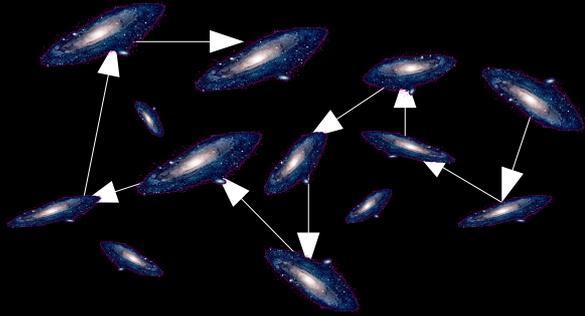


*Scheduling*  
o planificación de observaciones  
Clase 2

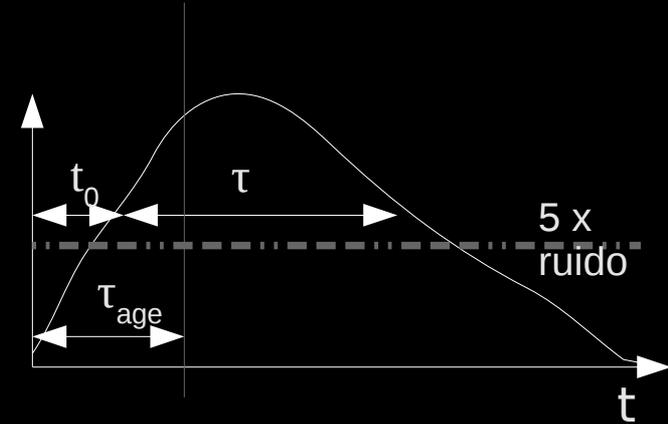
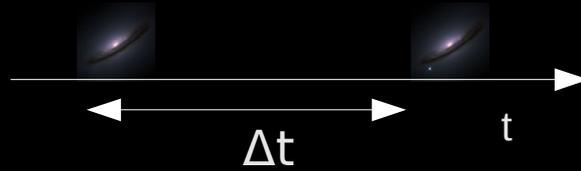
Francisco Förster Burón

Astroinformática – 24 Oct 2011

# Resumen clase 1



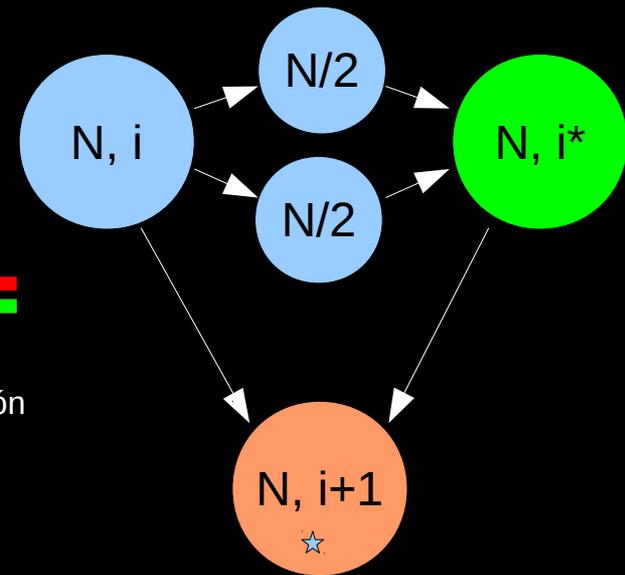
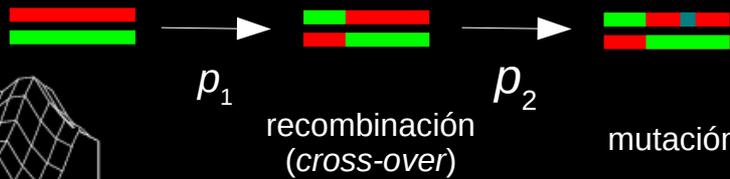
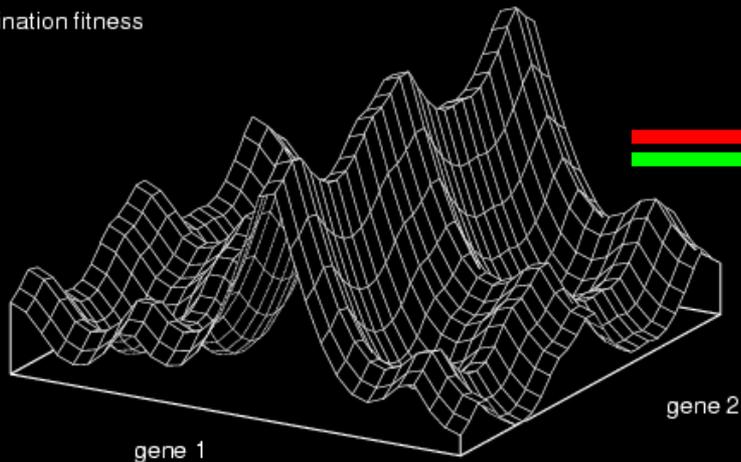
Telescopio robótico



$$\lambda_{D, \text{Total}}^{\text{age}} = \sum_i R^i \min\{\Delta t^i, \max(\tau_{\text{age}} - t_0^i, 0), \tau^i\}$$

Adaptive Landscape (Sewall Wright, 1932)

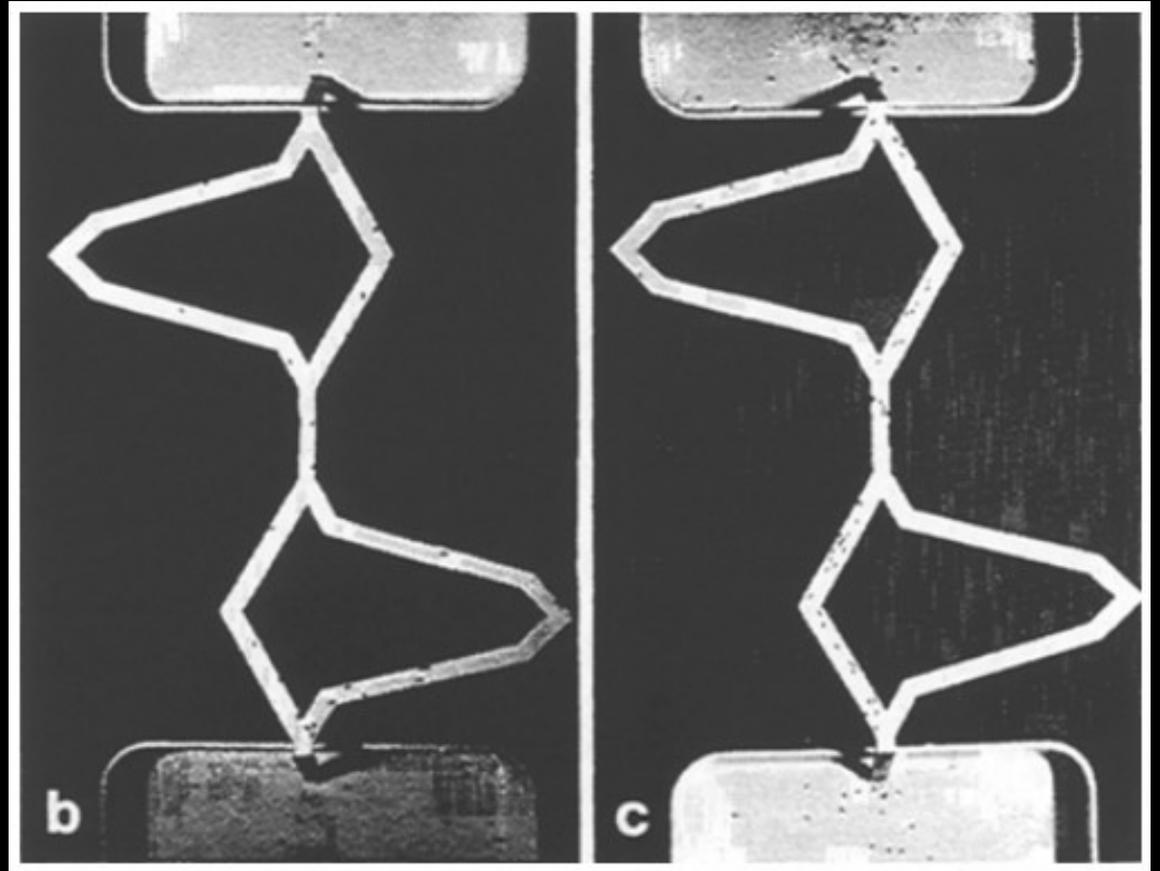
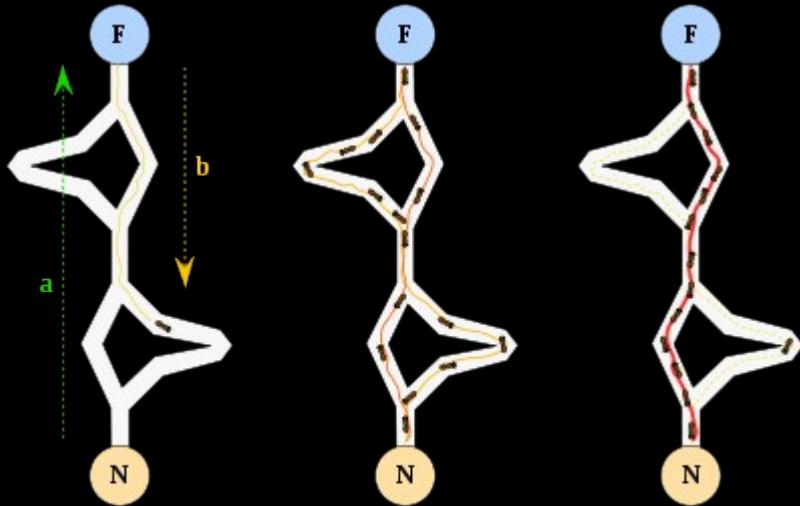
gene combination fitness



# Ant Colony Optimization



# Comportamiento de las hormigas



Las hormigas se comunican indirectamente modificando su ambiente con feromonas. Sistema de comunicación indirecto les permite optimizar la elección de caminos de forma distribuida

# Comportamiento de las hormigas

Deneubourg, Aron, Goss & Pasteels 1990

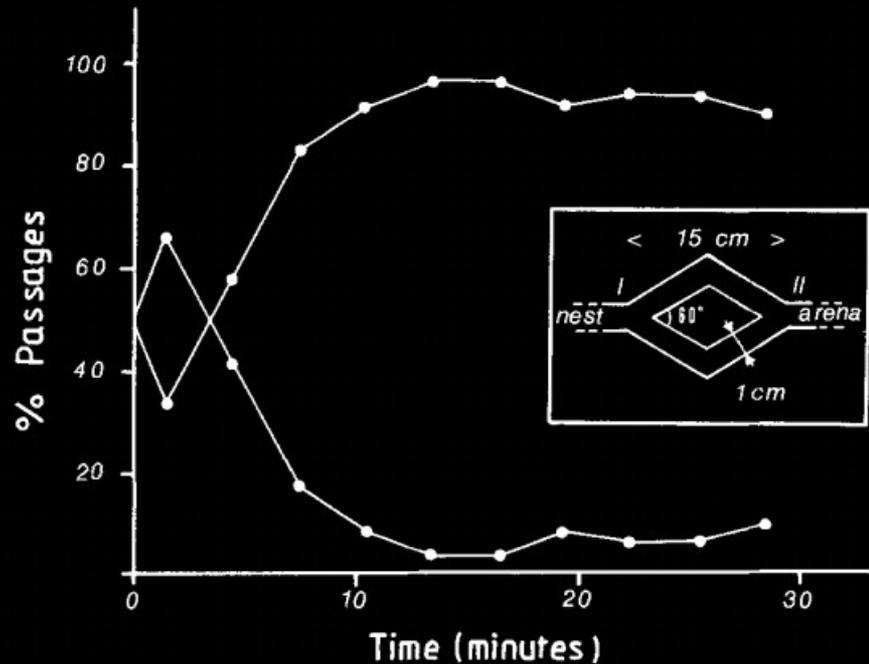
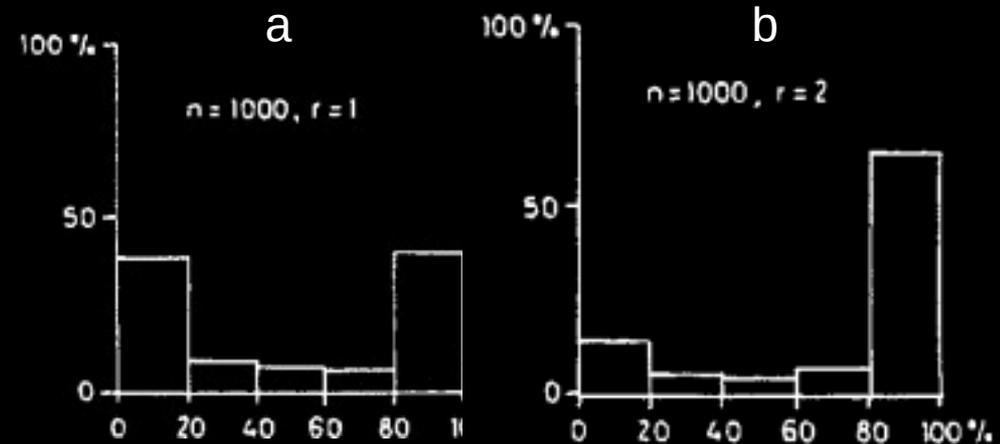


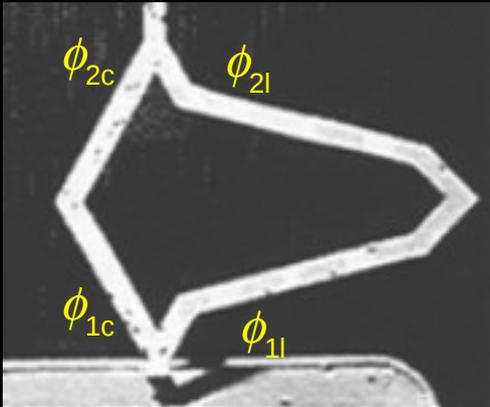
Fig. 2. Percentage of ants per 3-min period passing on the two branches of the bridge (inset). Colony of 1000 workers.



Frente a dos caminos de igual largo, eligen uno de los dos (a). Frente a dos caminos de distinto largo, eligen el más corto (b).

Retroalimentación positiva debido al depósito de feromonas

# Modelo hormigas virtuales



Expresión basada en experimentos para  $p_{ic}(t)$ , la probabilidad de elegir el camino corto (c) en la bifurcación i:

$$p_{ic}(t) = \frac{[t_c + \phi_{ic}(t)]^\alpha}{[t_c + \phi_{ic}(t)]^\alpha + [t_c + \phi_{il}(t)]^\alpha}$$

donde  $\phi_{ic}(t)$  es la cantidad de feromonas en la bifurcación i y en el camino c,  $t_c$  es el tiempo que demora una hormiga en cruzar el camino c, r es la razón entre la distancia de los caminos largo y corto.

La variación instantánea en un tiempo t de la cantidad de feromonas  $\phi$  en una bifurcación dada y en un camino dado es el flujo de hormigas que está saliendo del camino por la bifurcación más el flujo de hormigas que está entrando al camino por la bifurcación, donde  $\Psi$  es el flujo total de hormigas.

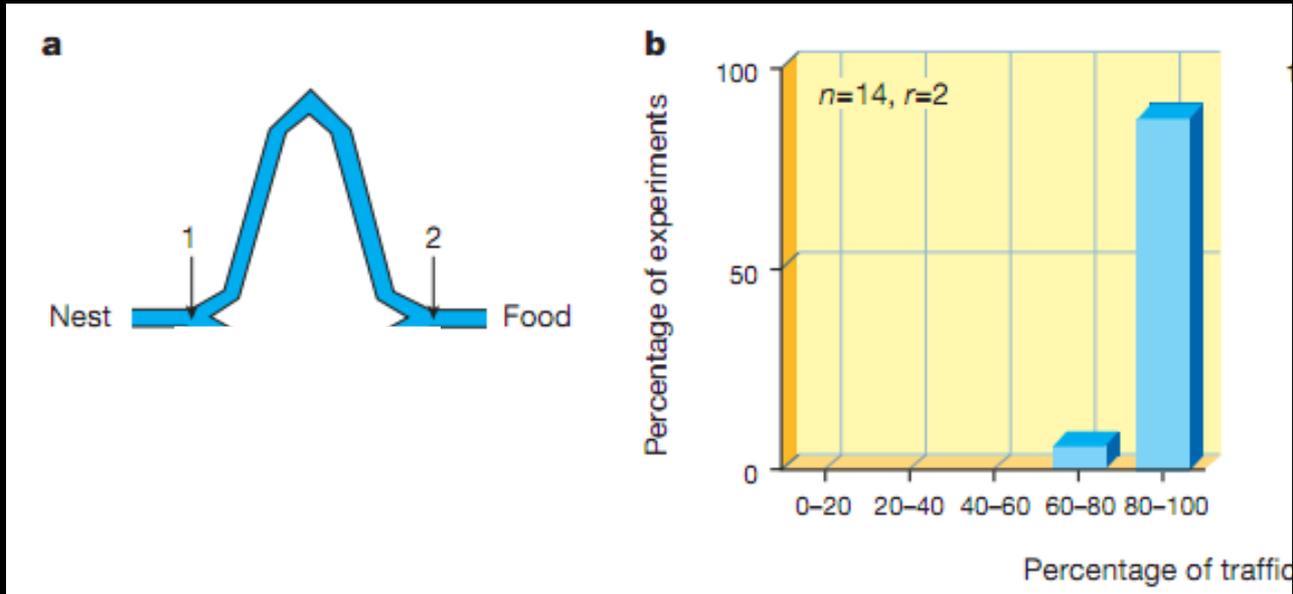
$$\begin{aligned} d\phi_{ic}/dt &= \underbrace{\psi p_{jc}(t-t_c)}_{\text{Hormigas saliendo}} + \underbrace{\psi p_{ic}(t)}_{\text{Hormigas entrando}}, & (i=1, j=2; i=2, j=1) \\ d\phi_{il}/dt &= \underbrace{\psi p_{jl}(t-r \cdot t_c)}_{\text{Hormigas saliendo}} + \underbrace{\psi p_{il}(t)}_{\text{Hormigas entrando}}, & (i=1, j=2; i=2, j=1) \end{aligned}$$

Hormigas  
saliendo

Hormigas  
entrando

# Comportamiento de las hormigas

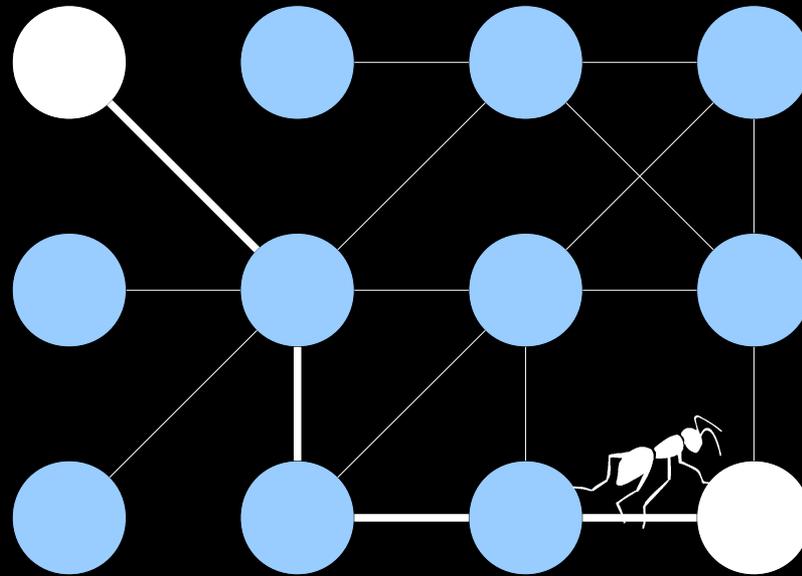
Bonabeau, Dorigo & Theraulaz, Nature 2000



Las hormigas escogen el camino más corto cuando se presentan ambos caminos de forma simultánea.

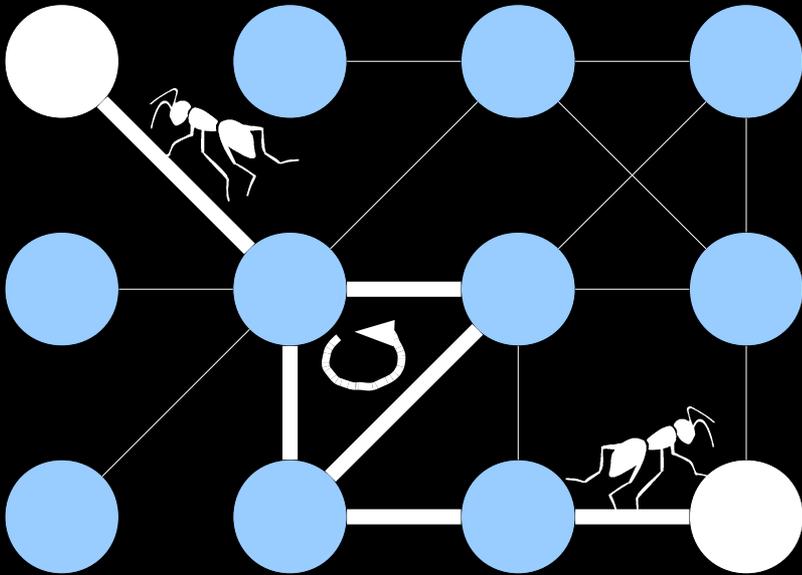
La evaporación de las feromonas es lenta

# Modelo hormigas virtuales



¿Cómo usar el modelo simple para 2 nodos a un modelo con muchos nodos y caminos que pueda ser utilizado para optimización de problemas combinatoriales?

# Modelo hormigas virtuales



Asumamos grafo  $G(N, A)$  de  $N$  nodos y un set  $A$  de caminos. Las hormigas debe ir desde el nodo hormiguero hasta el nodo donde se ubica el alimento de ida y de vuelta.

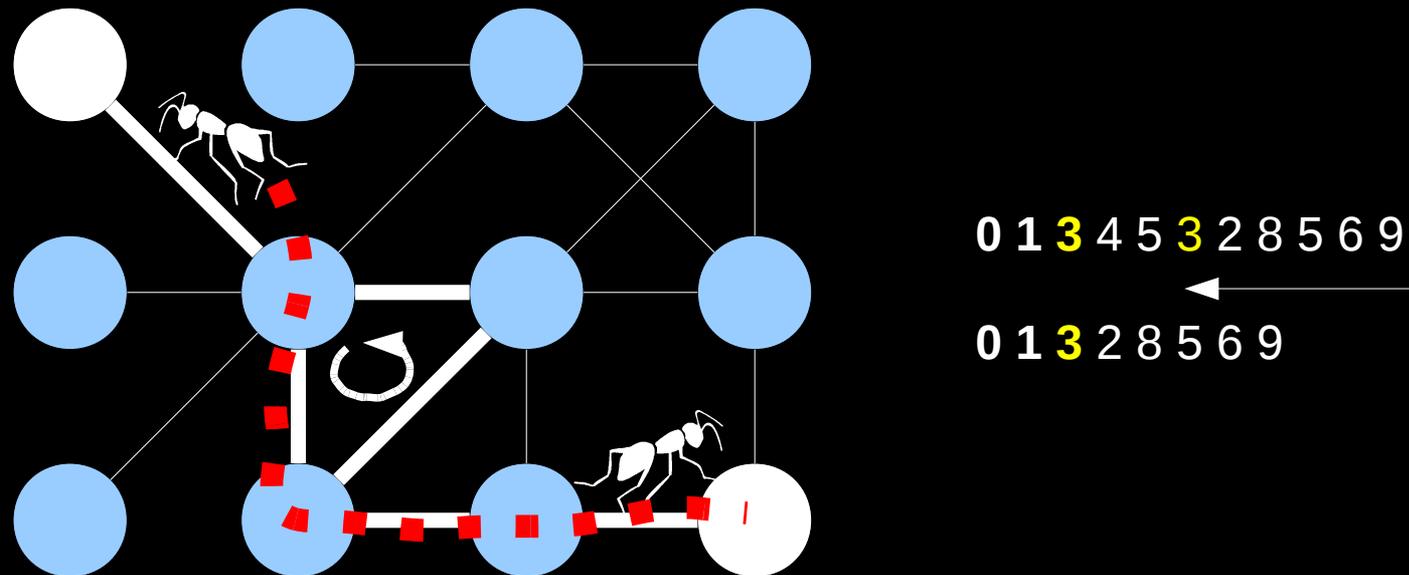
Producimos extensión del model anterior, i.e. tasa de cambio del nivel de feromonas en cada bifurcación en cada camino proporcional al número de hormigas en él.

**Problema:** esta extensión simple del modelo produce *loops* con retroalimentación positiva

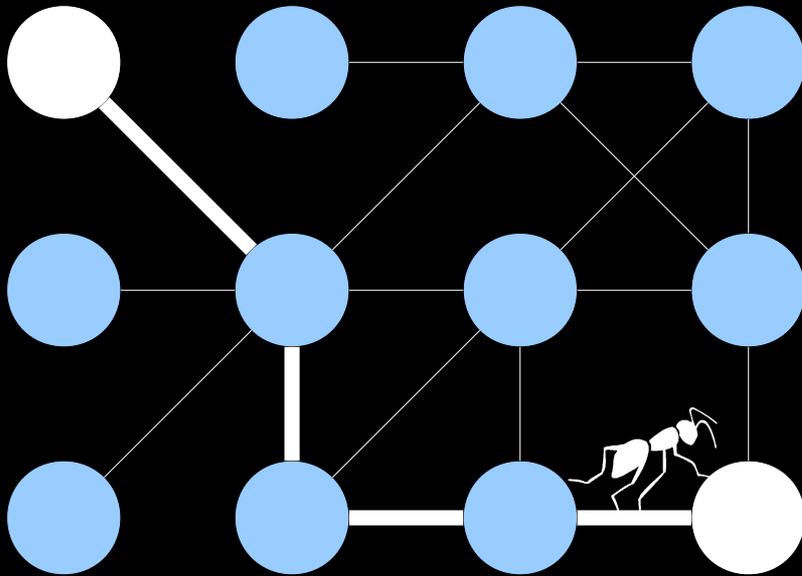
# Simple Ant Colony Optimization (S-ACO)

Para solucionar el problema de los *loops* y mejorar la convergencia se puede hacer lo siguiente:

1. caminos de ida y de vuelta se hacen de forma distinta
2. camino de ida probabilístico sin depositar feromonas, se mantiene registro del camino recorrido
3. camino de vuelta determinístico, eliminando *loops* del camino de ida y depositando feromonas
4. evaluación de la calidad de la solución para decidir cuántas feromonas depositar
5. evaporación artificial de las feromonas para evitar una convergencia prematura de la solución



# Ida: exploración probabilística



$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha}{\sum_{l \in N_i^k} \tau_{il}^\alpha} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases}$$

$p_{ij}^k$  es la probabilidad de que la hormiga  $k$  elija el camino  $ij$ , donde  $\tau_{ij}$  es el nivel de feromonas asociado a ese camino y  $N_i^k$  corresponde a los nodos en la vecindad de la hormiga  $k$ .

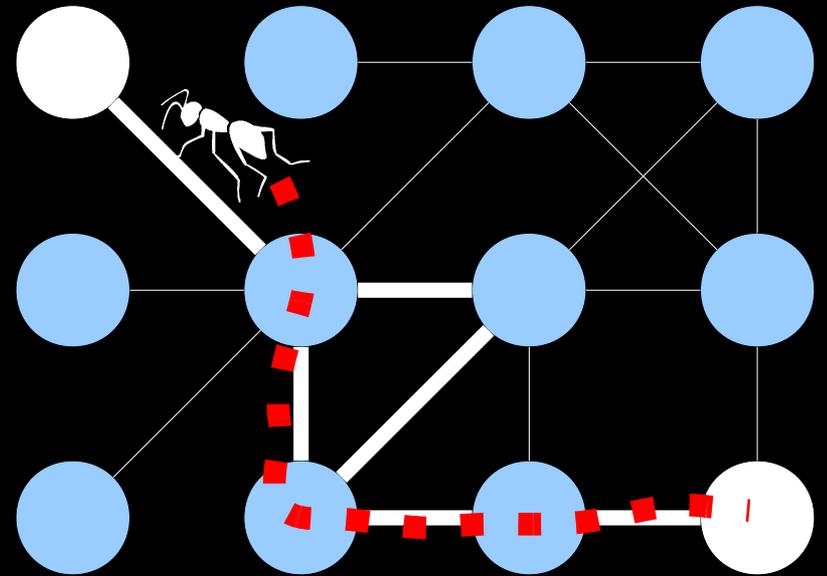
# Vuelta: actualización de las feromonas

Evaporación:

$$\tau_{ij} \leftarrow (1 - \rho) \tau_{ij}, \quad \rho \in [0, 1]$$

Hormigas volviendo al hormiguero:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta \tau^k$$



# Experimentos numéricos con S-ACO

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha}{\sum_{l \in N_i^k} \tau_{il}^\alpha} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases} \quad \begin{cases} \tau_{ij} \leftarrow (1 - \rho) \tau_{ij}, & \rho \in [0, 1] \\ \tau_{ij} \leftarrow \tau_{ij} + \Delta \tau^k \end{cases}$$

S-ACO es más efectivo con más hormigas virtuales

Se puede mejorar la calidad de las soluciones incluyendo una medida de la calidad de las soluciones en  $\Delta \tau^k$

Mientras mayor sea  $\alpha$  o  $\rho$ , S-ACO converge más rápido a soluciones sub-óptimas

El valor de  $\rho$  es crítico porque ayuda a encontrar buenas soluciones en problemas con grafos muy complicados, pero si  $\rho$  es muy alto S-ACO converge a soluciones sub-óptimas.

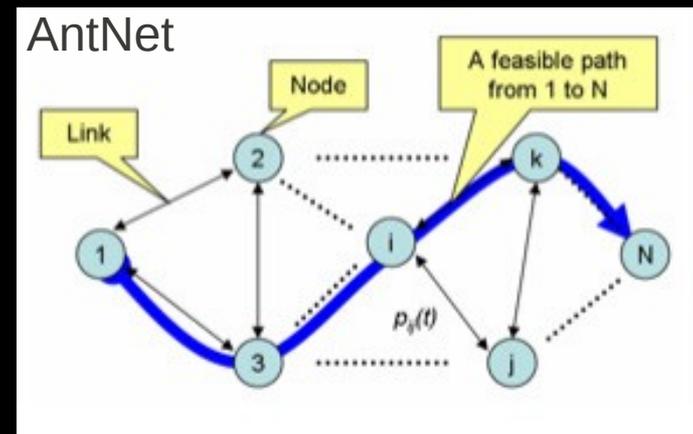
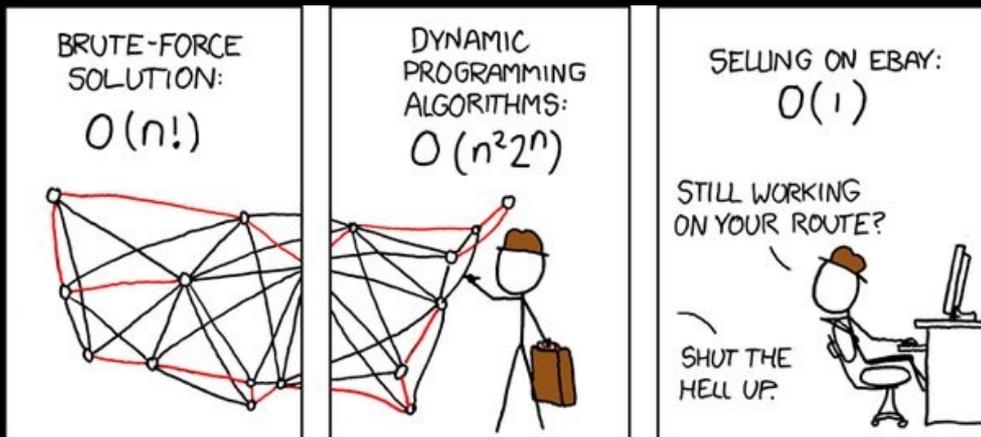
Un valor muy alto de  $\alpha$  generalmente resulta en un comportamiento malo, exagerando mucho las fluctuaciones aleatorias iniciales.

# Aplicaciones de ACO

Ant Colony Optimization (ACO) es un método *metaheurístico* en el que una colonia artificial de hormigas coopera para encontrar buenas soluciones en problemas de optimización discreta difíciles.

Las buenas soluciones son una propiedad que emerge de la interacción cooperativa de las hormigas.

ACO puede ser usado para resolver problemas *estáticos* o *dinámicos* de optimización combinatorial, e.g. travelling salesman problem vs network routing problem (AntNet)



# Representación en ACO

Al igual que en el caso de los algoritmos genéticos usados para resolver problemas de optimización combinatorial, **la eficacia de ACO depende de cómo se represente el problema.**

Consideremos el problema de minimizar el problema  $(S, f, \Omega)$ , donde  $S$  es el conjunto de soluciones candidato,  $f(s, t)$  es la función objetivo para las soluciones candidato y  $\Omega(t)$  es el conjunto de restricciones.

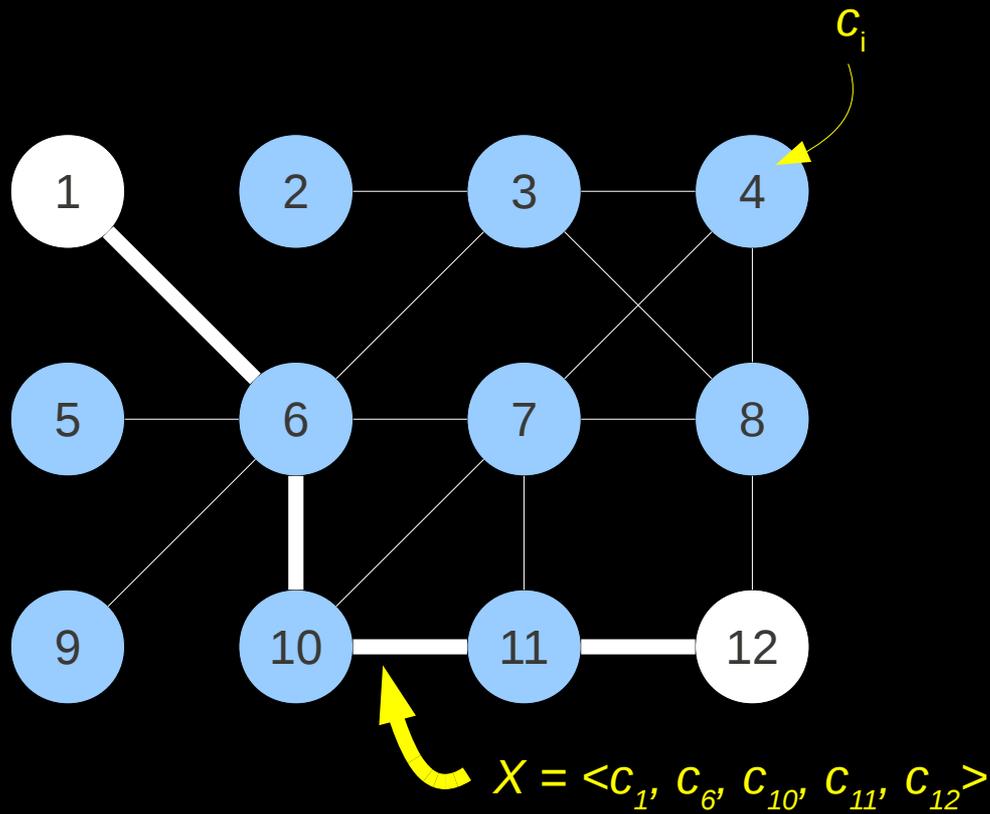
El objetivo es encontrar una solución globalmente óptima entre las soluciones candidato.

El problema combinatorial  $(S, f, \Omega)$  puede ser caracterizado por los siguientes elementos:

1. Un número finito de componentes  $C = \{c_1, c_2, \dots, c_{N_c}\}$ , donde  $N_c$  es el número de componentes
2. Diferentes estados del problema, dado por secuencias  $x = \langle c_i, c_j, \dots, c_h, \dots \rangle$ . El conjunto de estados posibles es  $X$ .
3. El conjunto de soluciones candidato  $S$ , que es un subconjunto de  $X$
4. Un conjunto de soluciones posibles  $\underline{X}$ , dado por las restricciones  $\Omega$
5. Un conjunto no vacío de soluciones óptimas  $S^*$
6. Una función costo  $g(s, t)$  para cada solución candidato. Generalmente  $g(s, t) = f(s, t)$
7. En algunos casos, una función de costo  $J(x, t)$  para soluciones que no sean candidatas.

Las hormigas virtuales construyen soluciones candidato realizando caminos aleatorios en un grafo  $G_c = (C, L)$ , con nodos  $C$  y caminos  $L$

Las restricciones se pueden imponer al momento de construir caminos o al momento de evaluar la calidad de las soluciones.



# Información heurística

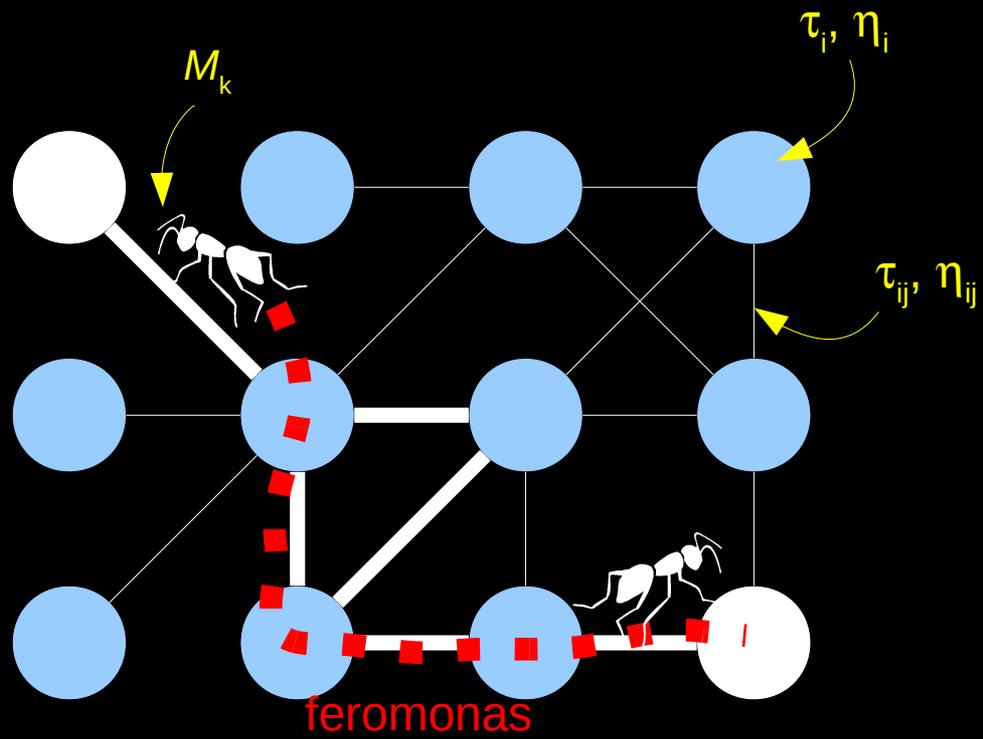
Los nodos y caminos tienen asociado un nivel de feromonas  $\tau_i$  o  $\tau_{ij}$  y un valor heurístico  $\eta_i$  o  $\eta_{ij}$

El nivel de feromonas indica el paso de hormigas virtuales por ese nodo o camino.

El valor heurístico indica información a priori acerca de la utilidad de un nodo dado. Es un valor aproximado de la utilidad de cada nodo o camino de acuerdo a la experiencia.

Cada hormiga de la colonia tendrá las siguientes propiedades:

1. Recorrerá el grafo  $G_c = (C, L)$
2. Tiene memoria  $M_k$  acerca del camino recorrido, que será utilizada para evaluar las restricciones del problema, asociar los valores heurísticos al camino, evaluar la calidad de la solución y realizar el camino de vuelta
3. Tiene un estado inicial y una condición de término
4. Se mueve de nodo en nodo aplicando decisiones probabilísticas en base al nivel local de feromonas y valor heurístico
5. Cuando agrega componentes a su estado, puede agregar feromonas al camino
6. Cuando se satisface la condición de término, puede realizar el camino de vuelta de forma determinista y actualizar el nivel de feromonas del grafo.



# Estructura típica de un algoritmo ACO

```
Procedure ACOMetaheuristic
```

```
    ScheduleActivities
```

```
        ConstructAntSolutions
```

```
        UpdatePheromones
```

```
        DaemonActions           # optional
```

```
    End-ScheduleActivities
```

```
end-procedure
```

# Network routing problem

Dado una red de telecomunicaciones definida por un conjunto de nodos  $N$ , un conjunto de enlaces entre nodos y costos asociados a los enlaces.

El problema de encontrar la ruta óptima dentro de la red se conoce como *network routing problem* (NRP).

Este problema se puede resolver con un tiempo polinomial usando el algoritmo de Dijkstra. Sin embargo, en la práctica los costos y la topología de la red varían en tiempo real, lo que hace impracticable usar el algoritmo de Dijkstra (similar al caso del planificador de observaciones).

ACO ha demostrado funcionar muy bien en NRP, e.g. AntNet.

La representación es la siguiente:

- 1. Grafo:** Los nodos y caminos de la red
- 2. Restricciones:** Las hormigas sólo pueden usar enlaces disponibles en un tiempo dado
- 3. Feromonas y valor heurístico:** Cada enlace tiene asociado un nivel de feromonas a medida que las hormigas lo recorren y un valor heurístico a priori, e.g. el inverso del tráfico en ese enlace.
- 4. Construcción de la solución:** cualquier realización del algoritmo, e.g. S-ACO

Cada hormiga tiene asociado un nodo origen y un nodo destino, moviéndose aleatoriamente desde el nodo origen hasta otros nodos hasta llegar al nodo destino según las probabilidades dadas por el nivel de feromona y valor heurístico locales.

# Implementaciones comunes de ACO

# Ant System (AS)

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases}$$

$$\eta_{ij} = \frac{1}{d_{ij}}$$

$$\alpha = 1, \beta = 2-5, \rho = 0.5, m = n, \tau_0 = m/C^{nn}$$

\*  $n$  es el número de nodos,  $m$  es el número de hormigas,  $\tau_0$  es el nivel inicial de feromonas asignado a todos los caminos,  $C^{nn}$  es el largo de una solución obtenida usando los mejores vecinos en cada nodo

$$\tau_{ij} \leftarrow (1 - \rho) \tau_{ij}$$

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k$$

$$\Delta \tau_{ij}^k = \begin{cases} 1/C^k & (i, j) \in T^k \\ 0 & \end{cases}$$

\*  $T^k$  es el secuencia de caminos de la hormiga  $k$ ,  $C^k$  es el largo de esta secuencia

# Elitist Ant System (EAS)

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases}$$

$$\eta_{ij} = \frac{1}{d_{ij}}$$

$$\alpha = 1, \beta = 2-5, \rho = 0.5, m = n, \tau_0 = (e+m)/\rho C^{nn}$$

\*  $n$  es el número de nodos ( $e = n$ ),  $m$  es el número de hormigas,  $\tau_0$  es el nivel inicial de feromonas asignado a todos los caminos,  $C^{nn}$  es el largo de una solución obtenida usando los mejores vecinos en cada nodo

$$\tau_{ij} \leftarrow (1 - \rho) \tau_{ij}$$

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k + e \Delta \tau_{ij}^{bs}$$

$$\Delta \tau_{ij}^k = \begin{cases} 1/C^k & (i, j) \in T^k \\ 0 & \end{cases}$$

\*  $T^k$  es el secuencia de caminos de la hormiga  $k$ ,  $C^k$  es el largo de esta secuencia.  $T^{bs}$  es la mejor secuencia hasta el momento (*best-so-far*)

# Rank-based Ant System (RAS)

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases}$$

$$\eta_{ij} = \frac{1}{d_{ij}}$$

$$\alpha = 1, \beta = 2-5, \rho = 0.1, m = n, \tau_0 = 0.5r(r-1)/\rho C^{nn}$$

\*  $n$  es el número de nodos,  $m$  es el número de hormigas,  $\tau_0$  es el nivel inicial de feromonas asignado a todos los caminos,  $C^{nn}$  es el largo de una solución obtenida usando los mejores vecinos en cada nodo,  $r$  es un ranking de los caminos.

$$\tau_{ij} \leftarrow (1 - \rho) \tau_{ij}$$

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{r=1}^{w-1} (w-r) \Delta \tau_{ij}^r + w \Delta \tau_{ij}^{bs}$$

$$\Delta \tau_{ij}^k = \begin{cases} 1/C^k & (i, j) \in T^k \\ 0 & \end{cases}$$

\*  $T^k$  es el secuencia de caminos de la hormiga  $k$ ,  $C^k$  es el largo de esta secuencia.  $T^{bs}$  es la mejor secuencia hasta el momento (*best-so-far*). Se toman en cuenta sólo las mejores  $w-1$  soluciones ( $w=6$ ) hasta el momento, con  $r$  la posición relativa de cada solución desde 1 hasta  $w-1$

# MIN-MAX Ant System

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases}$$

$$\eta_{ij} = \frac{1}{d_{ij}}$$

$$\alpha = 1, \beta = 2-5, \rho = 0.02, m = n, \tau_0 = 1/\rho C^{nn}$$

\*  $n$  es el número de nodos,  $m$  es el número de hormigas,  $\tau_0$  es el nivel inicial de feromonas asignado a todos los caminos,  $C^{nn}$  es el largo de una solución obtenida usando los mejores vecinos en cada nodo

$$\tau_{ij} \leftarrow (1 - \rho) \tau_{ij}$$

$$\tau_{ij} \leftarrow \max \left\{ \tau_{min}, \min \left\{ \tau_{max}, \tau_{ij} + \Delta \tau_{ij}^{best} \right\} \right\}$$

$$\Delta \tau_{ij}^k = \begin{cases} 1/C^k & (i, j) \in T^k \\ 0 & \end{cases}$$

\*  $T^{best}$  puede ser la mejor secuencia hasta el momento, o la mejor secuencia de la iteración. Se definen un niveles de feromonas máximo y mínimo (ver referencias para más información)

# Ant Colony System (ACS)

$$j = \begin{cases} \operatorname{argmax}_{l \in N_i^k} \tau_{il} [\eta_{il}]^\beta & \text{if } q \leq q_0 \\ J & q > q_0 \end{cases}$$
$$\eta_{ij} = \frac{1}{d_{ij}}$$

$$\beta = 2-5, \rho = 0.1, m = 10, \tau_0 = 1/nC^{nn}$$

\*  $j$  es el siguiente nodo a elegir,  $q$  es un número aleatorio entre 0 y 1,  $q_0$  es un número entre 0 y 1 (0.9) y  $J$  es un nodo aleatorio

$$\tau_{ij} \leftarrow (1 - \xi) \tau_{ij} + \xi \tau_0$$

$$\tau_{ij} \leftarrow (1 - \rho) \tau_{ij} + \rho \Delta \tau_{ij}^{bs}$$

$$\Delta \tau_{ij}^k = \begin{cases} 1/C^k & (i, j) \in T^k \\ 0 & \end{cases}$$

\* La primera actualización es local y se hace durante la formación del camino para aumentar la exploración de caminos alternativos ( $\xi = 0.1$ ), la segunda es global y se hace al final de cada iteración usando información sólo de la mejor secuencia hasta el momento

La ventaja de este método es que la actualización de feromonas es  $O(n)$  en lugar de  $O(n^2)$ , por lo que es apropiado para problemas combinatoriales muy grandes

# Aplicación al problema de scheduling

Se utilizó ACS con la siguiente representación:

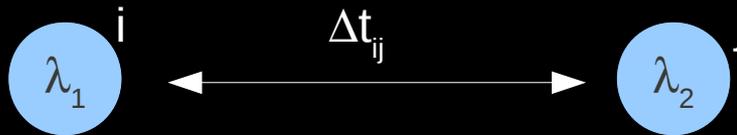
Se divide la noche en un número de intervalos  $k$

Cada galaxia en un intervalo de tiempo corresponde a un nodo

Para los nodos  $i$  y  $j$  se calcula el número esperado de detecciones de supernova,  $\lambda_i$  y  $\lambda_j$

Se calculan los tiempos de apunte entre nodos,  $\Delta t_{ij}$

Se asocia un nivel heurístico igual al número esperado de detecciones dividido por la mitad de la suma de los tiempos de exposición,  $T_1$  y  $T_2$ , más el tiempo de movimiento del telescopio entre nodos

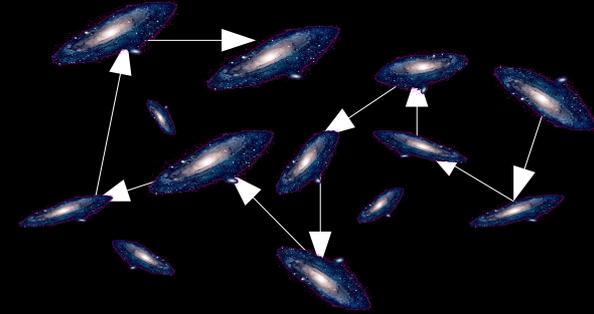


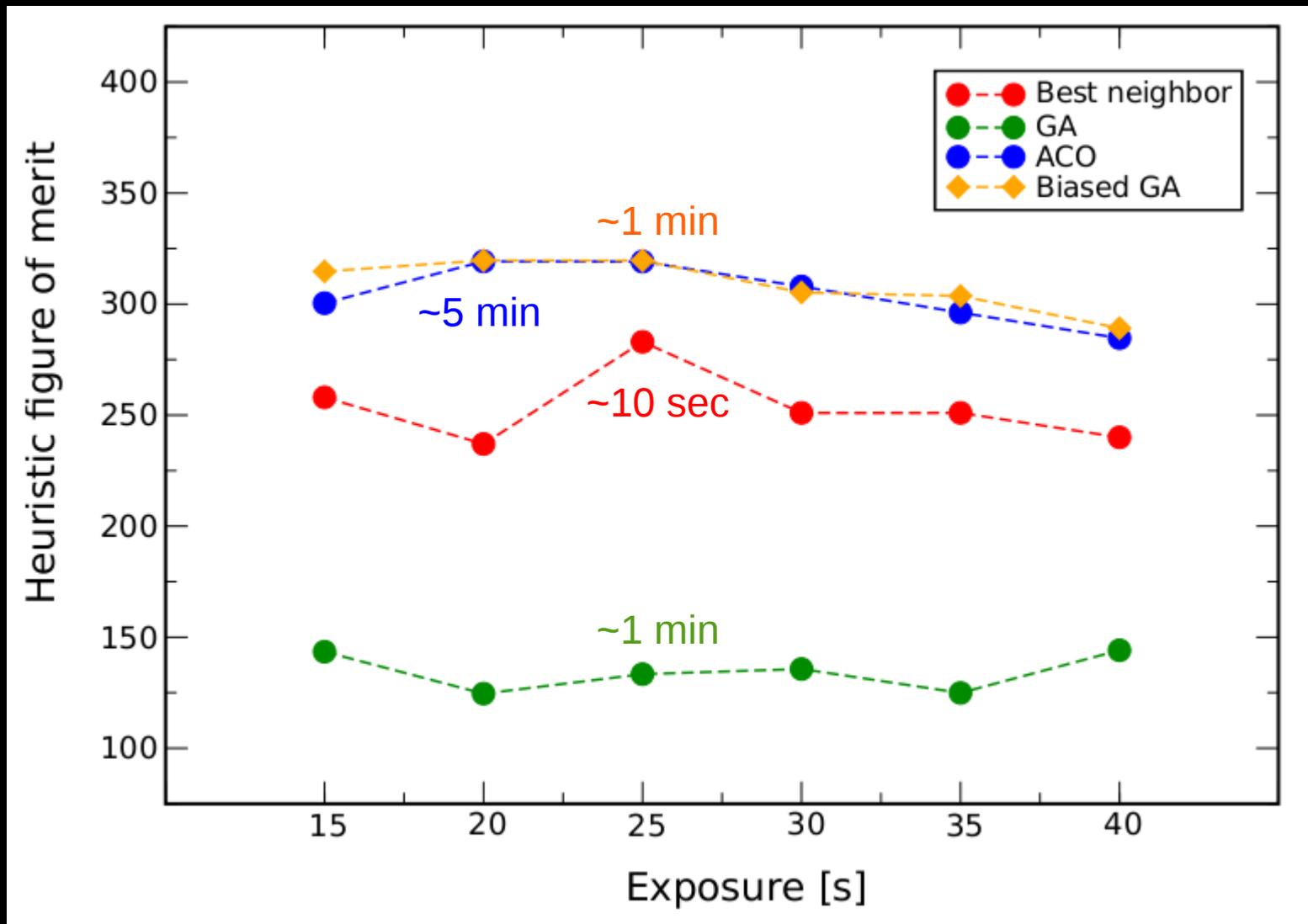
$$\eta_{ij} = \frac{\lambda_1 + \lambda_2}{T_1/2 + \Delta t_{ij} + T_2/2}$$

Se variaron los parámetros de ACS experimentalmente hasta encontrar aquellos que dieran mejores soluciones en un tiempo menor.

Se encontraron soluciones mejores a las encontradas con algoritmos genéticos!

Sin embargo, si usamos los valor heurísticos para sesgar la elección de caminos en un algoritmo genético encontramos soluciones mejores a ACO en un tiempo menor!





Crédito: César Parra

# Conclusiones y discusión

ACO es un método de optimización combinatorial recomendado para problemas dinámicos (e.g. AntNet)

Existen distintas implementaciones de ACO dependiendo de cómo se elijan los caminos y cómo se actualicen las feromonas (AS, EAS, RAS, MIN-MAX, ACS)

En general ACO consiste en un número de nodos y enlaces que pueden ser recorridos por hormigas virtuales

Las hormigas eligen un camino de acuerdo al nivel de feromonas y un valor heurístico para cada enlace

Las hormigas pueden actualizar el nivel de feromonas al regresar al hormiguero o durante la elección de caminos, para lo que deben ser capaces de memorizar los caminos recorridos

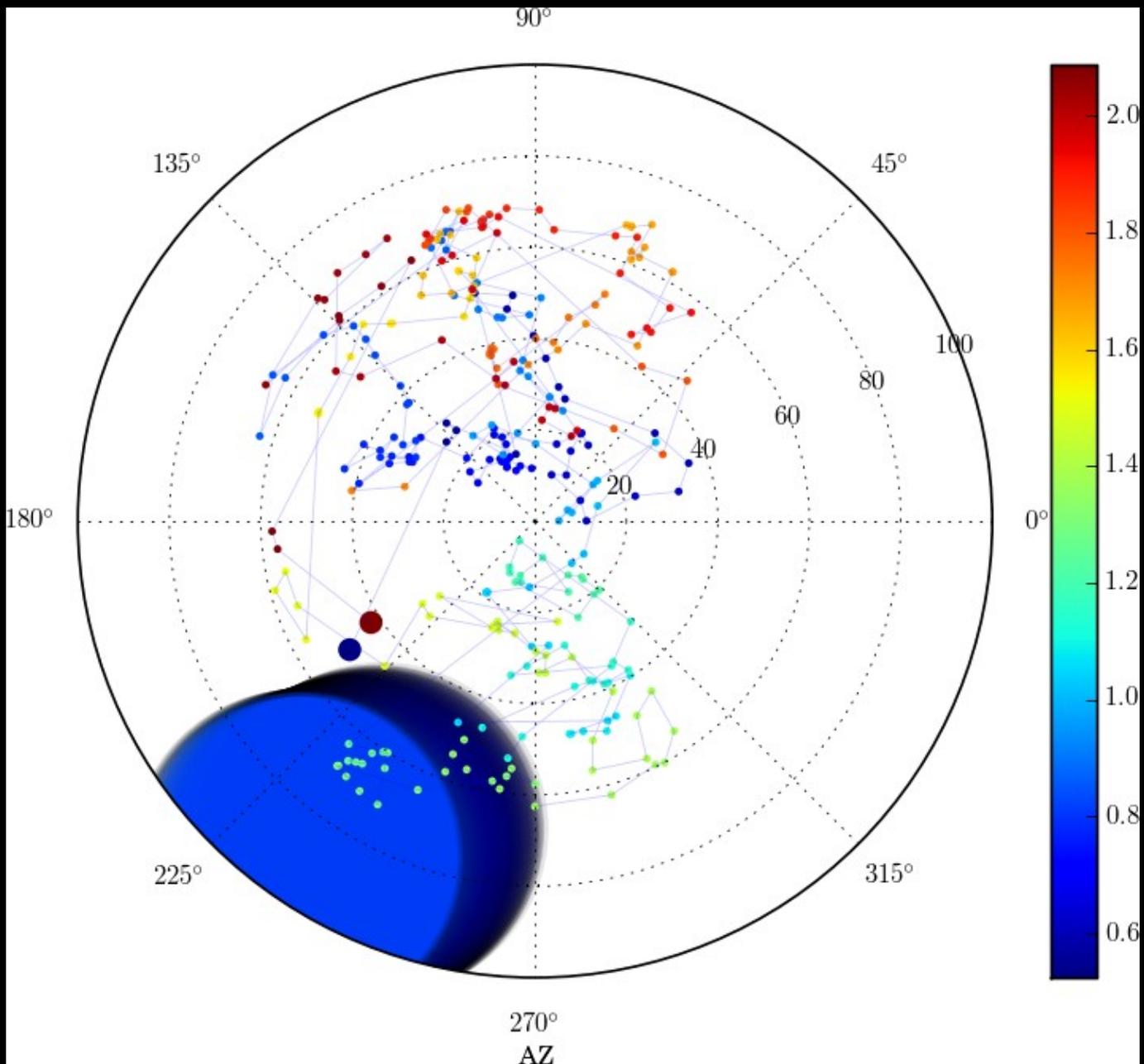
## Scheduling:

Para el problema de scheduling utilizamos algoritmos genéticos y ACO

Para representar el problema dividimos la noche en un número de intervalos fijo

Para acelerar la implementación precalculamos el número esperado de supernovas y los valores heurísticos de cada enlace para cada intervalo de tiempo durante la noche

En general, ACO con información heurística funcionó mejor que algoritmos genéticos sin información heurística, pero fue superado por un algoritmo genético que incorporaba información heurística.



Ejemplo de plan de observaciones para una noche dada obtenida usando un algoritmo genético sesgado. Los colores indican el tiempo, desde azul a rojo. La posición de la Luna está indicada por el círculo azul grande, que desaparece cuando la luna se pone debajo del horizonte. El plan generado tomó un minuto de cálculo usando un módulo para python escrito en FORTRAN 90, lo que es del orden del tiempo de exposición usado en la búsqueda.

# Referencias

“Introduction to Evolutionary Computing”, Agoston E. Eiben, J.E. Smith. Springer

“Ant Colony Optimization”, Marco Dorigo, Thomas Statzle. Springer

“Scheduling in Targeted Transient Surveys and a New Telescope for CHASE”, F. Förster, N. López, J. Maza, P. Kubánek, G. Pignata

F2PY: Fortran to Python interface generator, <http://cens.ioc.ee/projects/f2py2e/>

AntNet: <http://antnet.wordpress.com>