MAEB '07

Una Versión de ACO para Problemas con Grafos de muy Gran Extensión

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

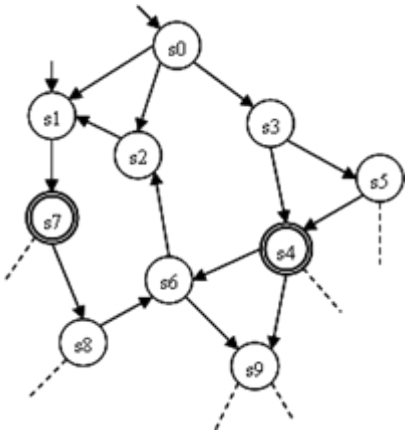
UNIVERSIDAD DE MÁLAGA

Enrique Alba y Francisco Chicano

MAEB '07

Introducción

- Los ACOs resuelven problemas planteados como **búsqueda de caminos mínimos en grafos**
- Existen problemas en los que el grafo subyacente es **desconocido y/o muy extenso**



$$\frac{\alpha \vee \beta,\ \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

- Los modelos actuales de ACO **no se pueden aplicar** a dichos problemas
  - En un grafo muy grande, la construcción de una solución completa puede requerir **mucha memoria y tiempo**
  - En algunos modelos **el número de nodos del grafo se usa para inicializar la matriz de feromonas**

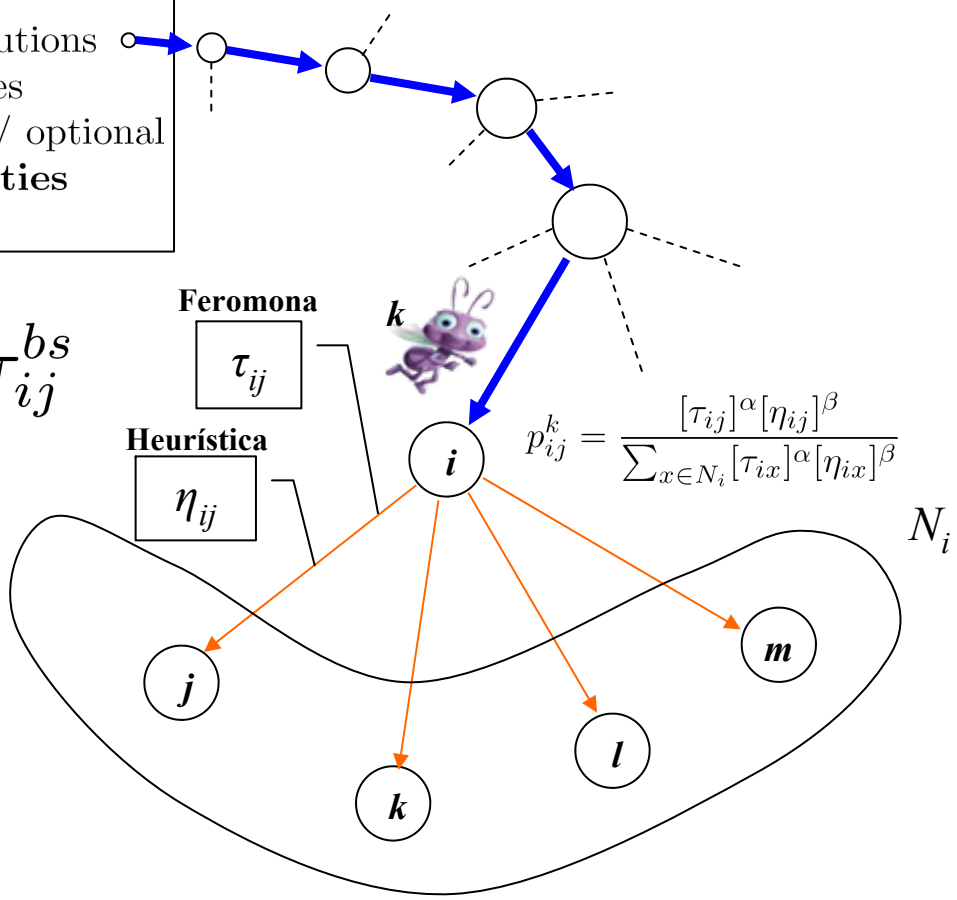
MAEB '07

Introducción

- **Necesitamos un nuevo modelo para abordar estas dificultades:**
  ACOhg (ACO for Huge Graphs)

```
procedure ACOMetaheuristic
    ScheduleActivities
        ConstructAntsSolutions
        UpdatePheromones
        DaemonActions // optional
    end ScheduleActivities
end procedure
```

$$\tau_{ij} \leftarrow \rho\tau_{ij} + \Delta\tau_{ij}^{bs}$$

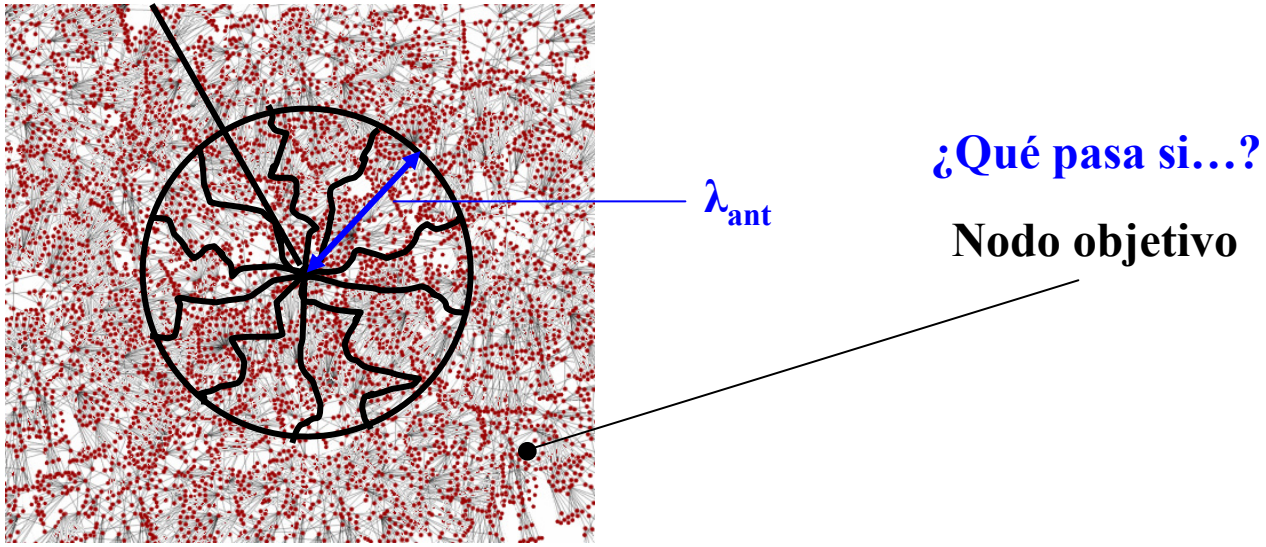


ACOhg extiende los modelos actuales con nuevas ideas pero mantiene la forma de construir las soluciones y de actualizar la feromona

MAEB '07

ACOhg: Longitud de los Caminos

- La longitud de los caminos de las hormigas se limita a λ_{ant}

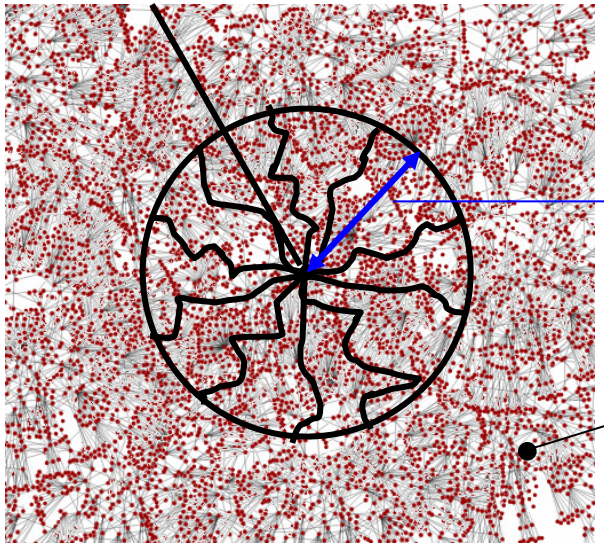


MAEB '07

ACOhg: Longitud de los Caminos

- La longitud de los caminos de las hormigas se limita a λ_{ant}

Nodo inicial



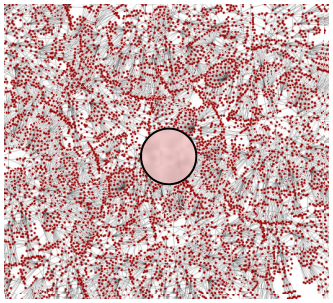
λ_{ant}

¿Qué pasa si…?

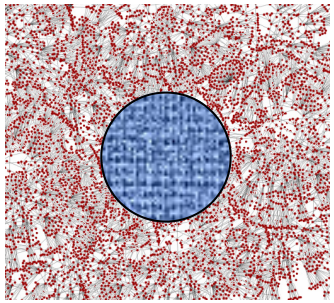
Nodo objetivo

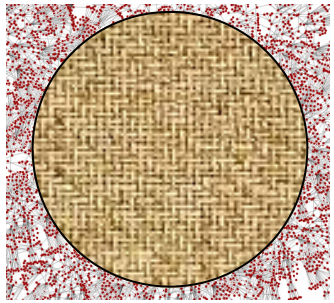
Dos alternativas

Técnica de Expansión: λ_{ant} cambia



Tras σ_i pasos





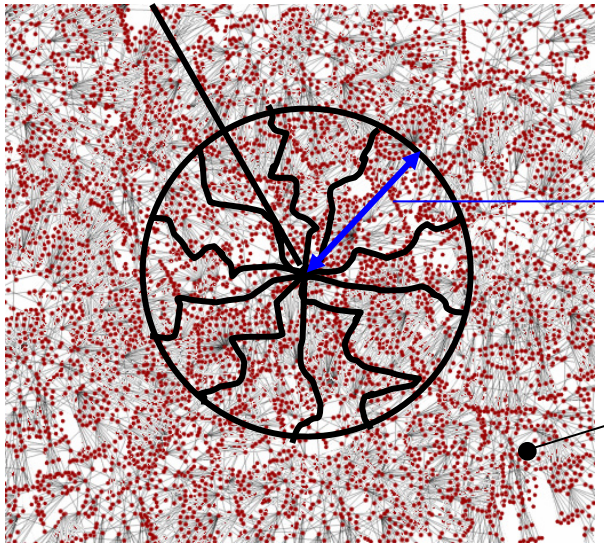
$\lambda_{ant} = \lambda_{ant} + \delta_l$

MAEB '07

ACOhg: Longitud de los Caminos

- La longitud de los caminos de las hormigas se limita a λ_{ant}

Nodo inicial

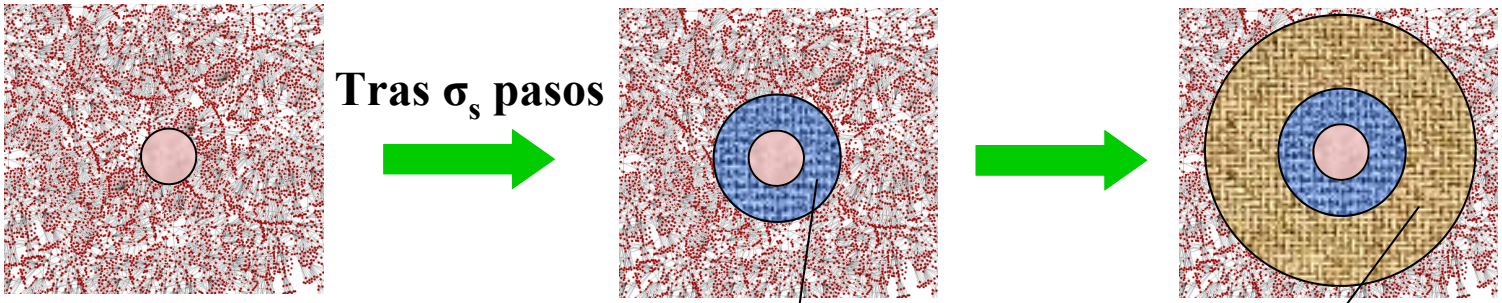


¿Qué pasa si…?

Nodo objetivo

Dos alternativas

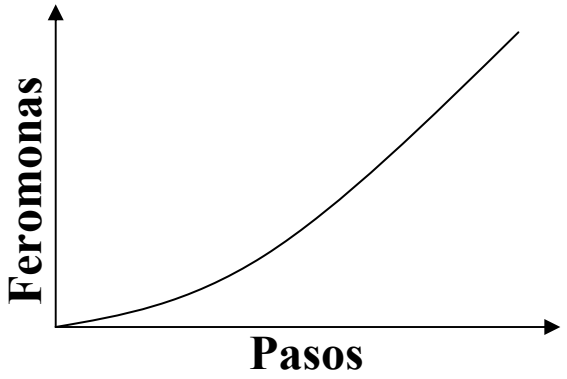
Técnica del Misionero: cambio de los nodos iniciales de los caminos



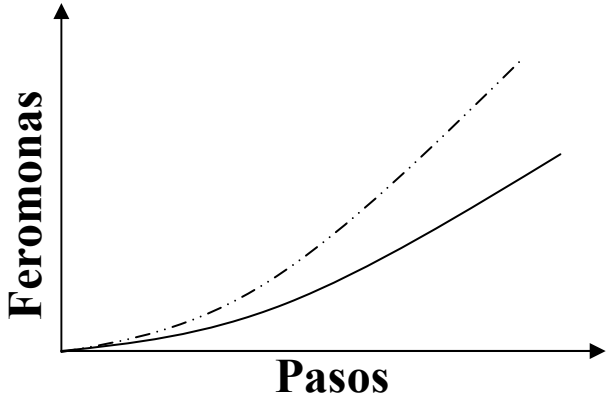
MAEB '07

ACOhg: Feromonas

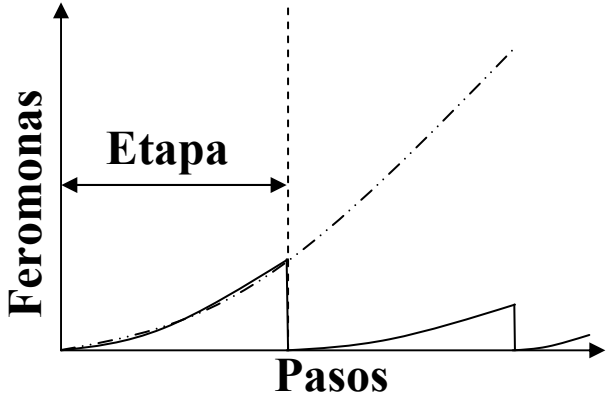
- El número de rastros de feromona aumenta durante la búsqueda
- Esto conduce a problemas de memoria
- Se deben eliminar algunos valores de feromona de la memoria



Eliminar valores de feromona τ_{ij} por debajo de un umbral τ_0



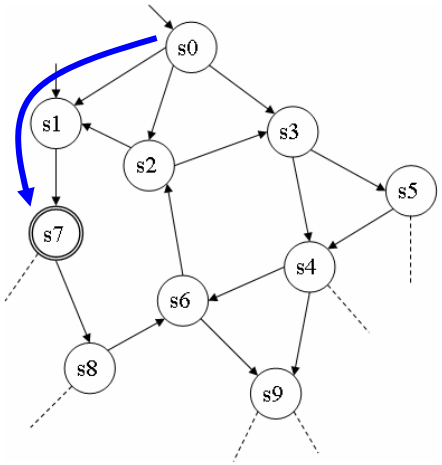
En la técnica del misionero, eliminar todos los valores de feromona tras cada etapa



MAEB '07

ACOhg: Función de Fitness

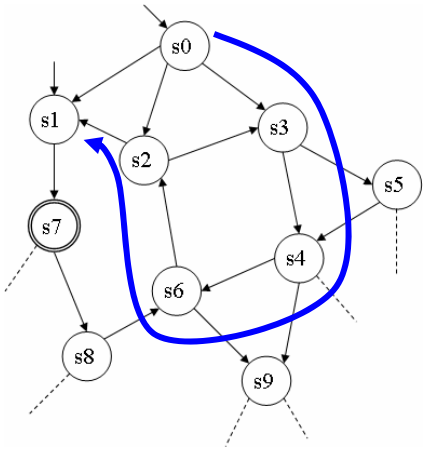
- La función de fitness debe ser capaz de evaluar soluciones parciales
- Se penalizan las soluciones parciales y las que contienen ciclos



Solución completa

$$p = 0$$

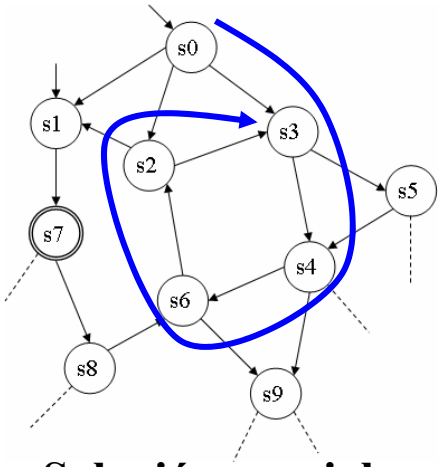
Penalización total



Solución parcial sin ciclo

$$p = p_p$$

Constante de penalización para soluciones parciales



Solución parcial con ciclo

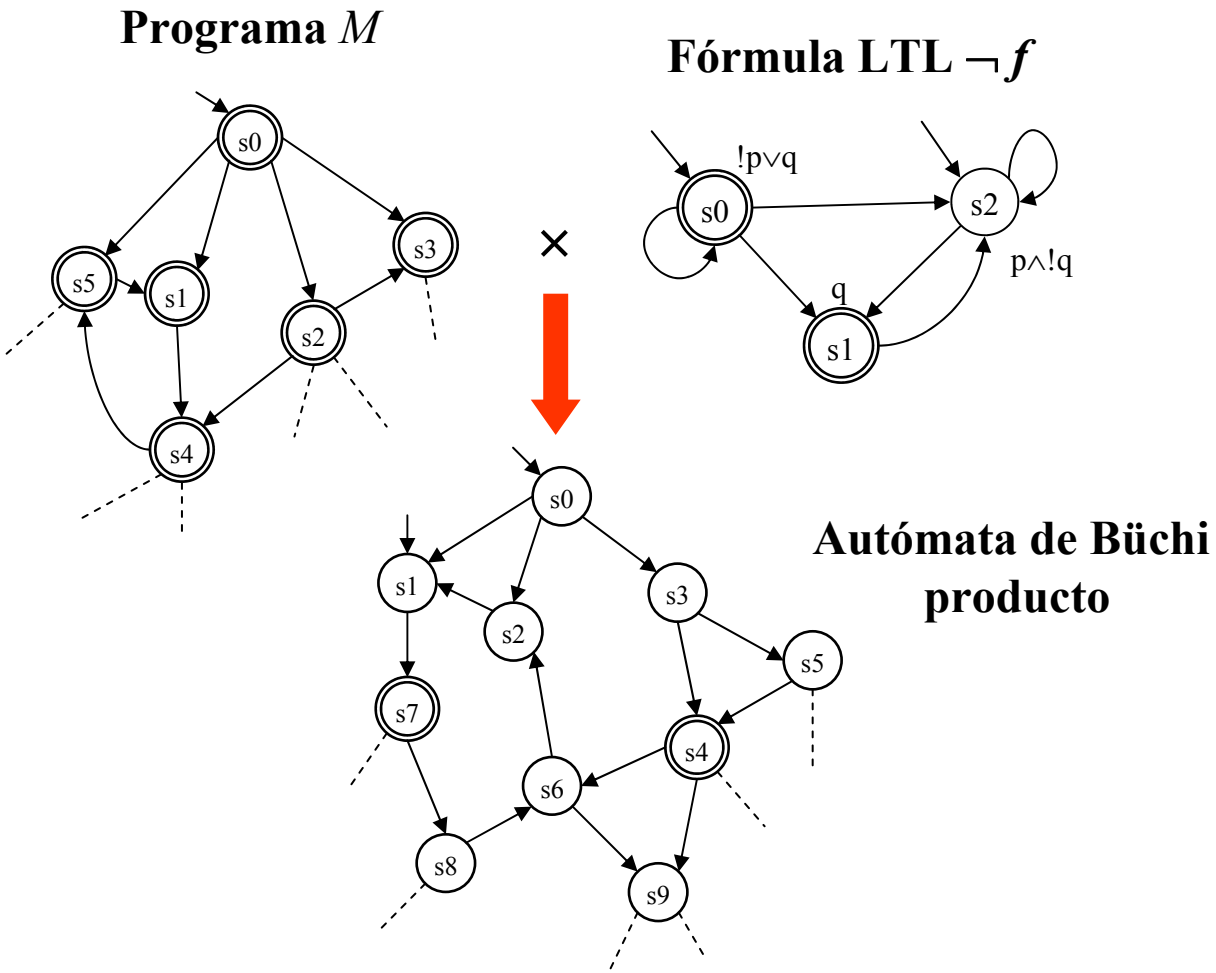
$$p = p_p + p_c \frac{\lambda_{ant} - l}{\lambda_{ant} - 1}$$

Constante de penalización para soluciones con ciclos

Longitud del camino

Problema Abordado (I)

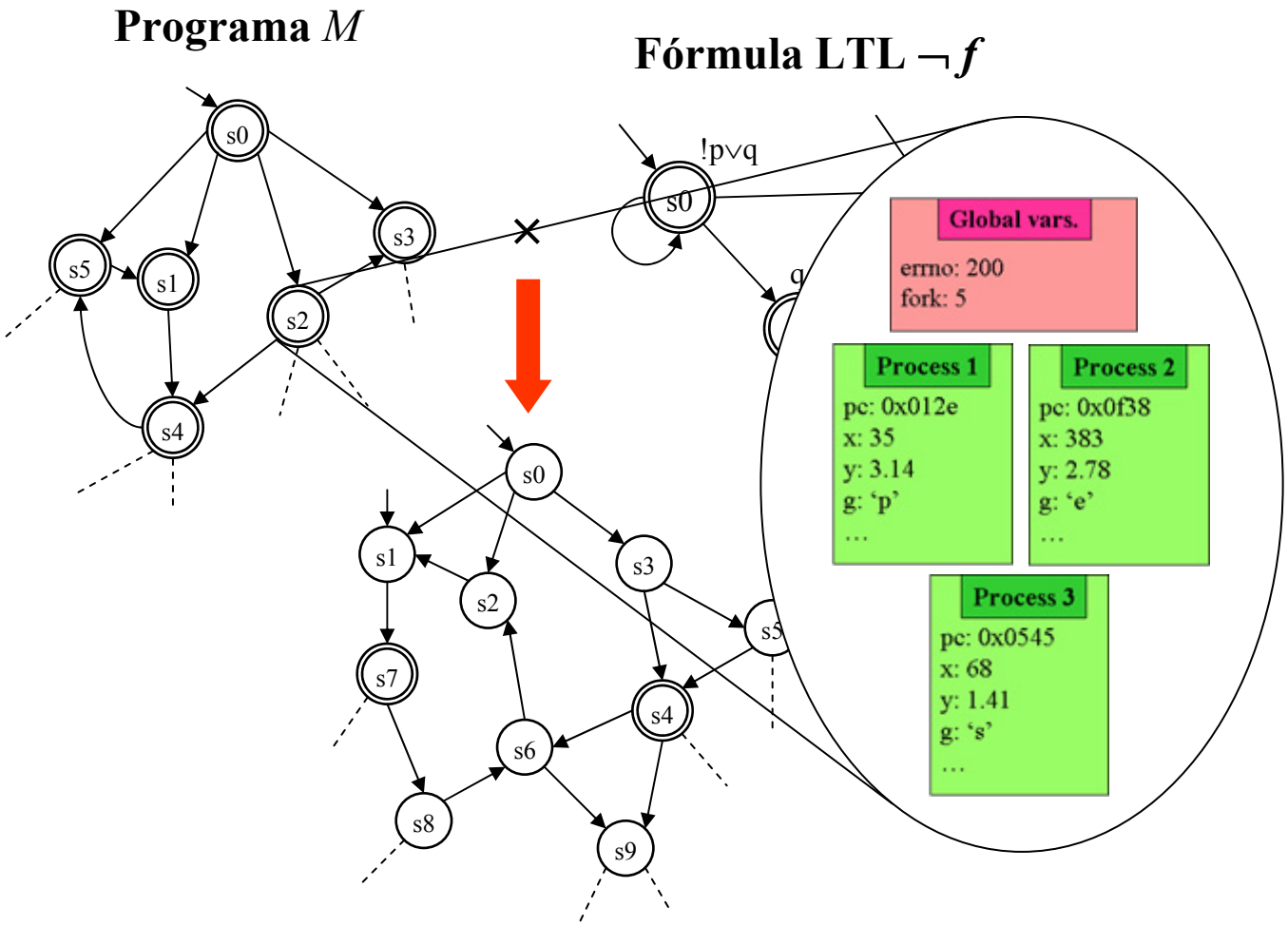
- **Objetivo**: encontrar errores en programas concurrentes
- **SPIN**: se especifica una propiedad usando una fórmula LTL f



MAEB '07

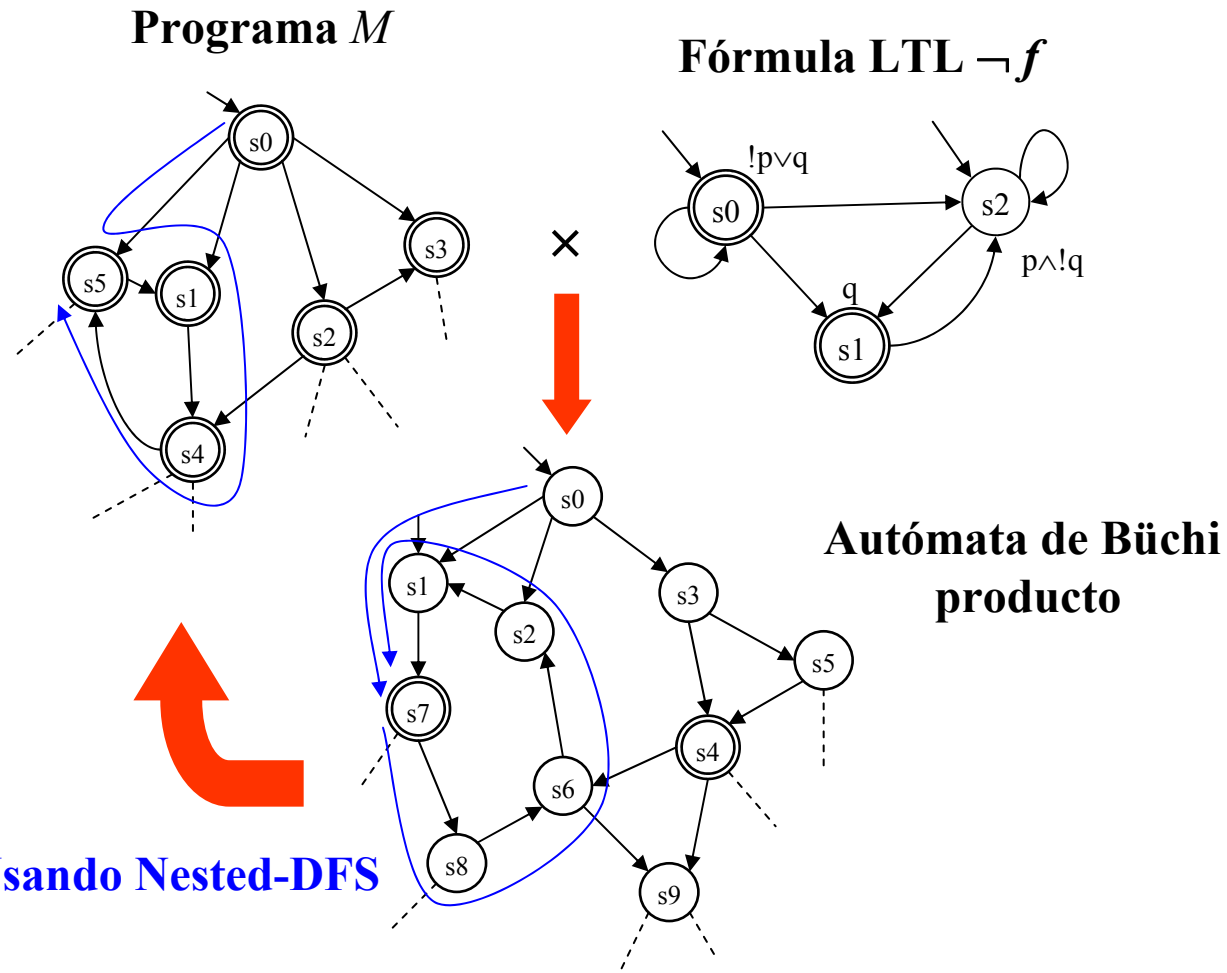
Problema Abordado (I)

- **Objetivo**: encontrar errores en programas concurrentes
- **SPIN**: se especifica una propiedad usando una fórmula LTL *f*



Problema Abordado (I)

- **Objetivo**: encontrar errores en programas concurrentes
- **SPIN**: se especifica una propiedad usando una fórmula LTL f



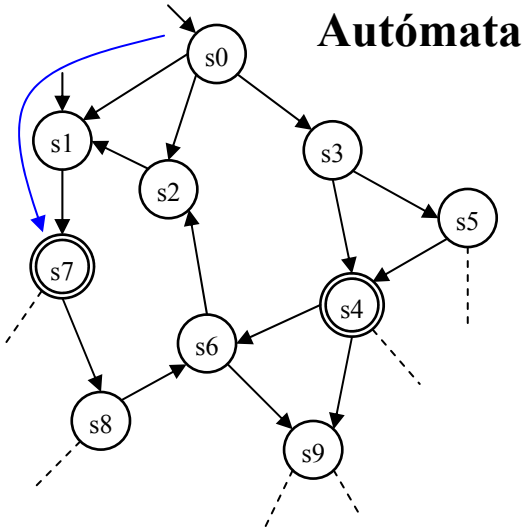
Problema Abordado (II)

- Las **propiedades de seguridad** son aquéllas que pueden expresarse con una fórmula LTL de la forma:

$$f = \Box p$$

donde p es una **fórmula pasada** (sólo con operadores pasados)

- Encontrar una traza de error ≡ encontrar un **estado de aceptación**



Propiedades de Seguridad

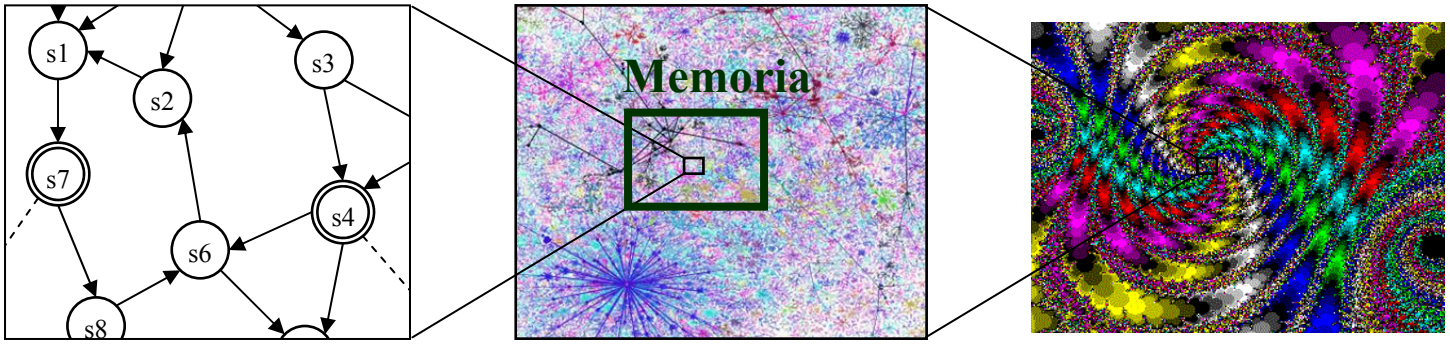
- Interbloqueos
- Invariantes
- Asertos
- …

- Pueden usarse algoritmos clásicos de exploración de grafos: ej., **DFS** y **BFS**

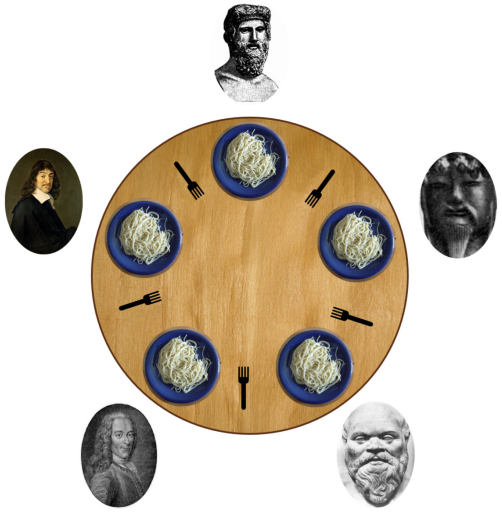
MAEB '07

Problema Abordado (y III)

- **Número de estados muy grande incluso para pequeños programas**



- **El programa usado en los experimentos modela el problema de los filósofos de Edsger Dijkstra**
  - ***n* filósofos → 3^n estados**
  - **20 filósofos → 1039 GB para almacenar los estados**



MAEB '07

Experimentos: Comparación

- Comparamos *MM*AShg con los algoritmos exhaustivos DFS y BFS

Programa	Aspectos	DFS	BFS	*MM*AShg
phi8	Longitud	1338	10	**10.00**
	Mem. (KB)	29696	17408	**4830**
	Tiempo (ms)	70	70	**27**
phi12	Longitud		-	**16.04**
	Mem. (KB)		-	**9344**
	Tiempo (ms)		-	**58**
phi16	Longitud	-	-	**25.40**
	Mem. (KB)	-	-	**18157**
	Tiempo (ms)	-	-	**161**
phi20	Longitud	-		**38.68**
	Mem. (KB)	-		**40628**
	Tiempo (ms)	-		**528**

NO RESUELTOS

- *MM*AShg es claramente mejor para los programas abordados

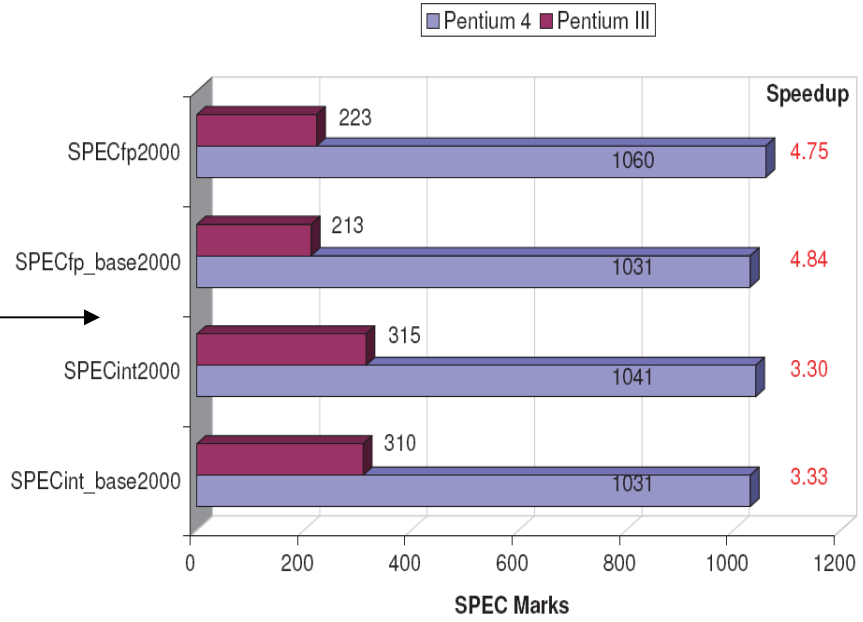
MAEB '07

Experimentos: ACOhg vs. GA

- GA es la única metaheurística aplicada previamente al problema
- Usamos `phi17` y `needham` (protocolo Needham-Schroeder) para la comparación (Godefroid & Khurshid, 2002)

Programa	Algoritmo	Éxito (%)	Tiempo (s)	Mem. (KB)
phi17	GA	52	197.00	n/a
	ACOhg	**100**	**0.28**	11274
needham	GA	3	3068.00	n/a
	ACOhg	**100**	**0.23**	4865

- Los resultados muestran que ACOhg tiene mayor eficacia y eficiencia que GA (incluso teniendo en cuenta la diferencia entre las máquinas)



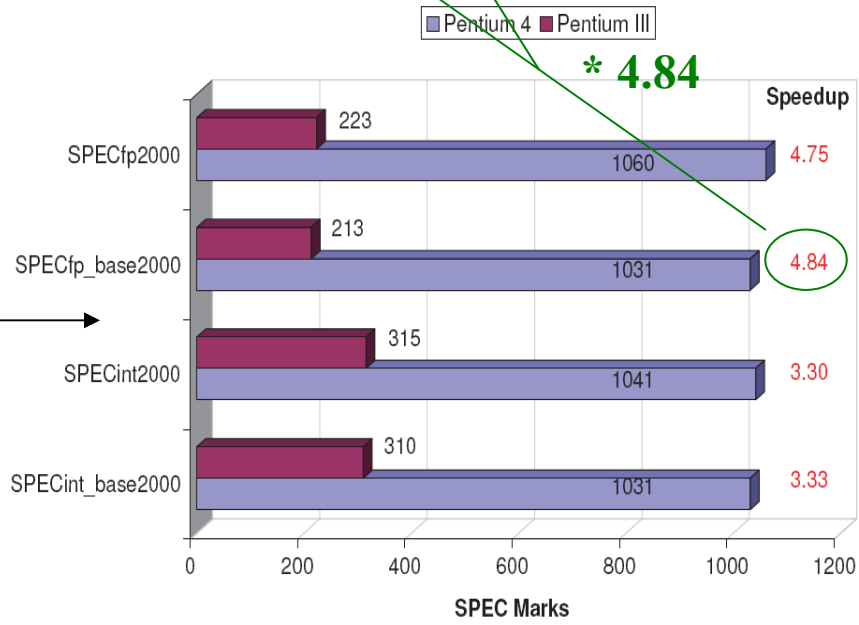
MAEB '07

Experimentos: ACOhg vs. GA

- GA es la única metaheurística aplicada previamente al problema
- Usamos `phi17` y `needham` (protocolo Needham-Schroeder) para la comparación (Godefroid & Khurshid, 2002)

Programa	Algoritmo	Éxito (%)	Tiempo (s)	Mem. (KB)
phi17	GA	52	197.00	n/a
	ACOhg	**100**	**1.36**	11274
needham	GA	3	3068.00	n/a
	ACOhg	**100**	**1.11**	4865

- Los resultados muestran que ACOhg tiene mayor eficacia y eficiencia que GA (incluso teniendo en cuenta la diferencia entre las máquinas)



MAEB '07

Experimentos: Influencia de λ_{ant}

- Analizamos la influencia de λ_{ant} en los resultados

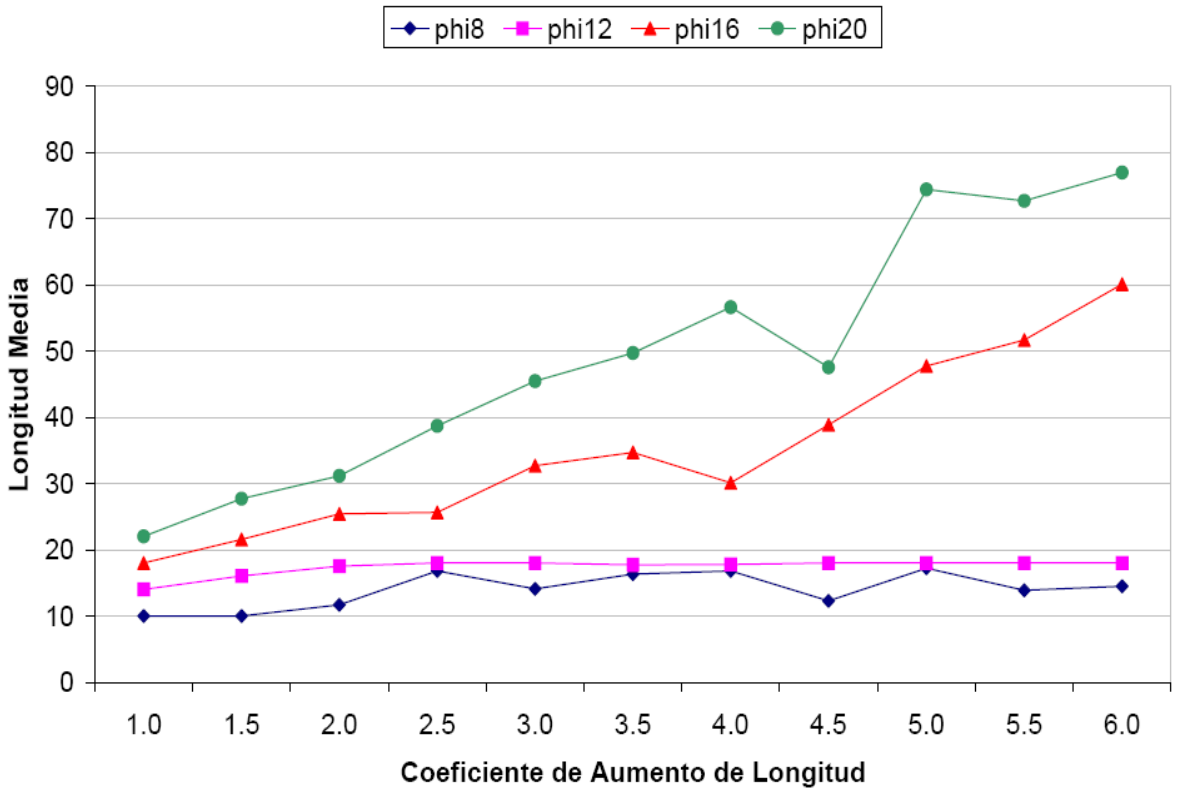
Número de Filósofos	Coeficiente de aumento de longitud (η)										
	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0
8	100	100	100	100	100	100	100	100	100	100	100
12	90	100	100	100	100	100	100	100	100	100	100
16	43	95	100	100	100	100	100	100	100	100	100
20	7	52	95	100	100	100	100	100	100	100	100

- La tasa de éxito aumenta con λ_{ant}

MAEB '07

Experimentos: Influencia de λ_{ant}

- Analizamos la influencia de λ_{ant} en los resultados

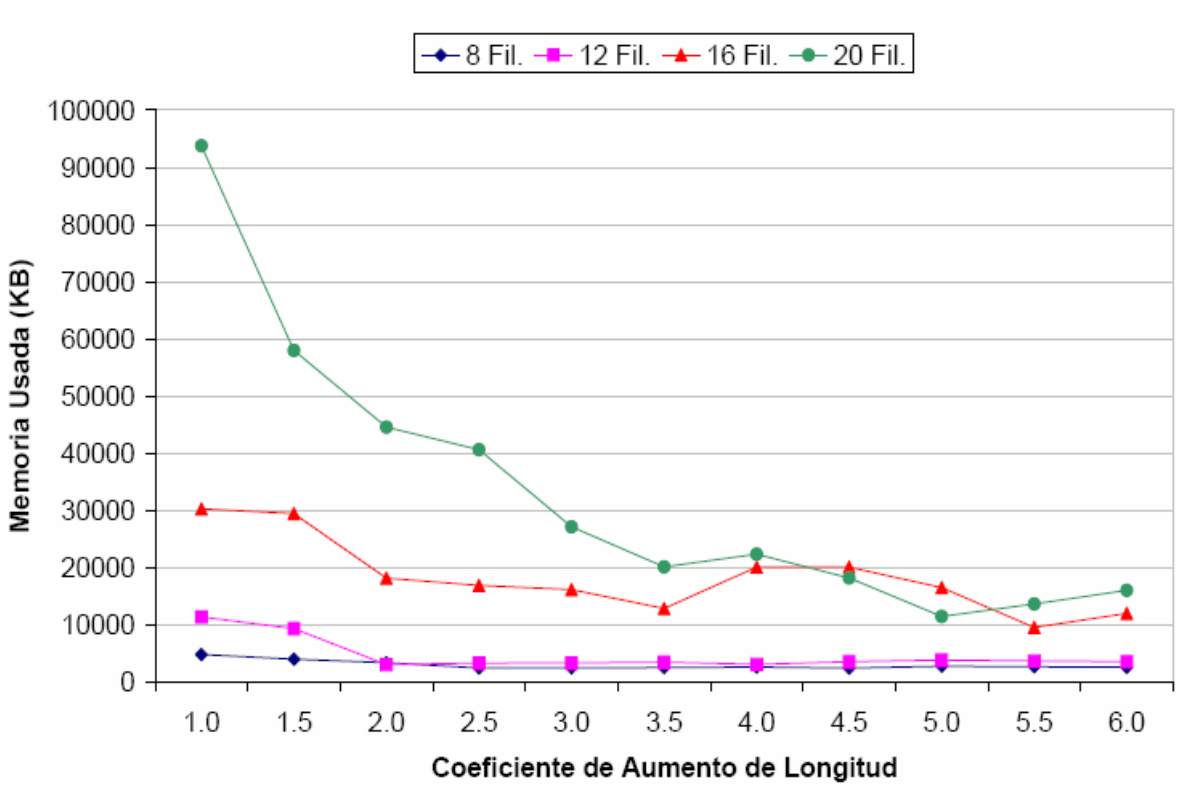


- La tasa de éxito aumenta con λ_{ant}
- La longitud media de los contraejemplos crece con λ_{ant}

MAEB '07

Experimentos: Influencia de λ_{ant}

- Analizamos la influencia de λ_{ant} en los resultados

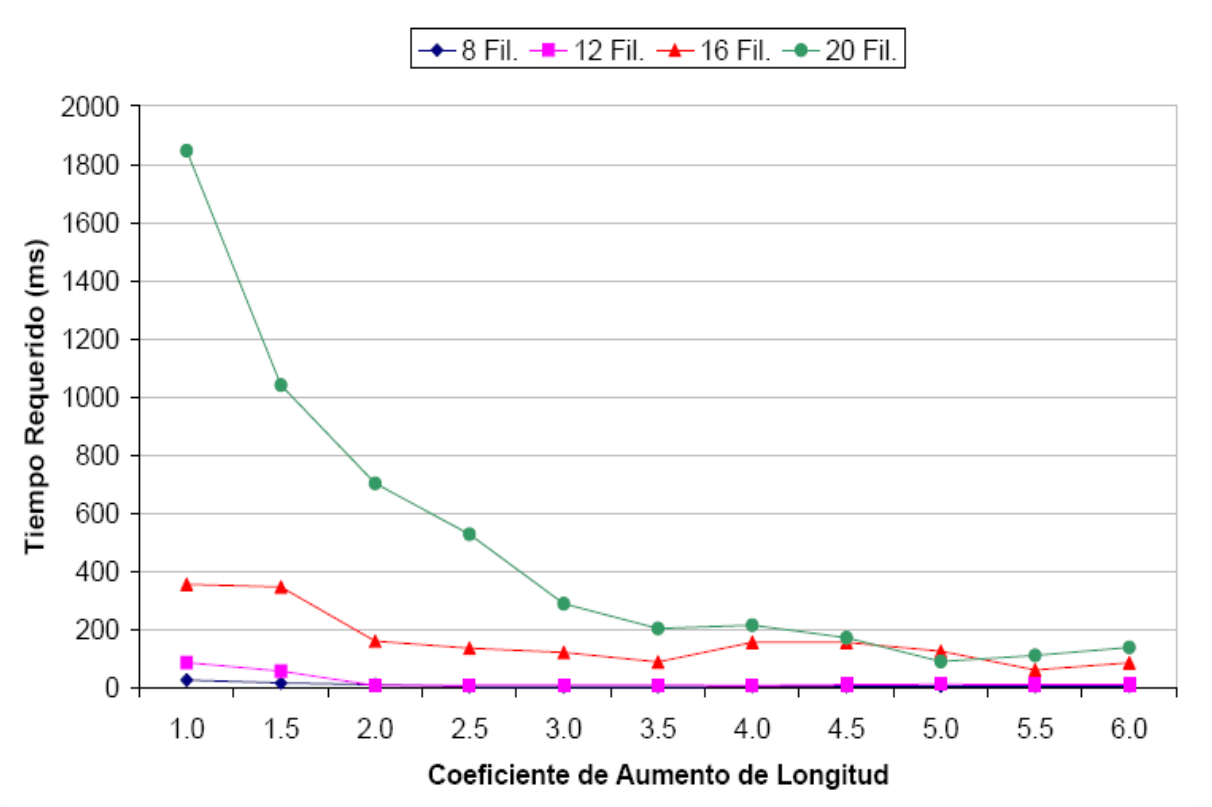


- La tasa de éxito aumenta con λ_{ant}
- La longitud media de los contraejemplos crece con λ_{ant}
- La memoria y el tiempo de cómputo disminuyen con λ_{ant}

MAEB '07

Experimentos: Influencia de λ_{ant}

- Analizamos la influencia de λ_{ant} en los resultados



- La tasa de éxito aumenta con λ_{ant}
- La longitud media de los contraejemplos crece con λ_{ant}
- La memoria y el tiempo de cómputo disminuyen con λ_{ant}

Conclusiones y Trabajo Futuro

Conclusiones

- ACOhg es capaz de superar las limitaciones de los modelos actuales de ACO cuando se enfrentan a problemas con grafos subyacentes desconocidos o de gran extensión
- ACOhg supera a algoritmos exhaustivos del dominio de la comprobación de modelos
- Asimismo obtiene mejores resultados que el GA usado en el pasado para este problema

Trabajo Futuro

- Estudiar el modelo ACOhg en profundidad, explorando todas las alternativas mencionadas
- Trasladar las ideas usadas en ACOhg a otras metaheurísticas para extender el conjunto de problemas al que se pueden aplicar
- Profundizar en el uso de ACOhg para la aplicación de encontrar errores en programas concurrentes

MAEB '07

Fin

Gracias por su atención !!!