

Autooptimización en esquemas paralelos iterativos

Autor:

Juan Pedro Martínez Gallar

Directores:

Domingo Giménez Cánovas
Universidad de Murcia

Francisco Almeida Rodríguez
Universidad La Laguna



ÍNDICE

1- Introducción.

- Situación histórica. Objetivos. Motivación.
- Optimización.
- Autooptimización.
- El problema del mapeo.
- Metodologías: Tesis, Modelos.
- Esquemas iterativos, paralelos, homogéneos y heterogéneos.

2.- Autooptimización en sistemas homogéneos.


3.- Autooptimización en sistemas heterogéneos.

4.- Metaheurísticas en la autooptimización.

5.- Conclusiones y trabajos futuros.

INTRODUCCIÓN

Situación histórica. Objetivos. Motivación.

- Evolución de los sistemas informáticos  Mayor complejidad
- Desarrollo de técnicas para garantizar uso eficiente de recursos
- Software diseñado, implementado y optimizado sobre sistemas concretos
- Necesidad de disponer de técnicas que aseguren el uso en sistemas para los que no se diseñaron originalmente sin intervención humana (autotuning): ATLAS, LFC, FFTW...
- Metodología para adaptación a sistemas homogéneos, heterogéneos

Situación histórica. Objetivos. Motivación.

→ Objetivo Tesis:

”Adaptación automática del software iterativo paralelo en sistemas para los que no fue propuesto inicialmente”

→ Objetivos Parciales:

- ❖ Desarrollo metodologías en sistemas homo y heterogéneos
- ❖ Uso de técnicas metaheurísticas en sist. heterogéneos
- ❖ Desarrollo metodología para otros esquemas computacionales

→ Reutilización



Mejora del ciclo vida del software

Optimización

→ Optimización: necesario para lograr uso eficiente de recursos

→ Optimización

{
Código
Uso del código

→ Diferentes niveles de optimización:

{
Diseño
Código fuente
Compilación
Tº Ejecución
...

Optimización

→ Optimización
uso de código



Trabajo adicional usuario
final, puede no experto
a nuevos sistemas

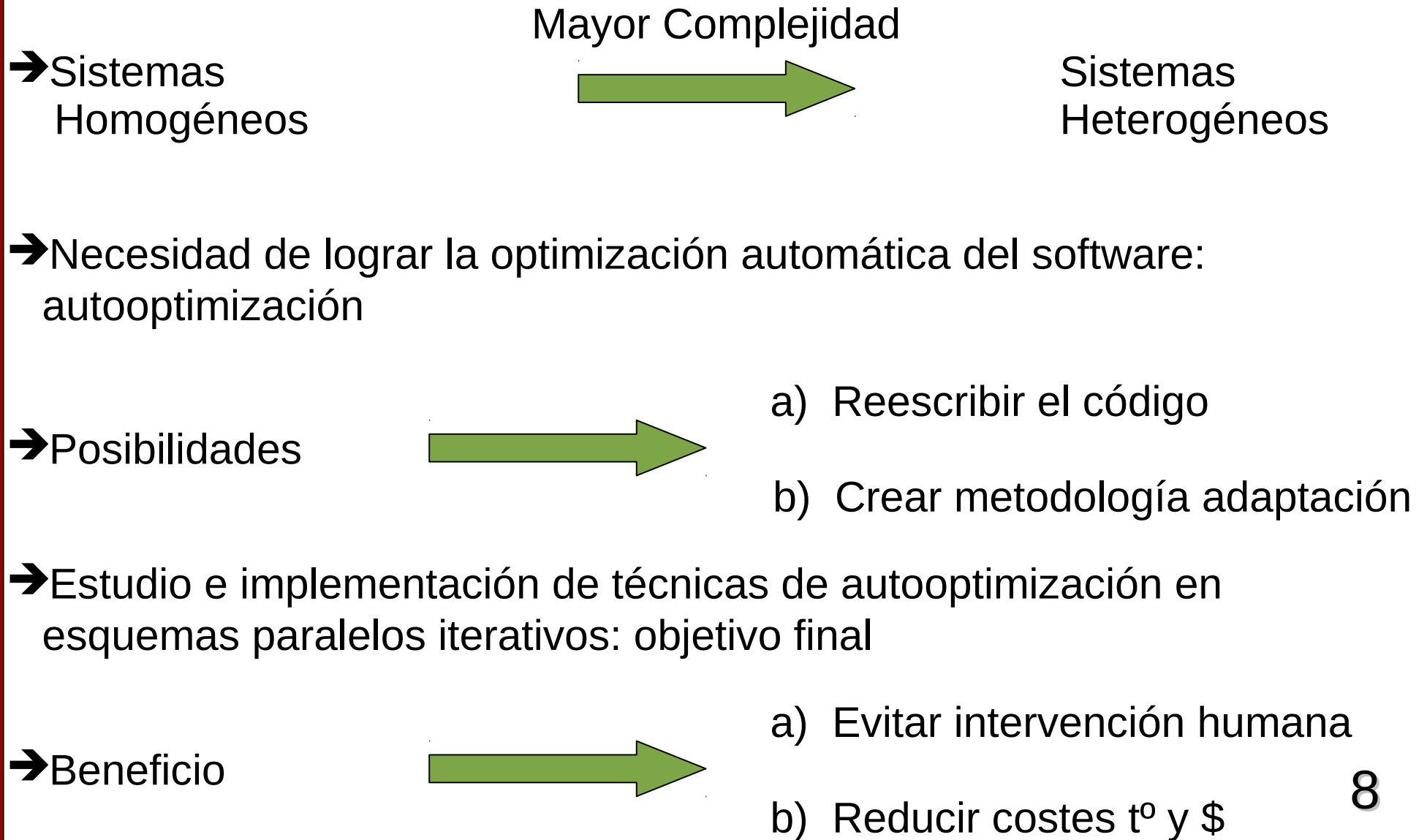
→ Desarrollar técnicas para lograr la adaptación del software a la
arquitectura de destino para reducir tiempos de ejecución de
forma automática

→ Autooptimización



Liberar al usuario final

Autooptimización



Autooptimización

→ Optimización basada en la modelización del tiempo de ejecución:

$$t(s, AP, SP)$$

- s: Tamaño del problema
- AP: Parámetros algorítmicos (tamaño bloque, n° procesos, n° procesadores, procesos/procesadores ...)
- SP: Parámetros del sistema (dependen de la arquitectura y el tipo de sistema: homogéneo / heterogéneo)

t_c : tiempo de operación básica computación

t_s : tiempo de inicio de comunicaciones

t_w : tiempo de envío de un dato básico

$$t_s + nt_w$$

El problema de Optimización

→ Objetivo:

$$\begin{array}{l} \min t(s, \text{vap}, \text{SP}) \\ \text{vap} \in \text{AP} \end{array}$$

→ Asignación de procesadores a las necesidades software

→ Técnicas:

→ Grafos de precedencia

→ Optimización analítica

→ Árboles de asignación

→ Estrategias heterogéneas (HoHe, HeHo)

→ Metaheurísticas

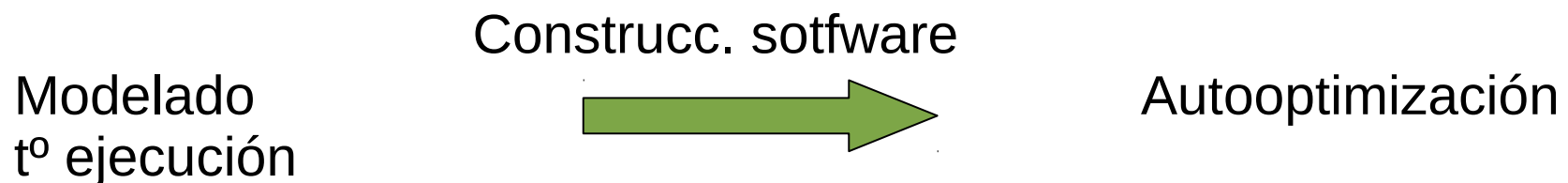
→

El problema de Optimización

→ Cuestiones relevantes:

- Particionado de datos
- Análisis de dependencias
- Asignación de recursos
- Equilibrado de carga

→ Hipótesis:



→ Necesidad de establecer una metodología de trabajo

Metodología Tesis

→Etapas de la tesis

Construir modelo tº ejecución algoritmos iterativos

Metodología Tesis

→Etapas de la tesis

Construir modelo tº ejecución algoritmos iterativos

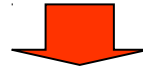


Aplicaciones en esquemas paralelos iterativos

Metodología Tesis

→Etapas de la tesis

Construir modelo tº ejecución algoritmos iterativos



Aplicaciones en esquemas paralelos iterativos



Desarrollo metodologías autoopt. sist. homogéneos

Metodología Tesis

→Etapas de la tesis

Construir modelo t^o ejecución algoritmos iterativos



Aplicaciones en esquemas paralelos iterativos



Desarrollo metodologías autoopt. sist. homogéneos



Desarrollo metodologías autoopt. sist. heterogéneos

Metodología Tesis

→Etapas de la tesis

Construir modelo tº ejecución algoritmos iterativos



Aplicaciones en esquemas paralelos iterativos



Desarrollo metodologías autoopt. sist. homogéneos



Desarrollo metodologías autoopt. sist. heterogéneos



Estrategias
heterogéneas
(HoHe, HeHo)

Metodología Tesis

→Etapas de la tesis

Construir modelo tº ejecución algoritmos iterativos



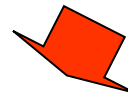
Aplicaciones en esquemas paralelos iterativos



Desarrollo metodologías autoopt. sist. homogéneos



Desarrollo metodologías autoopt. sist. heterogéneos



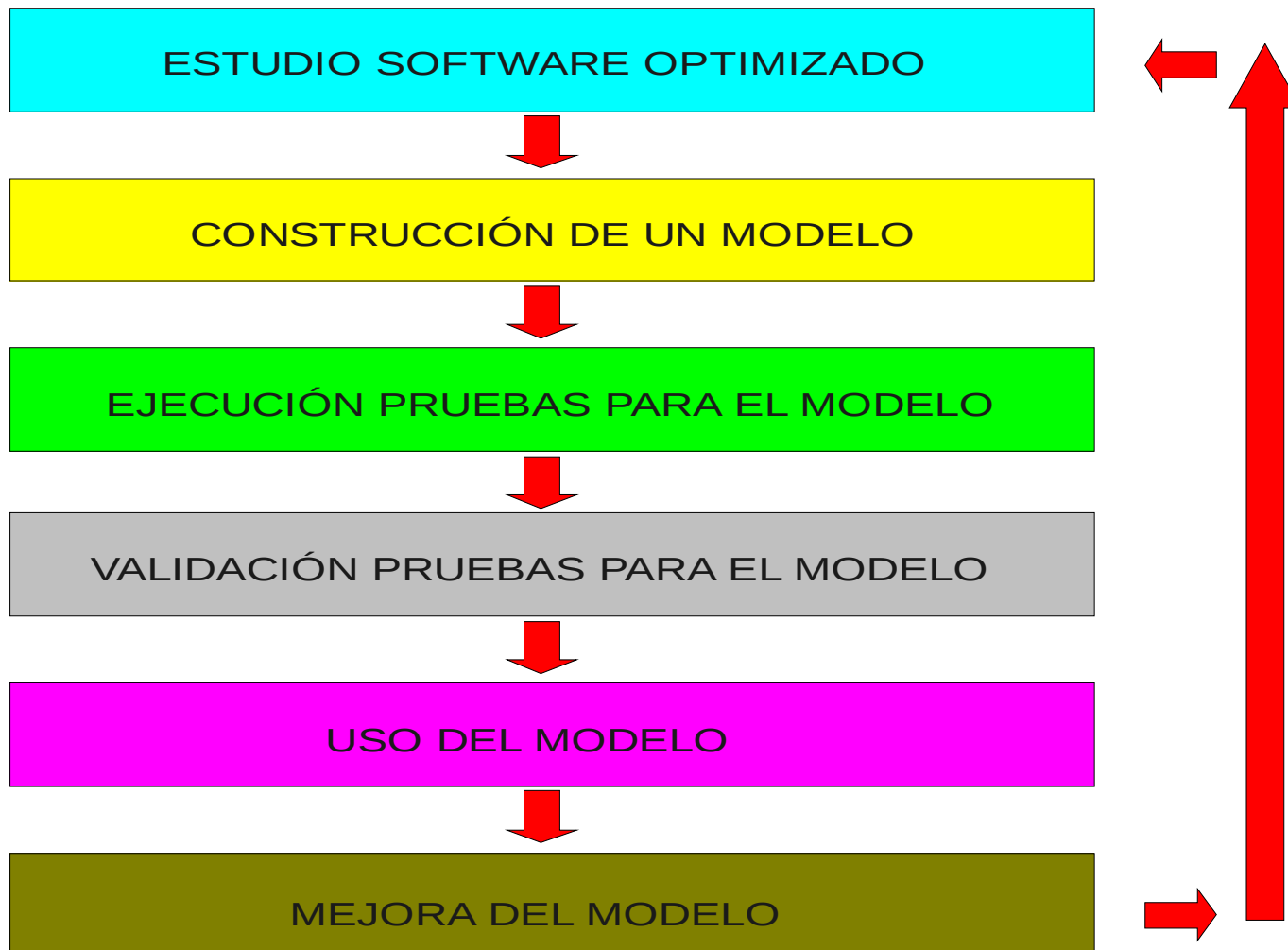
Estrategias
heterogéneas
(HoHe, HeHo)

Metaheurísticas


Metodología de autooptimización



Construcción modelo tº ejecución



Esquemas iterativos

- Esquema iterativos usados en multitud de problemas: Progr. Dinámica, Dijkstra, genéticos, sist. ecuaciones ...
- Ejecución de un conjunto de instrucciones de forma iterativa hasta la condición de fin
- Tipos
 - Secuenciales
 - Paralelos
 - Homogéneos
 - Heterogéneos
- Caso Prueba (Programación dinámica)  Diferentes Esquemas
- Diferentes formas de calcular los datos: por filas, columnas, diagonales

Programación dinámica

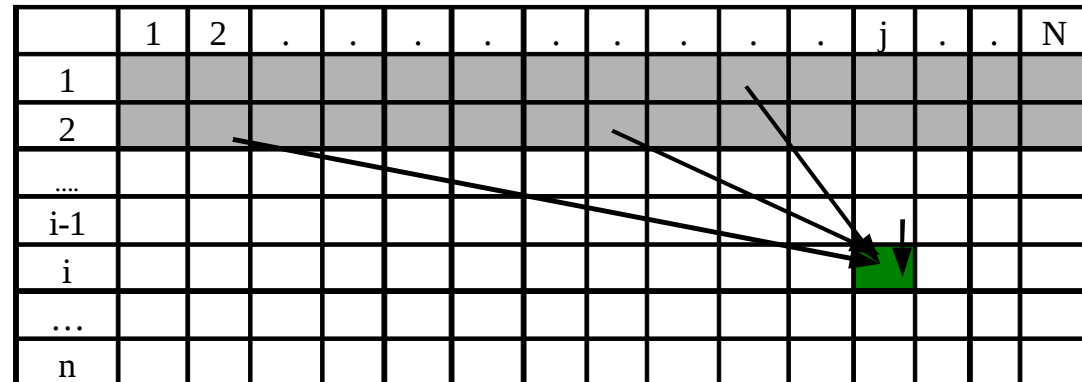
→ Por filas

	1	2	j	.	.	N
1															
2															
...															
i-1															
i															
...															
n															

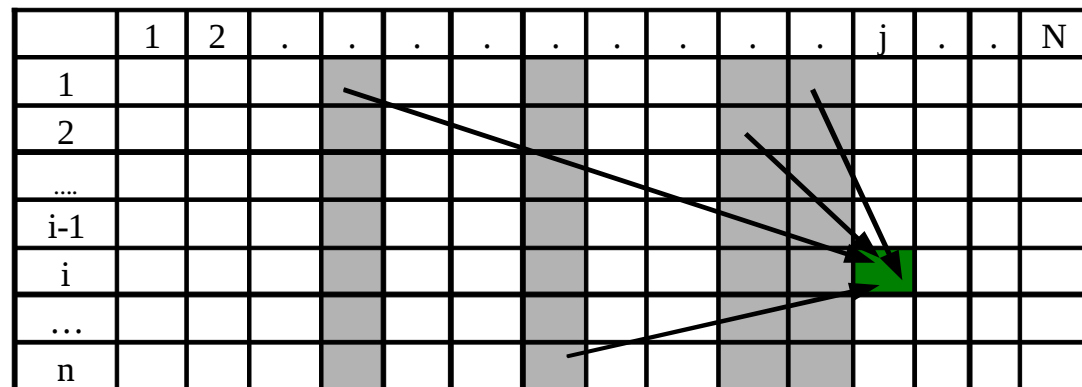
The diagram shows a grid representing a dynamic programming table. The columns are labeled 1, 2, ..., j, ..., N. The rows are labeled 1, 2, ..., i-1, i, ..., n. The first two rows are shaded gray. A green arrow points to the cell at row i, column j. Three black arrows point from the cells at (i-1, j), (i-1, j-1), and (i-1, j-2) to the green cell at (i, j).

Programación dinámica

→ Por filas

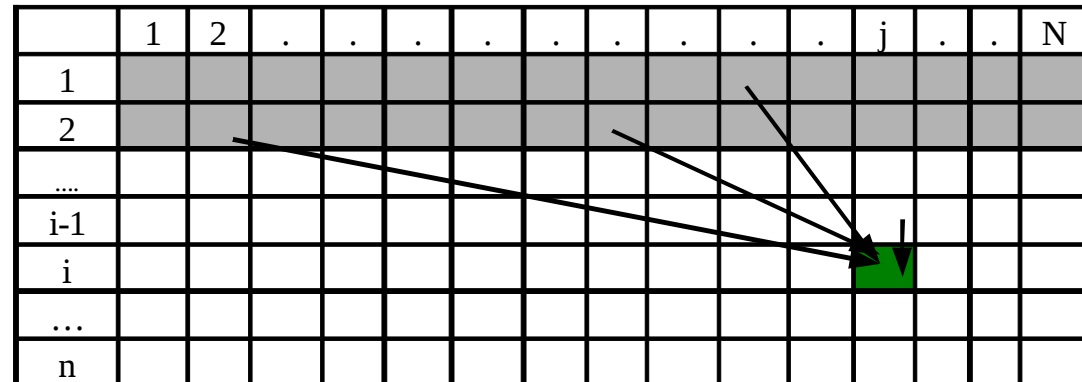


→ Por columnas

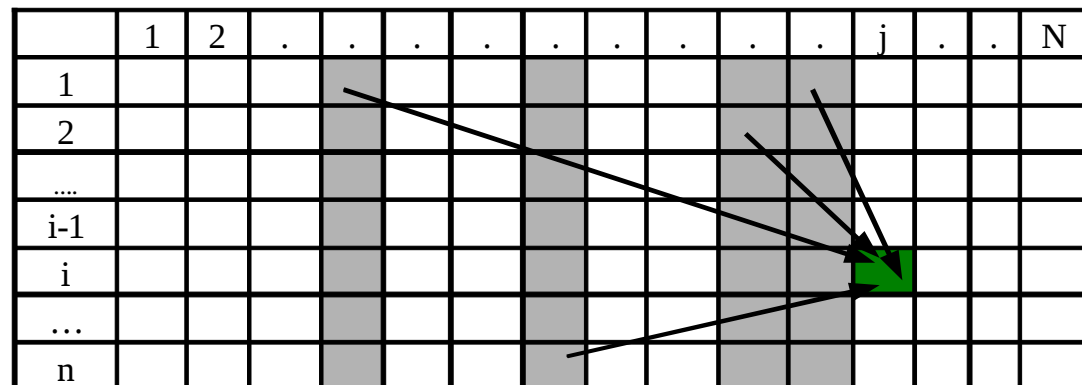


Programación dinámica

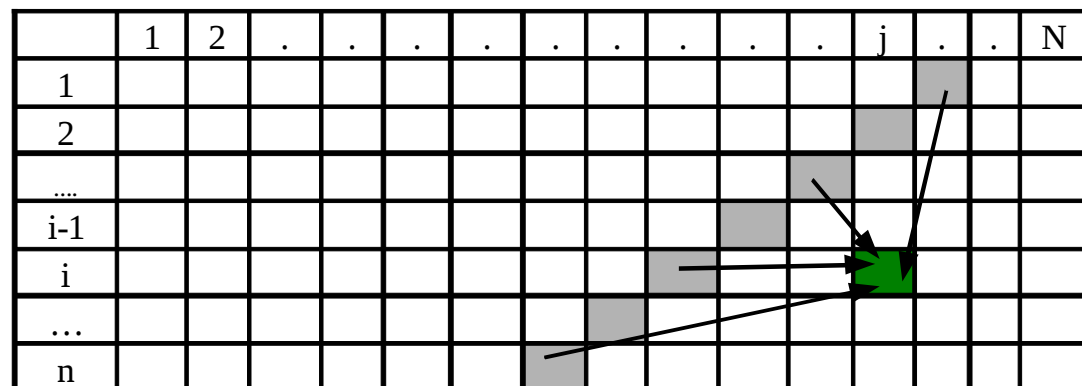
→ Por filas



→ Por columnas



→ Por diagonales



Programación dinámica

→ Usados como ejemplos: **Problema monedas**, Mochila

→ Definición

N: cantidad a devolver

n: número tipos de monedas

v_i : valor de la moneda de tipo i , $v_i > 0$

q_i : cantidad de monedas de tipo i , $q_i > 0$

$c[i,j]$ = mínimo número de monedas para devolver cantidad j usando

→ Ecuación hasta los i tipos de monedas

$$c[i,j] = \min \{ c[i-1, j-k*v_i] + k \}, \quad 1 \leq i \leq n, \quad 1 \leq j \leq N, \quad k=0, \dots, j/v_i$$



Objetivo: $c[n,N]$

Programación dinámica

→ Esquema Secuencial

for $i=1$ to *numero_de_decisiones*

for $j=1$ to *tamaño_problema*

obtener la solución óptima

con i decisiones y

tamaño de problema j

endfor

endfor

FILAS

COLUMNAS

	1	2	j	N
1																		
2																		
...																		
i-1																		
i																		
...																		
n																		

Esquemas iterativos

→ Esquema Paralelo (dependencia de datos)

for $i=1$ to *numero_de_decisiones*

En Paralelo:

for $j=1$ to *tamaño_problema*

obtener la solución óptima

con i decisiones y

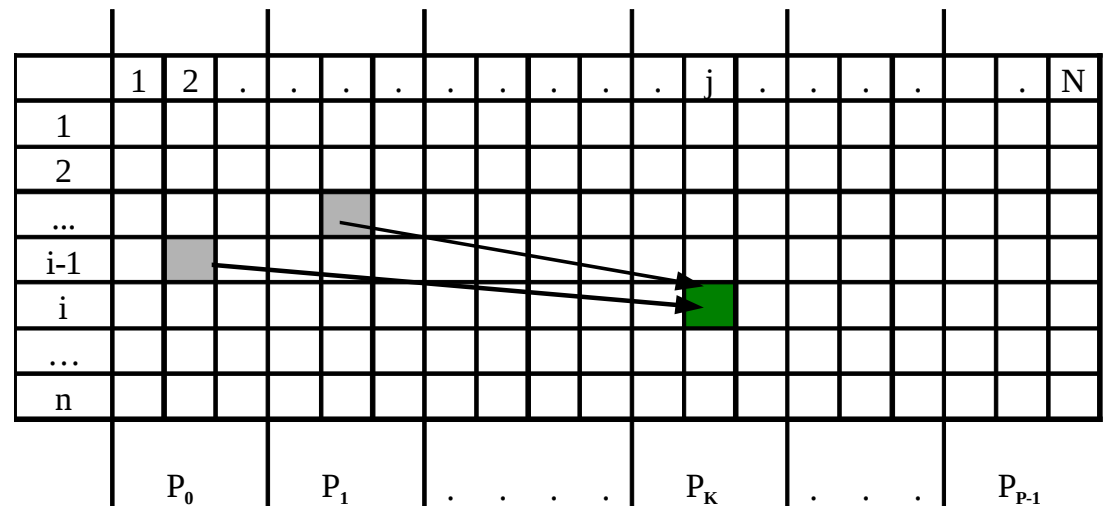
tamaño de problema j

endfor

Comunicaciones entre procesos

endEnParalelo

endfor



AUTOOPTIMIZACIÓN EN SISTEMAS HOMOGÉNEOS

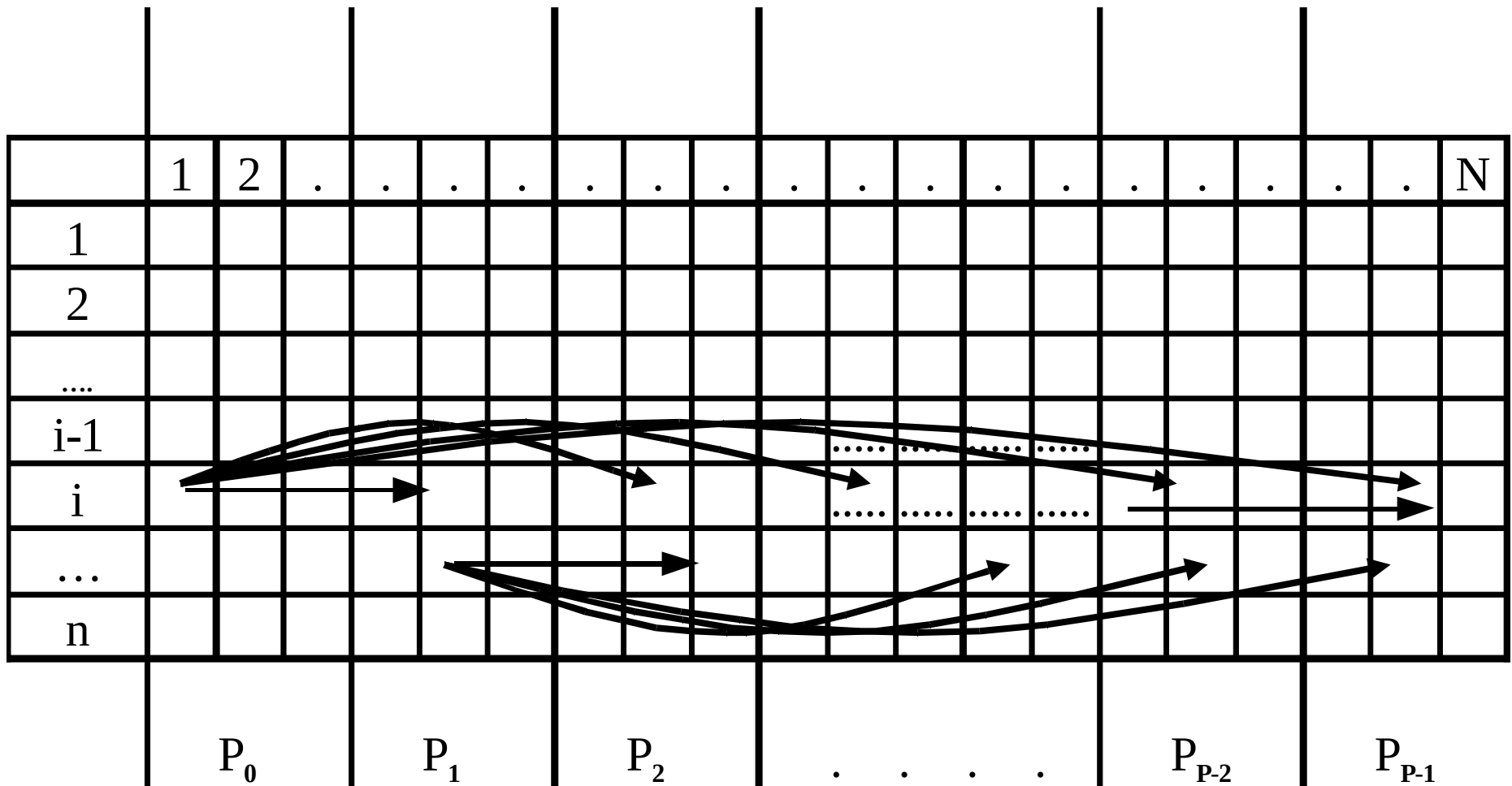
Autooptimización en sistemas homogéneos

- Homogéneos: características similares no exactamente iguales
- Relativa facilidad construir modelo tiempo ejecución
- Determinar número procesos (p) = procesadores (P)
- Pruebas realizadas sobre diferentes sistemas:

→ SOLARIS/SUN (SUNet)	(Un. Murcia)	} HOMOGÉNEOS
→ PenFE	(Un. Murcia)	
→ ORIGIN 2000	(Un. Polit. Cat.)	
→ HPC160 (Inter e intranodo)	(Un. Polit. Cart.)	
→ KIPLING	(Univ. Polit. Val.)	} HETEROGÉNEOS
→ TORC	(Univ. Tennessee)	

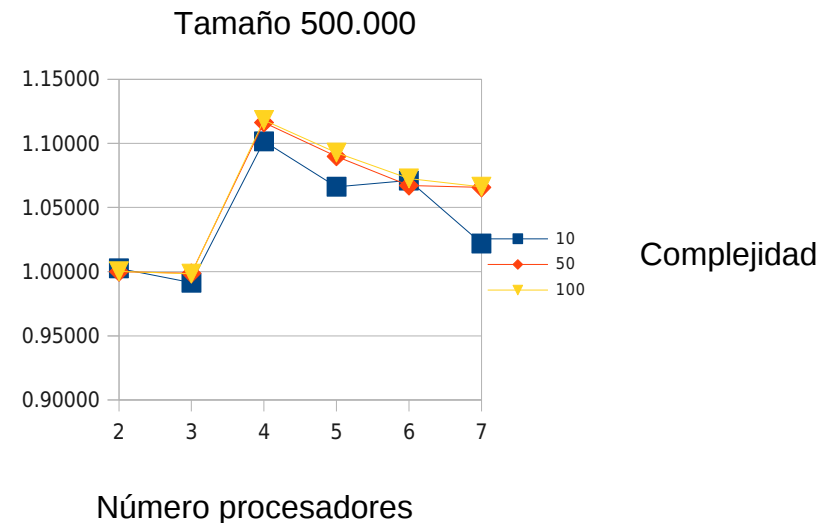
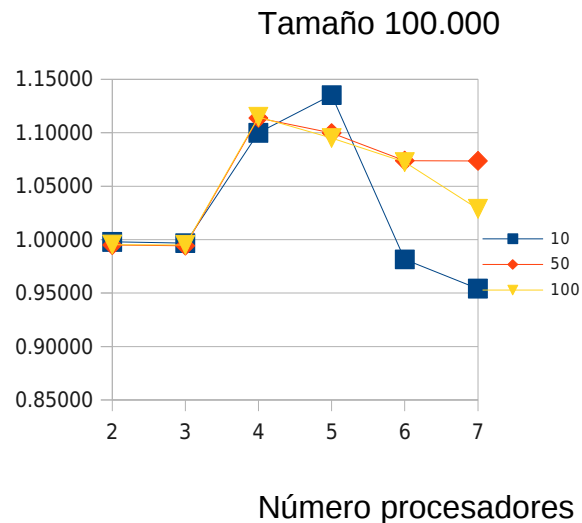
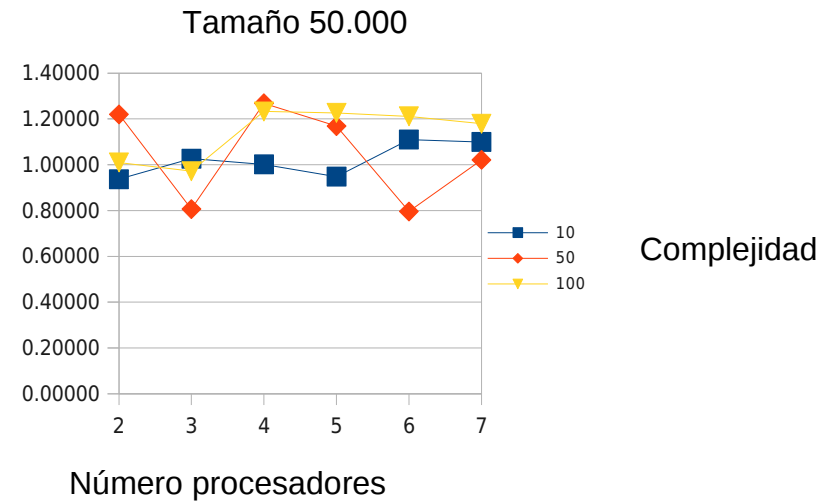
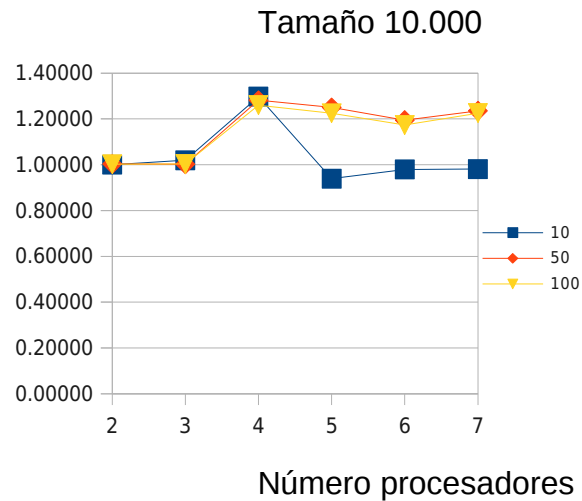
Autooptimización en sistemas homogéneos

DEPENDENCIA DE DATOS (VERSIONES A y B)



Autooptimización en sistemas homogéneos

Cociente de tiempos de ejecución entre versiones A y B en SUNEt



Autooptimización en sistemas homogéneos

→ Modelo teórico:

$$T_{\text{total}} = T_{\text{computación}} + T_{\text{comunicación}}$$

→ Coste secuencial: $O\left(\frac{N^2}{2v_i}\right)$

→ Coste computacional paralelo (q_i grande):

$$\sum_{j=N-\frac{N}{p}+1}^N \left(1 + \left\lfloor \frac{j}{v_i} \right\rfloor\right) \approx O\left(\frac{N^2}{pv_i}\right)$$

→ Coste comunicaciones: $\frac{p(p-1)}{2}t_s + \frac{N(p-1)}{2}t_w$

Proceso P_{p-1}

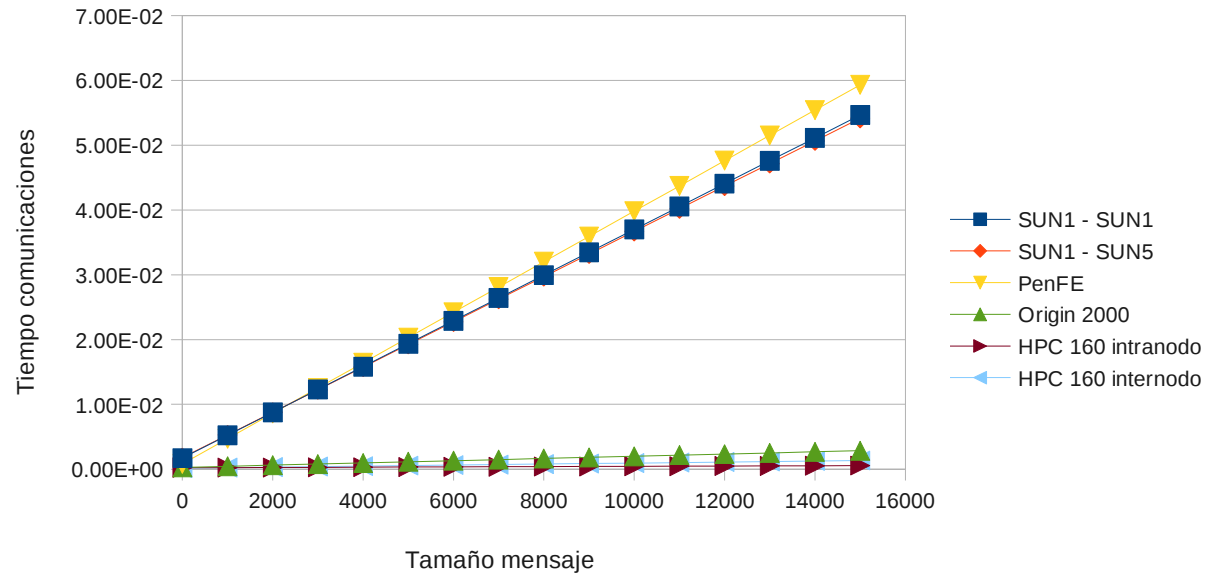
→ Parámetro de complejidad o granularidad para aumentar comput.

→ Los SP son t_c , t_s y t_w

→ El único AP es p

Autooptimización en sistemas homogéneos

$$t_s + nt_w \longrightarrow$$



→ Diferentes posibilidades estimar parámetros sistema

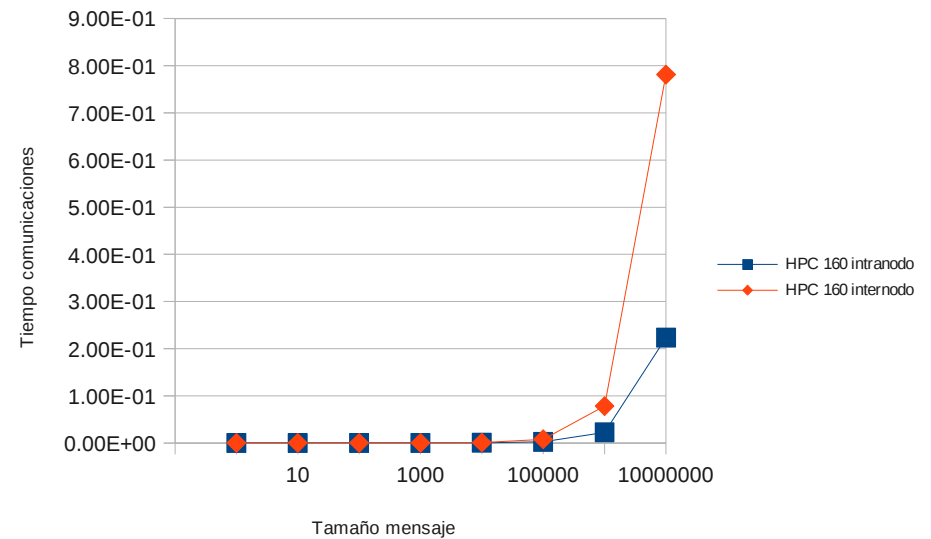
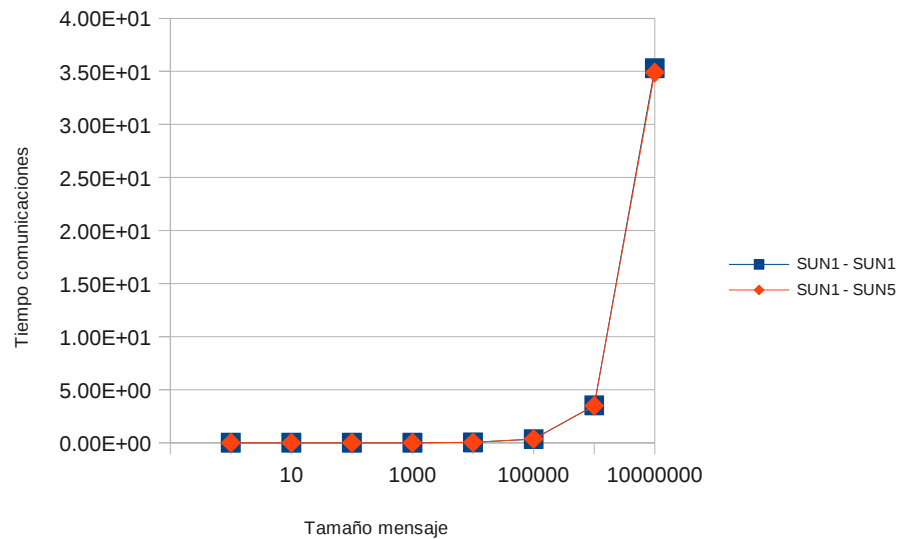
→ Diferencias en speed-up de sistemas



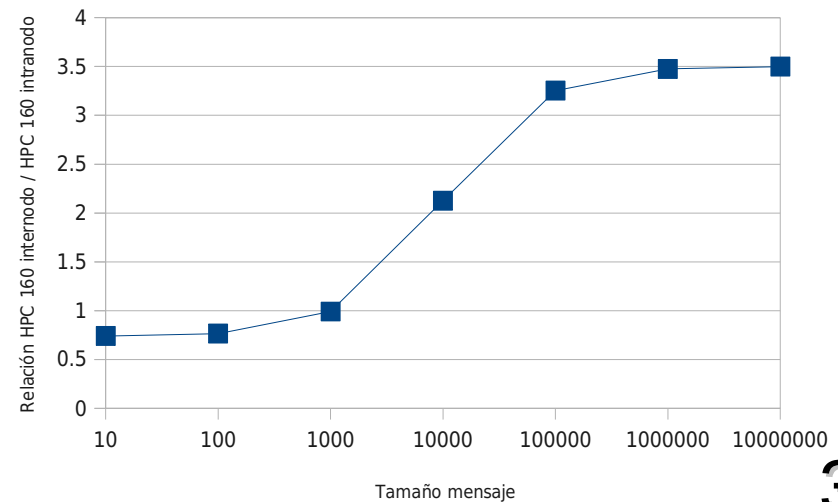
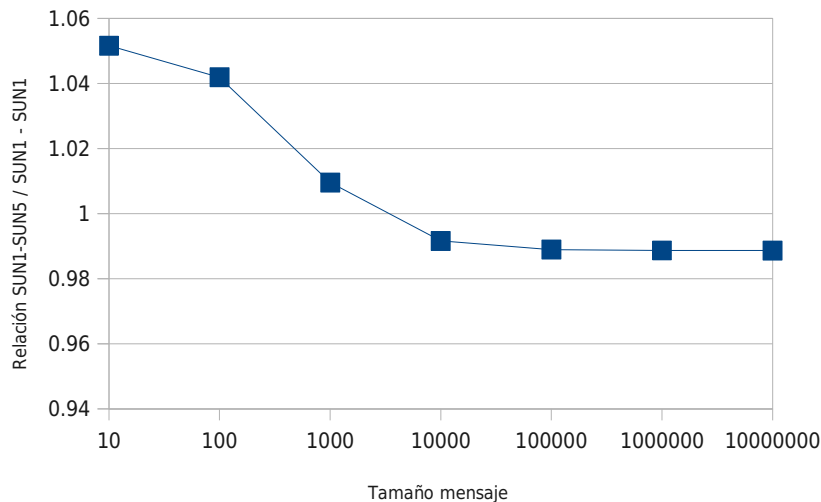
Utilidad autooptimización

Autooptimización en sistemas homogéneos

Tiempos de comunicaciones en SUNEt (izq) y HPC160 (der)



Ratios de tiempos de comunicaciones en SUNEt (izq) y HPC160 (der)



Autooptimización en sistemas homogéneos

Comparativa entre el número de procesos con los que se obtiene mejor tiempo de ejecución

Cantidad	Granularidad	SUNet	PenFE	ORIGIN 2000	HPC 160
10.000	10	3	1	4	5
10.000	50	6	5	5	6
10.000	100	6	6	5	6
50.000	10	4	1	6	5
50.000	50	6	6	6	6
50.000	100	6	5	6	6
100.000	10	3	1	5	5
100.000	50	3	4	6	5
100.000	100	3	6	6	5
500.000	10	3	1	5	5
500.000	50	3	3	5	6
500.000	100	3	6	6	6

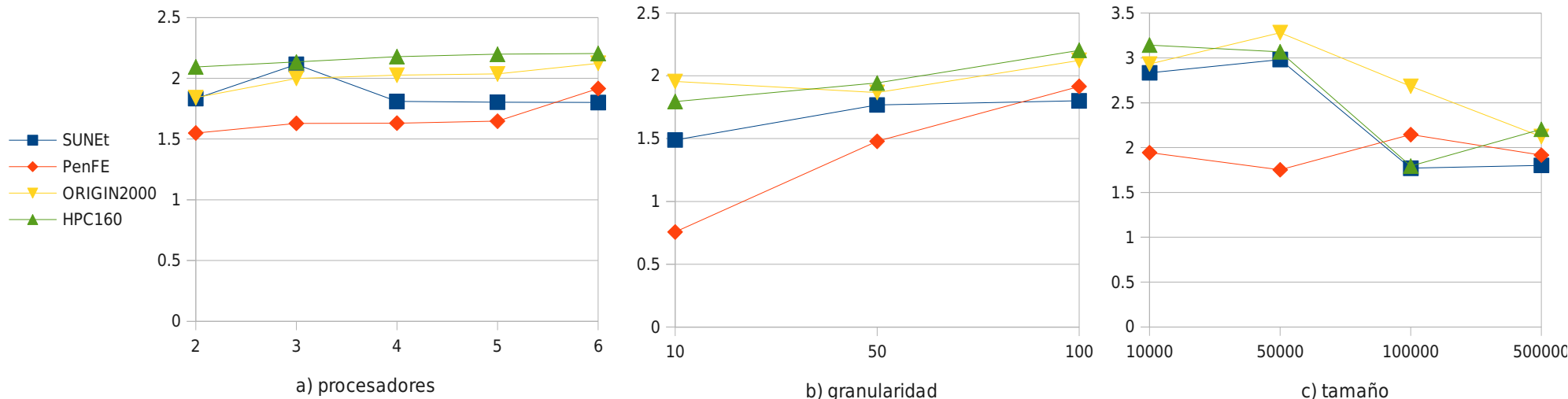
El sistema debe decidir cómo ejecutar el algoritmo

Autooptimización en sistemas homogéneos

Media de los speedups máximos variando tamaño y granularidad

Sistema	SUNet	PenFE	ORIGIN 2000	HPC intranodo	HPC internodo
Speedup máximo	1,08287	1,50725	2,39445	2,23616	2,83555

Speedup variando procesadores, granularidad y tamaño



Autooptimización en sistemas homogéneos

→ Estimación de SP aritméticos: resolviendo un problema reducido

→ Estimación de SP de comunicac:

- Ping-pong (CP1)
- Solución de problema reducido variando el número de procesadores (CP2)
- Solución de problema reducido variando el número de procesadores y el tamaño del problema (CP3)

Comparativa del número de procesadores seleccionados con diferentes métodos con respecto al tmb

Granularidad

	10	50	100
--	----	----	-----

SUNEt

Tamaño	tmb	cp1	cp2	cp3	tmb	cp1	cp2	cp3	tmb	cp1	cp2	cp3
10.000	1	1	1	1	6	6	1	6	6	6	1	6
50.000	1	1	1	1	1	6	1	4	6	6	1	5
100.000	1	1	1	1	1	6	1	4	5	6	1	5
500.000	1	1	1	1	1	6	1	4	1	6	1	5

PenFE

Tamaño	tmb	cp1	cp2	cp3	tmb	cp1	cp2	cp3	tmb	cp1	cp2	cp3
10.000	1	6	1	1	5	7	1	7	6	7	5	7
50.000	1	6	1	1	7	7	1	6	7	7	7	7
100.000	1	6	1	1	4	7	1	6	6	7	7	7
500.000	1	6	1	1	7	7	1	6	6	7	7	7

Autooptimización en sistemas homogéneos

Cocientes entre tiempos de ejecución obtenido con los diferentes métodos con respecto al tmb

Granularidad

	10	50	100
--	----	----	-----

SUNet

Tamaño	cp1	cp2	cp3	cp1	cp2	cp3	cp1	cp2	cp3
10.000	1,00	1,00	1,00	1,00	1,33	1,00	1,00	1,44	1,00
50.000	1,00	1,00	1,00	1,52	1,00	1,13	1,00	1,13	1,07
100.000	1,00	1,00	1,00	1,04	1,00	1,13	1,23	1,60	1,00
500.000	1,00	1,00	1,00	1,25	1,00	1,19	1,20	1,00	1,18

PenFE

Tamaño	cp1	cp2	cp3	cp1	cp2	cp3	cp1	cp2	cp3
10.000	1,64	1,00	1,00	1,02	1,60	1,02	1,02	1,02	1,02
50.000	1,56	1,00	1,00	1,00	1,61	1,04	1,00	1,00	1,00
100.000	2,03	1,00	1,00	1,07	1,65	1,10	1,10	1,10	1,15
500.000	1,32	1,00	1,00	1,00	1,50	1,01	1,15	1,15	1,15

RATIOS

cp1/tmb	cp2/rmb	cp3/tmb
1,173	1,149	1,044

Autooptimización en sistemas homogéneos

→ Comparación con usuarios:

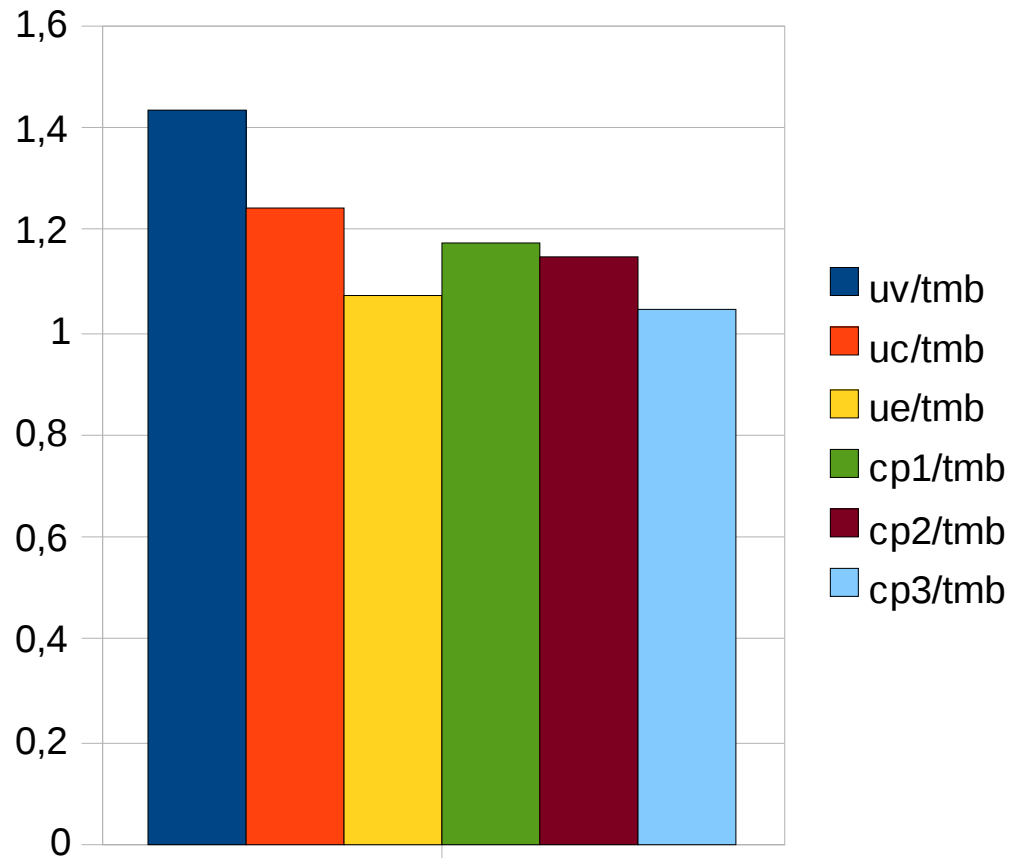
- Voraz (UV): usa todos los procesadores disponibles
- Conservador (UC): usa la mitad de los procesadores disponibles
- Experto (UE): usa un número de procesadores en función del tamaño del problema

Cocientes entre tiempos de ejecución obtenido con los diferentes usuarios con respecto al tmb

													Granularidad											
													10			50			100					
													SUNet											
Tamaño	uv	uc	ue	cp3	uv	uc	ue	cp3	uv	uc	ue	cp3												
10.000	2,31	2,23	1	1	1	1,33	1,33	1	1	1,41	1	1												
50.000	2,19	1,51	1	1	1,52	1,08	1,08	1,13	1	1,2	1	1,07												
100.000	3,78	1,45	1	1	1,04	1,15	1,15	1,13	1,23	1,19	1,23	1												
500.000	1,87	1,44	1	1	1,25	1,14	1,14	1,19	1,2	1,11	1,2	1,18												
													PenFE											
Tamaño	uv	uc	ue	cp3	uv	uc	ue	cp3	uv	uc	ue	cp3												
10.000	1,73	1,66	1	1	1,02	1,12	1,12	1,02	1,02	1,12	1,02	1,02												
50.000	1,53	1,57	1	1	1	1,08	1,08	1,04	1	1,1	1	1												
100.000	2,03	1,26	1	1	1,07	1,05	1,05	1,1	1,1	1,23	1,1	1,15												
500.000	1,35	1,27	1	1	1	1,01	1,01	1,01	1,15	1,18	1,15	1,15												
													RATIOS											
													uv/tmb		uc/tmb		ue/tmb		cp3/tmb					
													1,433		1,245		1,070		1,044					

Autooptimización en sistemas homogéneos

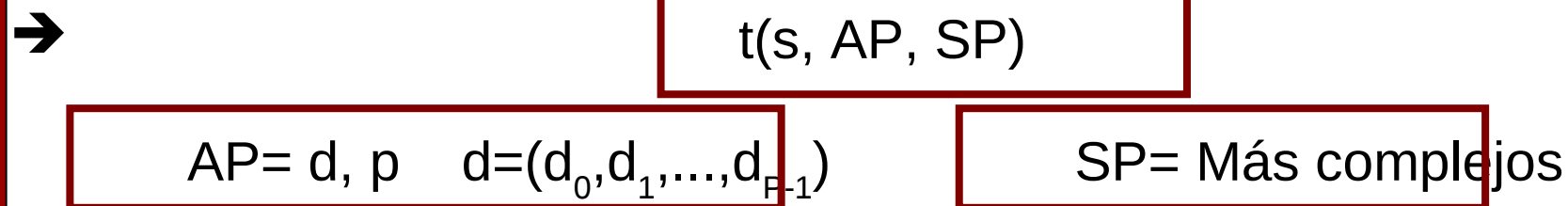
Cocientes entre tiempos de ejecución obtenidos con los diferentes usuarios y métodos con respecto al tmb



AUTOOPTIMIZACIÓN EN SISTEMAS HETEROGÉNEOS

Autooptimización en sistemas heterogéneos

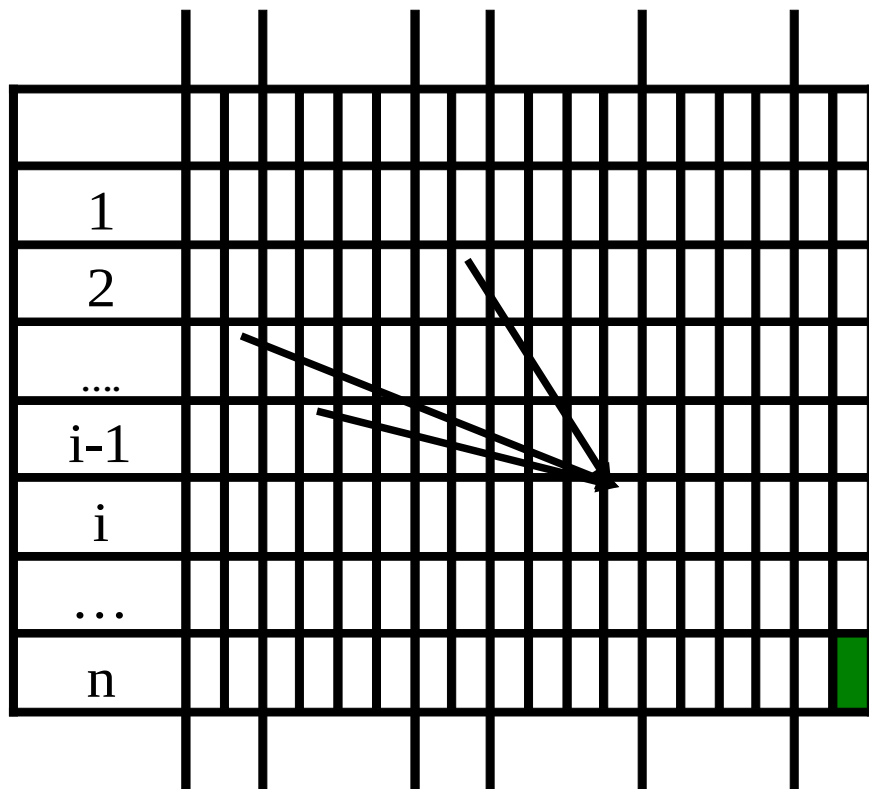
- Heterogéneos: agrupar elementos del sistema según características similares no exactamente iguales.
- Mayor dificultad construir modelo tiempo ejecución.
- Buscar parámetros algorítmicos / min t^0 ejec



- Diferentes posibilidades {
 - Construir algoritmos heterogéneos
 - Desarrollo métodos (exactos y/o aproximados de mapeo (asignar procesos a procesadores))

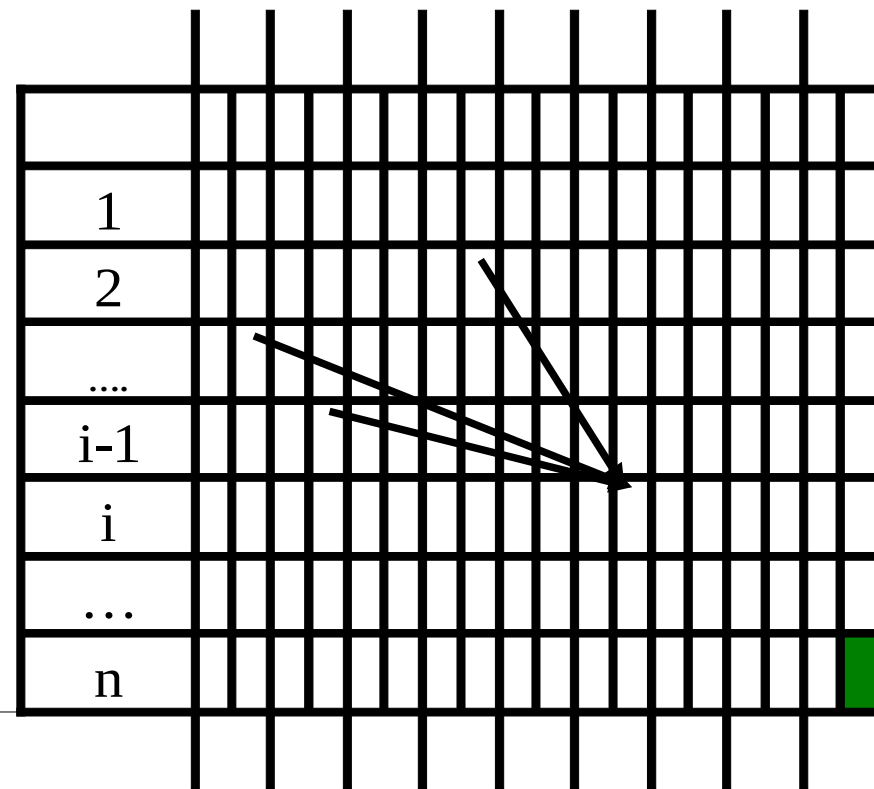
Autooptimización en sistemas heterogéneos

Distribución del trabajo y asignación de procesos a procesadores en un esquema de programación dinámica, en algoritmos heterogéneos (HeHo) (a) y homogéneos (HoHe) (b)



P_0 P_1 P_2 ... P_S ... P_{K-1} ... P_{P-1}

a)

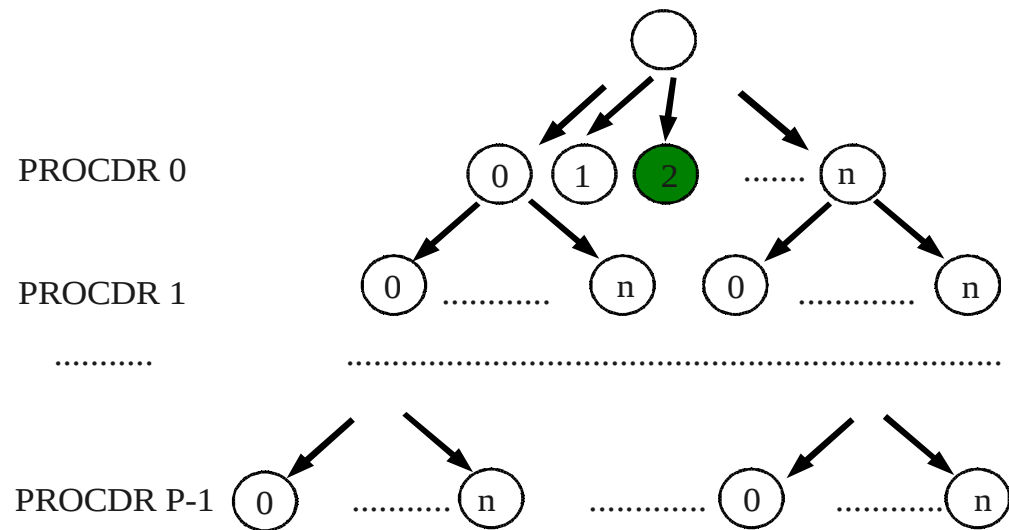
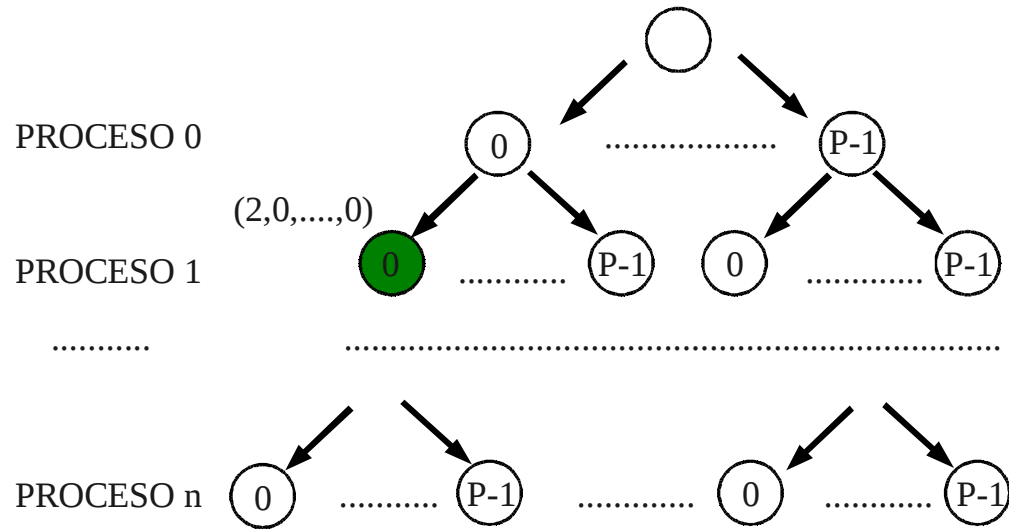


P_0 P_1 P_2 P_3 P_4 P_S .. P_{r-1} P_r
 P_0 P_1 P_1 P_2 ... P_S ... P_K P_K P_{P-1}

b)

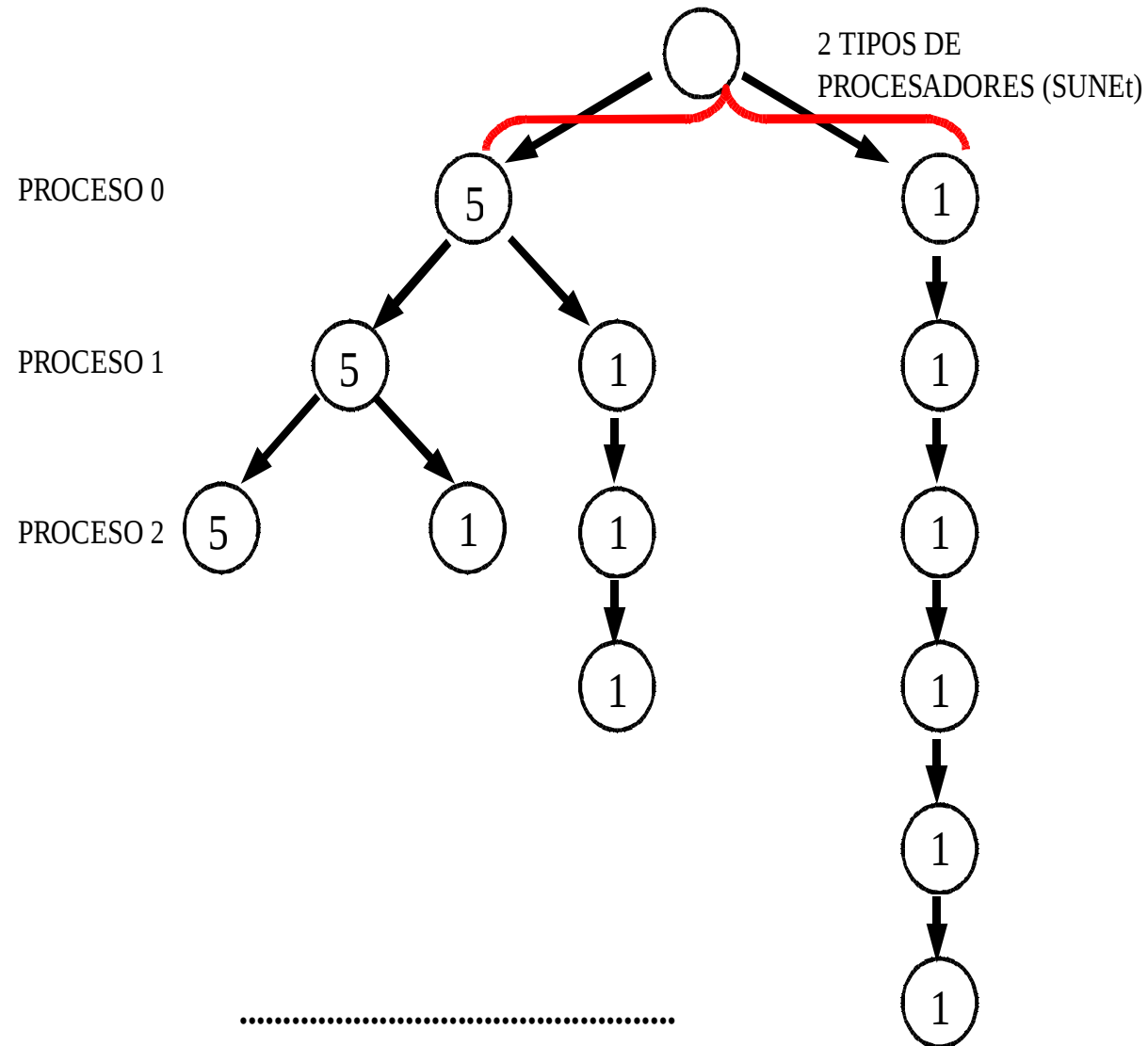
Autooptimización en sistemas heterogéneos

→ Posibilidades de representación del árbol de asignación:



Autooptimización en sistemas heterogéneos

- Árbol de asignación con 2 tipos de procesadores y p procesos
- Necesidad de limitar la altura del mismo
- Cada nodo lleva asociada 3 cotas: EET, LET, GET.
- Uso de técnicas de búsqueda en árboles para optimizar el modelo de t^o ejecución



Autooptimización en sistemas heterogéneos

Desviación con respecto al tmb obtenido experimentalmente, de los tiempos obtenidos con diferentes métodos y usuarios en una combinación de TORC 1 17P4 + 1 Ath + 1 SPIII + 8 DPIII (a) y 1 Ath + 1 SPIII + 8 DPIII (b)

Cantidad	Granularidad	cp1	cp2	ue	uc	uv
50.000	10	0,00	0,00	0,18	0,43	0,73
50.000	50	0,40	0,00	0,24	0,27	0,21
50.000	100	0,39	0,00	0,05	0,25	0,08
100.000	10	0,00	0,00	0,14	0,67	0,43
100.000	50	0,61	0,01	0,43	0,90	0,00
100.000	100	0,72	0,07	0,00	0,82	0,04
500.000	10	0,00	0,00	0,14	0,66	0,95
500.000	50	0,61	0,00	0,48	0,76	1,07
500.000	100	0,61	0,00	0,92	0,76	0,84
PROMEDIO		0,37	0,01	0,29	0,61	0,48
TOTAL		0,56	0,01	0,59	0,73	0,84

a)

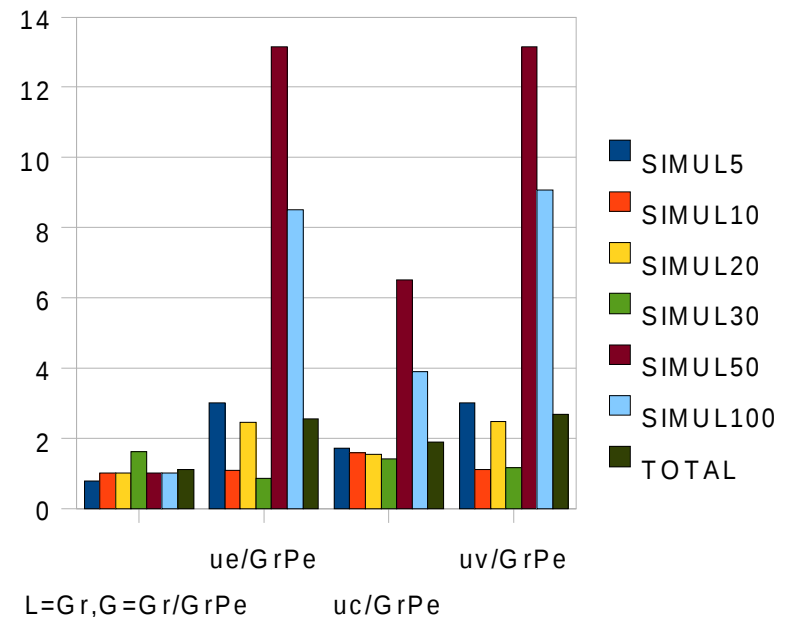
Cantidad	Granularidad	cp1	cp2	ue	uc	uv
50.000	10	0,35	0,38	0,40	0,51	0,00
50.000	50	0,88	0,48	0,48	0,37	0,07
50.000	100	0,88	0,49	0,02	0,33	0,10
100.000	10	0,07	0,07	0,09	0,34	0,26
100.000	50	0,83	0,53	0,53	0,93	0,04
100.000	100	1,02	0,68	0,07	1,13	0,09
500.000	10	0,00	0,09	0,06	0,32	0,41
500.000	50	0,01	0,00	0,00	0,24	0,32
500.000	100	0,00	0,12	0,18	0,36	0,32
PROMEDIO		0,45	0,32	0,20	0,50	0,18
TOTAL		0,14	0,12	0,18	0,36	0,32

b)

Autooptimización en sistemas heterogéneos

Resultados de simulaciones diversas

SISTEMA	L=Gr, G=Gr	GrPe	ue	uc	uv
SIMUL5	4,72	6,08	18,26	10,44	18,26
SIMUL10	5,72	5,72	6,16	9,03	6,29
SIMUL20	1,62	1,62	3,97	2,51	4,00
SIMUL30	10,14	6,28	5,44	8,86	7,30
SIMUL50	1	1,00	13,14	6,50	13,14
SIMUL100	1	1,00	8,51	3,91	9,06
TOTAL	24,2	21,70	55,48	41,25	58,05



METAHEURÍSTICAS EN LA AUTOOPTIMIZACIÓN

Metaheurísticas en la autooptimización

OBJETIVO FINAL



min tº ejec

Minimización analítica



Minimización numérica



Métodos exhaustivos



Métodos aproximados



Métodos heurísticos

Metaheurísticas en la autooptimización

Esquema General de las Metaheurísticas

Inicializar(S)

while no se cumple CondiciondeFin(S) **do**

 SS = ObtenerSubconjunto(S);

if |SS|>1 **then**

 SS1= Combinar(SS);

else

 SS1=SS;

end

 SS2 = Mejorar(SS1);

 S = IncluirSoluciones(SS2)

end

Metaheurísticas en la autooptimización

Esquema de la técnica de Búsqueda Dispersa (particularización)

CrearConjuntoInicial S;

GenerarConjuntoReferencia RS;

while Convergencia no alcanzada **do**

 Seleccionar elementos a combinar;

 Combinar elementos seleccionados;

 Mejorar elementos combinados;

 ActualizarConjuntoReferencia RS con los elementos más prometedores y los más dispersos

end

Metaheurísticas en la autooptimización

→ Mapeo como un problema de optimización

→ Representación y codificación soluciones: $(d_0, d_1, \dots, d_{p-1})$

$d_i = n^o$ procesos asignados al procesador i

→ Gran cantidad de alternativas a la hora de instanciar el algoritmo:

Generar el conjunto inicial de referencia	Resultados
Considerando número procesos	80%
Aleatoriamente	20%

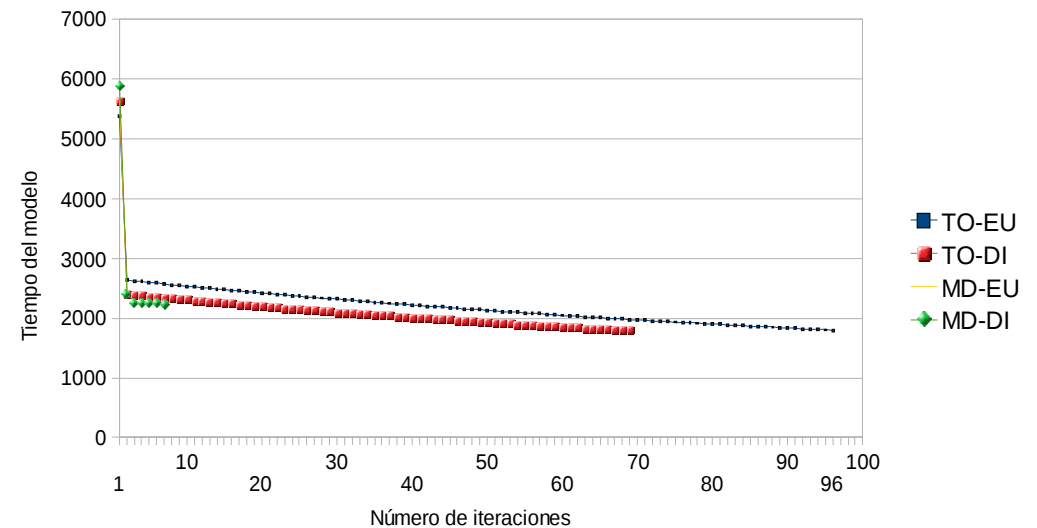
Convergencia alcanzada	Resultados
Media	80%
Mejores elementos	20%

Metaheurísticas en la autooptimización

Comparación entre las posibilidades de opciones de selección e inclusión. Porcentaje en que mejora la búsqueda dispersa respecto al backtracking con poda

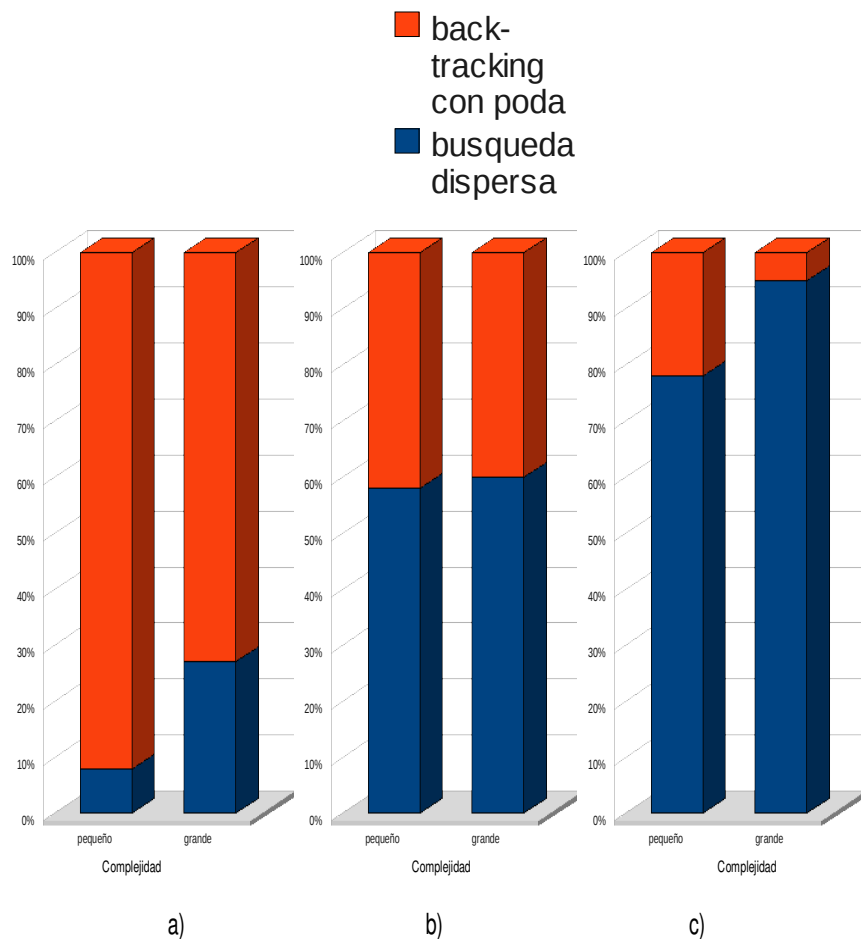
Incluir	Seleccionar elementos	
	Todos (TO)	Mejores/dispersos (MD)
Distancia euclídea (EU)	90%	85%
Diferencia de componenetes (DI)	91%	86%

Tiempos e iteraciones para diferentes métodos de selección e inclusión en el conjunto de referencia



Metaheurísticas en la autooptimización

→ Simulaciones para comparar Búsqueda dispersa y backtracking con poda

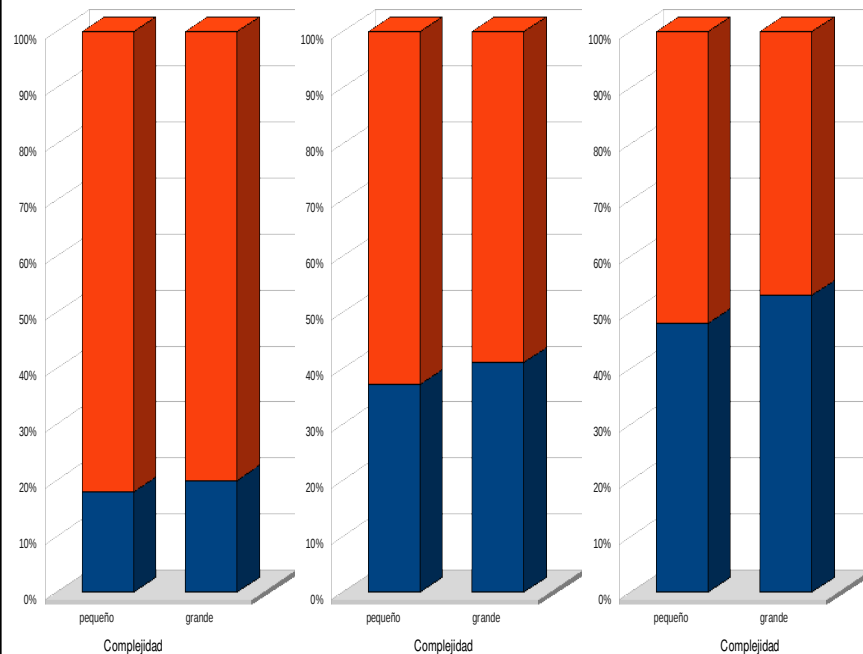


Tamaño	Complejidad	Backtracking	Búsqueda dispersa
Pequeño [1,...,100000]	Pequeño [1,...,100]	92%	8%
Pequeño [1,...,100000]	Grande [100,...,400]	73%	27%
Mediano [100000,...,500000]	Pequeño [1,...,100]	42%	58%
Mediano [100000,...,500000]	Grande [100,...,400]	40%	60%
Grande [500000,...,1000000]	Pequeño [1,...,100]	22%	78%
Grande [500000,...,1000000]	Grande [100,...,400]	5%	95%

Metaheurísticas en la autooptimización

→ Pruebas en Kipling

■ back-tracking con poda
■ búsqueda dispersa



a)

b)

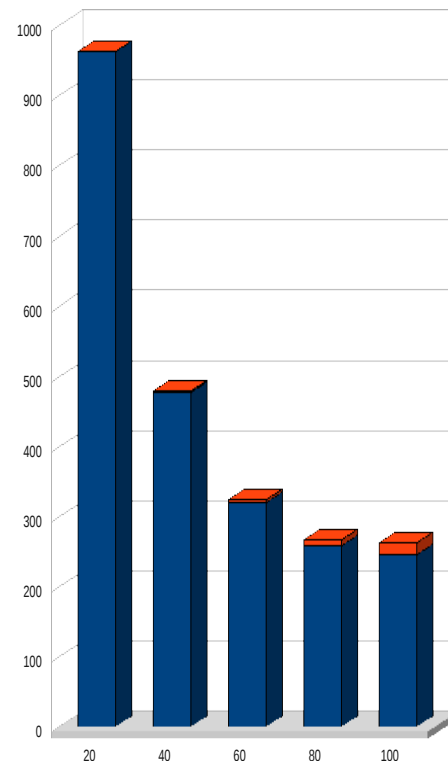
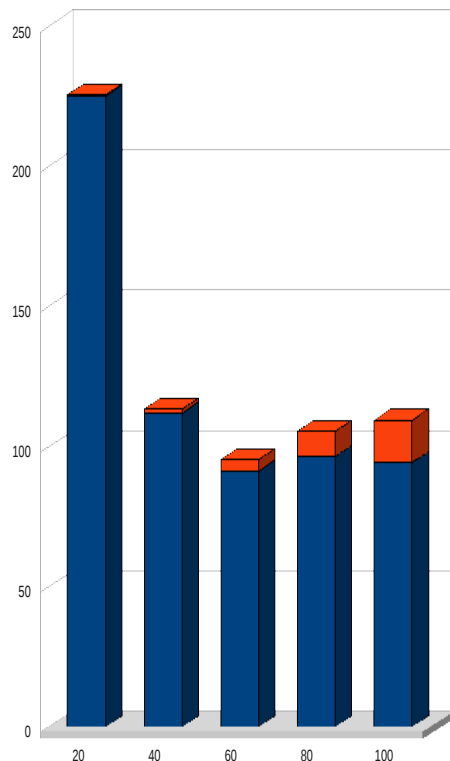
c)

Tamaño	Complejidad	Backtracking	Búsqueda dispersa
Pequeño [1,...,100000]	Pequeño [1,...,100]	82%	18%
Pequeño [1,...,100000]	Grande [100,...,400]	80%	20%
Mediano [100000,...,500000]	Pequeño [1,...,100]	63%	37%
Mediano [100000,...,500000]	Grande [100,...,400]	59%	41%
Grande [500000,...,1000000]	Pequeño [1,...,100]	52%	48%
Grande [500000,...,1000000]	Grande [100,...,400]	47%	53%

Metaheurísticas en la autooptimización

→ Relación entre tiempos de decisión y modelado en una simulación concreta

■ t decisión
■ t modelo



Procesadores	Complejidad 100		Complejidad 400	
	Tiempo modelo	Tiempo decisión	Tiempo modelo	Tiempo decisión
20	225,29	0,28	963,10	0,27
40	112,07	1,45	477,12	1,48
60	91,41	4,03	320,39	4,31
80	96,57	8,93	258,45	8,88
100	94,32	15,15	246,54	15,98

CONCLUSIONES Y TRABAJOS FUTUROS

Conclusiones y Trabajos Futuros

→ Optimización



Autooptimización

→ Hipótesis investigación



$t^{\circ} = f(s, AP, SP)$
+
Búsqueda óptimo

Alternativas



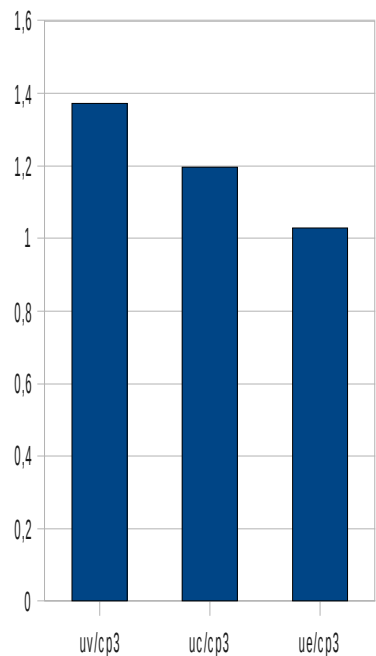
Métodos exactos (Búsqueda exhaustiva)

Métodos aproximados

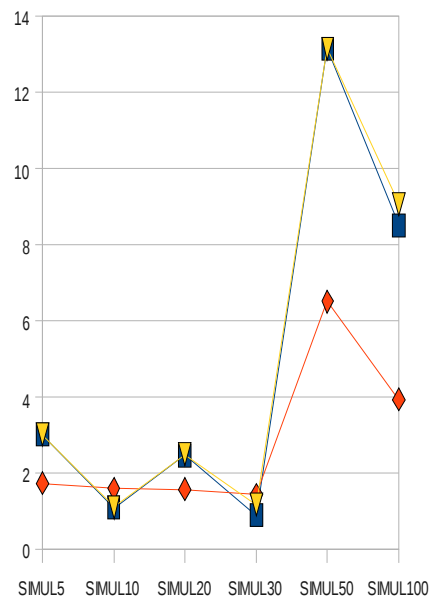
Métodos metaheurísticos

Conclusiones y Trabajos Futuros

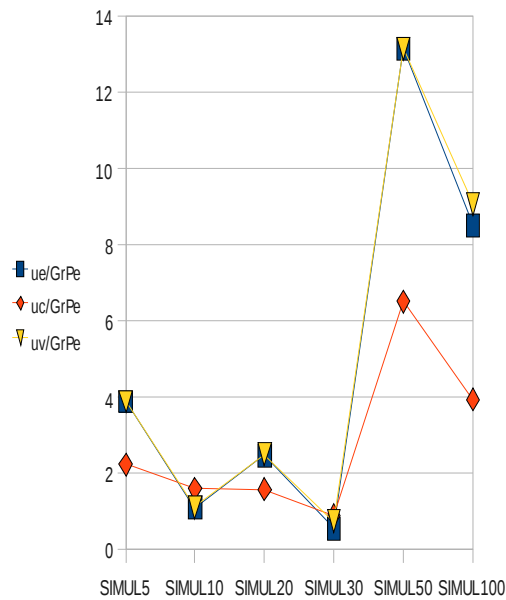
→ Resultados generales en sistemas homogéneos(a), heterogéneos (b) y con el uso de metaheurísticas(c)



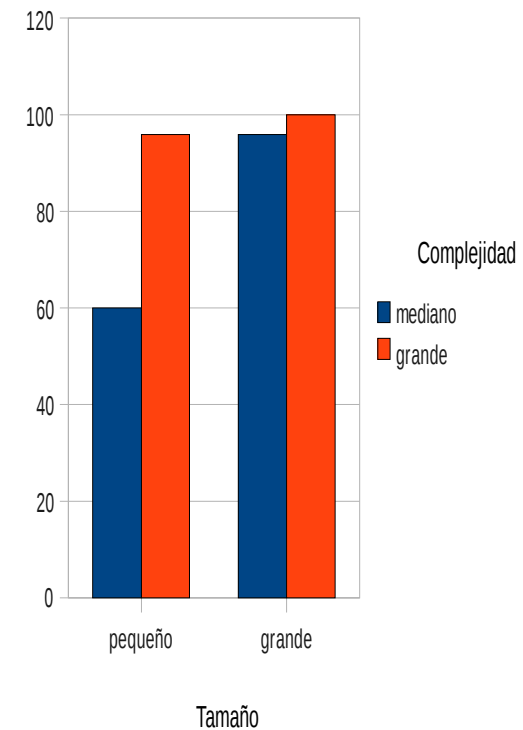
a)



a)



b)



c)

Conclusiones y Trabajos Futuros

Aportaciones

- VecPar04. Sistemas homogéneos.
- HeteroPar04, Parallel Computing04. Sistemas heterogéneos.
- Para06, Maeb07. Metaheurísticas.
- Cluster07.
- ICCS08.
- Journal of Supercomputing 09.

Trabajos futuros

- Mejorar el modelado el tiempo de ejecución, sobre todo en sistemas heterogéneos
- Aplicación en diferentes esquemas algorítmicos (divide & conquer, master - slave, backtracking...)
- Metaheurísticas diversas (tabú, temple simulado, genéticos ...)
- Construcción de un entorno para la prueba y evaluación de metaheurísticas.