

Parametrización de esquemas algorítmicos paralelos para autooptimización

Francisco Almeida, Univ. La Laguna

Juan Manuel Beltrán, Murilo do Carmo, Javier Cuenca, Domingo Giménez, Univ. Murcia

Luis Pedro García, Univ. Politécnica de Cartagena

Juan Pedro Martínez, Univ. Miguel Hernández de Elche

Antonio M. Vidal, Univ. Politécnica de Valencia



Contenido

- El problema
- Optimización por parametrización
- Esquema iterativo
- Esquema divide y vencerás
- Recorrido de árboles de soluciones
- Conclusiones y trabajo futuro



El problema

- Evolución continua del hardware
 - Gran coste (tiempo, humano, monetario) de desarrollo
- Usuarios no expertos
 - Mal uso de los recursos
 - No utilización de sistemas paralelos

El problema

- Desarrollo de rutinas con capacidad de autooptimización:
 - Se adecúan automáticamente a las condiciones del sistema
 - Uso eficiente sin participación del usuario
- En diversos campos:
 - FFTW
 - LFC
 - ATLAS
 - SOLAR
 - FIBER

Optimización por parametrización

- Propuesta: parametrización del modelo del tiempo de ejecución:
 - Se obtiene un modelo en tiempo de diseño
 - Se incluyen parámetros en el modelo:
 - Unos representan parámetros del sistema
 - Otros representan parámetros del algoritmo

$$t_{parallel}(n, AP, SP)$$

Optimización por parametrización

■ En tiempo de instalación:

- Se obtienen valores de los parámetros del sistema, por experimentación (guiada por el manager o decidida por el diseñador)
- Se incluyen los valores en el modelo (en el código o en fichero)
- Se instala la rutina o la librería en el sistema, incluyendo el modelo con los parámetros del sistema (se compila, incluye ejecución condicional)

⇒ **adecuación automática al sistema**

Optimización por parametrización

- En tiempo de ejecución:
 - A partir del modelo, de los valores de los parámetros del sistema y de la entrada a resolver, se obtienen valores óptimos de los parámetros del algoritmo
 - Se ejecuta la rutina con esos valores
- ⇒ ejecución óptima independiente de los conocimientos del usuario



Optimización por parametrización

- Trabajos previos:

- Rutinas básicas de álgebra lineal
- Adecuación a sistemas heterogéneos y dinámicos

- Trabajo actual:

- Estudio de la aplicación de la técnica de parametrización a esquemas algorítmicos paralelos

Esquema iterativo

Esquema paralelo:

for $i=1$ **to** *numero_de_decisiones*

In Parallel:

for $j=1$ **to** *tamaño_problema*

obtener solución

óptima con

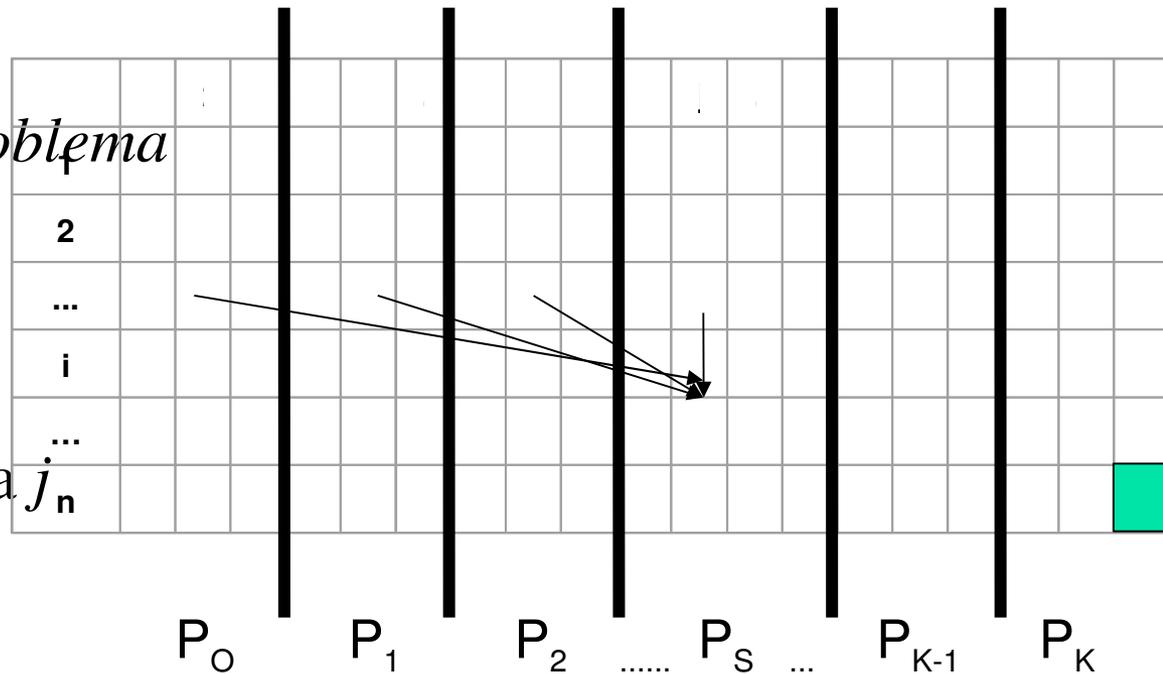
i decisiones y

tamaño de problema j_n

endfor

endInParallel

endfor



Esquema iterativo

Esquema de paso por mensajes

In each processor P_j

for $i=1$ to *numero_de_decisiones*

 paso de comunicación

 obtener solución

 óptima con

i decisiones

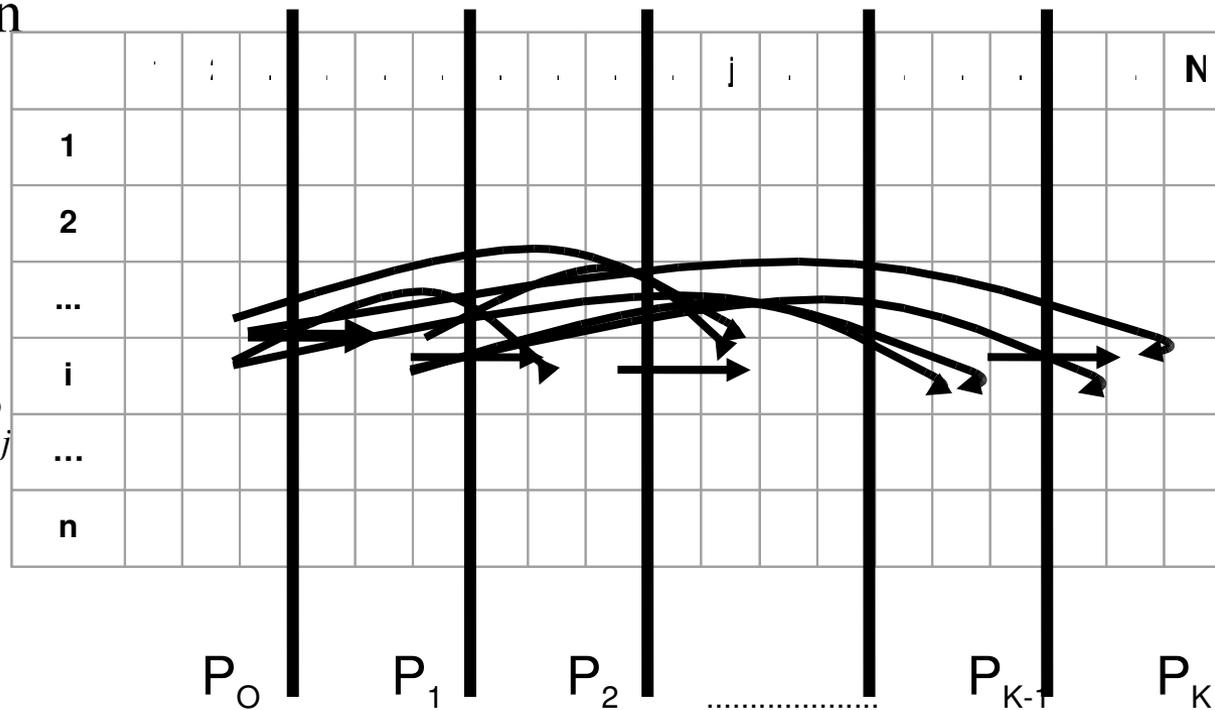
 y los tamaños

 de problema que P_j

 tiene asignados

endfor

endInEachProcessor



Esquema iterativo

- Modelo teórico:

$$t_{parallel}(n) = t_{arit,1} + t_{com,1} + t_{arit,2} + t_{com,2} + \dots$$

Secuencial: $C^2 / (2v_i)$

Paralelo aritmético: $C^2 / (pv_i)$

Paralelo comunicación: $(p(p-1)) / 2 t_s + (C(p-1)) / 2 t_w$

- Modelo teórico:

Tamaño problema: C, v_i

Parámetro algorítmico: p

Parámetros del sistema: t_s, t_w

Esquema iterativo

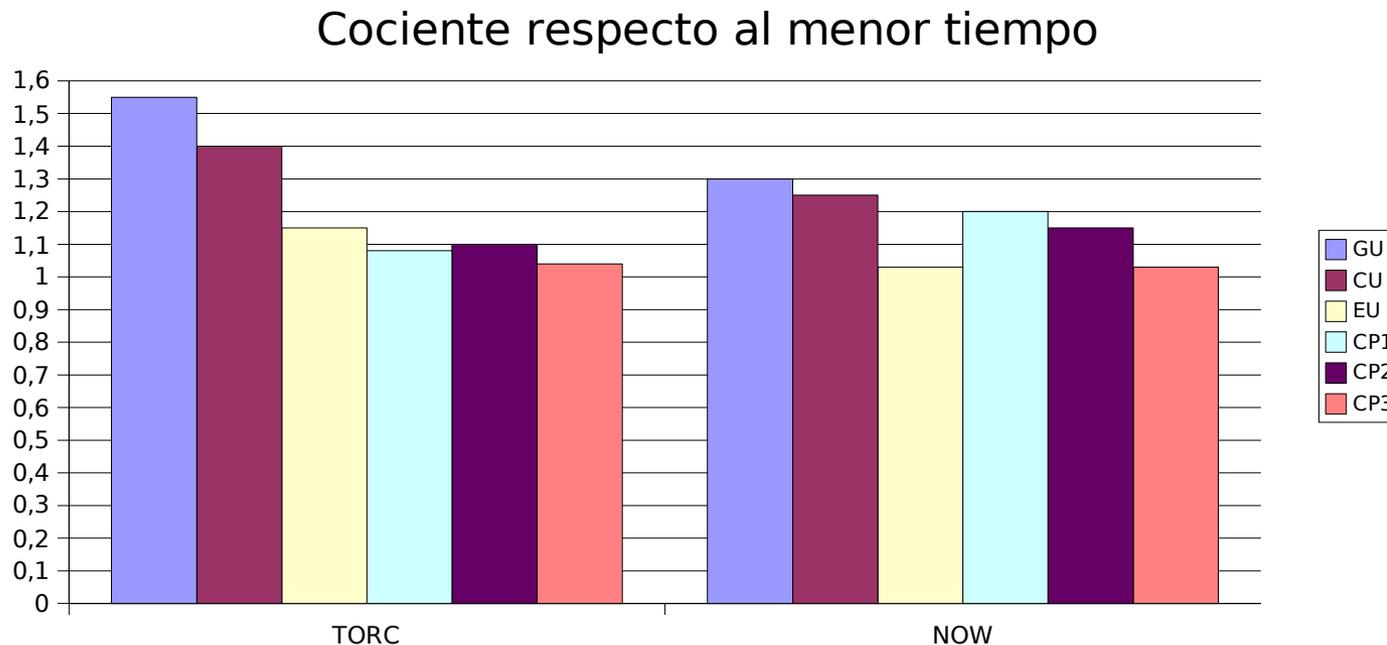
- Estimación de SP aritméticos: resolviendo un problema reducido
- Estimación de SP de comunicaciones:
 - Ping-pong (CP1)
 - Solución de problema reducido variando el número de procesadores (CP2)
 - Solución de varios problemas con número variado de procesadores y ajuste por mínimos cuadrados (CP3)

Esquema iterativo

- Comparación con usuarios:
 - Greedy (GU): usa todos los procesadores disponibles
 - Conservative (CU): usa la mitad de los procesadores disponibles
 - Expert (EU): usa un número de procesadores en función del tamaño del problema

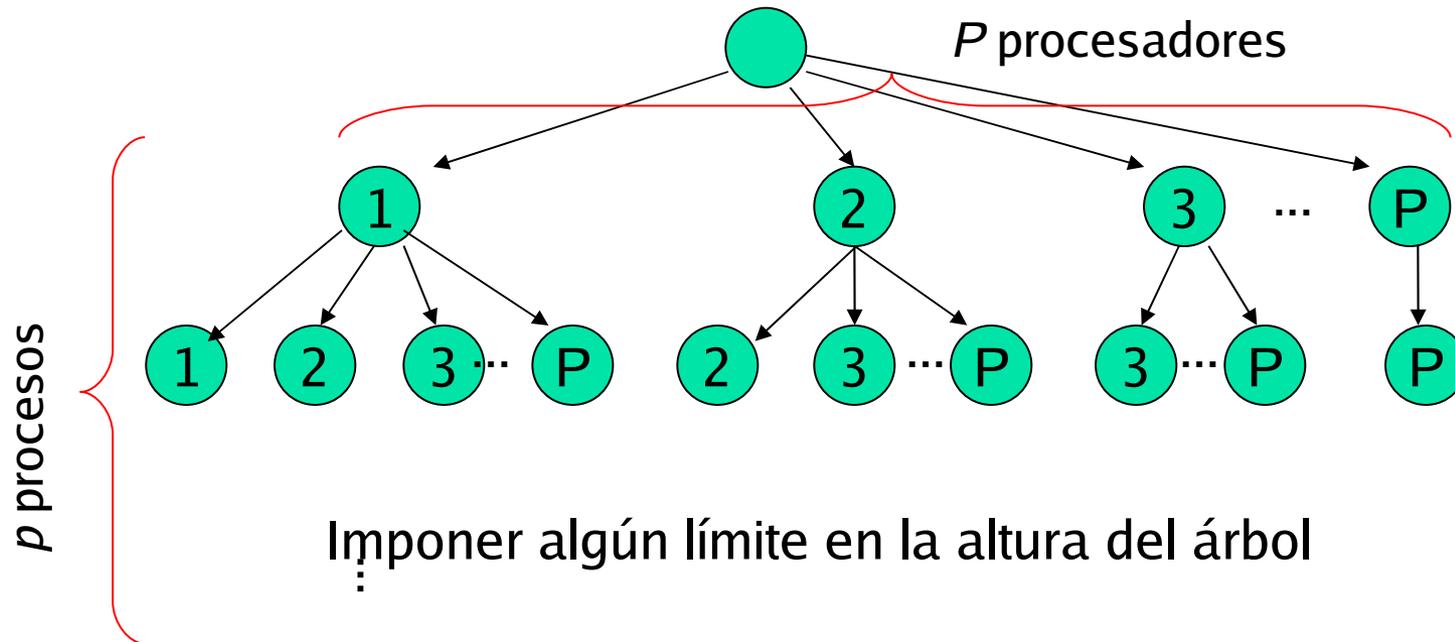
Esquema iterativo

- Cociente respecto al menor tiempo de ejecución experimental: en TORC (red de 8 biprocesadores) y NOW (red de 6 estaciones SUN)



Esquema iterativo

- Asignación en sistema heterogéneo:

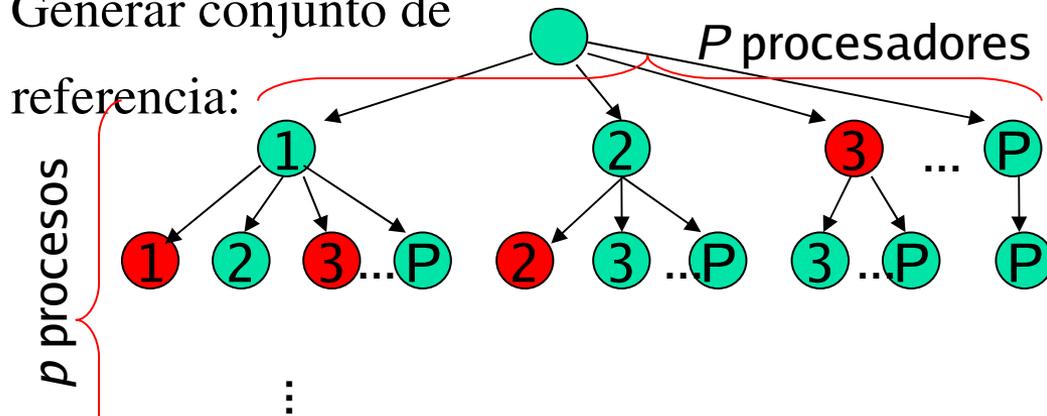


Avance rápido: asignación rápida, lejos de la óptima
Backtracking: asignación lenta, óptima

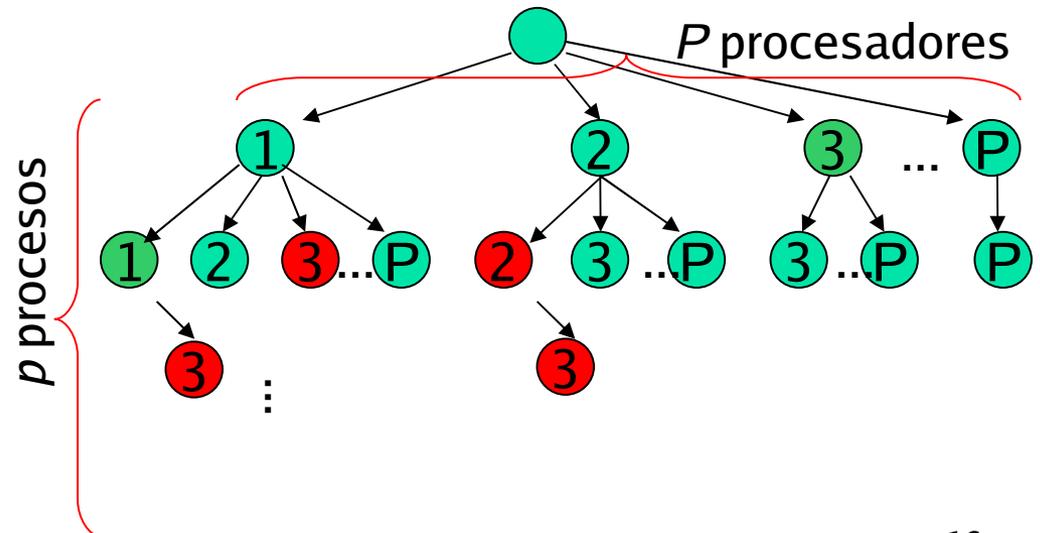
Esquema iterativo

- Utilizar métodos heurísticos (Scatter Search):

Generar conjunto de referencia:



Combinar elementos del conjunto y quedarse con los mejores y los más “distantes”



Esquema divide y vencerás

Esquema_DV_Paralelo (p : problema, n : tamaño):
solución

IF MASTER

dividir p en subproblemas p_1, p_2, \dots, p_k

distribuir subproblemas a procesos SLAVE

recibir resultados de SLAVES

combinar soluciones de subproblemas

ELSE

recibir del MASTER subproblema p_i

$s_i = \text{Esquema_DV_Secuencial}(p_i, n_i)$

enviar s_i a MASTER

ENDIF

Esquema divide y vencerás

- Ordenación por mezcla
 - Tamaño de problema: número de datos a ordenar
 - Parámetros del sistema: operaciones aritméticas básicas (resolviendo un problema pequeño), parámetros de comunicaciones (ping-pong, resolver varios problemas con número de procesadores variable y ajuste por mínimos cuadrados)
 - Parámetros del algoritmo: método directo, tamaño base, número de procesadores

Esquema divide y vencerás

CONFIGURACIÓN DEL SISTEMA

Método de Ordenación Directa

1 Burbuja
 Inserción
 Selección

Rango de Experimentación:

2 Rango Inicial
Rango Final
Incremento

Cantidades de Experimentaciones:

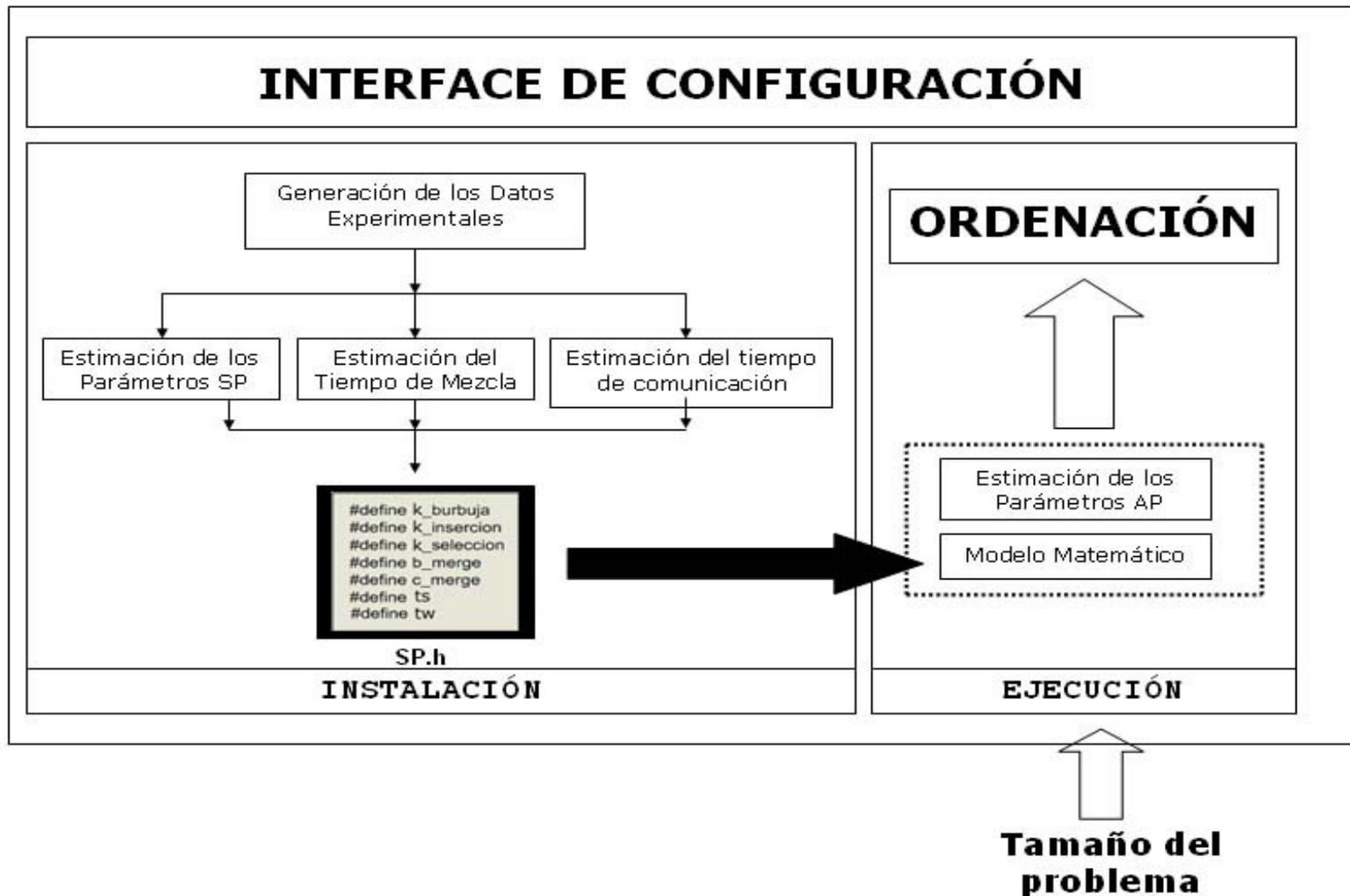
3 3

Métodos de Aproximación:

4

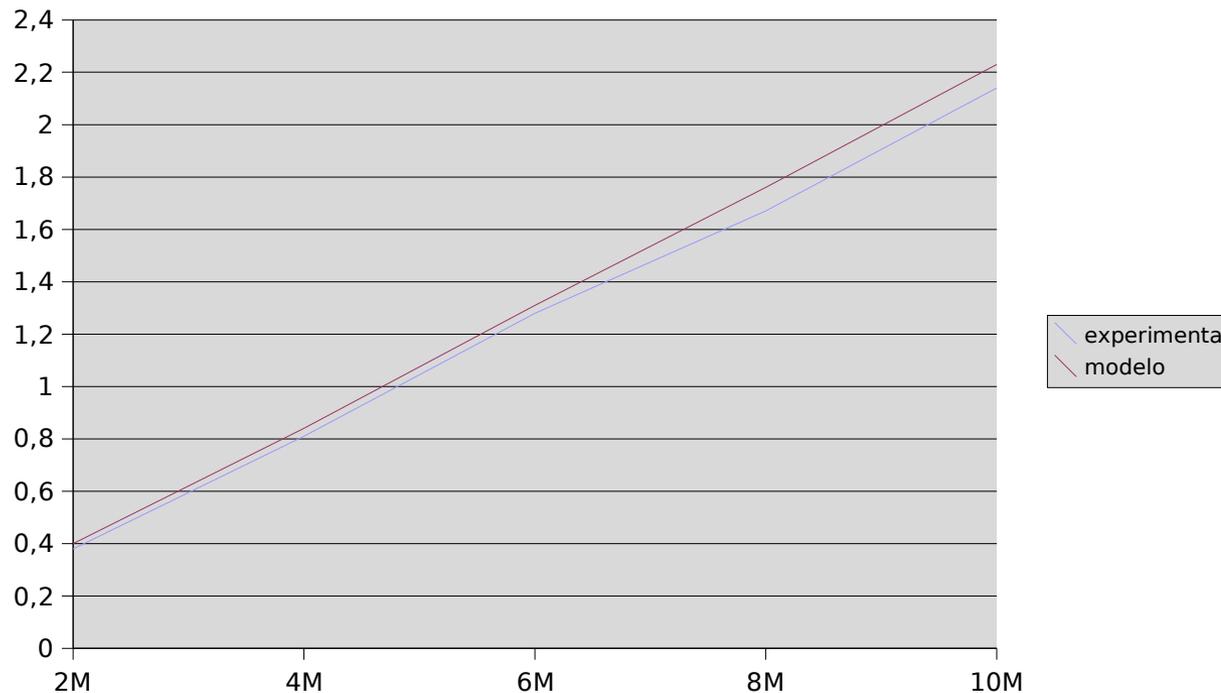
Limpia Pantalla **5** Configuración

Esquema divide y vencerás



Esquema divide y vencerás

- Comparación del menor tiempo experimental y el obtenido con los parámetros proporcionados por el modelo:



Recorrido de árboles de soluciones

Ejecutar Esquema Secuencial Hasta Nivel l

Repartir Trabajos Iniciales a P_1, \dots, P_{p-1}

SI P_i ya acabado y Más Trabajos Pendientes

Enviar Nuevo Trabajo a P_i

REPETIR

Recibir de P_i valores óptimos locales y si
acabado

Comparar óptimos recibidos con global

Enviar a P_i óptimo global y trabajo

MIENTRAS No Todos Acabados

Recorrido de árboles de soluciones

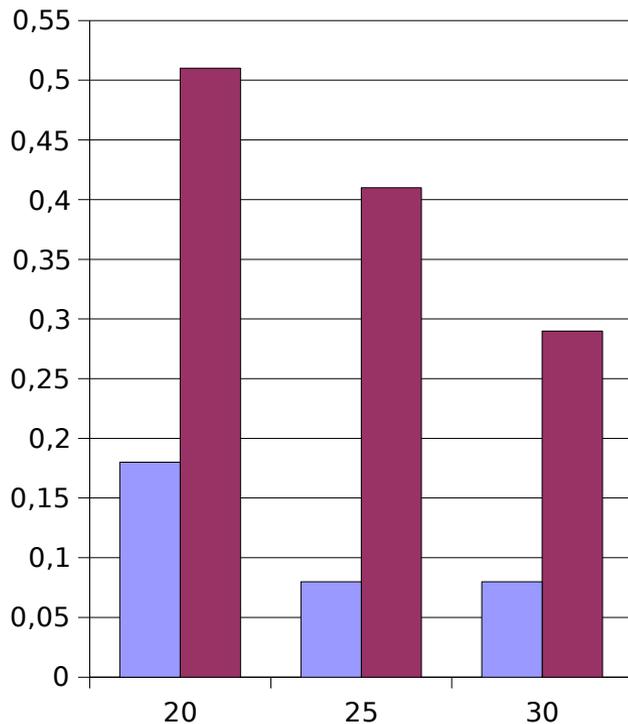
- Mochila 0/1 por backtracking, con eliminación de nodos
 - Tamaño de problema: número de datos, y valores de entrada (no conocidos hasta el momento de ejecución)
 - Parámetros del sistema: operaciones aritméticas básicas (resolviendo un problema pequeño), parámetros de comunicaciones (ping-pong)
 - Parámetros del algoritmo: nivel inicial, frecuencia de intercambio, número de procesadores

Recorrido de árboles de soluciones

- Aparece un parámetro del problema: porcentaje de nodos que se elimina
 - Depende de los valores de entrada. Estimarlos en tiempo de ejecución:
 - Resolviendo problema agrupando datos
 - Generando aleatoriamente posibles soluciones y estimando altura media del árbol
 - O haber estimado un valor resolviendo varios problemas en la instalación, y rectificar estimación con estimación reducida

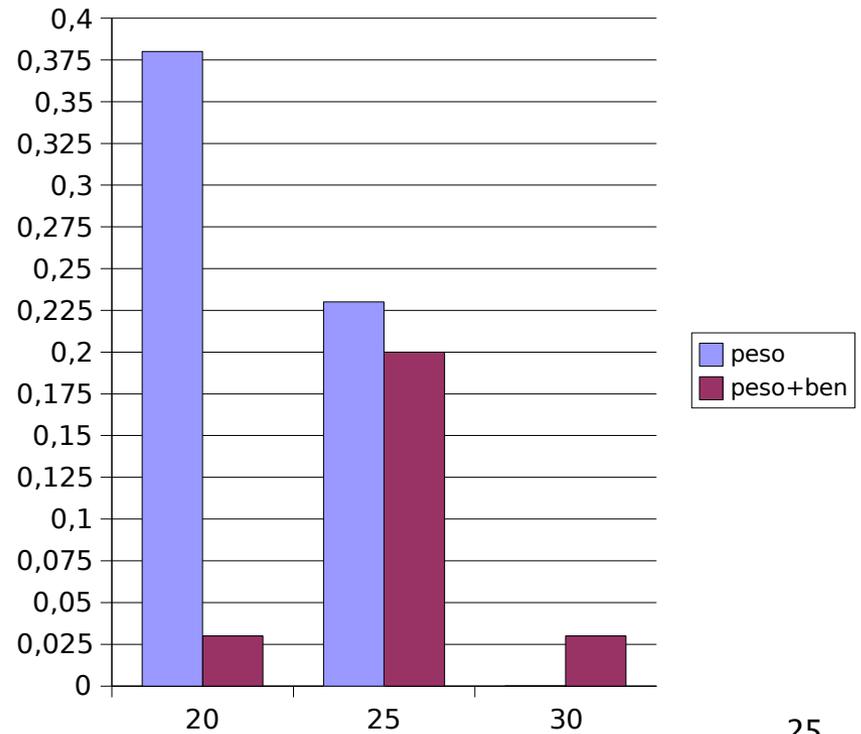
Recorrido de árboles de soluciones

- Puede haber grandes diferencias en la estimación



- Pero la decisión es satisfactoria

error relativo respecto al menor experimental



Conclusiones y trabajo futuro

- Ideas iniciales de introducción de técnicas de autooptimización por parametrización en esquemas algorítmicos paralelos
- Resultados iniciales satisfactorios
- Continuación de los experimentos con más problemas, problemas no académicos, y con más esquemas
- Estudiar la combinación con esqueletos paralelos