Introducción

ROS

Evaluación

Conclusiones y Trabajo Futuro

ROS: Servicio de Optimización en Internet

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

UNIVERSIDAD DE MÁLAGA

Enrique Alba, José Nieto y Francisco Chicano

Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Optimización Combinatoria

- Un problema de optimización combinatoria está formado por:
  - Variables: $x_1, x_2, \ldots, x_n$
  - Dominios para las variables: $D_1, D_2, \ldots, D_n$
  - Restricciones entre variables
  - Función objetivo a minimizar $f : D_1 \times \ldots \times D_n \rightarrow \mathbf{R}^+$
- Espacio de soluciones (asignaciones factibles)

$$S = \{ s = \{(x_1, v_1), \ldots, (x_n, v_n)\} \mid v_i \in D_i, s \text{ satisface las restricciones}\}$$

- El objetivo es encontrar un mínimo global $s^* \in S$ tal que

$$f(s^*) \leq f(s) \; \forall s \in S$$

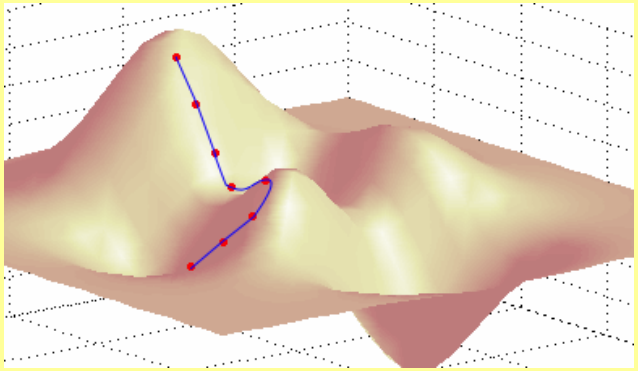
Algoritmos para COPs

- **Exactos**: aseguran la solución óptima
- **Aproximados**: no la aseguran, pero pueden encontrar una solución "buena" en un tiempo "razonable"

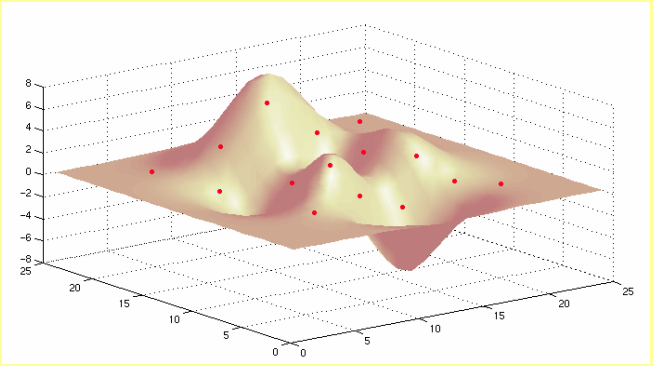
Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Metaheurísticas

Trayectoria



Población



Algoritmos para COPs

- **Exactos: aseguran la solución óptima**
- **Aproximados: no la aseguran, pero pueden encontrar una solución "buena" en un tiempo "razonable"**

Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

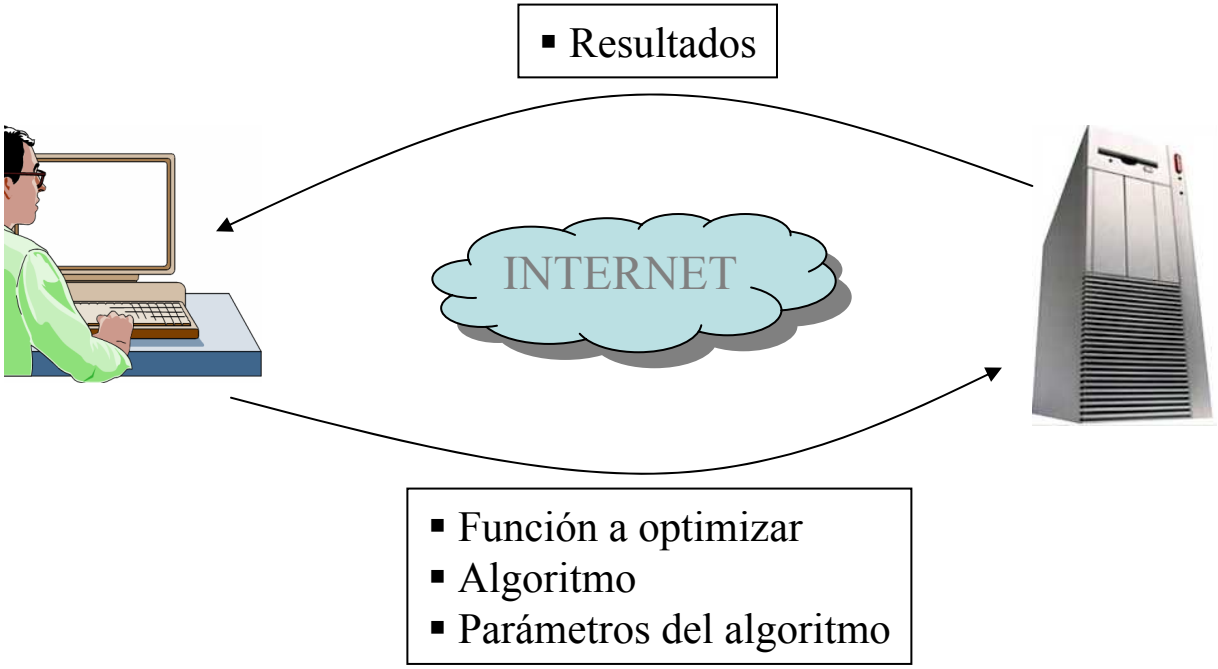
Metaheurísticas

Trayectoria	**Población**
• Simulatd Annealing • Tabu Search • Variable Neighborhood Search • Iterated Local Search • Guided Local Search • GRASP	• Evolutionary Algorithms • Ant Colony Optimization • Scatter Search • EDAs • Particle Swarm Optimization

Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Objetivo

- **Existe una gran variedad de bibliotecas de Metaheurísticas**
  - **MALLBA (C++), GAlib (C++), EO (C++), paradisEO (C++), JEO (Java), parSA (C++), toolbox para GAs y DS (MATLAB), GEATbx (MATLAB), OPEAL (Perl), JCLEC (Java)**
- **Objetivo: hacer disponibles a través de internet una gran variedad de algoritmos metaheurísticos**



Remote Optimization Service

- ROS ofrece un servicio de ejecución de algoritmos de optimización para resolver problemas definidos por el usuario

Características

- Implementado en Java (multiplataforma)
- Sigue el modelo Cliente/Servidor
- Usa XML para el intercambio de información
- Dos versiones: Sockets y SOAP
- Sin limitación respecto al lenguaje de los algoritmos (wrappers)

Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Introducción

ROS

Evaluación

Conclusiones y Trabajo Futuro

Estructura (I)

- Compuesto por cuatro componentes:
  - Cliente (Client) 
  - Servidor Primario (Primary Server) 
  - Servidor de Distribución (Server) 
  - Servidor de Proceso (Worker) 

CLIENTE

- Conecta con el Servidor Primario para gestionar la identificación y registro de usuarios
- Obtiene del Servidor Primario información sobre los servidores de distribución y tipos de algoritmos
- Conecta con el Servidor de distribución para enviar y recibir los datos de la ejecución (XML)
- Ofrece la interfaz gráfica para la selección de datos y E/S de información

Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Estructura (II)

SERVIDOR PRIMARIO

- Gestiona el registro e identificación de usuarios
- Sirve al cliente el fichero de configuración de la red conteniendo:
  - Direcciones de los servidores de distribución
  - Tipos de algoritmos a los que pueden acceder estos servidores

SERVIDOR DE DISTRIBUCIÓN

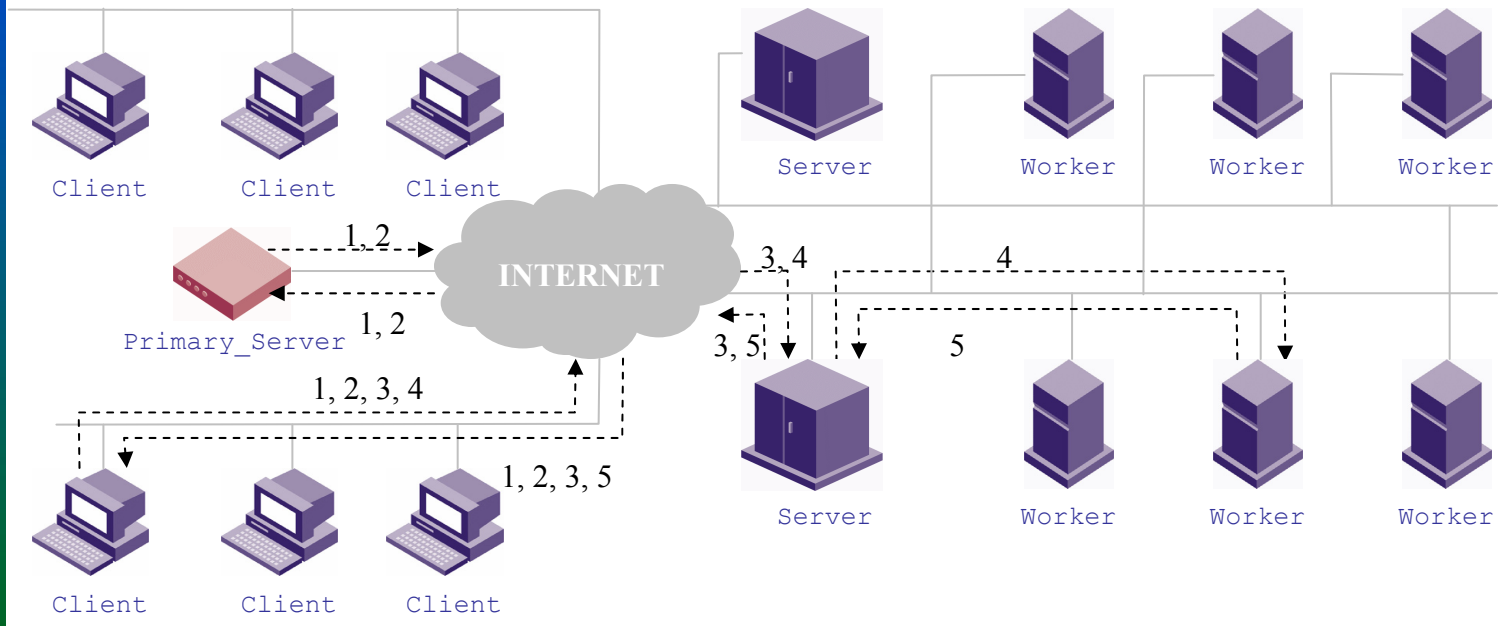
- Nexo de unión entre el cliente y el servidor de proceso
- Proporciona al cliente el fichero XML correspondiente al algoritmo seleccionado
- Obtiene del cliente los datos de ejecución y parámetros (XML) y los envía al worker
- Recoge los resultados (XML) y los envía al cliente

Estructura (y III)

Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

SERVIDOR DE PROCESO

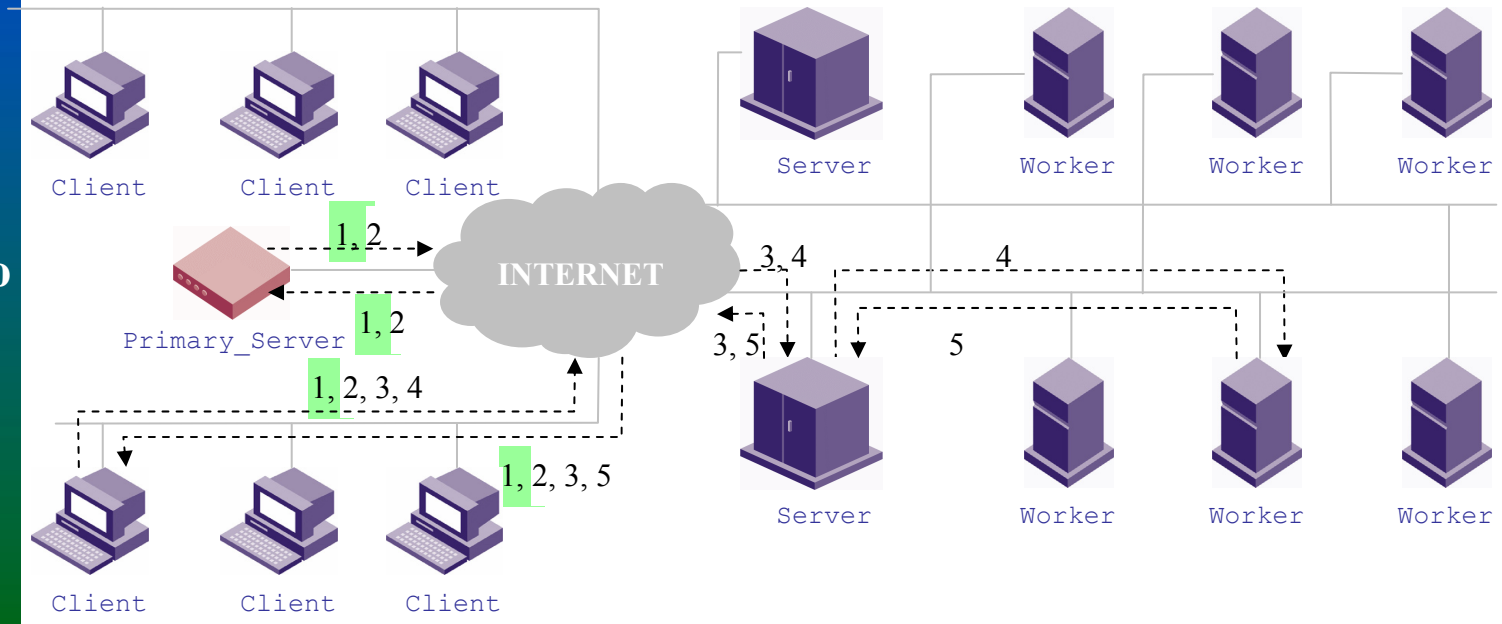
- Recibe del servidor de distribución los datos de ejecución (XML)
- Ejecuta los algoritmos
- Envía los resultados al servidor de distribución (XML)



Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Funcionamiento

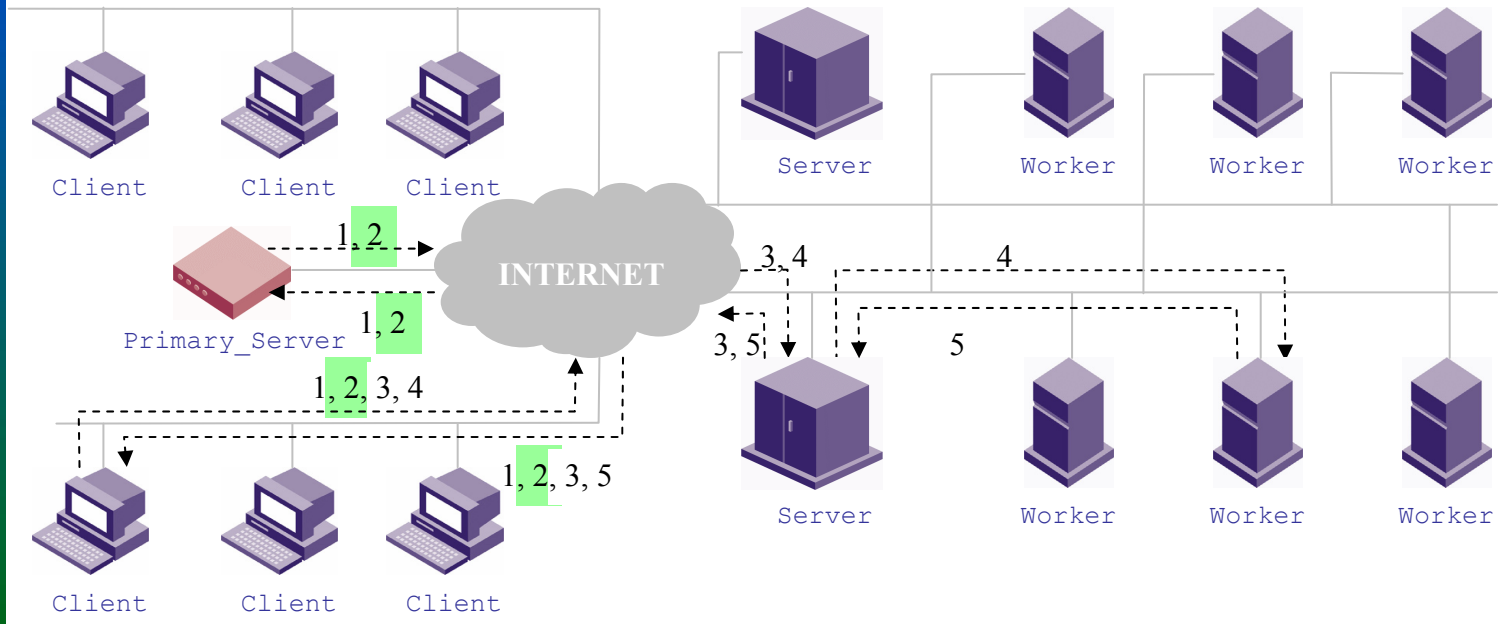
1. **Identificación de usuario**
2. **Selección del algoritmo y servidor de distribución**
3. **Petición de parámetros del algoritmo**
4. **Ejecución del algoritmo**
5. **Obtención de resultados**



Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Funcionamiento

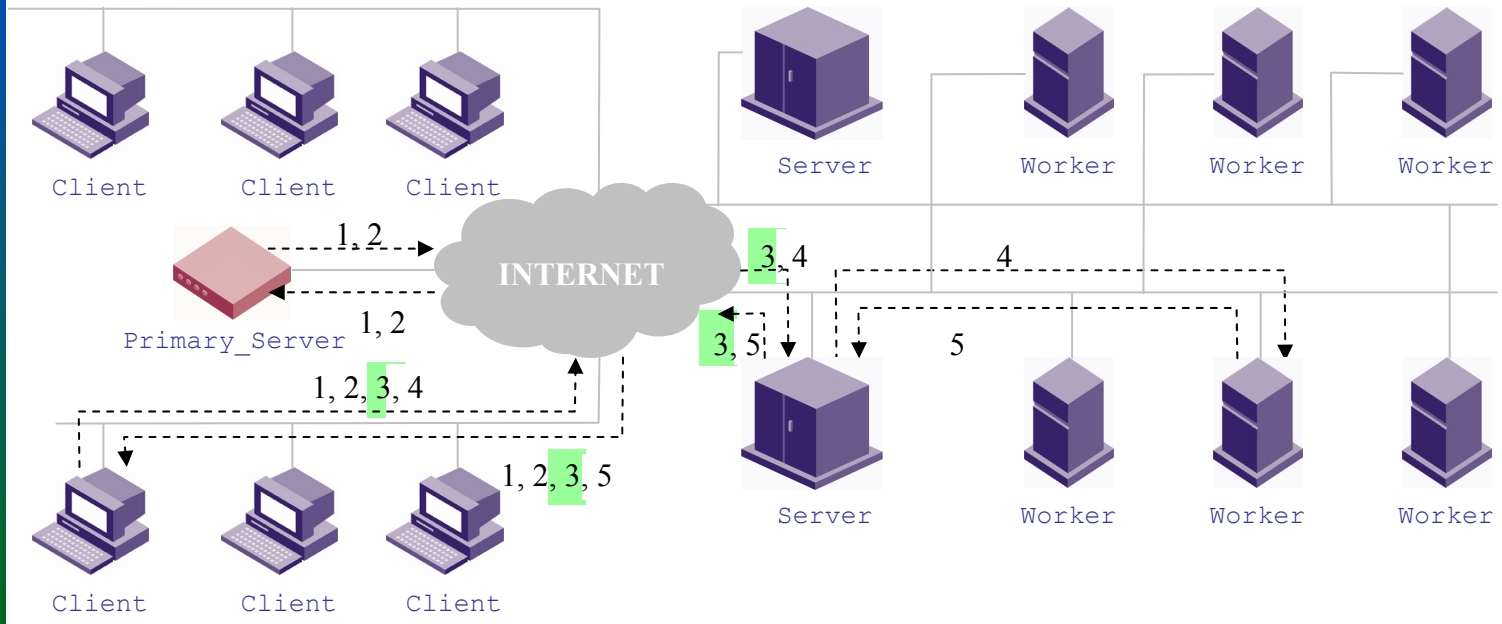
1. Identificación de usuario
2. Selección del algoritmo y servidor de distribución
3. Petición de parámetros del algoritmo
4. Ejecución del algoritmo
5. Obtención de resultados



Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Funcionamiento

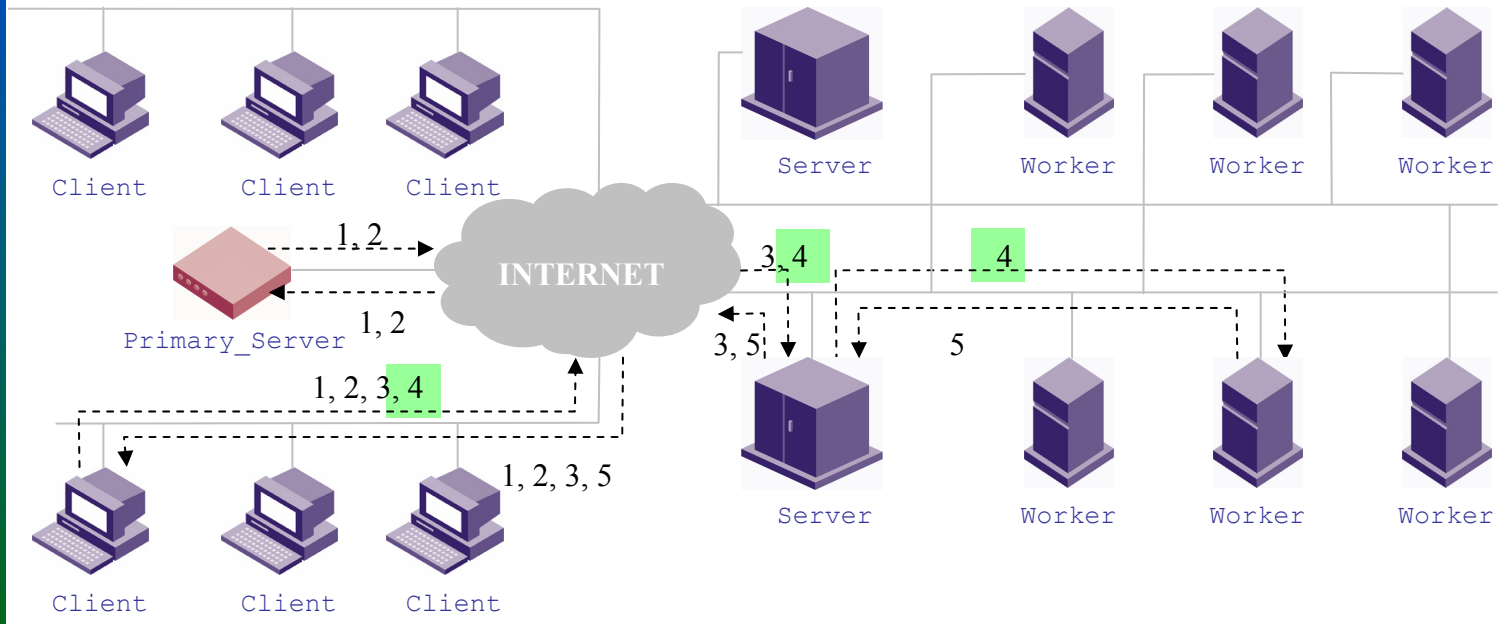
1. **Identificación de usuario**
2. **Selección del algoritmo y servidor de distribución**
3. **Petición de parámetros del algoritmo**
4. **Ejecución del algoritmo**
5. **Obtención de resultados**



Funcionamiento

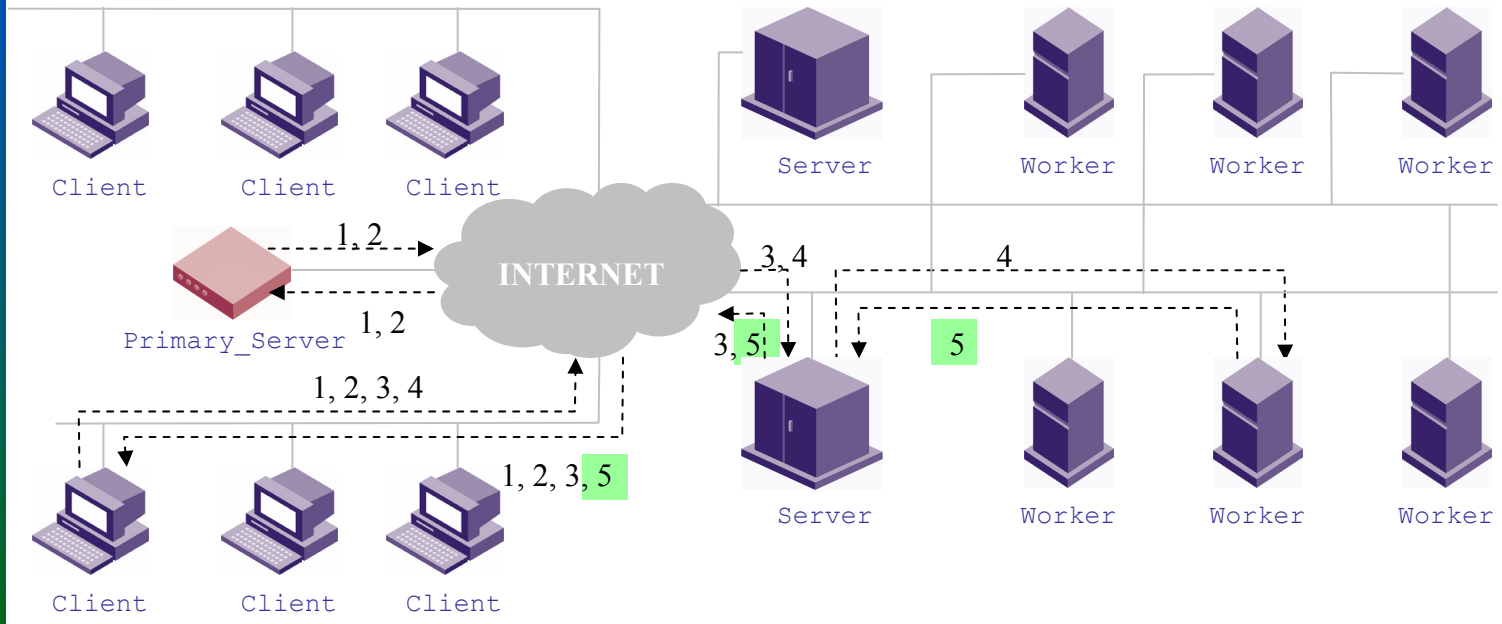
1. **Identificación de usuario**
2. **Selección del algoritmo y servidor de distribución**
3. **Petición de parámetros del algoritmo**
4. **Ejecución del algoritmo**
5. **Obtención de resultados**

Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro



Funcionamiento

1. Identificación de usuario
2. Selección del algoritmo y servidor de distribución
3. Petición de parámetros del algoritmo
4. Ejecución del algoritmo
5. Obtención de resultados



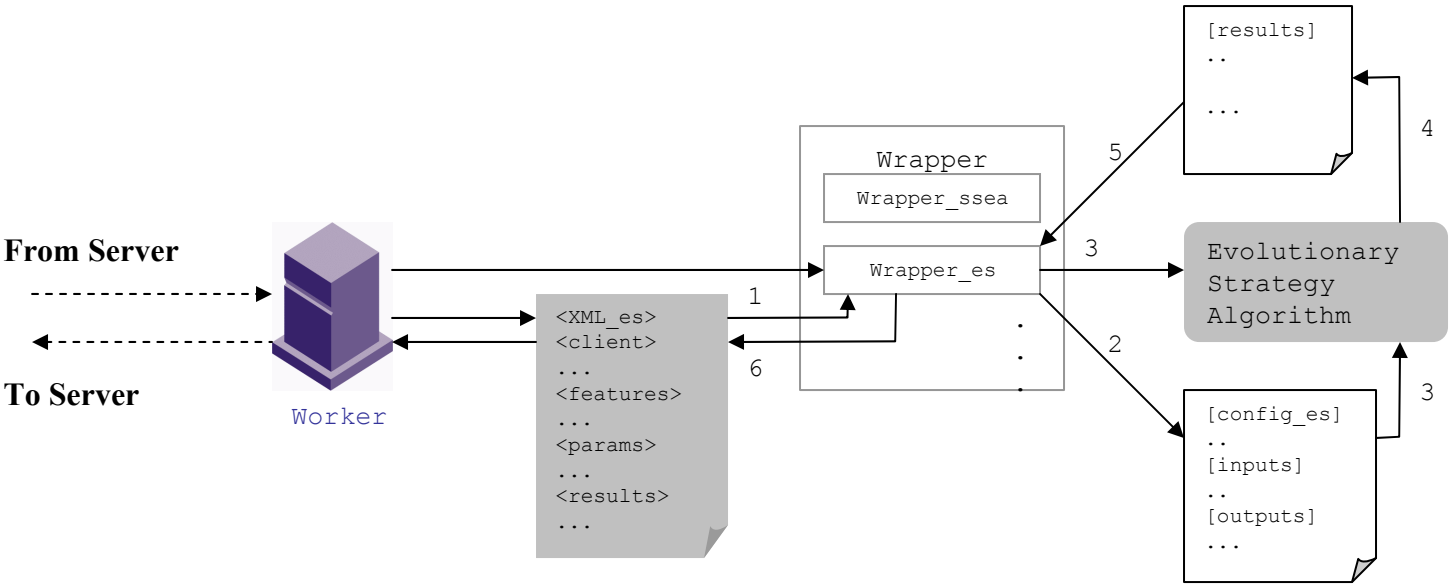
Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Detalles Internos

WRAPPERS

- Los algoritmos disponibles pueden ser muy diferentes (en paradigma, lenguaje, interfaz, etc.)
- El servidor de proceso interacciona con ellos usando un interfaz común (clase `Wrapper`)
- El desarrollador del algoritmo debe implementar una subclase de `Wrapper` específica para su algoritmo



Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Detalles Internos

XML

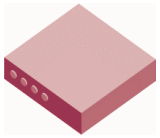
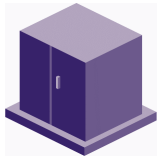
- Usado en la comunicación entre componentes
- Documentos XML restringidos y validados respecto a un DTD
- Elementos principales (a partir de <OptAlgorithm>):
  - <client>
  - <features>
  - <params>
  - <results>

```
<?xml version="1.0" encoding="UTF-8" ?>
  <!DOCTYPE optimization_algorithm SYSTEM "optimization_algorithm.dtd">
<optimization_algorithm>
client      <client>
              <client_name>jnieto</client_name>
              <client_ip>150.214.214.26</client_ip>
              <client_id>null</client_id>
            </client>
features    <features>
              <language>C++</language>
              <compilation>make MainSeq</compilation>   } Órdenes de
              <execution>make SEQ</execution>           } compilación y
              <online />                                } ejecución
              <comment />
            </features>
params      <params type="es">
              <population_size>100</population_size>
              <offsprings_size>50</offsprings_size>
              <crossover>
                <crossover_prob>1.0</crossover_prob>
              </crossover>
              <mutation>
                <mutation_prob>0.1</mutation_prob>
              <number_of_generations>100</number_of_genertons>
              <migration>
                <migration_rate>4</migration_rate>
                <migration_number>20</migration_number>
              </migration>
              <fitness_function>rastrigin</fitness_function>
              <replacement type="inclusion" />
              <selection type="roulette_wheel" />
            </params>
results     <results>
              <best_fitness>0.00288847</best_fitness>
              <worst_fitness>546.003</worst_fitness>
              <generation>72 </generation>
              <computation_time>1751ms</computation_time>
              <error>null</error>
            </results>
</optimization_algorithm>
```

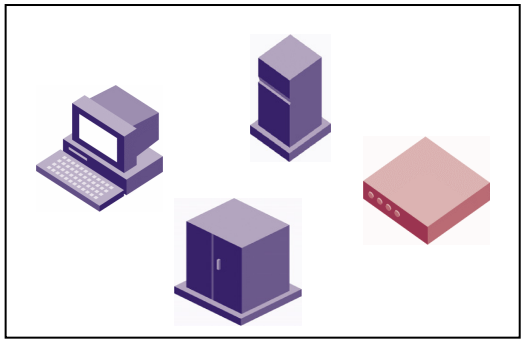
Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Evaluación: Configuraciones

LAN

Cliente	Servidor Primario	Servidor de Distribución	Servidor de Proceso 1	Servidor de Proceso 2
				
Pentium 4 2,6 GHz 512MB	Pentium 4 2,4 GHz 512MB	Pentium 4 2,4 GHz 512MB	Pentium 4 2,4 GHz 512MB	Pentium 4 2,4 GHz 512MB

Local



Pentium 4
2,6 GHz
512MB

Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Evaluación: Resultados

Configuración LAN							
Alg.	Problema	SO	Lenguaje	te (ms)	σ_{te}	tc (ms)	σ_{tc}
ssGA	OneMax	Linux	Java	296.28	68.94	1510.00	549.91
cGA	Rastrigin	Win	Modula 2	**1104.20**	7.76	**1778.10**	39.53
dGA	Rastrigin	Win	Modula 2	**1124.90**	4.73	1744.60	57.64
ES	Rastrigin	Linux	C++	3440.00	92.88	**52.84**	256.07
SA	OneMax	Linux	C++	126.00	43.53	264.64	28.91
GA	Series Temp.	Win	C++	**148320.00**	699.41	**652.42**	139.70
Configuración Local							
ssGA	OneMax	Linux	Java	201.76	18.08	995.96	61.35
cGA	Rastrigin	Win	Modula 2	**1336.40**	42.13	**1838.20**	165.73
dGA	Rastrigin	Win	Modula 2	**1311.70**	23.96	1630.40	70.52
ES	Rastrigin	Linux	C++	2278.10	20.45	**101.52**	24.72
SA	OneMax	Linux	C++	70.74	5.08	99.90	40.25
GA	Series Temp.	Win	C++	**276620.00**	49361.00	**760.96**	151.02

Introducción
ROS
Evaluación
Conclusiones y Trabajo Futuro

Conclusiones y Trabajo Futuro

Conclusiones

- ROS permite a los usuarios (clientes) hacer uso de algoritmos y máquinas para resolver problemas de optimización
- Emplea XML para la comunicación y es multiplataforma
- El sistema puede escalarse añadiendo más servidores de proceso

Trabajo Futuro

- Añadir nuevos algoritmos al servicio
- Crear una interfaz web para acceder al servicio a través de cualquier navegador web

URL: http://tracer.lcc.uma.es/ros

Introducción

ROS

Evaluación

Conclusiones y Trabajo Futuro

FIN

Gracias por su Atención !!!