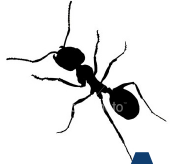


Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Laboratorio de Inteligencia Artificial

Framework basado en Colonias de Hormigas artificiales para la resolución de problemas de optimización

Enrique Areyán
Tutores:
Ignacio Calderón, Haydemar Nuñez

Enero, 2010



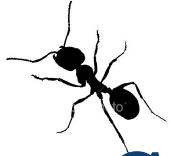
Agenda

- ▶ Introducción
- ▶ Sistemas bioinspirados
- ▶ Colonias de Hormigas (biológicas)
- ▶ Modelo de Colonias de Hormigas (software)
- ▶ Propuesta de Framework
- ▶ Experimentos
- ▶ Demostración
- ▶ Conclusiones
- ▶ Trabajos Futuros



Introducción

- ▶ Sistemas gobernados por reglas muy simples
- ▶ A simple vista pueden parecer caóticos, sin sentido ni aplicación práctica
- ▶ Sin embargo, para algunos sistemas es justo lo contrario. Por Ejemplo: Colonias de Hormigas (CH)



Sistemas Bioinspirados

- ▶ Emulan el comportamiento de sociedades de seres vivos
- ▶ Conjunto de reglas simples + conjunto de organismos simples que obedecen dichas reglas = estructura compleja
- ▶ Un sólo miembro no hubiera podido conseguir un resultado similar



Colonias de Hormigas (biológicas)

- ▶ Sociedad de seres vivos capaces de organizarse y cooperar entre sí
- ▶ Sistema distribuido con miembros muy simples pero que en conjunto presentan una organización social estructurada y compleja
- ▶ Las CH realizan tareas complejas que exceden la capacidad individual de cada hormiga



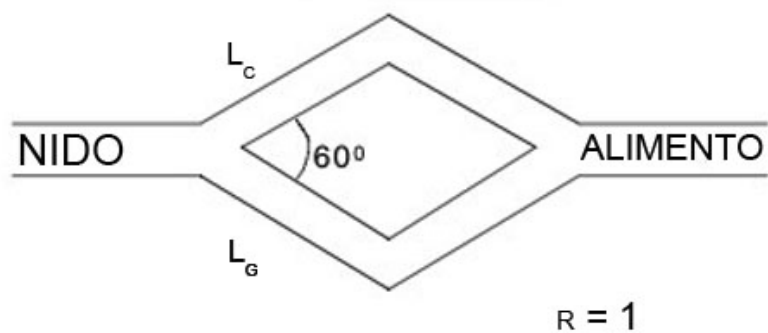
Colonias de Hormigas (biológicas)

- ▶ *Stigmergy*: mecanismo por medio del cual las hormigas coordinan sus actividades
- ▶ Forma indirecta de comunicación a través de pequeñas modificaciones al medio ambiente
- ▶ Una hormiga deposita feromona en el suelo aumentando la probabilidad de que otras hormigas sigan el mismo camino que ésta

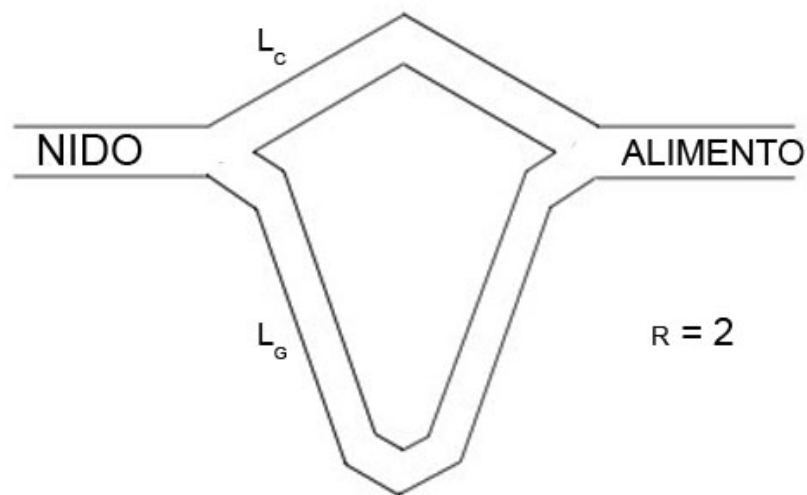


Colonias de Hormigas (biológicas)

CASO 1: $L_g = L_c$



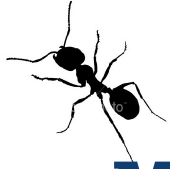
CASO 2: $L_g = 2 * L_c$





Modelo de Colonias de Hormigas (software)

- ▶ Modelo Artificial que emula el comportamiento de las CH naturales en búsqueda de alimento
- ▶ Emplean algún tipo de *stigmergy* artificial o rastros de feromona para coordinar la interacción entre los agentes artificiales
- ▶ Se compone de:
 - ▶ Feromona artificial (matriz)
 - ▶ Hormigas artificiales (agentes, procedimiento estocástico)



Modelo de Colonias de Hormigas (software)

- ▶ Metaheurística Optimización por Colonias de Hormigas (OCH)
- ▶ Enmarca a los principales algoritmos
- ▶ Se compone de:
 - ▶ Construir soluciones de hormigas
 - ▶ Actualizar Feromonas
 - ▶ Actividades misceláneas



Modelo de Colonias de Hormigas (software)

- ▶ Principales algoritmos basados OCH
 - ▶ Sistema de Hormigas (SH)
 - ▶ Sistema de Hormigas Élite (SHE)
 - ▶ Sistema de Hormigas basado en rangos (SH_{rango})
 - ▶ Sistema de Hormigas Máximo-Mínimo (SHMM)
 - ▶ Sistema de Colonias de Hormigas (SCH)
 - ▶ Hormiga-Q



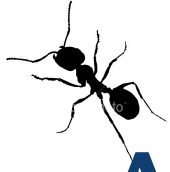
Modelo de Colonias de Hormigas (software)

Regla proporcional aleatoria de SH

$$p(k)_{ij} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \text{ si } j \in N_i^k$$

Características

- ▶ k = hormiga
- ▶ (i, j) = nodo i , nodo j
- ▶ N_i^k = conjunto de nodos accesibles para la hormiga k situada en el nodo i
- ▶ η_{ij} = info. Heurística
- ▶ τ_{ij} = rastro de feromonas



Algunas Aplicaciones

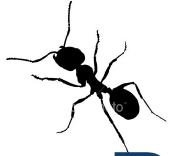
- ▶ Minería del uso de la Internet utilizando Agrupamiento por el SCH Abraham (A. & Ramos, V., 2003)
- ▶ Sistema Colonia de Hormigas en un Ambiente Paralelo Asíncrono (Barán, B. & Almirón, M., 2002)
- ▶ Optimización por Colonias de Hormigas para enrutamiento de vehículos en sistemas de logística avanzados (Gambardella L.M., Rizzoli A.E., Oliverio F., Casagrande N., Donati A.V., Montemanni R. & Lucibello E., 2003)



Propuesta de Framework

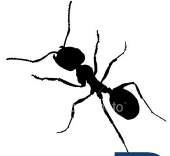
Objetivo General

- ▶ Construir una herramienta basada en software libre que incorpore los distintos algoritmos de Colonias de Hormigas y que pueda ser utilizado por la comunidad científica en general.



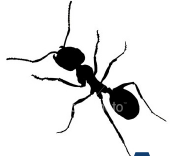
Propuesta de Framework

- ▶ Se basará en la construcción de caminos desde un nodo origen hacia un nodo destino (búsqueda de alimento)
- ▶ Construido bajo el paradigma de programación orientado a objetos
- ▶ Proveerá al programador un API con los principales algoritmos

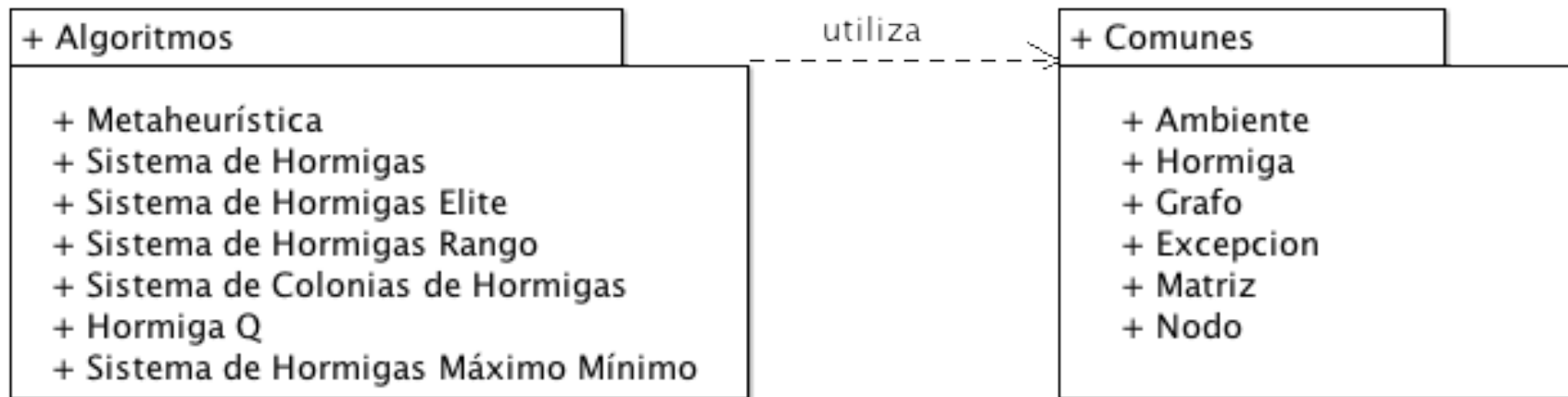


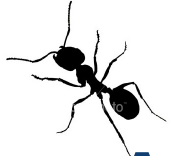
Propuesta de Framework

- ▶ El Framework podrá ser extendido para adaptarse a necesidades particulares
- ▶ Se construirá sobre la última versión del lenguaje de programación JAVA (JAVA SE 1.6)
- ▶ No requerirá ninguna librería externa que no se distribuya con la versión estándar

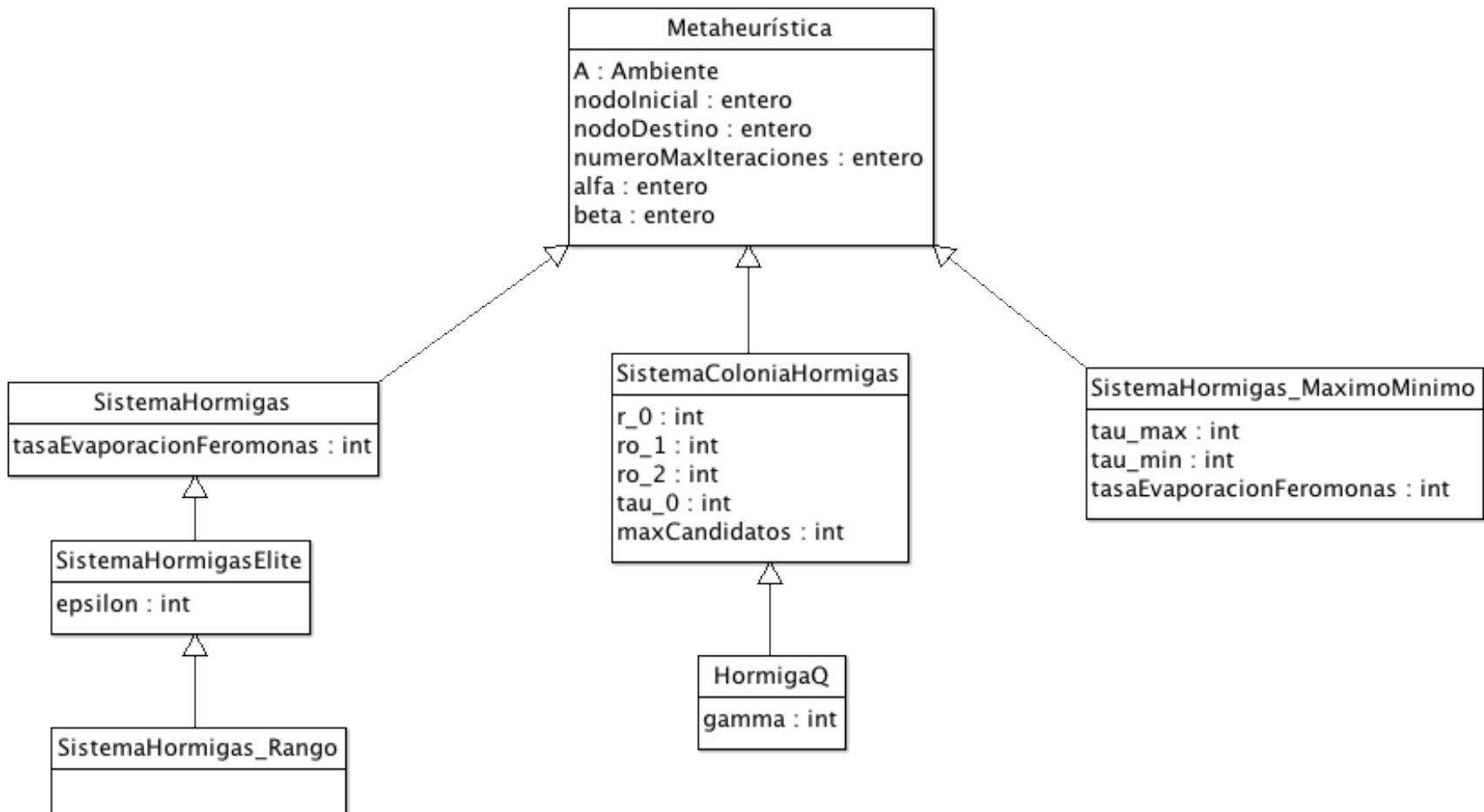


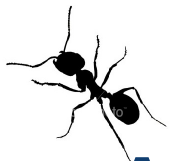
Arquitectura del Framework





Arquitectura del Framework





Arquitectura del Framework

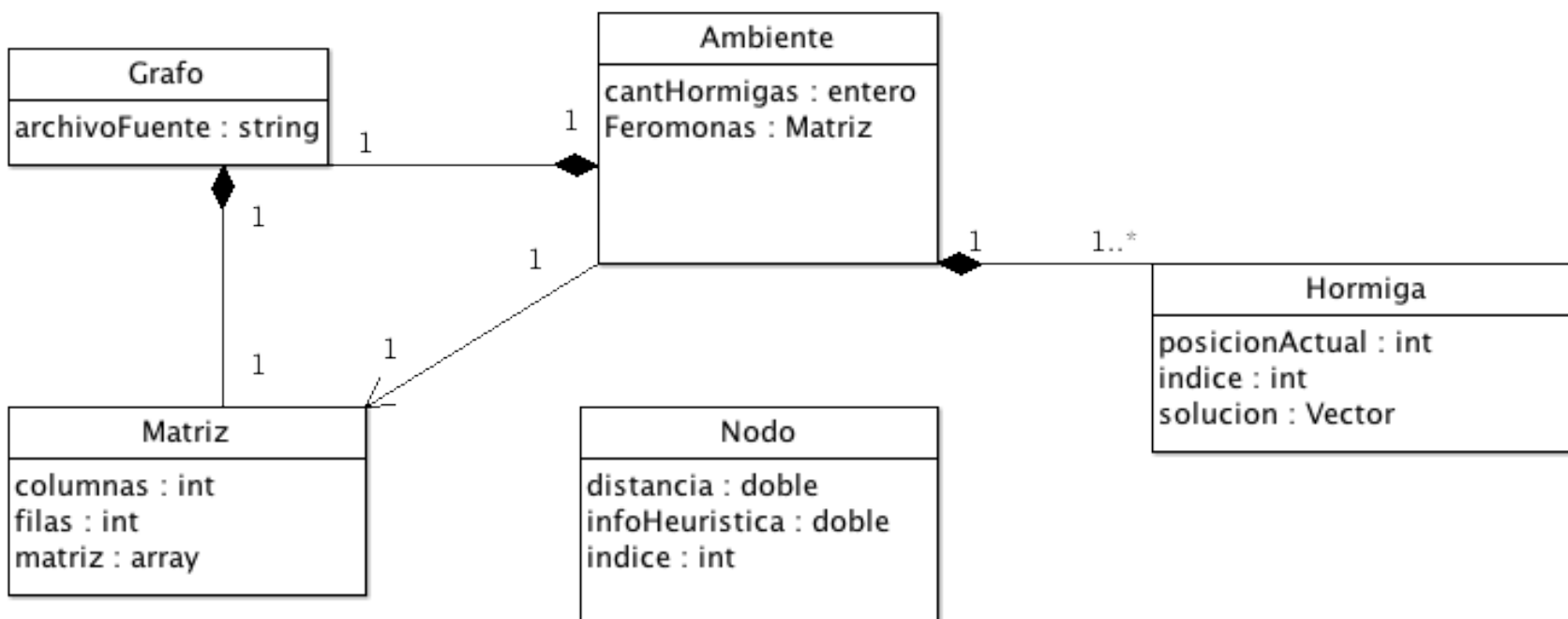




Diagrama de Colaboración

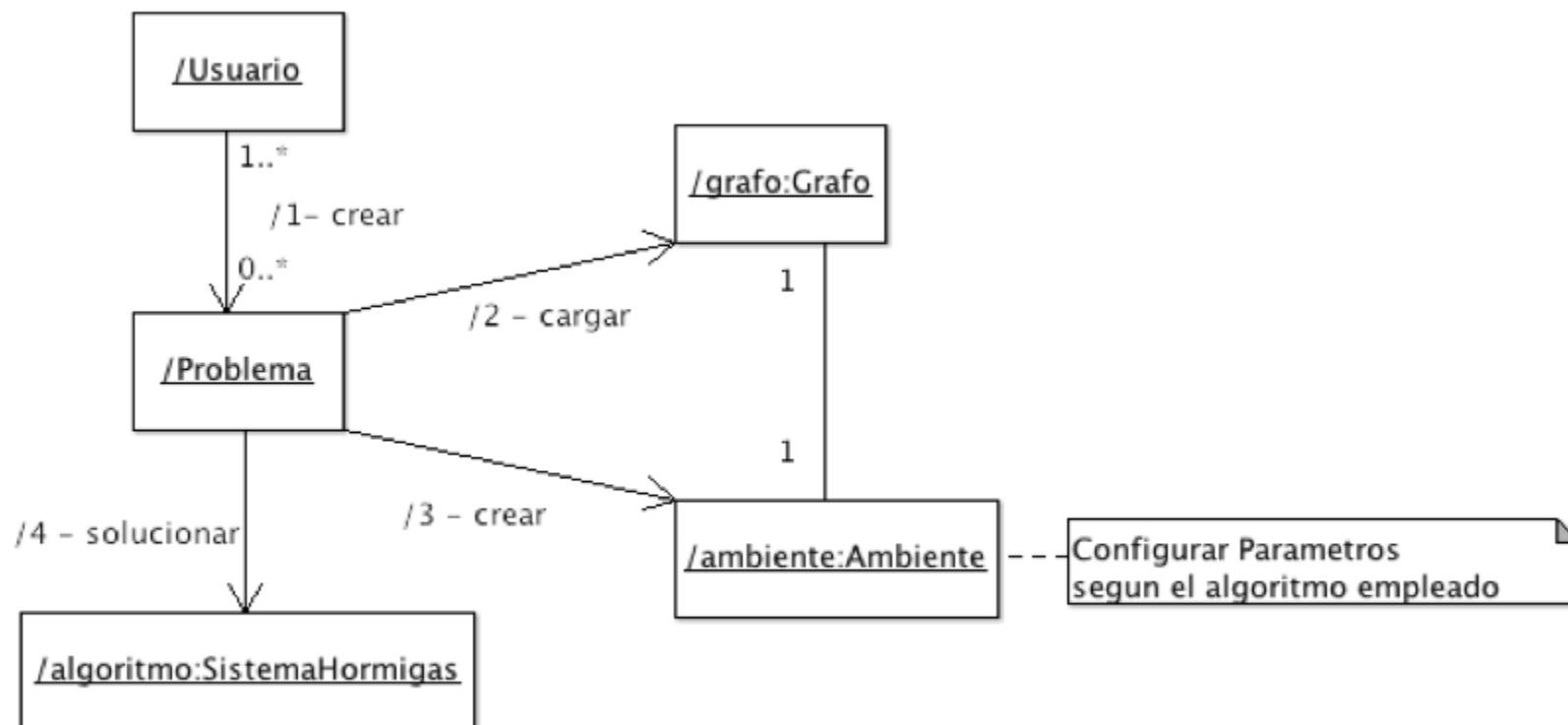




Diagrama de Secuencia

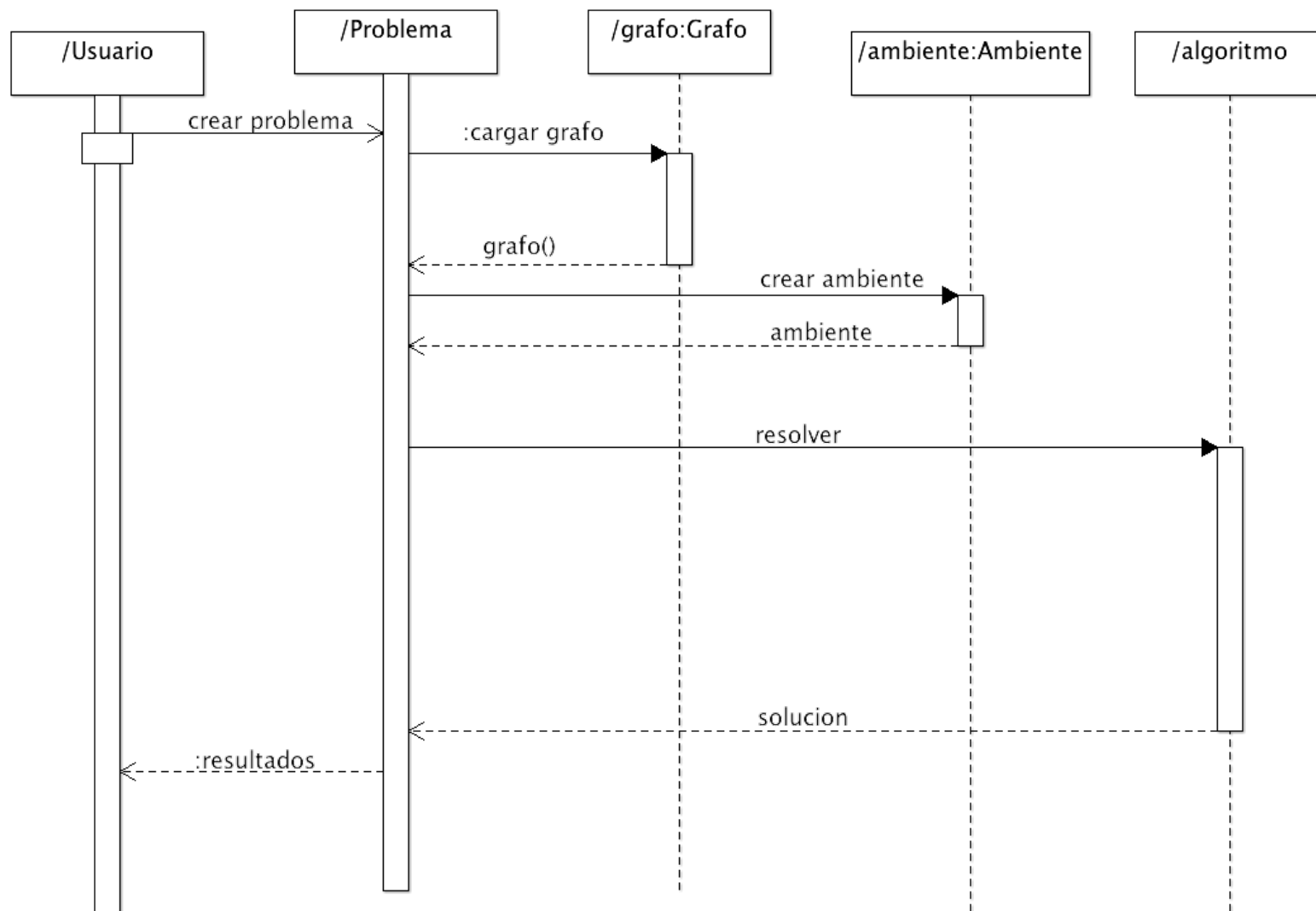




Diagrama de Secuencia para SH

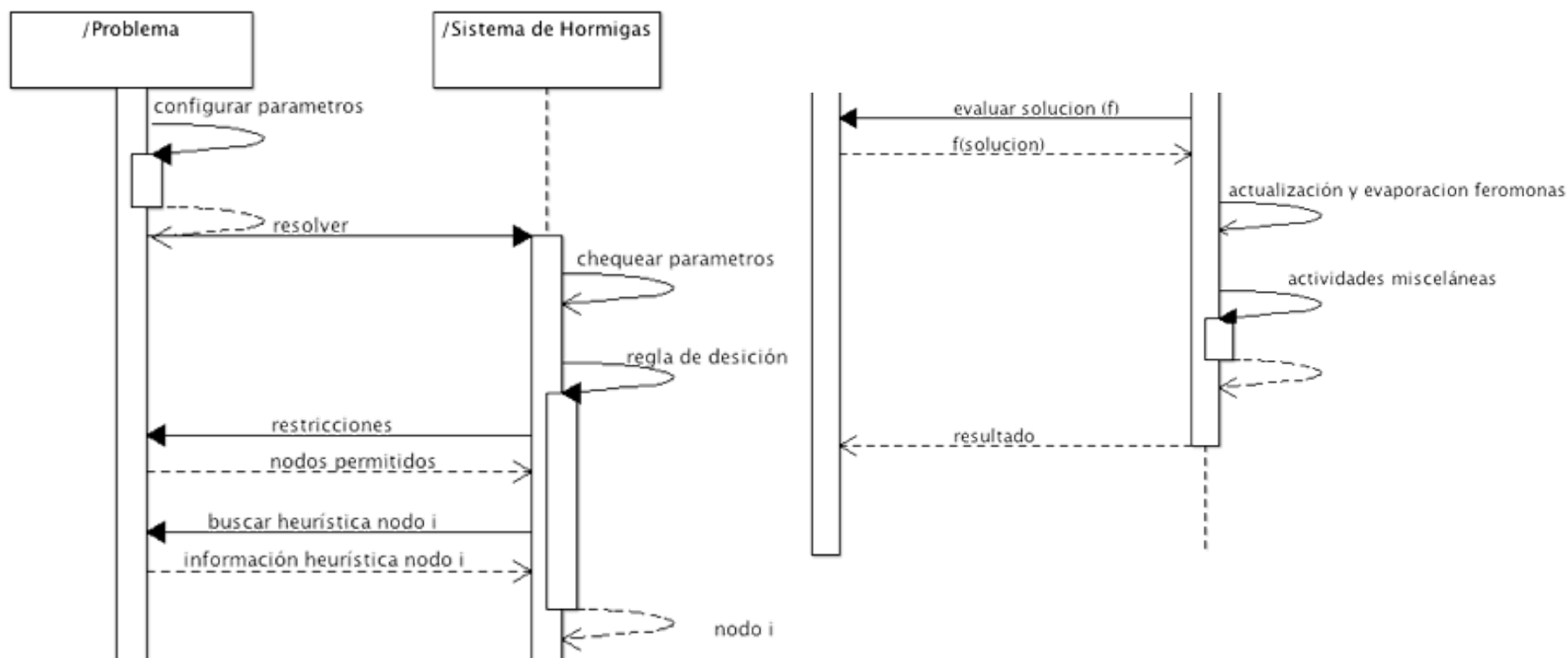




Diagrama de Secuencia para SHE

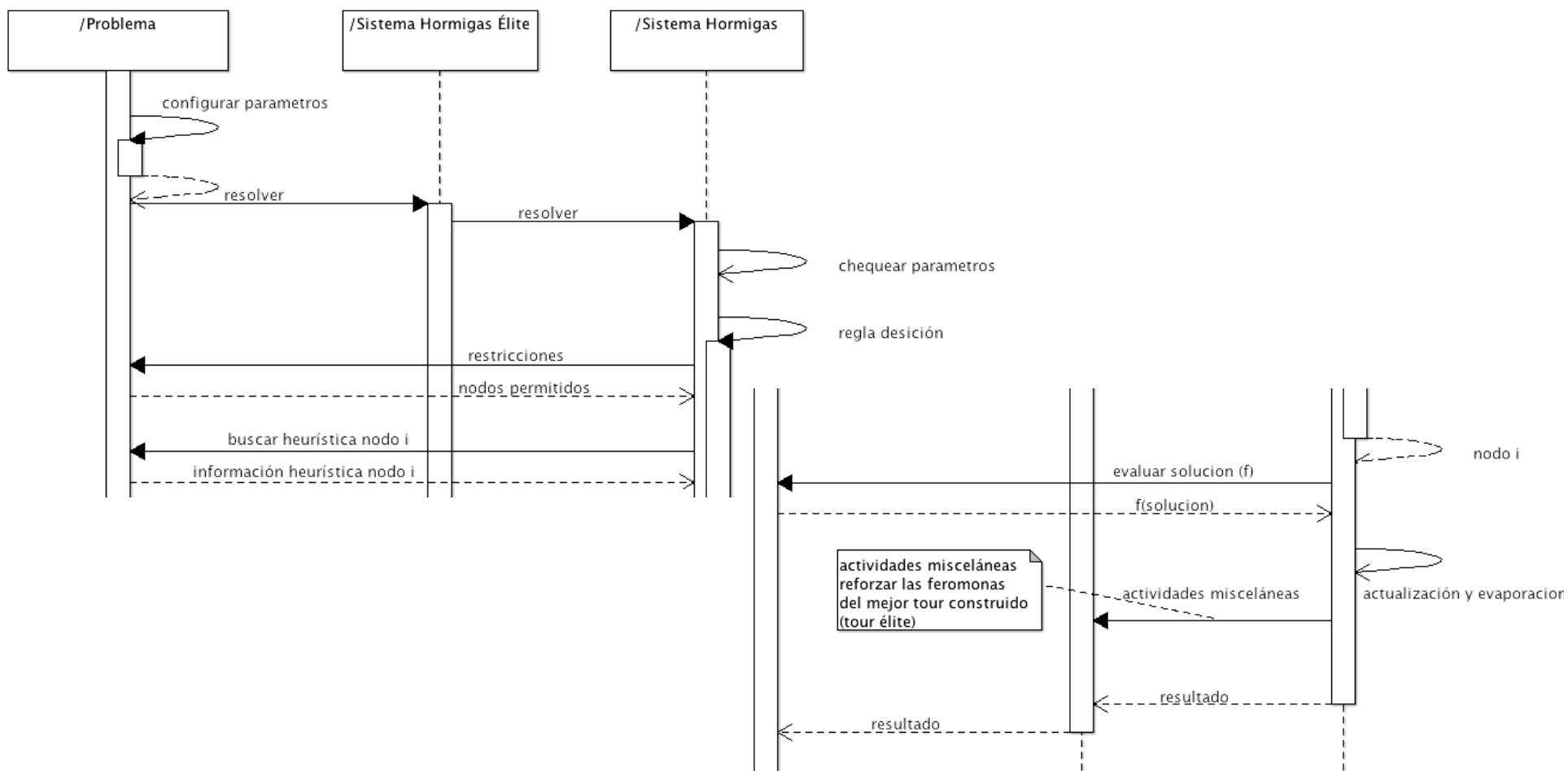




Diagrama de Secuencia para MMAS

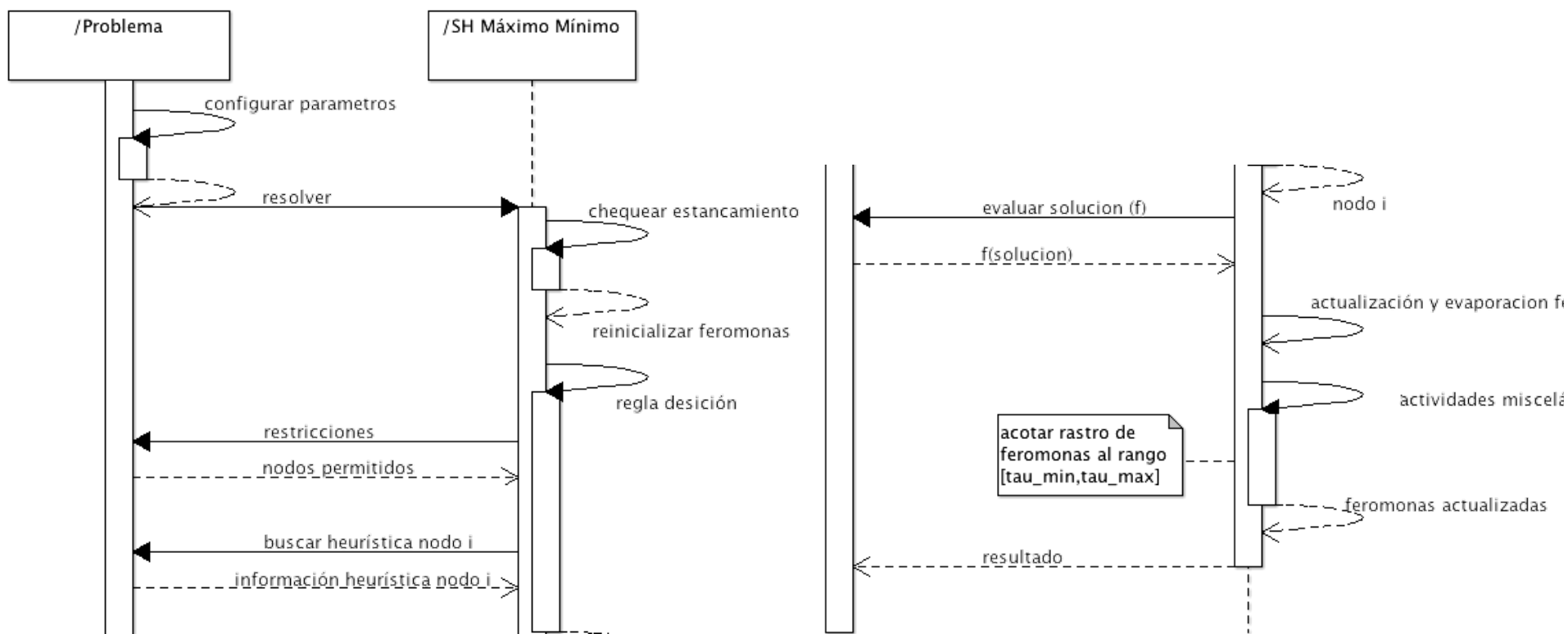
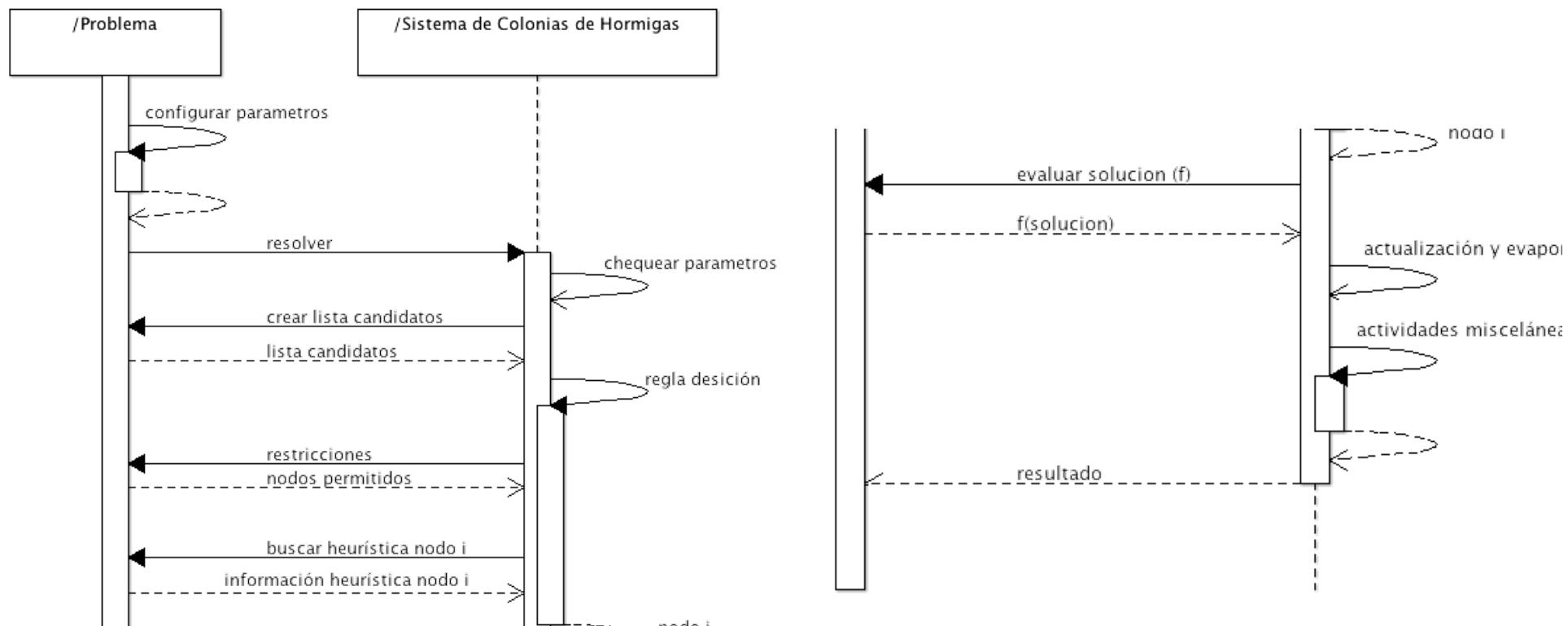
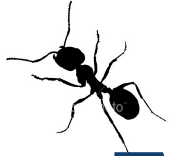




Diagrama de Secuencia para ACS



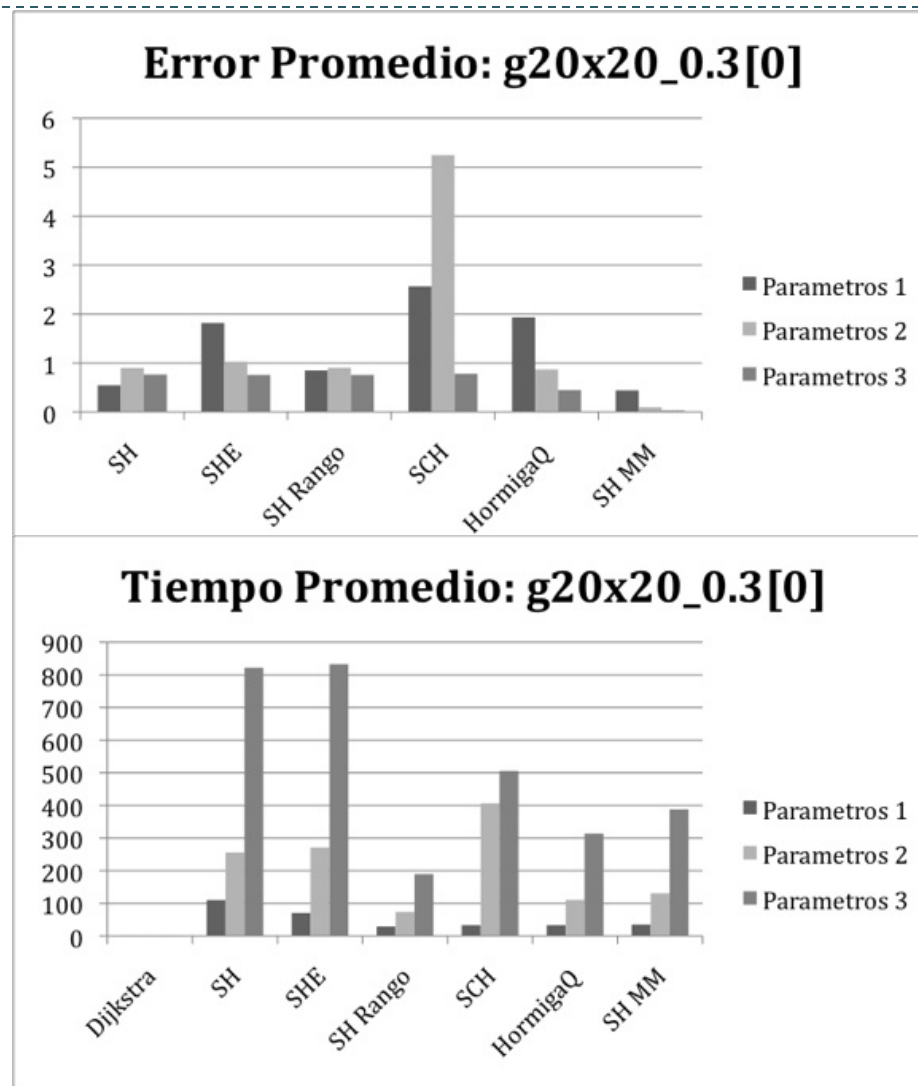


Experimentos

- ▶ Dimensiones: calidad y velocidad
- ▶ Ruta más corta
 - ▶ Dijkstra
- ▶ Recorrido de Caballo
 - ▶ NP-completo con heurísticas altamente eficientes
- ▶ Problema del Agente Viajero
 - ▶ *Backtracking* completo
 - ▶ Algoritmo genético

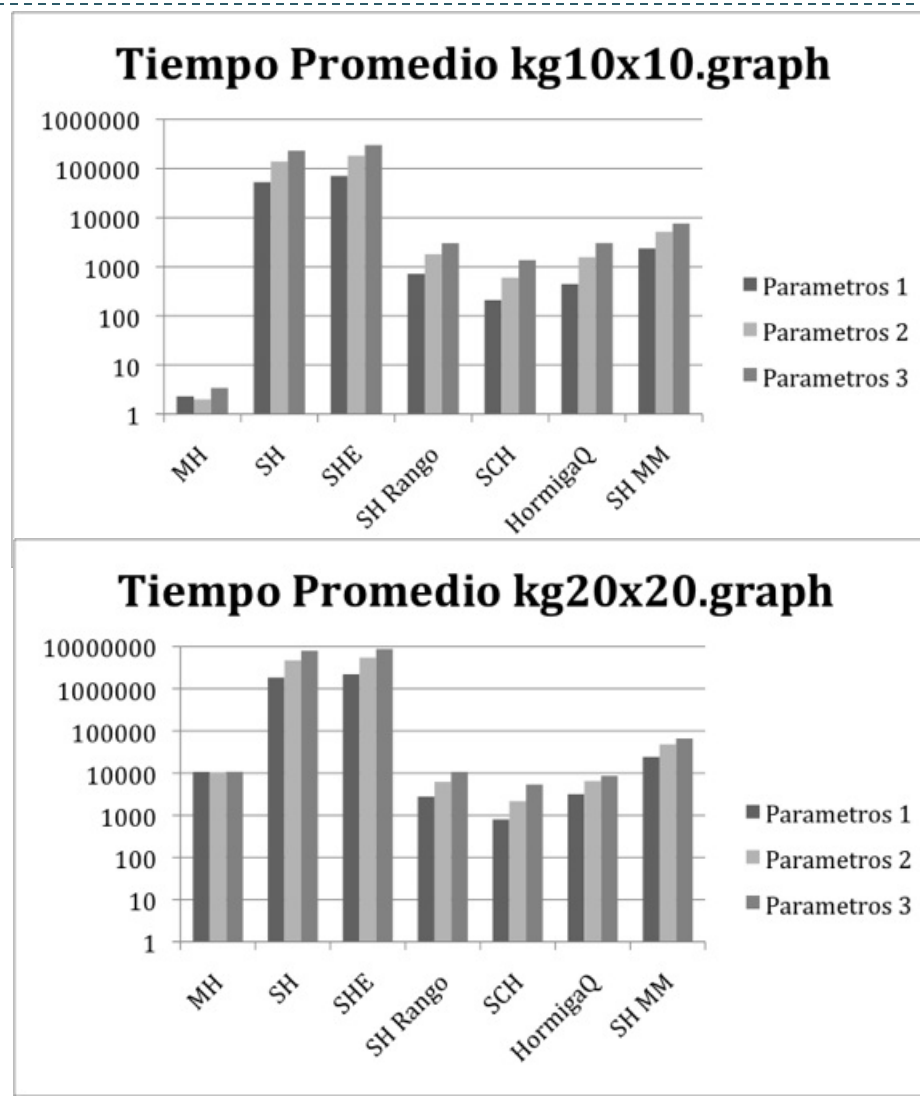


Experimentos: Ruta más corta



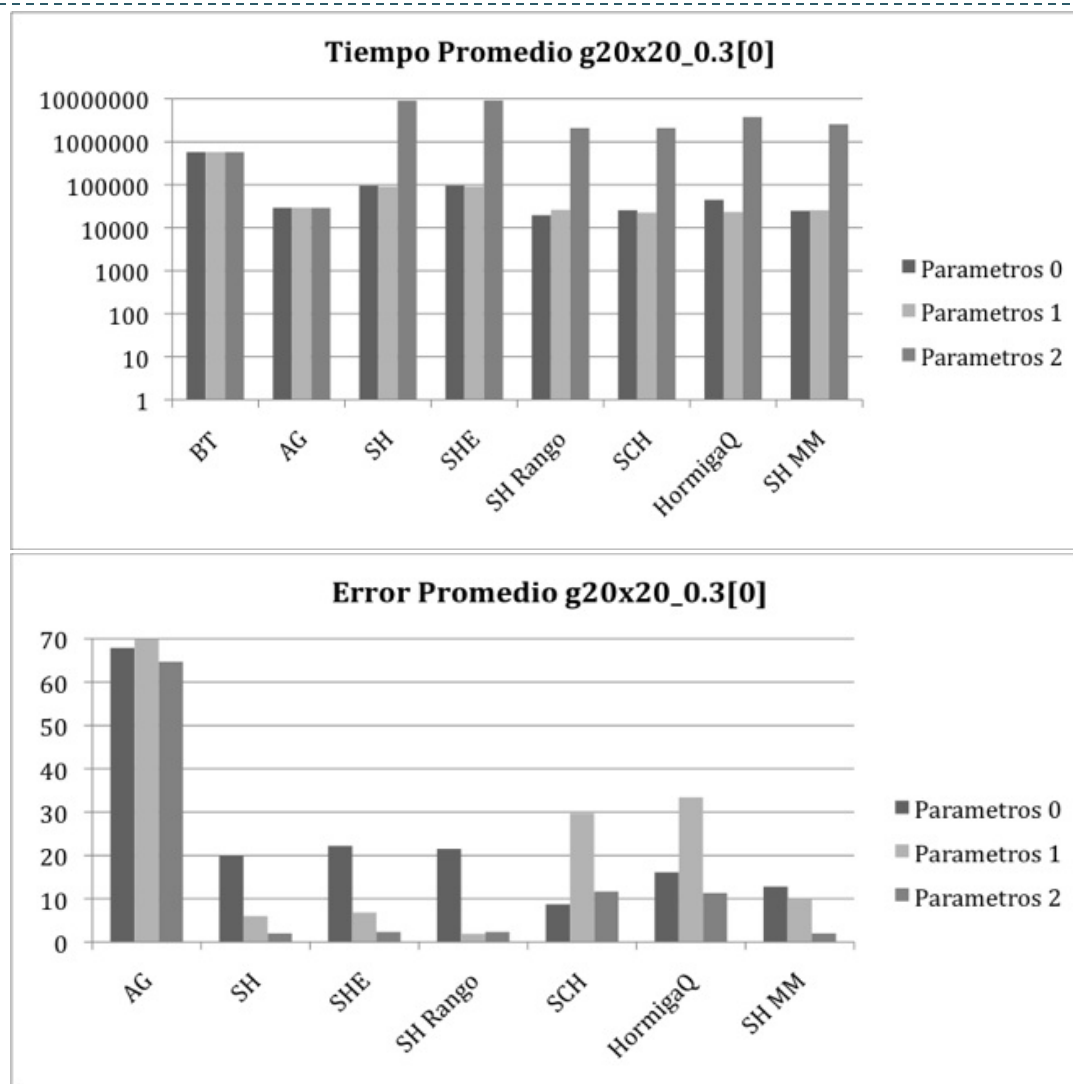


Experimentos: Recorrido de Caballo



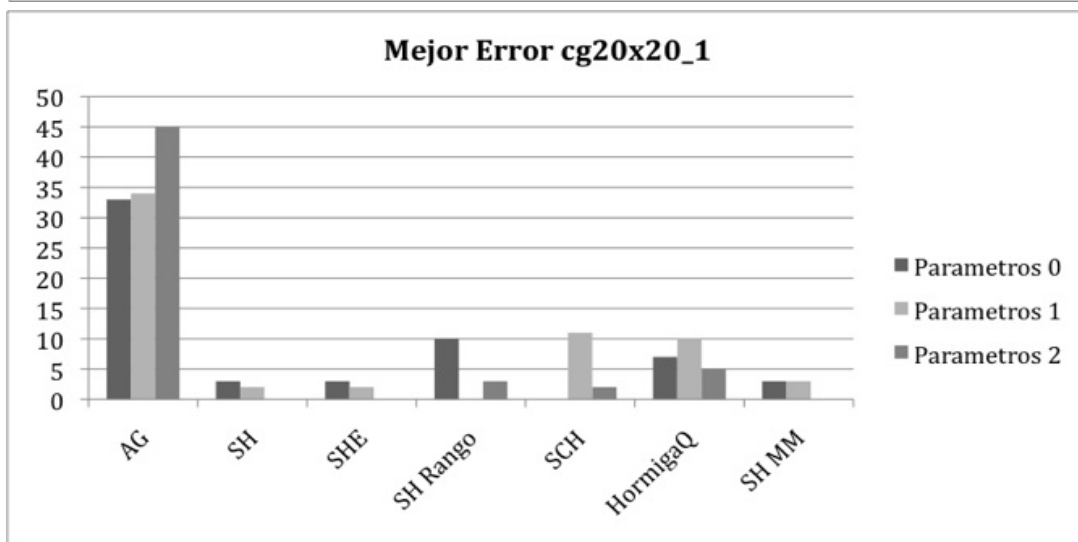
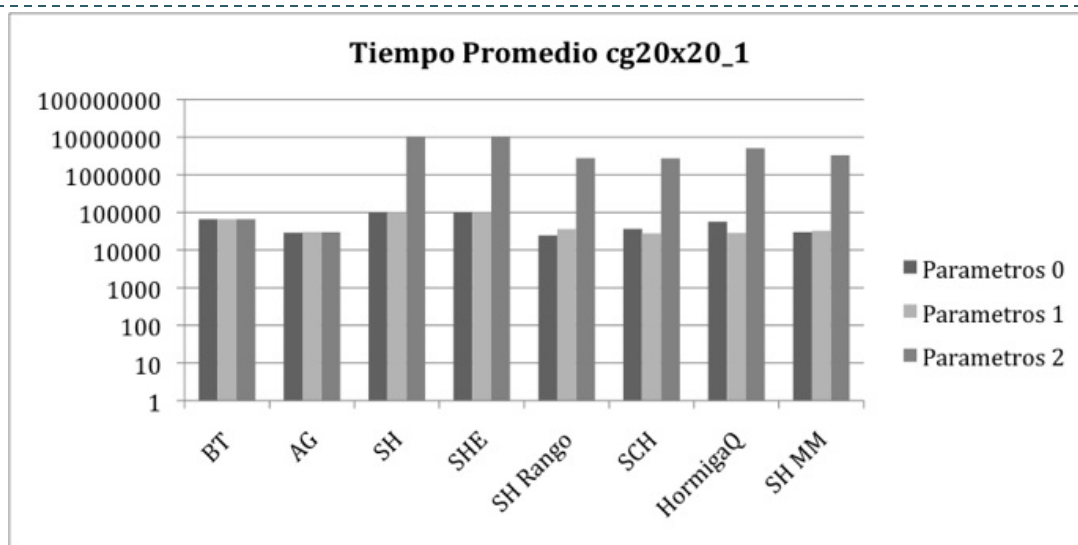


Experimentos: Agente Viajero





Experimentos: Agente Viajero





Demostración

AntFramework v1.0

Opciones

Inicio g20x20_0.3[0].graph

Parametros

Algoritmo:
(SH) Sistema de Hormigas

Iteraciones: 1,000

Hormigas: 100

Nido: 10

Fuente: 15

Otros parametros:

Parametro	Valor
Tasa de evaporacion	0.6
Alfa	0.1
Beta	0.8

Ejecutar

Grafo

0	1	2	3	4	5
-1.0	9.0	11.0	6.0	19.0	12.0
-1.0	-1.0	20.0	-1.0	18.0	-1.0
20.0	6.0	-1.0	-1.0	12.0	2.0
10.0	5.0	14.0	-1.0	16.0	2.0
-1.0	13.0	-1.0	13.0	-1.0	2.0
15.0	20.0	-1.0	18.0	4.0	-1.0

Guardar

Resultados

Algoritmo	Solucion	f(Solucion)	Velocidad (segs)
-----------	----------	-------------	------------------

Ver Trazas

Limpiar

Cerrar Pestaña



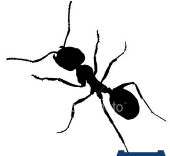
Conclusiones

- ▶ Es posible realizar una abstracción de las variantes de los algoritmos de las CH, con el fin de organizar una jerarquía de clases y objetos
- ▶ Dichas clases y objetos se conjugaron para formar un Framework en el cual se abstraen las funcionalidades básicas de cualquier algoritmo



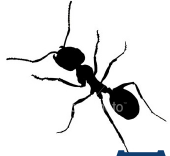
Conclusiones

- ▶ Bajo ciertas condiciones los algoritmos son una opción viable en contraste con otras opciones como algoritmos NP-completos y algoritmos genéticos
- ▶ La calidad de la solución está directamente relacionada a la calidad de la heurística utilizada



Trabajos Futuros

- ▶ Ampliar el Framework para incluir nuevos algoritmos actualmente en investigación.
- ▶ Extender el Framework para que funcione en aplicaciones distribuidas y/o en paralelo.
- ▶ Explorar el rendimiento de los algoritmos en la resolución de otros problemas de optimización



Trabajos Futuros

- ▶ Verificar la aplicabilidad de los algoritmos en el área de minería de datos: agrupamiento, minería de textos o descubrimiento de conocimiento.