7	119.56	0.85	4.5
SCH + Re	225243	226501	1059.57	1.06	2	51828	51986.8	105.80	2.28	5
SMPH	225310	225721.1	476.88	0.72	5.3	50890	50986.3	74.87	0.36	17.2

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (9)

- Por otro lado, el MMAS con la búsqueda local 3-opt:
 - obtiene el óptimo para los casos de hasta 500 ciudades en pocos segundos/minutos
 - en los casos grandes (se ha probado en instancias de hasta 3038 ciudades), se obtienen soluciones con un error de un 0.25% con respecto al óptimo en tiempo razonable (menos de 1 hora)
- El rendimiento es cercano al de los algoritmos state-of-the-art para el TSP

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- El Sistema de Hormigas
- Ejemplo de aplicación del SH al TSP
- Otros algoritmos de OCH
- Estudio comparativo de la OCH en el TSP
- Algoritmos de OCH con búsqueda local
- **La metaheurística OCH**

LA METAHEURÍSTICA DE OPTIMIZACIÓN BASADA EN COLONIAS DE HORMIGAS

- A posteriori, cuando la mayoría de los algoritmos de OCH estaban ya propuestos, Dorigo y Di Caro propusieron un marco de trabajo general que define la Metaheurística OCH:

M. Dorigo, G. Di Caro, Ant Algorithms for Discrete Optimization. Artificial Life, Vol. 5, n° 2, 1999, 137-172

- Instanciando el algoritmo general con componentes concretas (reglas de transición, actualización, ...), se pueden obtener distintas variantes de algoritmos OCH
- Los distintos algoritmos se pueden implementar de forma **secuencial** o **paralela** para ser aplicados respectivamente a **problemas estáticos** o **dinámicos**

LA METAHEURÍSTICA DE OPTIMIZACIÓN BASADA EN COLONIAS DE HORMIGAS (2)

- En general, para aplicar la OCH a un problema, es necesario que pueda ser representado en forma de grafo con pesos
- Cada arco a_{rs} del grafo contendrá dos tipos de información:
 - Información heurística: **preferencia heurística** del arco, η_{rs} , dependiente del caso concreto del problema. **Las hormigas no la modifican durante la ejecución del algoritmo, aunque puede variar a lo largo del tiempo en problemas dinámicos**
 - Información memorística: medida de la **“deseabilidad”** del arco, τ_{rs} , representada por la cantidad de **feromona** depositada en él y **modificada durante el algoritmo**

LA METAHEURÍSTICA DE OPTIMIZACIÓN BASADA EN COLONIAS DE HORMIGAS (3)

ESTRUCTURA GENERAL DE UN ALGORITMO DE OCH

1. Inicialización de parámetros (p.e., asignación de una cantidad inicial de feromona τ_0 a cada arco)
2. *Planificación_de_actividades*
 - generación_y_activación_de_hormigas
 - evaporación_de_feromona
 - acciones_del_demonio (**OPCIONAL**)

El constructor **Planificación_de_actividades** define la planificación de las tres tareas, ya sea secuencialmente (problemas estáticos) o en paralelo (problemas dinámicos)

LA METAHEURÍSTICA DE OPTIMIZACIÓN BASADA EN COLONIAS DE HORMIGAS (4)

ESTRUCTURA GENERAL DE UN ALGORITMO DE OCH

1. Inicialización de parámetros (p.e., asignación de una cantidad inicial de feromona τ_0 a cada arco)
2. *Planificación de actividades*
 - generación_y_activación_de_hormigas
 - evaporación_de_feromona
 - acciones_del_demonio (**OPCIONAL**)

El **demonio** es un elemento opcional que realiza una serie de funciones adicionales centralizadas sin contrapunto natural (aplicar búsqueda local, ...)

3. APLICACIONES DE LA OCH

- Aplicación de la OCH al Enrutamiento en Redes de Telecomunicaciones: AntNet
- Planificación de rutas para el transporte de mercancías: Ant@ptima
- Otras Aplicaciones

APLICACIÓN AL ENRUTAMIENTO DE PAQUETES EN REDES DE TELECOMUNICACIONES: ANTNET

G. Di Caro, M. Dorigo, *AntNet: Distributed Stimergetic Control for Communication Networks*. Journal of Artificial Intelligence Research, Vol. 9, 1998, 317-365

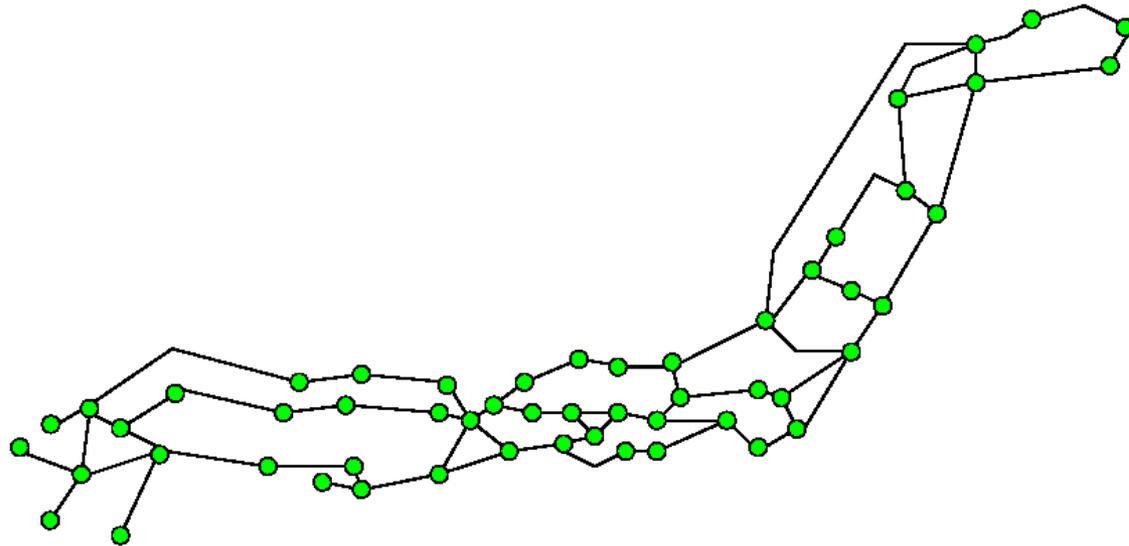
- El **enrutamiento** es la tarea consistente en determinar el camino que seguirán los paquetes en una red de telecomunicaciones cuando llegan a un nodo para alcanzar su nodo destino de la forma más rápida posible
- **AntNet** es un algoritmo de hormigas adaptativo y distribuido para enrutamiento de paquetes en redes

APLICACIÓN AL ENRUTAMIENTO DE PAQUETES EN REDES DE TELECOMUNICACIONES: ANTNET (2)

- Las redes se modelan mediante un grafo dirigido con N nodos de procesamiento/destino
- Los arcos del grafo están caracterizados por el ancho de banda (bits/segundo) y el retardo de transmisión (segundos) del enlace físico
- Se consideran dos tipos de paquetes: enrutamiento y datos. Los de enrutamiento tienen una mayor prioridad

APLICACIÓN AL ENRUTAMIENTO DE PAQUETES EN REDES DE TELECOMUNICACIONES: ANTNET (3)

- Una de las redes consideradas, la NNTnet de Japón:



APLICACIÓN AL ENRUTAMIENTO DE PAQUETES EN REDES DE TELECOMUNICACIONES: ANTNET (4)

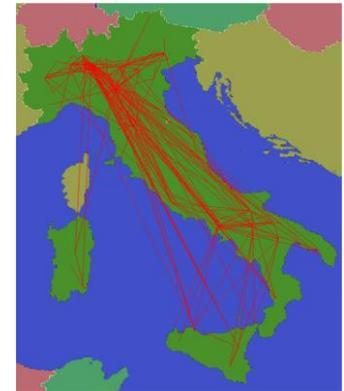
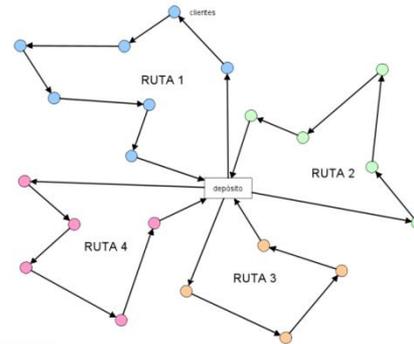
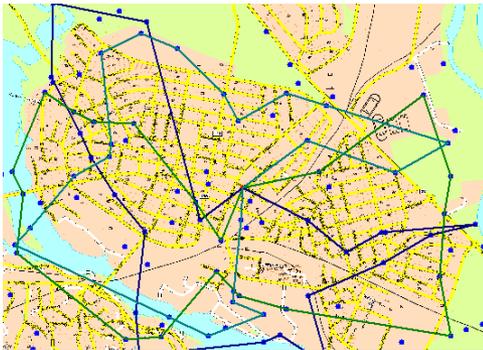
- Las hormigas (paquetes de enrutamiento) se lanzan asíncronamente a la red hacia nodos destino aleatorios
- Cada hormiga busca un camino de coste mínimo entre su nodo de partida y su nodo destino
- Se mueve paso a paso por la red (grafo). En cada nodo intermedio, lanza la regla de transición para decidir a qué nodo se dirige
- Para ello, considera la feromona (almacenado en los nodos y función del tiempo consumido en el envío de los paquetes) y la preferencia heurística (dependiente del estado actual) de los enlaces de la red

APLICACIÓN AL ENRUTAMIENTO DE PAQUETES EN REDES DE TELECOMUNICACIONES: ANTNET (5)

- El estado de la red varía con el tiempo (caída de enlaces, congestión, ...). El algoritmo maneja adecuadamente este hecho gracias a su naturaleza distribuida y su capacidad de adaptación
- Cuando la hormiga llega al nodo destino, vuelve sobre sus pasos y actualiza las tablas de enrutamiento de los nodos de acuerdo al tiempo que tardó en hacer el camino (**refuerzo positivo o negativo**)
- En un estudio experimental en el que se comparó su funcionamiento contra el de seis algoritmos de enrutamiento diferentes, AntNet proporcionó el mejor comportamiento

Planificación de Rutas para Transporte de Mercancías

- Hoy en día es difícil encontrar empresas que gestionen las **operaciones de logística** sin la ayuda del ordenador
- El problema típico es diseñar las rutas más adecuadas de transporte/recogida de productos entre un almacén central y unos destinos dispersos geográficamente



- Su resolución de forma adecuada puede suponer **ahorros muy significativos para la empresa**

Planificación de Rutas para Transporte de Mercancías

- Esta tarea se lleva a cabo empleando una flota de vehículos pertenecientes o no a la empresa



- Un **sistema de planificación de vehículos** debe proporcionar un conjunto de **rutas de reparto** a los conductores
- Las mercancías deben ser entregadas cuándo y donde se requieran, con el mínimo coste posible y verificando todas las restricciones legales y políticas de la empresa
- Los **algoritmos de hormigas (AntRoute)** son una herramienta muy potente para la planificación de rutas



Ant@ptima

<http://www.antoptima.com>

Planificación de Rutas para Transporte de Mercancías

- **AntRoute** planifica diariamente las rutas de reparto desde el almacén central de **Migros**, una gran cadena suiza con 600 supermercados, localizado en **Suhr (AG)**, a toda **Suiza**
- Migros dispone de una **flota de entre 150 y 200 vehículos** con tres tamaños: camiones (capacidad de 17 palés), trailers (35 palés) y unidades tractoras (33 palés)
- Esto provoca restricciones de acceso a los almacenes de los supermercados, restricciones de uso de ciertas carreteras, ...
- Los repartos tienen de realizarse a horas específicas, todos ellos en un solo día (**productos perecederos**) y el último tiene que hacerse lo más lejos posible del almacén (**servicios extra**)



Ant@ptima

<http://www.antoptima.com>

- Por ejemplo, en un reparto de 52000 palés a 6800 clientes en un periodo de 20 días, AntRoute obtuvo el diseño diario de rutas en menos de 5 minutos en un PC estándar
- Los expertos de la empresa necesitaron tres horas...
- Las soluciones de AntRoute fueron de mucha mejor calidad en cuanto al número de rutas necesario, la distancia total recorrida y al aprovechamiento de los vehículos:

	Human Planner	AR-RegTW	AR-Free
Total number of tours	2056	1807	1614
Total km	147271	143983	126258
Average truck loading	76.91%	87.35%	97.81%

OTRAS APLICACIONES DE LA OCH

- Los algoritmos de OCH se han aplicado a otros muchos problemas:
 - Secuenciación de Tareas
 - Coloreo de Grafos
 - Enrutamiento de Vehículos
 - Ordenación Secuencial
 - “Pooling” de vehículos
 - Líneas de producción de coches
 - Problemas de Agrupamiento (Clustering)
 - Aprendizaje de Reglas Clásicas y Difusas
 - Bioinformática: plegado de proteínas 2D
 - etc.

4. COMENTARIOS FINALES



- La OCH es una metaheurística bioinspirada que permite diseñar algoritmos:
 - Sencillos de entender
 - Rápidos
 - Con buen rendimiento

para problemas de optimización que se puedan representar en forma de grafo con pesos

4. COMENTARIOS FINALES (2):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

DEFINICIÓN DE LOS RASTROS DE FEROMONA

- Tarea totalmente dependiente del problema a resolver
- Depende directamente de la representación de soluciones considerada y de su significado
- Una mala elección provoca un mal rendimiento del algoritmo de OCH

4. COMENTARIOS FINALES (3):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

BALANCE EXPLORACIÓN-EXPLOTACIÓN

- El algoritmo de OCH tendrá el mejor rendimiento si se consigue un balance adecuado para el problema
- En OCH, son los rastros de feromona los que establecen ese equilibrio definiendo una distribución de probabilidad para muestrear el espacio de búsqueda
- Componentes de OCH para intensificación:
 - Actualización de feromona por la calidad de las soluciones
 - Estrategias “elitistas”
 - Regla de transición pseudo-aleatoria del SCH

4. COMENTARIOS FINALES (4):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

- Componentes de OCH para diversificación:
 - Construcción aleatoria de soluciones por las hormigas
 - Mecanismos para evitar el estancamiento de la búsqueda:
 - Regla de actualización local del SCH
 - Límites a los valores de feromona en el SHMM/MMAS
 - Mutación de los rastros de feromona en el SHMP
 - Reinicialización de los rastros de feromona

- También es importante para este balance el papel de los pesos α y β de la regla de transición

4. COMENTARIOS FINALES (5):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

OCH y BÚSQUEDA LOCAL

- En muchos problemas NP-duros, la inclusión de optimizadores locales en el algoritmo de OCH consigue el mejor rendimiento
- Los dos enfoques son complementarios, convirtiendo la OCH en un algoritmo de búsqueda multiarranque
- Sin embargo, la OCH puede ser muy efectiva en aquellos problemas en los que no es sencillo aplicar una búsqueda local

4. COMENTARIOS FINALES (6):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

USO DE LA INFORMACIÓN HEURÍSTICA

- Ayuda a explotar el conocimiento específico del problema a resolver
- Distinción entre información heurística en problemas estáticos y dinámicos. La primera es más efectiva
- En algunos problemas, no es posible usarla y el algoritmo de OCH puede obtener buenos resultados guiado sólo por los rastros de feromona (Ejemplo: QAP)
- Su importancia disminuya cuando se usa OCH con búsqueda local

4. COMENTARIOS FINALES (7):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

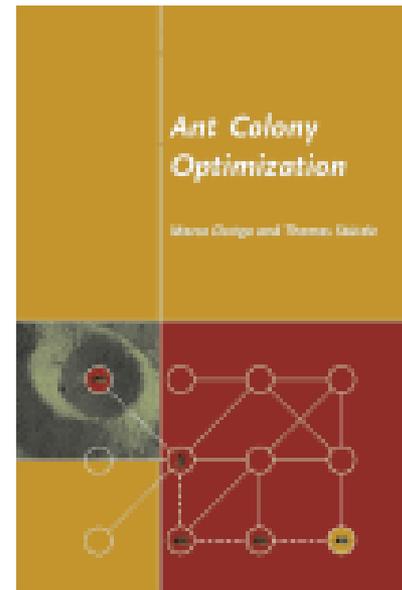
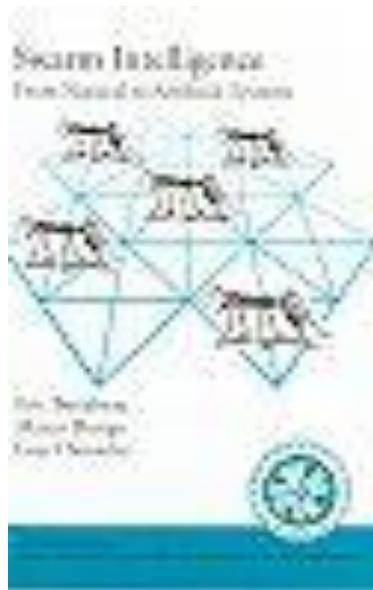
NÚMERO DE HORMIGAS EN LA COLONIA

- ¿Por qué hay que usar una colonia de hormigas y no una sola?
- Hay un algoritmo que maneja una única hormiga: el Sistema de Hormigas Rápido (Fast Ant System)
- Experimentalmente se ha demostrado que el uso de varias hormigas permite obtener mejores resultados
- Aún así, los algoritmos de OCH son robustos en cuanto al número de hormigas considerado, es decir, obtiene buenos resultados independientemente de este valor

5. BIBLIOGRAFÍA Y RECURSOS

BIBLIOGRAFÍA BÁSICA

- E. BONABEAU, M. DORIGO, G. THERAULAZ, **Swarm Intelligence. From Natural to Artificial Systems**, Oxford University Press, 1999.



- M. DORIGO, T. STÜTZLE, **Ant Colony Optimization**, The MIT Press, 2004.

5. BIBLIOGRAFÍA Y RECURSOS (2)

BIBLIOGRAFÍA BÁSICA

- M. DORIGO, G. DI CARO, *Ant Algorithms for Discrete Optimization*, **Artificial Life** 5:2, 1999, pp. 137-172.
- M. DORIGO, T. STÜTZLE, *The Ant Colony Optimization Metaheuristic: Algorithms, Applications and Advances*, En: F. Glover, T. Kochenberger (Eds.), **Handbook of Metaheuristics**, pp. 251-285, 2003, Kluwer.
- O. CORDÓN, F. HERRERA, T. STÜTZLE, *A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends*, *Mathware & Soft Computing* 9:2-3, 2002, pp. 141-175.

<http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html>

5. BIBLIOGRAFÍA Y RECURSOS (3)

BIBLIOGRAFÍA DE APOYO

- M. DORIGO, V. MANIEZZO, A. COLORNI, *The Ant System: Optimization by a Colony of Cooperating Agents*, **IEEE Transactions on Systems, Man, and Cybernetics** 26:1, 1996, pp. 29-41.
- M. DORIGO, L.M. GAMBARDELLA, *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*, **IEEE Transactions on Evolutionary Computation** 1:1, 1997, pp. 53-66.

METAHEURÍSTICAS

TEMA 6. METAHEURÍSTICAS BASADAS EN ADAPTACIÓN SOCIAL

1. INTRODUCCIÓN A LOS MODELOS BASADOS EN ADAPTACIÓN SOCIAL: SWARM INTELLIGENCE
2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS
3. OPTIMIZACIÓN BASADA EN NUBES DE PARTÍCULOS (PARTICLE SWARM OPTIMIZATION)
4. CONCLUSIONES

Bibliografía

E. Bonabeau, M. Dorigo, G. Theraulaz. *Swarm Intelligence. From Nature to Artificial Systems*. Oxford University Press, 1999.

Kennedy, J., Eberhart, R. C., and Shi, Y.. *Swarm intelligence*. San Francisco: Morgan Kaufmann Publishers, 2001.

METAHEURÍSTICAS

OPTIMIZACIÓN BASADA EN NUBES DE PARTÍCULAS (PARTICLE SWARM)

- 1. INTRODUCCIÓN Y RÁPIDO RESUMEN**
- 2. FUNCIONAMIENTO DEL ALGORITMO PSO**
- 3. ASPECTOS AVANZADOS**
- 4. APLICACIONES Y RECURSOS ELECTRÓNICOS**

Kennedy, J., Eberhart, R.C. Swarm Intelligence. Morgan Kauffmann, 2001.

1. INTRODUCCIÓN Y RÁPIDO RESUMEN

- La “Particle Swarm Optimization” (PSO) es una metaheurística poblacional inspirada en el comportamiento social del vuelo de las bandadas de aves y el movimiento de los bancos de peces.
- La población se compone de varias partículas (*nube de partículas = particle swarm*) que se mueven (“*vuelan*”) por el espacio de búsqueda durante la ejecución del algoritmo.
- Este movimiento de cada partícula p depende de:
 - Su mejor posición desde que comenzó el algoritmo ($pBest$),
 - la mejor posición de las partículas *de su entorno* ($lBest$) o de *toda la nube* ($gBest$) desde que comenzó el algoritmo.

En cada iteración, se cambia aleatoriamente la velocidad de p para acercarla a las posiciones $pBest$ y $lBest/gBest$.

1. INTRODUCCIÓN Y RÁPIDO RESUMEN (2)

- Desarrollo: USA, en 1995.
- Primeros autores: Russ C. Eberhart y James Kennedy
Kennedy, J. and Eberhart, R. (1995). "Particle Swarm Optimization", Proc. 1995 IEEE Intl. Conf. on Neural Networks, pp. 1942-1948, IEEE Press.
- Aplicación típica:
 - Optimización continua (optimización de parámetros reales, numérica).
- Características atribuidas:
 - Asume un intercambio de información (**interacciones sociales**) entre los agentes de búsqueda.
 - **Idea básica**: guardar información del mejor propio y global.
 - Implementación muy sencilla, pocos parámetros.
 - Convergencia rápida a buenas soluciones.

2. FUNCIONAMIENTO DEL ALGORITMO PSO

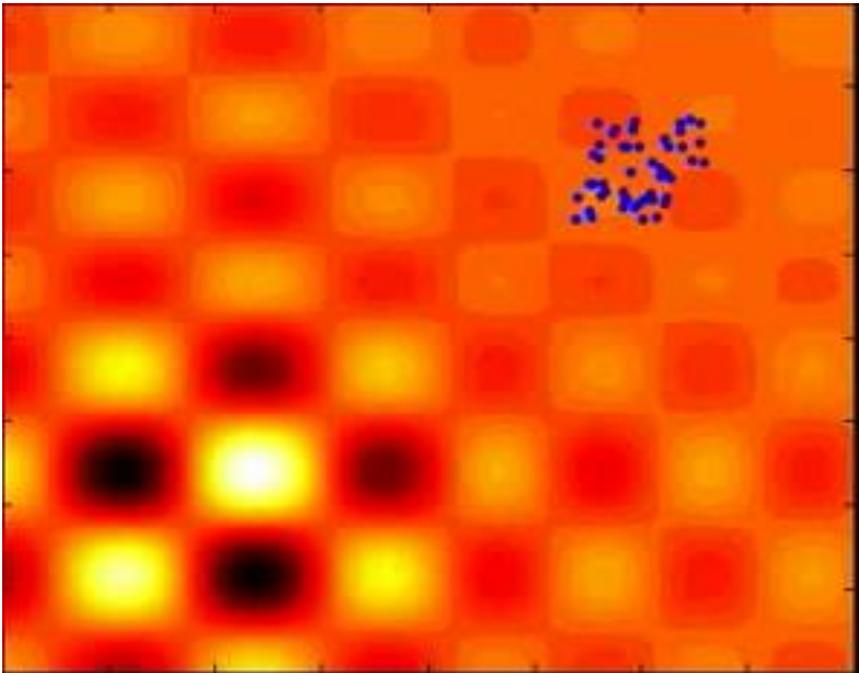
- **FUNCIONAMIENTO BÁSICO**
- **ANATOMÍA DE UNA PARTÍCULA**
- **INICIALIZACIÓN DE LA NUBE DE PARTÍCULAS**
- **MOVIMIENTO DE LAS PARTÍCULAS**
- **PSEUDOCÓDIGOS**
- **VALORES DE LOS PARÁMETROS**
- **TOPOLOGÍAS DE LA NUBE DE PARTÍCULAS**

Funcionamiento Básico

- PSO simula el comportamiento de las bandadas de aves.
- Supongamos que una de estas bandadas busca comida en un área y que solamente hay una pieza de comida en dicha área.
- Los pájaros no saben donde está la comida pero sí conocen su distancia a la misma.
- La estrategia más eficaz para hallar la comida es seguir al ave que se encuentre más cerca de ella.

Funcionamiento Básico (2)

- PSO emula este escenario para resolver problemas de optimización. Cada solución (**partícula**) es un "*ave*" en el espacio de búsqueda que está siempre en continuo movimiento y que nunca muere.



Funcionamiento Básico (2)

- La nube de partículas es un **sistema multiagente**. Las partículas son agentes simples que se mueven por el espacio de búsqueda y que guardan (y posiblemente comunican) la mejor solución que han encontrado.

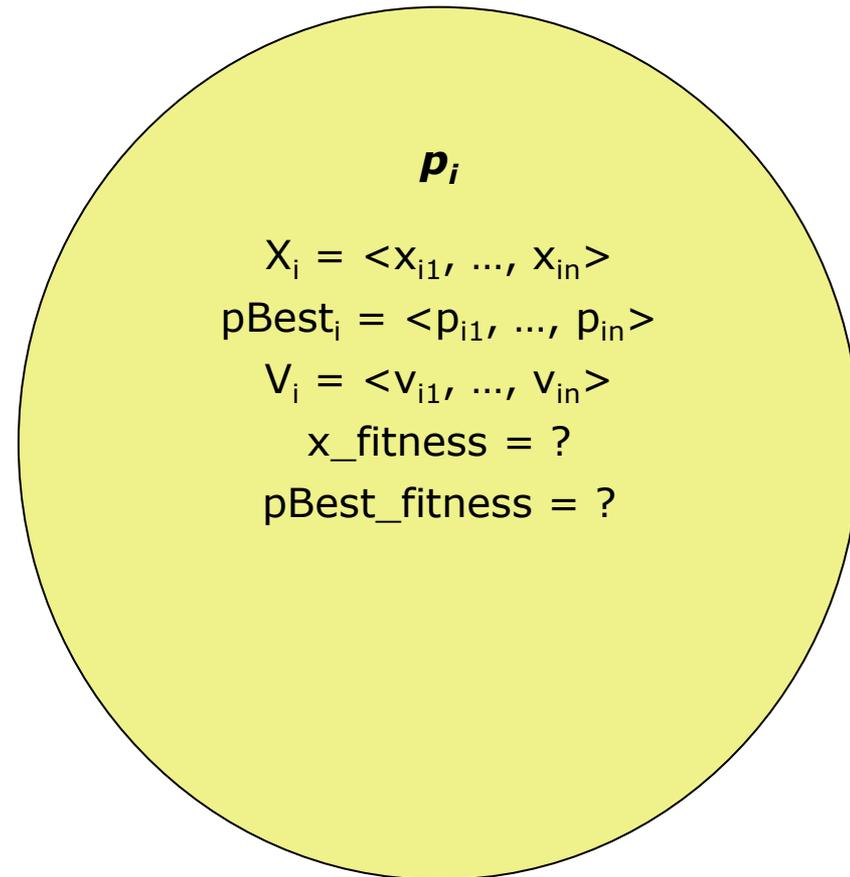


- Cada partícula tiene un **fitness**, una **posición** y un **vector velocidad** que dirige su "vuelo". El movimiento de las partículas por el espacio está guiado por las partículas óptimas en el momento actual.

Anatomía de una Partícula

Una partícula está compuesta por:

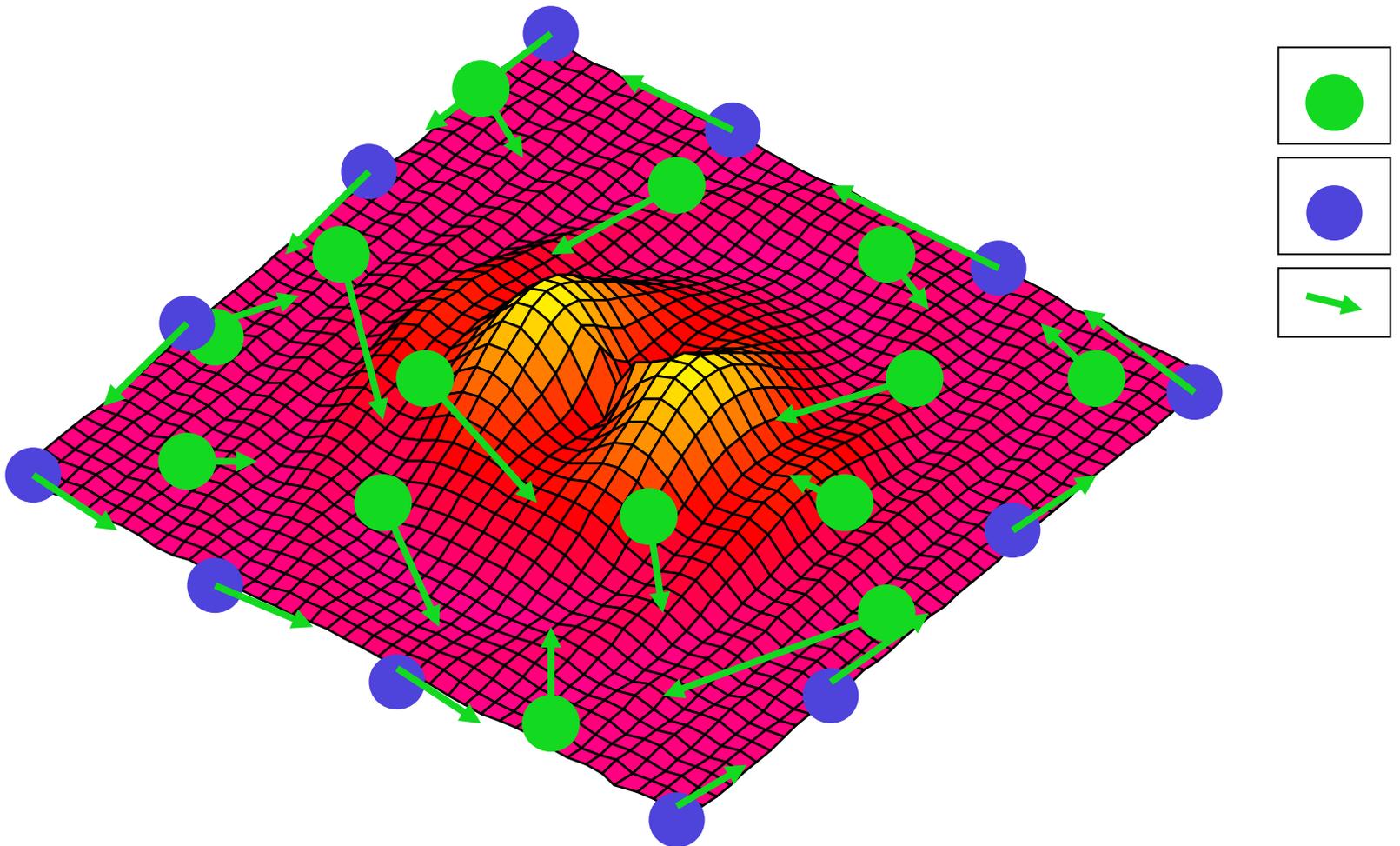
- Tres vectores:
 - El **vector X** almacena la posición actual (localización) de la partícula en el espacio de búsqueda,
 - El **vector pBest** almacena la localización de la mejor solución encontrada por la partícula hasta el momento, y
 - El **vector V** almacena el gradiente (dirección) según el cuál se moverá la partícula.
- Dos valores de fitness:
 - El **x_fitness** almacena el fitness de la solución actual (vector X), y
 - El **p_fitness** almacena el fitness de la mejor solución local (vector *pBest*).



Inicialización de la Nube de Partículas

- La nube se inicializa generando las posiciones y las velocidades iniciales de las partículas.
- Las posiciones se pueden generar aleatoriamente en el espacio de búsqueda, de forma regular, o con una combinación de ambas.
- Las velocidades se generan aleatoriamente, con cada componente en el intervalo $[-V_{\max}, V_{\max}]$.
No es conveniente fijarlas a cero, no se obtienen buenos resultados.
 V_{\max} será la velocidad máxima que pueda tomar una partícula en cada movimiento.

Inicialización de la Nube de Partículas (2)



Movimiento de las Partículas

¿Cómo se mueve una partícula de una posición del espacio de búsqueda a otra?

- Se hace simplemente añadiendo el vector velocidad V_i al vector posición X_i para obtener un nuevo vector posición:

$$X_i \leftarrow X_i + V_i$$

- Una vez calculada la nueva posición de la partícula, se evalúa ésta. Si el nuevo fitness es mejor que el que la partícula tenía hasta ahora, $pBest_fitness$, entonces:

$$pBest_i \leftarrow X_i \quad ; \quad pBest_fitness \leftarrow x_fitness.$$

Movimiento de las Partículas (2)

- De este modo, el primer paso es ajustar el vector velocidad, para después sumárselo al vector posición.
- Las fórmulas empleadas son las siguientes:

$$v_{id} = v_{id} + \underbrace{\varphi_1 \cdot \text{rnd}() (pBest_{id} - x_{id})}_{\text{COGNITIVO}} + \underbrace{\varphi_2 \cdot \text{rnd}() (g_{id} - x_{id})}_{\text{SOCIAL}}$$
$$x_{id} = x_{id} + v_{id}$$

donde:

- p_i es la partícula en cuestión, $pBest_{id}$ es la mejor solución encontrada por la partícula.
- φ_1, φ_2 son ratios de aprendizaje (pesos) que controlan los componentes **cognitivo** y **social**,
- g representa el índice de la partícula con el mejor $pBest_fitness$ del entorno de p_i ($iBest$) o de toda la nube ($gBest$),
- los **rnd()** son números aleatorios generados en $[0,1]$, y
- d es la d -ésima dimensión del vector.

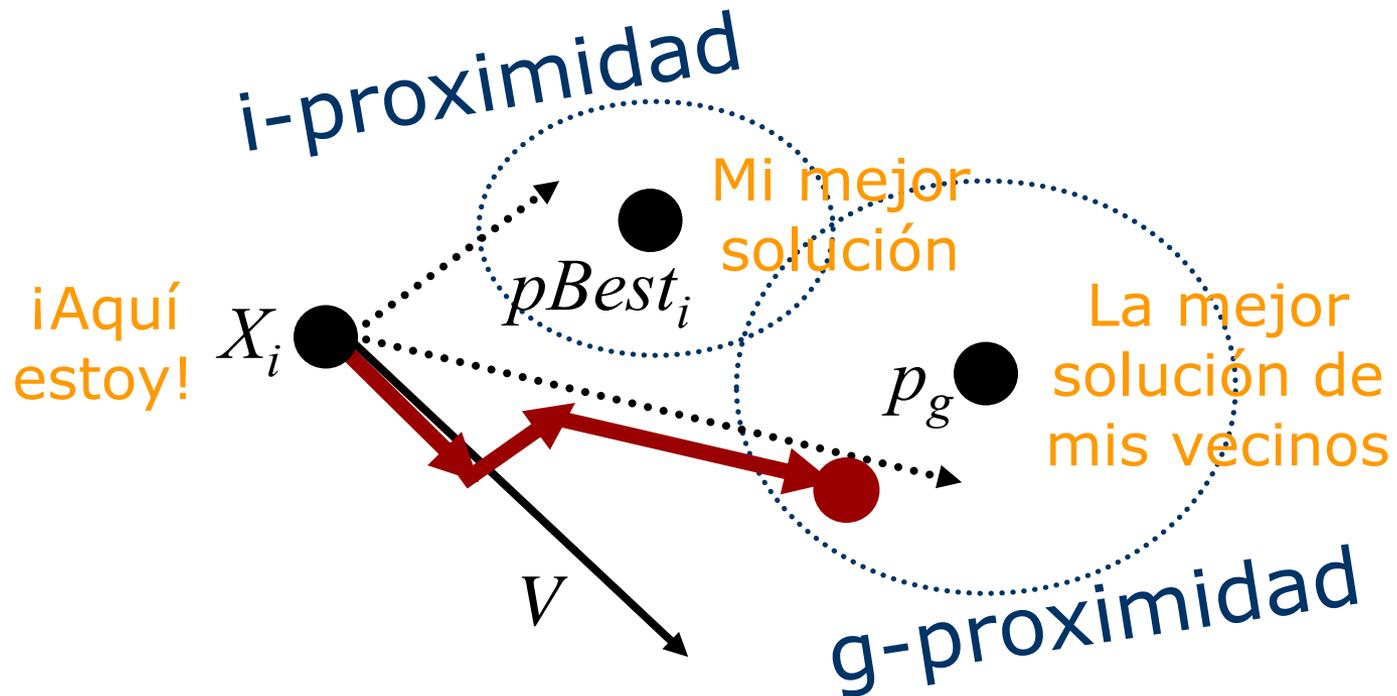
Movimiento de las Partículas (3)

TIPOS DE ALGORITMOS DE PSO:

- Kennedy identifica cuatro tipos de algoritmos de PSO en función de los valores de φ_1 y φ_2 :
 - Modelo completo: $\varphi_1, \varphi_2 > 0$.
 - Sólo Cognitivo: $\varphi_1 > 0$ y $\varphi_2 = 0$.
 - Sólo Social: $\varphi_1 = 0$ y $\varphi_2 > 0$.
 - Sólo Social exclusivo: $\varphi_1 = 0$, $\varphi_2 > 0$ y $g \neq i$ (la partícula en sí no puede ser la mejor de su entorno).

Movimiento de las Partículas (4)

REPRESENTACIÓN GRÁFICA:



Pseudocódigo PSO Local

$t = 0;$

Para $i=1$ hasta Número_partículas
 inicializar X_i y V_i ;

Mientras (no se cumpla la condición de parada) hacer
 $t \leftarrow t + 1$

 Para $i=1$ hasta Número_partículas
 evaluar X_i ;

 Si $F(X_i)$ es mejor que $F(pBest)$ entonces
 $pBest_i \leftarrow X_i$; $F(pBest_i) \leftarrow F(X_i)$

 Para $i=1$ hasta Número_partículas

 Escoger $lBest_i$, la partícula con mejor fitness del entorno de X_i

 Calcular V_i , la velocidad de X_i , de acuerdo a $pBest_i$ y $lBest_i$

 Calcular la nueva posición X_i , de acuerdo a X_i y V_i

Devolver la mejor solución encontrada

Pseudocódigo PSO Global

$t = 0;$

Para $i=1$ hasta Número_partículas
inicializar X_i y V_i ;

Mientras (no se cumpla la condición de parada) hacer
 $t \leftarrow t + 1$

Para $i=1$ hasta Número_partículas

evaluar X_i ;

Si $F(X_i)$ es mejor que $F(pBest)$ entonces

$pBest_i \leftarrow X_i$; $F(pBest_i) \leftarrow F(X_i)$

Si $F(pBest)$ es mejor que $F(gBest)$ entonces

$gBest \leftarrow pBest_i$; $F(gBest_i) \leftarrow F(pBest_i)$

Para $i=1$ hasta Número_partículas

Calcular V_i , la velocidad de X_i , de acuerdo a $pBest_i$ y **$gBest_i$** ;

Calcular la nueva posición X_i , de acuerdo a X_i y V_i

Devolver la mejor solución encontrada

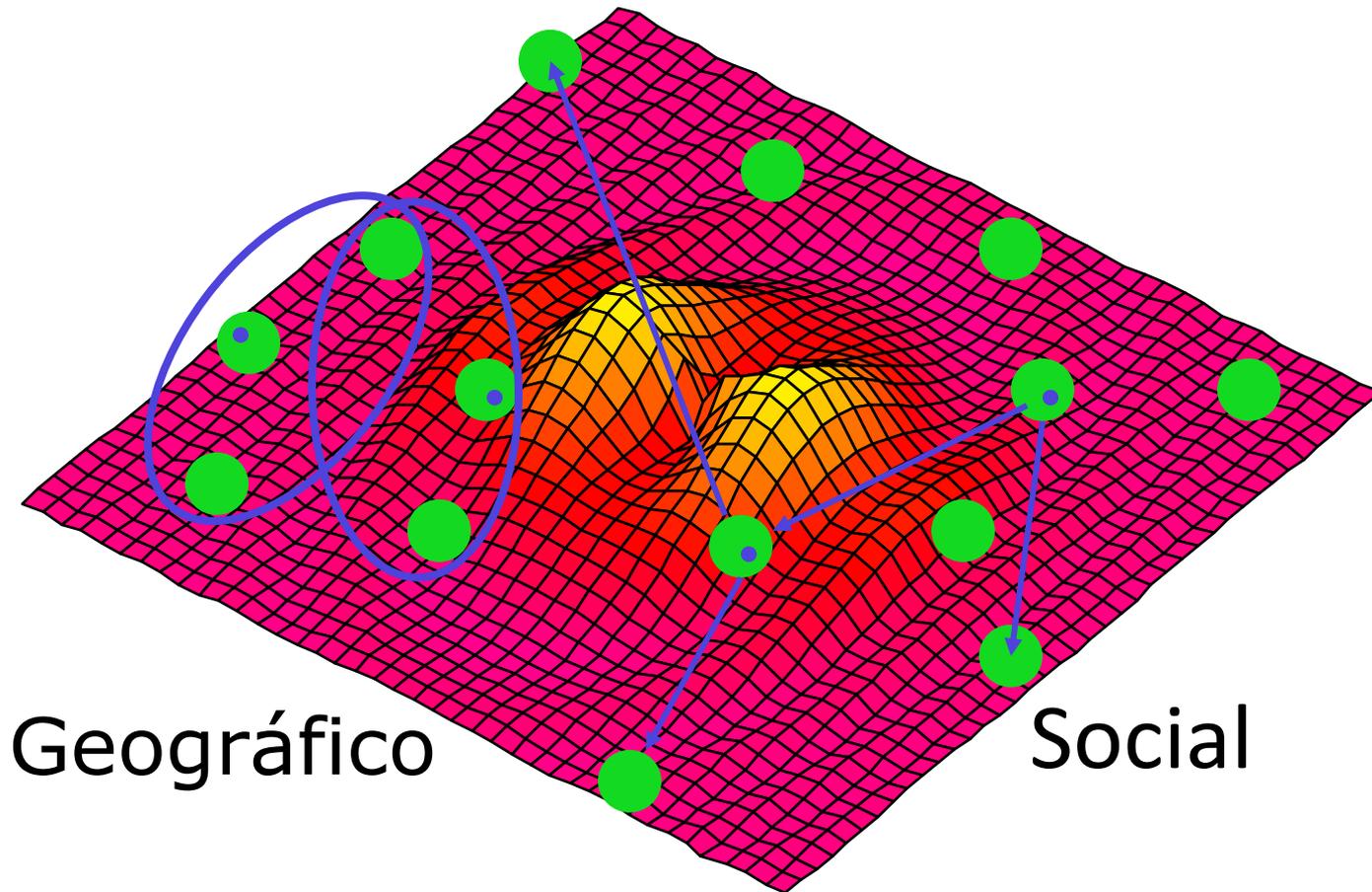
Valores de los Parámetros

- **Tamaño de la nube:** Entre 20 y 40 partículas (problemas simples, 10; problemas muy complejos, 100-200).
- **Velocidad máxima:** V_{\max} se suele definir a partir del intervalo de cada variable.
- **Ratios de aprendizaje:** Habitualmente, $\varphi_1 = \varphi_2 = 2$.
- **PSO Global vs. PSO Local:** La versión global converge más rápido pero cae más fácilmente en óptimos locales y viceversa.

Topologías de la Nube de Partículas

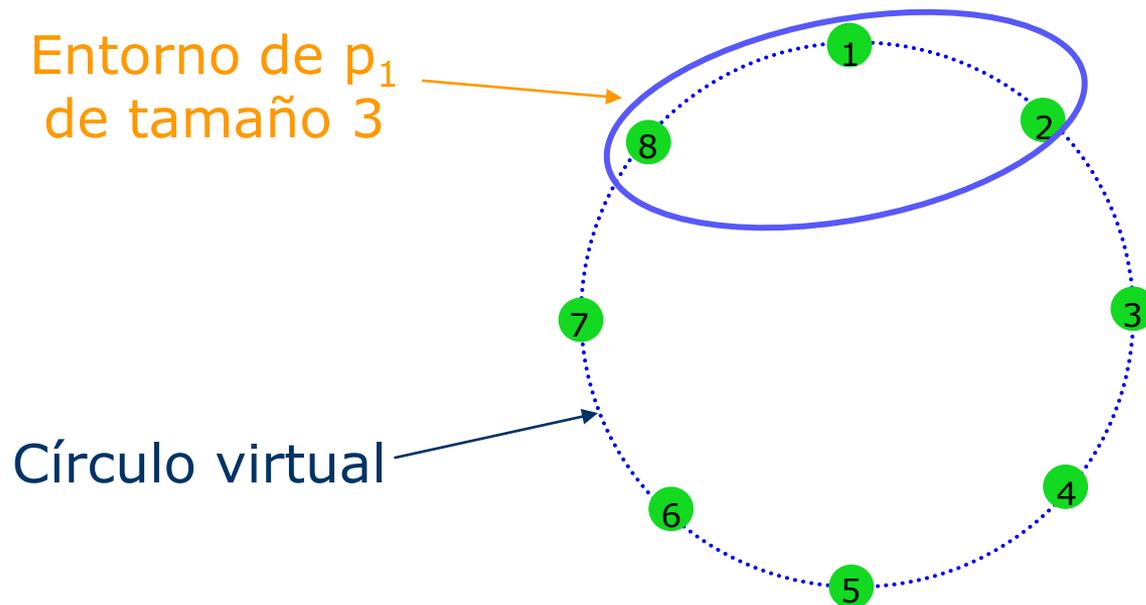
- Las topologías definen el entorno de cada partícula individual. La propia partícula siempre pertenece a su entorno.
- Los entornos pueden ser de dos tipos:
 - Geográficos: se calcula la distancia de la partícula actual al resto y se toman las más cercanas para componer su entorno.
 - Sociales: se define a priori una lista de vecinas para partícula, independientemente de su posición en el espacio.
- Los entornos sociales son los más empleados.
- Una vez decidido el entorno, es necesario definir su tamaño. El algoritmo no es muy sensible a este parámetro (3 o 5 son valores habituales con buen comportamiento).
- Cuando el tamaño es toda la nube de partículas, el entorno es a la vez geográfico y social, y tenemos la PSO global.

Topologías de la Nube de Partículas (2)



Topologías de la Nube de Partículas (3)

- La topología social más empleada es la de anillo, en la que se considera un vecindario circular.
- Se numera cada partícula, se construye un círculo virtual con estos números y se define el entorno de una partícula con sus vecinas en el círculo:



3. ASPECTOS AVANZADOS

- **CONTROL DE LA VELOCIDAD DE LAS PARTÍCULAS**
- **TAMAÑO DE LA NUBE DE PARTÍCULAS**
- **INFLUENCIA DEL TIPO DE ENTORNO**
- **ACTUALIZACIÓN DE LAS PARTÍCULAS**
- **ELECCIÓN DE VALORES ADAPTATIVOS PARA φ_1 Y φ_2**

Control de la Velocidad de las Partículas

- Un problema habitual de los algoritmos de PSO es que la magnitud de la velocidad suele llegar a ser muy grande durante la ejecución, con lo que las partículas se mueven demasiado rápido por el espacio.
- El rendimiento puede disminuir si no se fija adecuadamente el valor de V_{\max} , la velocidad máxima inicial de cada componente del vector velocidad.
- Se han propuesto dos métodos para controlar el excesivo crecimiento de las velocidades:
 - Un factor de inercia, ajustado dinámicamente, y
 - Un coeficiente de constricción.

Control de la Velocidad de las Partículas (2)

Factor de Inercia

- En este caso, la ecuación de adaptación de la velocidad pasa a ser la siguiente:

$$v_{id} = \omega \cdot v_{id} + \varphi_1 \cdot \text{rnd}() (pBest_{id} - x_{id}) + \varphi_2 \cdot \text{rnd}() (lBest_{id} - x_{id})$$

donde ω se inicializa a 1.0 y se va reduciendo gradualmente a lo largo del tiempo (medido en iteraciones del algoritmo).

- ω debe mantenerse entre 0.9 y 1.2. Valores altos provocan una búsqueda global (más diversificación) y valores bajos una búsqueda más localizada (mas intensificación).

Control de la Velocidad de las Partículas (3)

Coefficiente de Constricción

- De nuevo, se realiza una modificación en la ecuación de adaptación, la siguiente:

$$v_{id} = K \cdot [v_{id} + \varphi_1 \cdot \text{rnd}() (pBest_{id} - x_{id}) + \varphi_2 \cdot \text{rnd}() (lBest_{id} - x_{id})]$$

donde:

- $K = \frac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}}$
- $\varphi = \varphi_1 + \varphi_2$
- $\varphi > 4$ (normalmente $\varphi = 4.1$, $\varphi_1 = \varphi_2$)

Tamaño de la Nube de Partículas

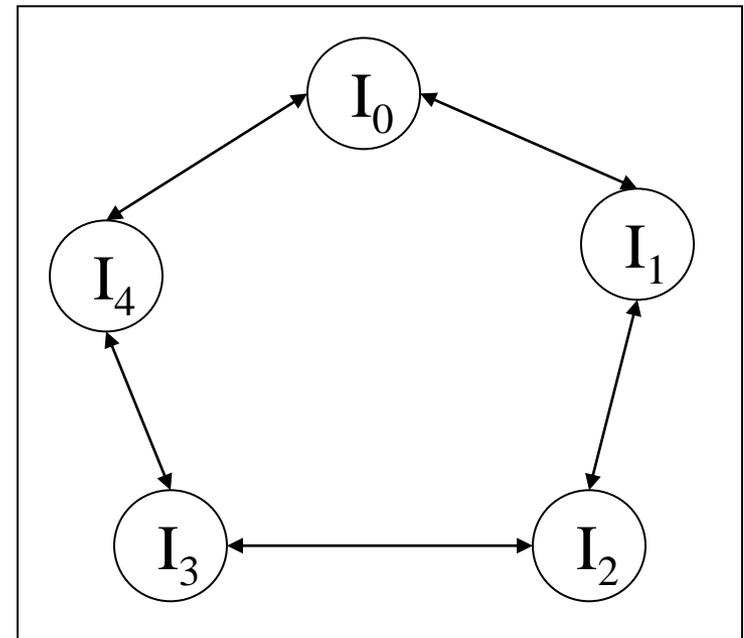
- El tamaño de la nube de partículas determina el equilibrio entre la calidad de las soluciones obtenidas y el coste computacional (número de evaluaciones necesarias).
- Hace poco, se han propuesto algunas variantes que adaptan heurísticamente el tamaño de la nube:
 - Si la calidad del entorno de la partícula ha mejorado pero la partícula es la peor de su entorno, se elimina la partícula.
 - Si la partícula es la mejor de su entorno pero no hay mejora en el mismo, se crea una nueva partícula a partir de ella.
- Las decisiones se toman de forma probabilística en función del tamaño actual de la nube.

Influencia del Tipo de Entorno

- Los entornos globales parecen obtener mejores resultados desde el punto de vista del coste computacional.
- El rendimiento es similar a la topología de anillo y al del uso de entornos con tamaño mayor que 3.
- Se ha investigado poco en los efectos de la topología de la nube en el comportamiento de la búsqueda del algoritmo.
- Por otro lado, el tamaño del vecindario también se puede adaptar con la misma heurística del tamaño de la nube.

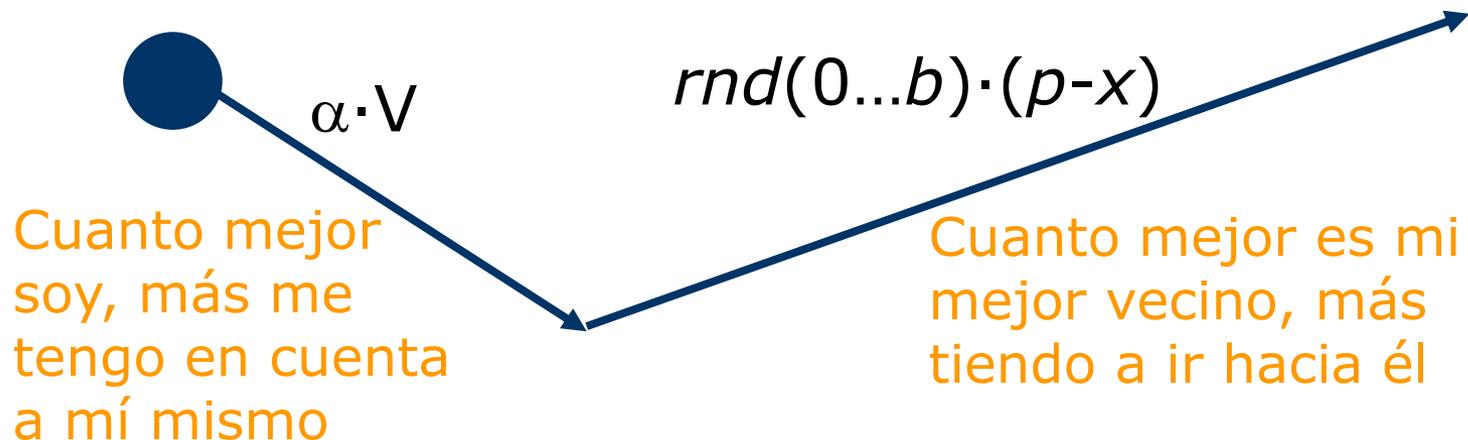
Actualización de las Partículas

- La actualización de las partículas se puede efectuar de dos formas distintas:
 - Síncrona
 - Asíncrona
- La actualización asíncrona permite considerar las soluciones nuevas más rápidamente.
- El método asíncrono puede representarse por el gráfico siguiente.



Elección de Valores Adaptativos para ϕ_1 Y ϕ_2

- Los pesos que definen la importancia de las componentes cognitiva y social pueden definirse dinámicamente según la calidad de la propia partícula y del entorno:



4. APLICACIONES Y RECURSOS ELECTRÓNICOS

- Optimización de funciones numéricas.
- Entrenamiento de Redes Neuronales.
- Aprendizaje de Sistemas Difusos.
- Registrado de Imágenes.
- Viajante de Comercio.
- Control de Sistemas.
- Ingeniería Química.
- ...
- Web site de PSO: <http://www.swarmintelligence.org/>

4. APLICACIONES Y RECURSOS ELECTRÓNICOS

- Versión discreta de PSO:

**Chen, W.-N. Zhang, J. Chung, H. S. H. Zhong, W.-L. Wu,
W.-G. Shi, Y.-h.**

**A Novel Set-Based Particle Swarm Optimization Method for
Discrete Optimization Problems**

**IEEE Transactions on Evolutionary Computation, 2010
14:2, 278 - 300**

Implementación en:



<http://sci2s.ugr.es/EAMHCO/#Software>

METAHEURÍSTICAS

TEMA 6. METAHEURÍSTICAS BASADAS EN ADAPTACIÓN SOCIAL

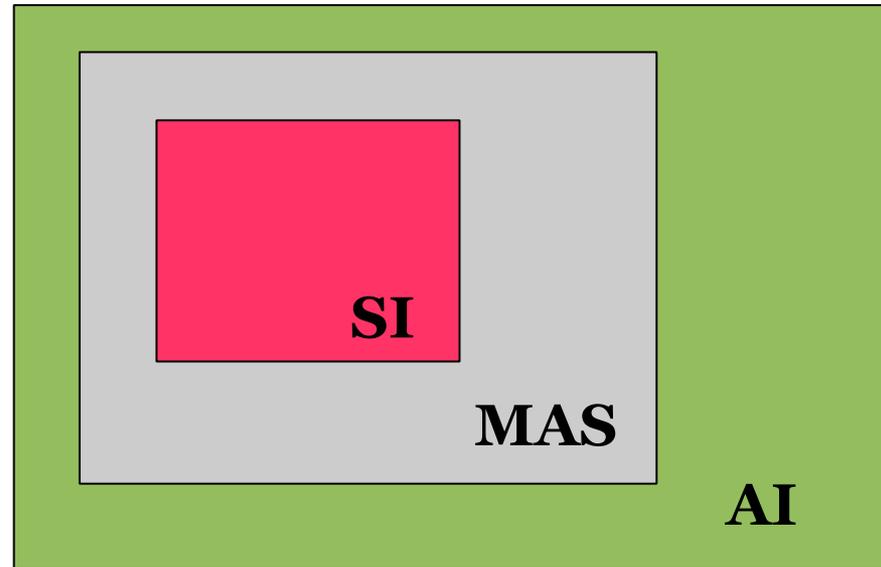
1. INTRODUCCIÓN A LOS MODELOS BASADOS EN ADAPTACIÓN SOCIAL: SWARM INTELLIGENCE
2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS
3. OPTIMIZACIÓN BASADA EN NUBES DE PARTÍCULOS (PARTICLE SWARM OPTIMIZATION)
4. CONCLUSIONES

Bibliografía

E. Bonabeau, M. Dorigo, G. Theraulaz. *Swarm Intelligence. From Nature to Artificial Systems*. Oxford University Press, 1999.

Kennedy, J., Eberhart, R. C., and Shi, Y.. *Swarm intelligence*. San Francisco: Morgan Kaufmann Publishers, 2001.

Swarm Intelligence e Inteligencia Artificial



- AI** - Artificial Intelligence
- MAS** - Multi Agent Systems
- SI** - Swarm Intelligence

Una definición de contexto de Swarm Intelligence en el ámbito de la Inteligencia Artificial

Conclusiones

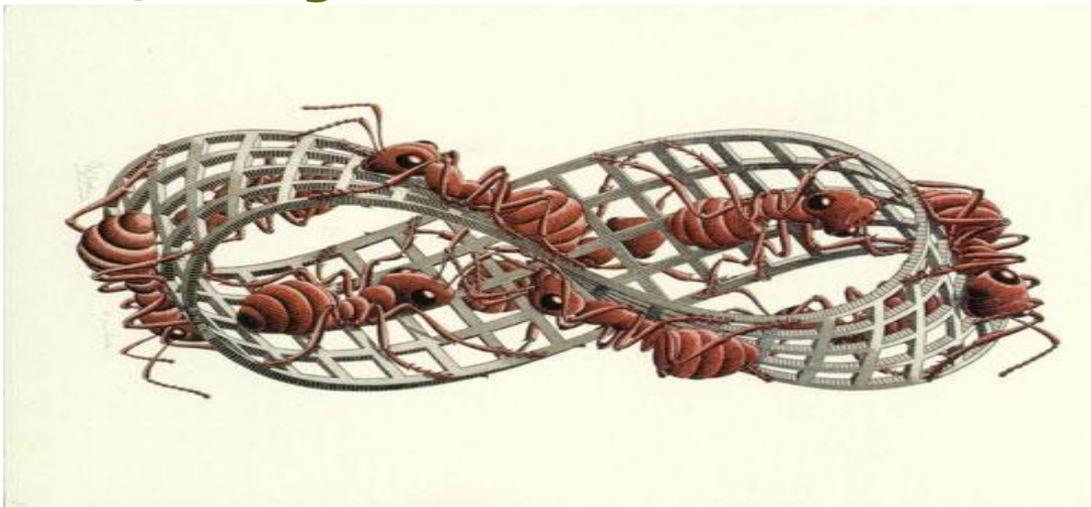
Swarm Intelligence (SI). Comportamiento colectivo de agentes (no sofisticados) que interactúan localmente con el entorno proporcionando un patrón global de funcionamiento (imitando a las sociedades de insectos).

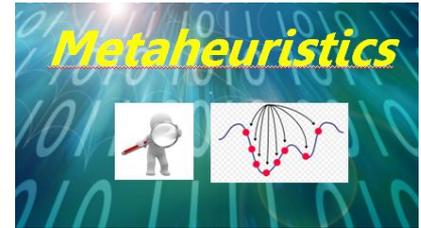
SI proporciona una base con la cual es posible explorar la resolución de problemas colectivamente (o de forma distribuida) sin un control centralizado ni un modelo global de comportamiento.

Conclusiones

"Dumb parts, properly connected into a swarm, yield smart results".

"Partes tontas/mudas, conectadas adecuadamente en un enjambre, producen resultados elegantes/inteligentes".





METAHEURÍSTICAS

2020-2021

- **Tema 1. Introducción a las Metaheurísticas**
- **Tema 2. Modelos de Búsqueda: Entornos y Trayectorias vs Poblaciones**
- **Tema 3. Metaheurísticas Basadas en Poblaciones**
- **Tema 4: Algoritmos Meméticos**
- **Tema 5. Metaheurísticas Basadas en Trayectorias**
- **Tema 6. Metaheurísticas Basadas en Adaptación Social**
- **Tema 7. Aspectos Avanzados en Metaheurísticas**
- **Tema 8. Metaheurísticas Paralelas**