

ALGORÍTMICA

2012 - 2013

- **Parte I. Introducción a las Metaheurísticas**
 - Tema 1. Metaheurísticas: Introducción y Clasificación
- **Parte II. Métodos Basados en Trayectorias y Entornos**
 - Tema 2. Algoritmos de Búsqueda Local Básicos
 - Tema 3. Algoritmos de Enfriamiento Simulado
 - Tema 4. Algoritmos de Búsqueda Tabú
 - Tema 5. Métodos Basados en Trayectorias Múltiples I: Métodos Multiarranque Básicos y GRASP
 - Tema 6. Métodos Basados en Trayectorias Múltiples II: ILS y VNS
- **Parte III. Métodos Basados en Poblaciones**
 - Tema 7. Algoritmos Genéticos
- **Parte IV. Intensificación y Diversificación**
 - Tema 8. Estudio del Equilibrio entre Intensificación y Diversificación
- **Parte V. Metaheurísticas Híbridas: Poblaciones y Trayectorias**
 - Tema 9. Algoritmos Meméticos
 - Tema 10. Scatter Search
- **Parte VI. Paralelización de Metaheurísticas**
 - **Tema 11. Metaheurísticas en Sistemas Descentralizados**
- **Parte VII. Conclusiones**
 - Tema 12. Algunas Consideraciones sobre la Adaptación de Metaheurísticas a la Resolución de Problemas

ALGORÍTMICA

TEMA 11: METAHEURÍSTICAS EN SISTEMAS DESCENTRALIZADOS (Metaheurísticas Paralelas)

- 1. INTRODUCCIÓN**
- 2. SISTEMAS DE CÁLCULO PARALELOS**
- 3. PROGRAMACIÓN PARALELA/DISTRIBUIDA**
- 4. METAHEURÍSTICAS PARALELAS**
- 5. DESCOMPOSICIÓN FUNCIONAL DE BAJO NIVEL**
- 6. DESCOMPOSICIÓN DE DOMINIO**
- 7. MÚLTIPLES BÚSQUEDAS INDEPENDIENTES**
- 8. MÚLTIPLES BÚSQUEDAS COOPERATIVAS**

Bibliografía

- E. Alba (ed.), "Parallel Metaheuristics", John Wiley & Sons, 2005
- T.G. Crainic, M. Toulouse, Chap. 17: "Parallel Strategies for Metaheuristics." In F. Glover, G.A. Kochenberber (Eds.) Handbook of Metaheuristics, Kluwer Academic Publisher (2003)
- T.G. Crainic, M. Toulouse, "Parallel Meta-Heuristics" Technical Report (<http://www.cirrelt.ca/DocumentsTravail/CIRRELT-2009-22.pdf>)

Bibliografía (programación paralela)

- S. Akhter, J. Roberts, "Increasing Performance through Software Multithreading", Intel Press, 2003
- V. Kumar, A. Grana, A. Gupta, G. Karypis, "Introduction to Parallel Programming", Benjamin/Cummings Publishing Company, 2003
- OpenMP (www.openmp.org)
- OpenMPI (www.open-mpi.org)
- M.J. Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw-Hill, 2003

Contenido

1. Objetivos de la paralelización
2. Sistemas de cálculo paralelos. Programación paralela
3. Metaheurísticas paralelas
4. Descomposición funcional de bajo nivel
5. Descomposición de dominio
6. Múltiples búsquedas independientes
7. Múltiples búsquedas cooperativas

1. Introducción

- Las metaheurísticas son, típicamente, algoritmos de cálculo intensivo: supercomputación
- Tendencias en el diseño de ordenadores y supercomputadores
- Enfoque de cálculo cooperativo

1. Introducción

Objetivos de la paralelización

- Reducir el tiempo de cálculo
- Resolver problemas de tamaño mayor en un tiempo dado
- Obtener soluciones de mejor calidad sin incrementar el tiempo de cálculo:
 - Aumento del número de iteraciones
 - Incremento de la diversidad y evitación de la convergencia prematura

2. Sistemas de cálculo paralelos

- Evolución en la potencia de cálculo
- Cálculo paralelo *versus* distribuido
- Elementos de un sistema paralelo

2. Sistemas de cálculo paralelos

Evolución de la potencia de cálculo

- La mayoría de los ordenadores actuales (construidos según el modelo de von Neuman) son secuenciales
- La ley (empírica) de Moore ha marcado la evolución en la potencia de cálculo, pero se aproxima a su fin
- Los diseñadores de ordenadores buscan soslayar la ley de Moore buscando paralelismo a distintos niveles
- Equipos para altas prestaciones: supercomputadores (www.top500.org)

2. Sistemas de cálculo paralelos

Tendencias en sistemas de cálculo

- En arquitectura de microprocesadores
 - Multihebra: Pentium IV/Xeon
 - Multinúcleo: (Core 2 Quad)
 - Combinados: Procesador i7
- En arquitectura de sistema
 - Multiprocesador: 2, 4, 8, ... procesadores
 - Sistemas clúster: ordenadores interconectados por redes de bajo coste
- Arquitecturas específicas
 - Procesadores sistólicos, vectoriales, ...

2. Sistemas de cálculo paralelos

Cálculo paralelo *versus* distribuido

- Sólo se puede hablar de paralelismo real cuando existen dos ejecuciones al mismo tiempo, y no de forma intercalada (tiempo compartido)
- El paralelismo puede ser *total* (procesadores o núcleos independientes) o *parcial* (hebras)
- De forma estricta, se habla de **procesamiento paralelo** cuando los procesadores usan memoria compartida (comunicación rápida)
- Se habla de **procesamiento distribuido** cuando los procesadores usan memoria distribuida (nodos interconectados en red → comunicación más lenta)

2. Sistemas de cálculo paralelos

Elementos de un sistema paralelo

- **Nodos de cálculo:**
 - Procesadores
 - Arquitectura interna
- **Red de comunicaciones:**
 - Tecnología: Bus de sistema, red (Ethernet, Myrinet, ...)
 - Topología:
 - Anillo
 - Estrella
 - Grid (celulares, toros)
 - Interconexión total

3. Programación paralela/distribuida

- Concepto
- Aspectos clave
- Herramientas:
 - OpenMP
 - OpenMPI
- Rendimiento de la paralelización

3. Programación paralela/distribuida

Conceptos

- Procesamiento paralelo/distribuido significa que varios procesos trabajan simultáneamente en procesadores independientes para resolver un caso concreto de un problema
- El paralelismo surge de una descomposición de la carga de trabajo y de su distribución

3. Programación paralela/distribuida

Algoritmos y software paralelo

- La paralelización de un programa o algoritmo no es trivial. De modo genérico, no se puede automatizar
- Para algunos procesos concretos existen herramientas que pueden ayudar, pero siempre es necesaria la intervención del programador
- Lenguajes y herramientas para desarrollo en paralelo (OpenMP y OpenMPI)

3. Programación paralela/distribuida

Aspectos clave

- Los dos aspectos más importantes a tener en cuenta en el diseño de un algoritmo paralelo son:
 - Dependencias temporales entre los datos (uso de mecanismos de sincronización)
 - Acceso en exclusión mutua a datos compartidos (puede producir interbloqueos)
- Elementos muy importantes:
 - Sistemas homogéneos o heterogéneos
 - Síncronos *versus* asíncronos
 - Grano de paralelismo: fino *versus* grueso

3. Programación paralela/distribuida

Programación paralela en C/C++: OpenMP

- Los lenguajes de programación C/C++ no incorporan mecanismos propios para programación paralela → extensiones
- OpenMP: API que soporta programación paralela con memoria compartida, multiplataforma, en C/C++ y Fortran
- Extensión del lenguaje de programación básico a partir de #pragmas

3. Programación paralela/distribuida

MPI

- MPI: biblioteca para paso de mensajes (programación paralela y distribuida)
- Soporte para el desarrollo de plataformas de comunicación y aplicaciones paralelas/distribuidas
- Es multiplataforma (hardware) y multilenguaje
- Existen diversas implementaciones:
 - LAM MPI
 - MPICH
 - OpenMPI

3. Programación paralela/distribuida

Rendimiento de un algoritmo paralelo

- La medida básica de rendimiento es la **ganancia** (speed-up):

$$ganancia = \frac{t_{\text{secuencial}}}{t_{\text{paralelo}}}$$

- El límite máximo es el número de procesadores
- Pueden darse ganancias "super-lineales"
- Necesidad de adaptar estos conceptos al ámbito de las metaheurísticas por la dificultad de «realizar el mismo cálculo» (no se obtiene el óptimo).

4. Metaheurísticas paralelas

- Fuentes de paralelismo
- Metaheurísticas basadas en entornos y en poblaciones
- Clasificación de las metaheurísticas

4. Metaheurísticas paralelas

Fuentes de paralelismo

- El paralelismo surge de una descomposición del trabajo y reparto entre las unidades de ejecución
- La descomposición puede afectar a:
 - El algoritmo (paralelismo funcional)
 - Los datos del problema (paralelismo de datos o descomposición del dominio)
 - a estructura del problema (descomposición del problema en base a atributos)
- Desde un punto de vista algorítmico: paralelización de las evaluaciones en el bucle interno (población o vecindad)

4. Metaheurísticas paralelas

Metaheurísticas basadas en entornos y en poblaciones

- El proceso de descentralización es diferente según el tipo de metaheurística
- Básicamente, podemos distinguir entre dos tipos:
 1. Algoritmos de [Búsqueda basados en Entornos](#): sólo mantienen una solución actual en cada momento
 2. Algoritmos de [Búsqueda basados en Poblaciones](#): mantienen un conjunto de soluciones en cada momento

4. Metaheurísticas paralelas

Clasificación de heurísticas paralelas

- Taxonomía de Crainic y Nourredine
- Dimensiones:
 - Control global del proceso (cardinalidad del control de la búsqueda)
 - Intercambio de información entre los procesos (tipo de control y comunicaciones de búsqueda)
 - Variedad de los métodos de búsqueda empleados
- Con esta taxonomía, cada metaheurística paralela se denota con una terna (A/B/C). Por ejemplo: 1C/RS/SPSS

4. Metaheurísticas paralelas

Cardinalidad del control de la búsqueda

- Hay dos posibilidades para el control global de la búsqueda:
- Control ejercido por un único proceso: 1C
Uno sólo proceso monitoriza toda la búsqueda
- Control por varios procesos: pC
Control distribuido entre varios procesos independientes

4. Metaheurísticas paralelas

Comunicación y control de búsqueda

- Comunicación en entornos paralelos:
 - Síncrona: implica esperas entre los interlocutores
 - Asíncrona: sin esperas, necesita de almacenamiento temporal para la información recibida y no procesada
- Se distinguen cuatro categorías:
 - Rigid Synchronization (RS)
 - Knowledge Synchronization (KS)
 - Collegial (C)
 - Knowledge Collegial (KC)

4. Metaheurísticas paralelas

Diversidad en los métodos de solución

- Este criterio se refiere a la diversidad en los puntos de inicio y en los métodos de búsqueda empleados:
- Posibilidades:
 - SPSS: Same initial Point/Population, Same search Strategy
 - SPDS: Same initial Point/Population, Different search Strategy
 - MPSS: Mutiple initial Points/Populations, Same search Strategies
 - MPDS: Multiple initial Points/Populations, Different search Strategies

4. Metaheurísticas paralelas

Principales grupos de metaheurísticas paralelas

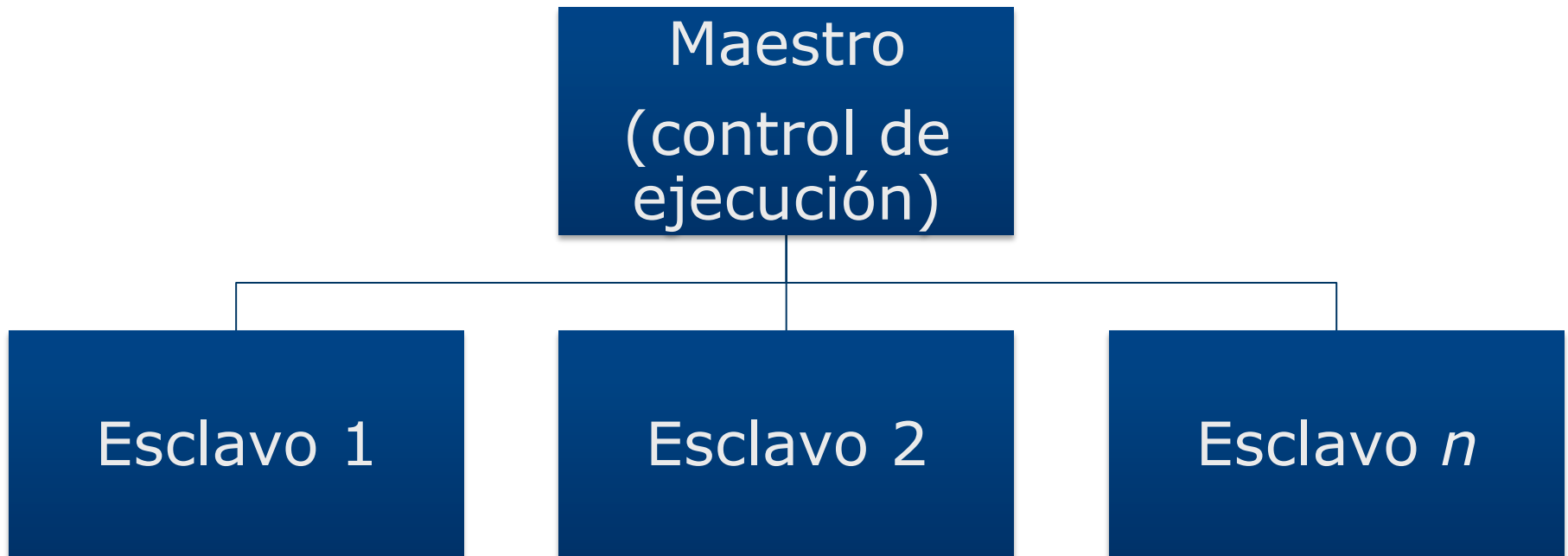
- Grupo 1: Descomposición funcional de bajo nivel
- Grupo 2: Descomposición de dominio
- Grupo 3: Múltiples búsquedas independientes
- Grupo 4: Múltiples búsquedas cooperativas

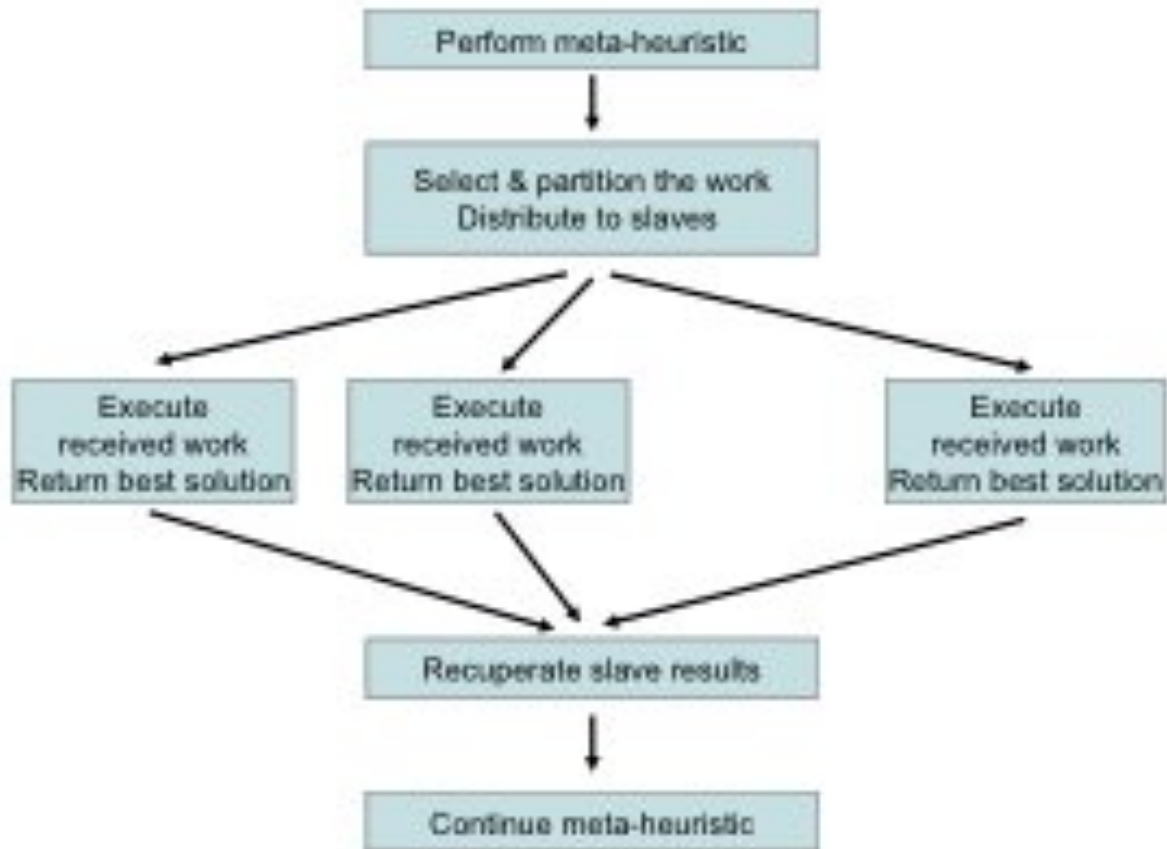
5. Descomposición funcional de bajo nivel

- No hay cambios en la lógica del algoritmo ni en el espacio de búsqueda
- La mayoría se corresponden con el modelo 1C/RS/SPSS. Responden al modelo clásico maestro-esclavo
- El «maestro» lleva el control total de la búsqueda y reparte las tareas de cálculo intensivo entre los «esclavos». Recoge los resultados de sus cálculos y los integra. Inicia la mayoría de las comunicaciones.
- No hay comunicación entre los esclavos

5. Descomposición funcional de bajo nivel

Modelo maestro-esclavo





5. Descomposición funcional de bajo nivel

Estrategias 1C/RS/SPSS para entornos

- Paralelización de la exploración de la vecindad: cada iteración del bucle en un procesador distinto. Si la vecindad es grande se reparte por bloques para amortizar el coste de comunicación
- Máxima flexibilidad y equilibrado automático de carga
- Distribución por grupos cuando la diversidad no es amplia

5. Descomposición funcional de bajo nivel

Estrategias 1C/RS/SPSS para poblaciones

- Para algoritmos genéticos, lo habitual es realizar la evaluación en paralelo de la población o por bloques, si el coste de evaluación no es muy elevado
- Para Scatter Search, se paraleliza el proceso de búsqueda local
- Un enfoque distinto consiste en evolucionar en paralelo varios subconjuntos del conjunto de referencia. El maestro realiza intercambios de individuos seleccionados entre los esclavos

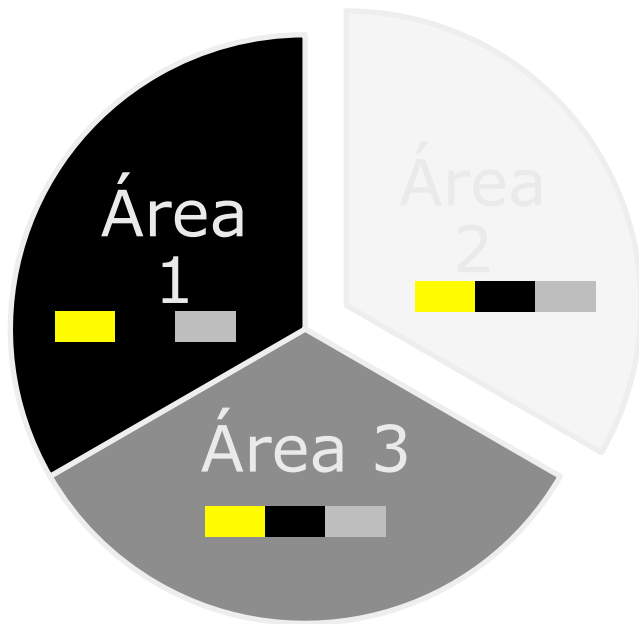
-
- Las ventajas obtenidas con estas paralelizaciones producen, en la mayoría de los casos, reducciones en el tiempo de ejecución, pero no en mejoras en la calidad de las soluciones
 - La calidad en las soluciones se mejora con estrategias de control más elaboradas

6. Descomposición del dominio

- Descomposición del espacio de búsqueda en zonas más pequeñas (disjuntas) y resolver el subproblema en cada zona (divide y vencerás)
- La descomposición puede ser espacial (por regiones) o estructural (subvectores de las soluciones)
- También se diferencia entre particionamiento estricto o con solapamiento
- Metaheurísticas habituales: 1C/KS/MPSS o 1C/KS/MPDS

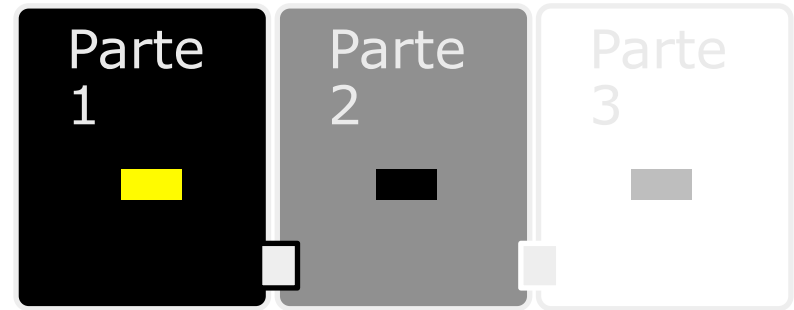
6. Descomposición del dominio

Descomposición del espacio de búsqueda



Subdominios con la misma estructura

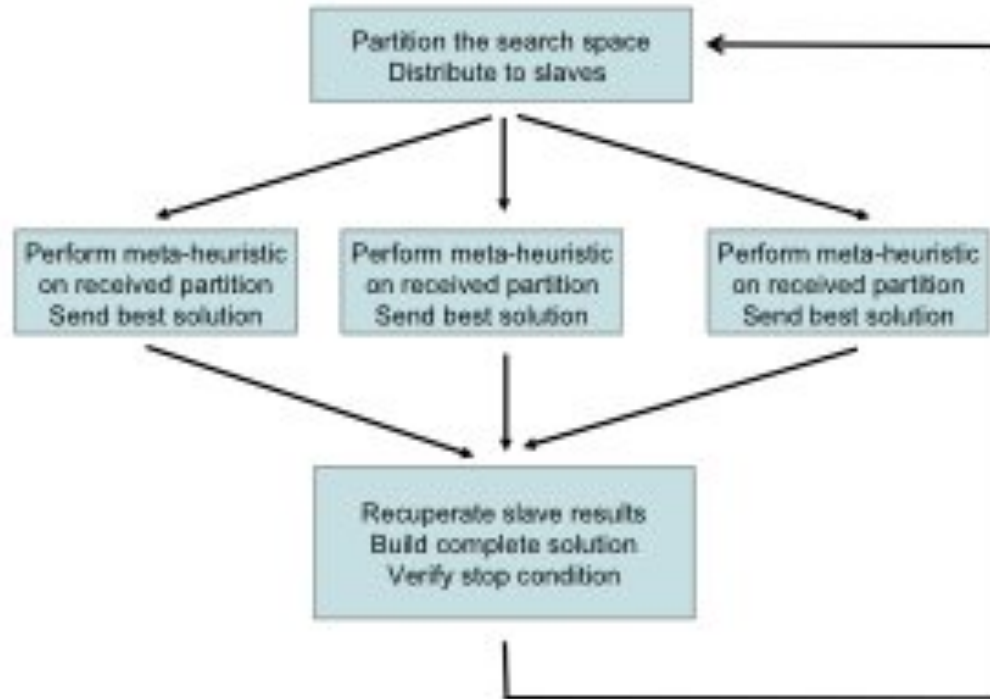
Soluciones: 



Subdominios con estructura distinta

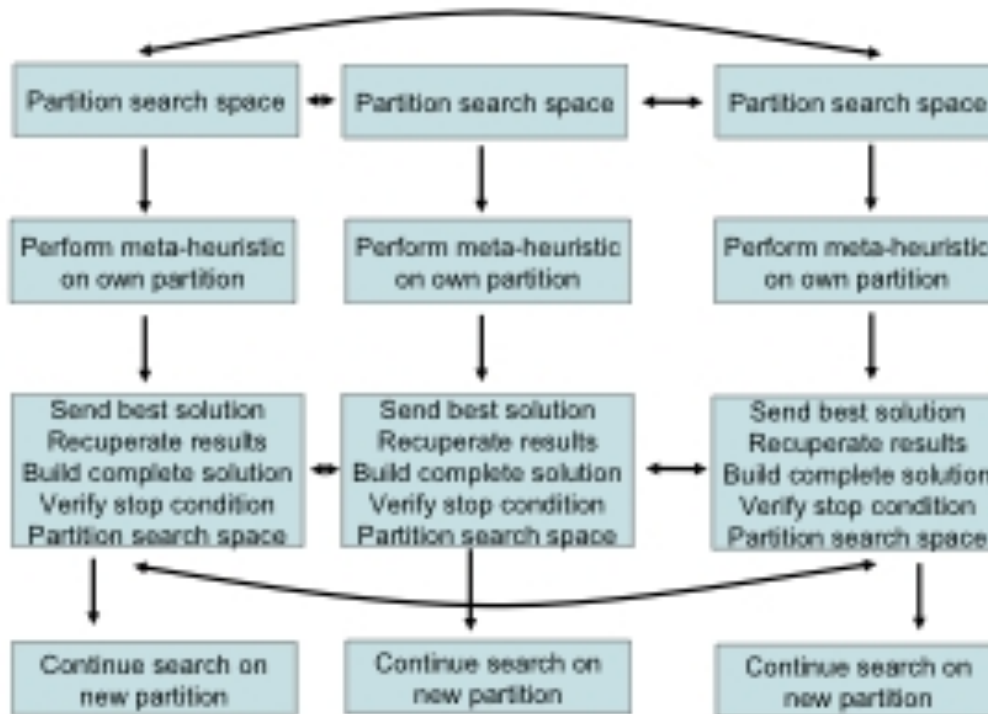
6. Descomposición del dominio

Descomposición de dominio: 1C



6. Descomposición del dominio

Descomposición de dominio: pC



7. Múltiples búsquedas independientes

- Esquema simple: varias ejecuciones paralelas e independientes de las metaheurísticas: pC/RS. Lo sorprendente es que es habitualmente da buenos resultados
- La principal desventaja es que no hay intento de intercambio de información entre las ejecuciones independientes, con lo que no se pueden aprovechar del «conocimiento» que van descubriendo sobre el problema
- Se ha aplicado a búsqueda tabú, GRASP, ES, y VNS.

-
- También se ha aplicado a Scatter Search
 - Sin embargo, no es muy popular para AG porque al distribuir la población inicial de n individuos entre p procesadores surgen poblaciones de tamaño n/p , que no son tan efectivas como un algoritmo secuencial con una población de tamaño n
 - Las pequeñas poblaciones aceleran el cálculo pero tienen efectos adversos en la diversidad

8. Múltiples búsquedas cooperativas

- Suponen un paso más en la política de intercambio de información, pues se realiza durante el proceso de búsqueda y no sólo al final. El resultado se traduce en soluciones de mejor calidad que las obtenidas con múltiples ejecuciones paralelas independientes
- El intercambio de información cooperativa especifica cómo interaccionan las metaheurísticas independientes para que el comportamiento emergente en la búsqueda global sea mejor

8. Múltiples búsquedas cooperativas

Aspectos importantes

- ¿Qué información intercambiar?
- ¿Entre qué procesos se intercambia la información?
- ¿Cuándo?
- ¿Cómo se intercambia (directo o diferido)?
- ¿Cómo se usa la información importada?

8. Múltiples búsquedas cooperativas

Ejemplo

- Una estrategia de cooperación podría establecer que un conjunto de metaheurísticas independientes se reinicializasen periódicamente desde la mejor solución actual.
- *Ejercicio*: Identificar las respuestas para todas las preguntas anteriores

8. Múltiples búsquedas cooperativas

¿Qué información intercambiar?

- La opción más simple es enviar la mejor solución encontrada hasta el momento, pero a lo largo del proceso de búsqueda se tiene mucha más información (p.ej. las memorias en la búsqueda tabú)
- Intercambiar sólo la mejor solución puede ser perjudicial por llevar a una pérdida de diversidad
- La información contextual es importante: información recogida durante la exploración

8. Múltiples búsquedas cooperativas

¿Entre qué procesos se intercambia?

- Intercambio directo entre procesos y definido por la topología de comunicación (estrella, anillo, grid, interconexión total). Comunicación síncrona o asíncrona con buffers.
- Uso de repositorios «centrales» o distribuidos de información (pizarras, *pools*, *data warehouse*). Los procesos envían y cogen información de estos repositorios en lugar de interactuar con otros procesos. Comunicación asíncrona

8. Múltiples búsquedas cooperativas

¿Cuándo y cómo?

- Comunicación **síncrona** (con esperas) y asíncrona. En cualquier caso, la práctica dice que no debe ser muy frecuente para que los retrasos por comunicaciones no penalicen la ganancia en tiempo y para que no se produzcan convergencias prematuras
- Cooperación **síncrona**: conseguir información completa del proceso de búsqueda global.
- Cooperación **asíncrona**: totalmente distribuida, más flexible y permite el desarrollo de un comportamiento emergente más efectivo

8. Múltiples búsquedas cooperativas

Múltiples búsquedas cooperativas basadas en Poblaciones

- Una caso especialmente importante por la repercusión habida en artículos de investigación y aplicaciones son las metaheurísticas basadas en poblaciones. Se les puede aplicar cualquiera de los enfoques de paralelización visto en las categorías previas. Pero realmente, son más efectivas cuando se trata de búsquedas cooperativas

Múltiples búsquedas cooperativas basadas en poblaciones

1. Tipo de descentralización
2. Modelos Distribuidos: Modelo de Isla
3. Modelos Celulares o Masivamente Paralelos
4. Relación entre Modelos Distribuidos y Celulares

Múltiples búsquedas cooperativas basadas en poblaciones

1. Tipos de descentralización

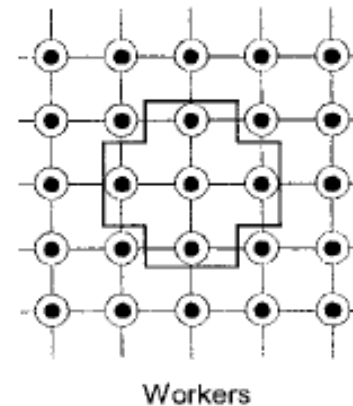
■ **Distribuidos**

- Se definen subpoblaciones
- Comunicación mediante intercambio de individuos



■ **Celulares**

- Sólo hay una población
- Comunicación mediante vecindad de individuos



Múltiples búsquedas cooperativas basadas en poblaciones

2. Modelos Distribuidos: Modelo de Isla

Fundamento

- En entornos aislados, tales como las islas, se encuentran especies animales que se adaptan más eficazmente a las peculiaridades de su entorno que las correspondientes a superficies de mayor amplitud, esto ha dado lugar a los llamados nichos

Hipótesis

- La competición entre varias subpoblaciones podría proporcionar una búsqueda más efectiva que la evolución de una gran población en la que todos los miembros coexistieran

Propuesta

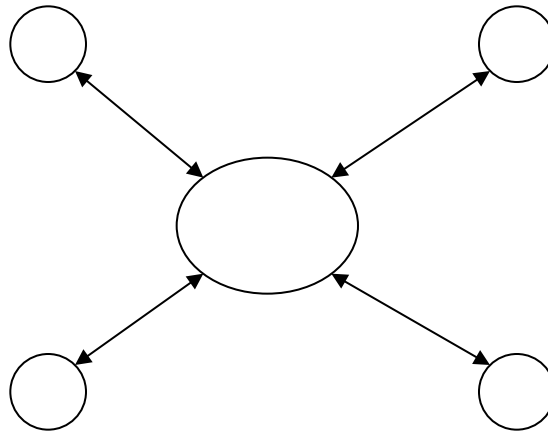
- **Modelo Isla:** Tener varias poblaciones aisladas que evolucionan en paralelo y periódicamente intercambian por migración sus mejores individuos con las subpoblaciones vecinas

Múltiples búsquedas cooperativas basadas en poblaciones

2. Modelos Distribuidos: Modelo de Isla

Estructuras de Intercomunicación

- **Estrella:** la subpoblación con mayor promedio objetivo se selecciona como maestra y las demás como subordinadas. Todas las subpoblaciones subordinadas envían sus mejores individuos a la maestra, y a su vez, ésta envía también sus mejores individuos a cada una de las subordinadas

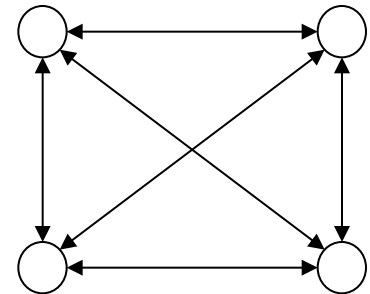
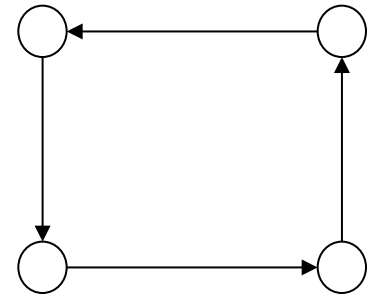


Múltiples búsquedas cooperativas basadas en poblaciones

2. Modelos Distribuidos: Modelo de Isla

Estructuras de Intercomunicación

- **Anillo:** Cada subpoblación envía sus mejores individuos a la subpoblación vecina más próxima en un único sentido de flujo
- **Red:** Todas las subpoblaciones envían sus mejores individuos a todas las demás



Múltiples búsquedas cooperativas basadas en poblaciones

3. Modelos Celulares o Masivamente Paralelos

Fundamento

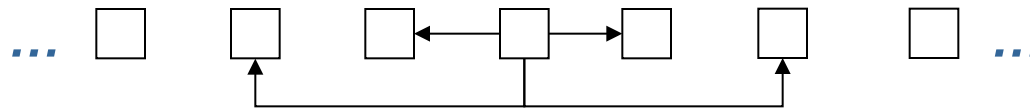
- Se trabaja con una única población y se pretende paralelizar las operaciones que realiza una BP clásica
- Cada individuo es colocado en una celda de un plano cuadrículado. La selección y el cruce se aplican entre individuos vecinos sobre la cuadrícula de acuerdo a una estructura de vecinos preestablecida
- **Función de evaluación**: Cada procesador elemental debe tener acceso sólo a aquellos individuos para los que calculará su función de evaluación
- **Cruce**: Cada procesador elemental que cree un nuevo individuo debe tener acceso a todos los otros individuos puesto que cada uno de ellos se puede seleccionar como padre
- **Mutación**: Cada procesador elemental necesita sólo los individuos con los que trate

Múltiples búsquedas cooperativas basadas en Poblaciones

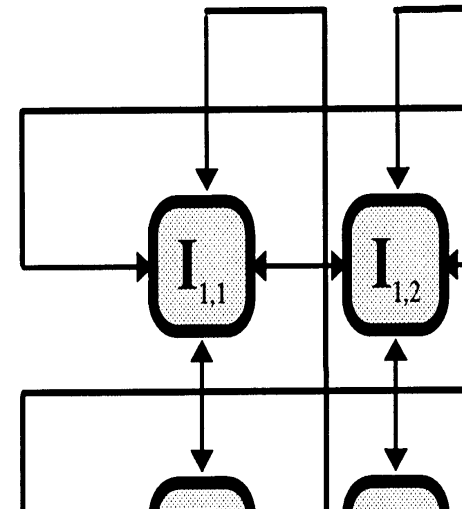
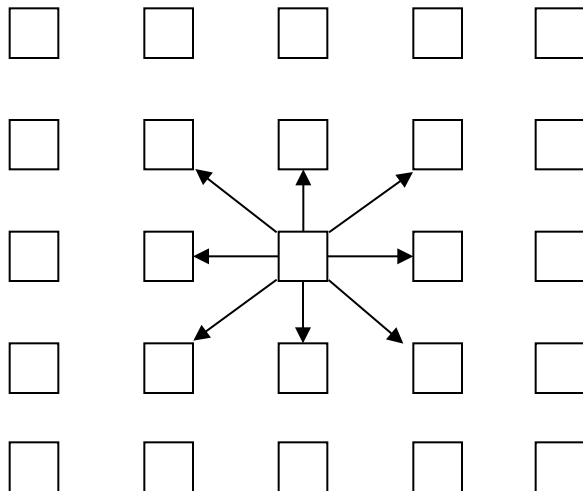
3. Modelos Celulares o Masivamente Paralelos

Estructuras de Intercomunicación

- Lista Circular



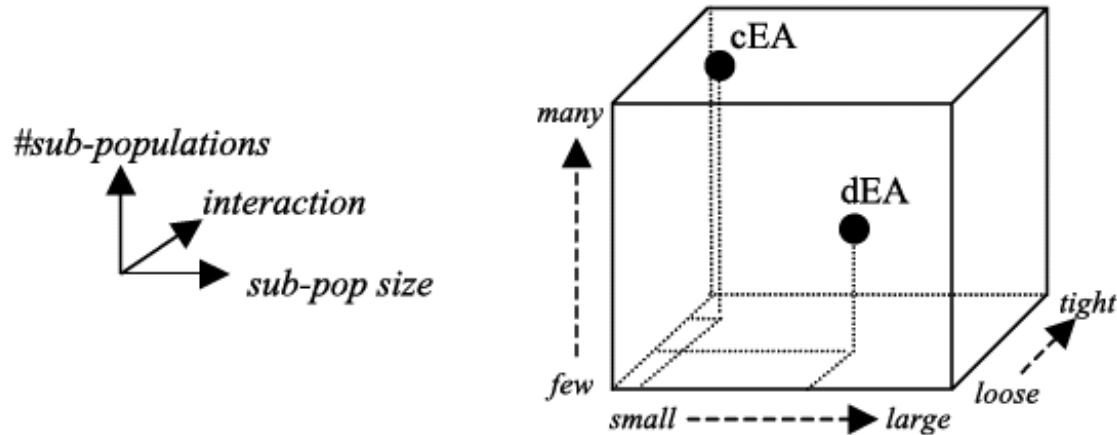
- Matriz bidimensional



Múltiples búsquedas cooperativas basadas en Poblaciones

3. Relación entre Modelos Distribuidos y Celulares

- dEA = Modelos Distribuidos
- cEA = Modelos Celulares

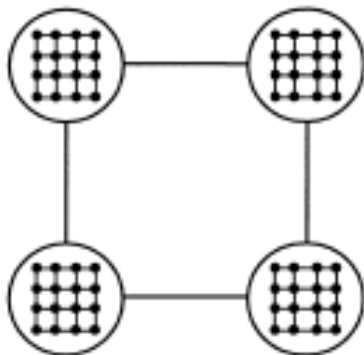


Múltiples búsquedas cooperativas basadas en poblaciones

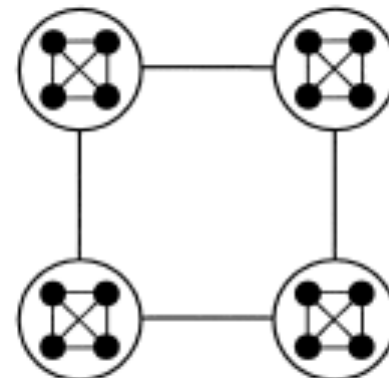
4. Otras Cuestiones

- Plataformas homogéneas o heterogéneas
- Comunicación síncrona o asíncrona
- Hibridaciones según sean subpoblaciones de grano (número de individuos por procesador) grueso o fino

Grano grueso y
grano fino



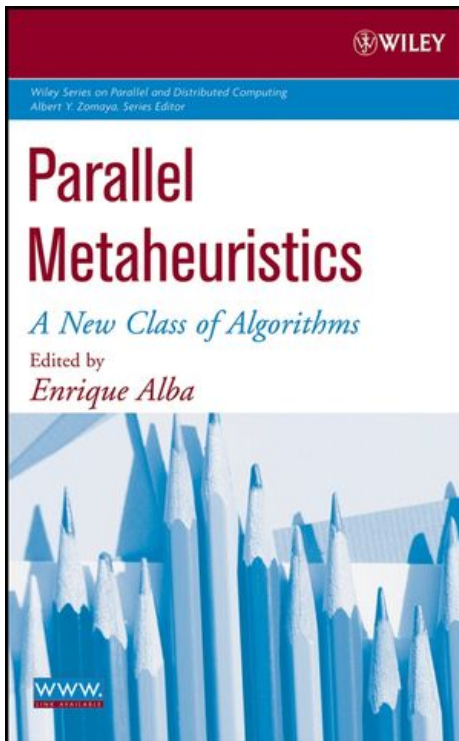
Grano grueso y
grano grueso



Conclusiones

- Se identifican cuatro clases fundamentales de metaheurísticas paralelas:
 - Descomposición funcional de bajo nivel de tareas de cálculo intensivas sin modificaciones al algoritmo original
 - Descomposición directa del espacio de búsqueda
 - Múltiples búsquedas independientes
 - Múltiples búsquedas cooperativas
- Cada clase tiene su ámbito de aplicación
- Las técnicas de cooperación asíncrona proporcionan un marco potente, flexible y adaptable que proporciona consistentemente buenos resultados en términos de eficiencia y calidad de las soluciones

Bibliografía



E. Alba,
Parallel Metaheuristics: A New Class of Algorithms,
Wiley, ISBN 0-471-67806-6, 2005

- * **Part One: Introduction to Metaheuristics and Parallelism**, including an Introduction to Metaheuristic Techniques, Measuring the Performance of Parallel Metaheuristics, New Technologies in Parallelism, and a head-to-head discussion on Metaheuristics and Parallelism
- * **Part Two: Parallel Metaheuristic Models**, including Parallel Genetic Algorithms, Parallel Genetic Programming, Parallel Evolution Strategies, Parallel Ant Colony Algorithms, Parallel Estimation of Distribution Algorithms, Parallel Scatter Search, Parallel Variable Neighborhood Search, Parallel Simulated Annealing, Parallel Tabu Search, Parallel GRASP, Parallel Hybrid Metaheuristics, Parallel Multi-Objective Optimization, and Parallel Heterogeneous Metaheuristics
- * **Part Three: Theory and Applications**, including Theory of Parallel Genetic Algorithms, Parallel Metaheuristics Applications, Parallel Metaheuristics in Telecommunications, and a final chapter on Bioinformatics and Parallel Metaheuristics

ALGORÍTMICA

2012 - 2013

- **Parte I. Introducción a las Metaheurísticas**
 - Tema 1. Metaheurísticas: Introducción y Clasificación
- **Parte II. Métodos Basados en Trayectorias y Entornos**
 - Tema 2. Algoritmos de Búsqueda Local Básicos
 - Tema 3. Algoritmos de Enfriamiento Simulado
 - Tema 4. Algoritmos de Búsqueda Tabú
 - Tema 5. Métodos Basados en Trayectorias Múltiples I: Métodos Multiarranque Básicos y GRASP
 - Tema 6. Métodos Basados en Trayectorias Múltiples II: ILS y VNS
- **Parte III. Métodos Basados en Poblaciones**
 - Tema 7. Algoritmos Genéticos
- **Parte IV. Intensificación y Diversificación**
 - Tema 8. Estudio del Equilibrio entre Intensificación y Diversificación
- **Parte V. Metaheurísticas Híbridas: Poblaciones y Trayectorias**
 - Tema 9. Algoritmos Meméticos
 - Tema 10. Scatter Search
- **Parte VI. Paralelización de Metaheurísticas**
 - Tema 11. Metaheurísticas en Sistemas Descentralizados
- **Parte VII. Conclusiones**
 - **Tema 12. Algunas Consideraciones sobre la Adaptación de Metaheurísticas a la Resolución de Problemas**