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. Modelos Híbridos II: *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 5. Métodos Basados en Trayectorias Múltiples I: Métodos Multiarranque Básicos y GRASP

1. Introducción a la Búsqueda Multiarranque
2. Algoritmos Multiarranque Básicos
3. Modelos Multiarranque
4. Algoritmo GRASP
5. Aplicación de GRASP

- *F. Glover, G.A. Kochenberber. Handbook of Metaheuristics. Kluwer Acad., 2003. Cap. 12. Multi-start Methods. (Rafael Martí), 355-368.* Chapter 8: Greedy Randomized Adaptive Search Procedure. *(* M.G.C. Resende, C.S. Ribeiro), 331-240.
- *R. Martí, J. Marcos Moreno. Métodos Multiarranque. Inteligencia Artificial 19 (2003) 49-60*
- *T.A. Feo, M.G.C. Resende. Greedy Randomized Adaptive Search Procedures. Journal of Global Optimization 6 (1995) 109-133*
- *M.G.C. Resende, J.L. González, GRASP: Procedimientos de Búsqueda Miopes, aleatorizados y adaptativos. Inteligencia Artificial 19 (2003) 61-76*

1. INTRODUCCIÓN A LA BÚSQUEDA MULTIARRANQUE

Problemas de la Búsqueda Local (Tema 2)

Suele caer en óptimos locales, que a veces están bastante alejados del óptimo global del problema

SOLUCIONES: 3 opciones para salir de los óptimos locales

- **Permitir movimientos de empeoramiento de la solución actual (Ejemplo: Enfriamiento Simulado, Búsqueda Tabú, …)**
- **Modificar la estructura de entornos (Ejemplo: Búsqueda Tabú, Búsqueda Descendente Basada en Entornos Variables: VND, Búsqueda en Entornos Variables: VNS, …)**
- **Volver a comenzar la búsqueda desde otra solución inicial (Ejemplo: Búsquedas Multiarranque, GRASP, ILS,VNS, …)**

1. INTRODUCCIÓN A LA BÚSQUEDA MULTIARRANQUE

- **Una Búsqueda con Arranque Múltiple es un algoritmo de búsqueda global que itera las dos etapas siguientes:**
  - **Generación de una solución inicial: Se genera una solución S de la región factible**
  - **Búsqueda Local: Se aplica una BL desde S para obtener una solución optimizada S'**
- **Estos pasos se repiten hasta que se satisfaga algún criterio de parada**
- **Se devuelve como salida del algoritmo la solución S' que mejor valor de la función objetivo presente**
- **La Búsqueda Multiarranque Básica se caracteriza porque las soluciones iniciales se generan de forma aleatoria**

1. INTRODUCCIÓN A LA BÚSQUEDA MULTIARRANQUE

Procedimiento Búsqueda con Arranque Múltiple

COMIENZO

S_{act} ← Genera Solución () (ETAPA 1)
***Mejor_Solución* ← S_{act}**
REPETIR
S' ← Búsqueda Local (S_{act}) (ETAPA 2)
Sl S' es mejor que *Mejor_Solución* ENTONCES
***Mejor_Solución* ← S'**
S_{act} ← Genera Solución () (ETAPA 1)
HASTA (criterio de parada)
Devolver *Mejor_Solución*

FIN

1. INTRODUCCIÓN A LA BÚSQUEDA MULTIARRANQUE

- **En algunas aplicaciones, la Etapa 1 se limita a la simple generación aleatoria de las soluciones, mientras que en otros modelos se emplean sofisticados métodos de construcción que consideran las características del problema de optimización para obtener soluciones iniciales de calidad**

- **En cuanto a la Etapa 2, se puede emplear una búsqueda local básica, o procedimientos de búsqueda basados en trayectorias más sofisticados**

- **En cuanto a la condición de parada, se han propuesto desde criterios simples, como el de parar después de un número dado de iteraciones, hasta criterios que analizan la evolución de la búsqueda**

2. ALGORITMOS MULTIARRANQUE BÁSICOS
2.1. Generación de Soluciones

Búsqueda Multiarranque Básica

- **El algoritmo multiarranque más básico que podemos considerar es aquel en el que las soluciones iniciales se generan al azar en la región factible del problema, y la etapa de búsqueda se realiza mediante algún procedimiento de búsqueda local BL**
- **Este método converge al óptimo global del problema con probabilidad 1 cuando el número de puntos generados tiende a infinito**
- **El procedimiento es muy ineficiente puesto que se pueden generar muchos puntos cercanos entre sí, de modo que al aplicarles el procedimiento de búsqueda BL se obtenga repetidamente el mismo óptimo local**

2. ALGORITMOS MULTIARRANQUE BÁSICOS
2.1. Generación de Soluciones

Método Multi-Level Single Linkage (MLSL) **(Rinnooy Kan y Timmer, 1987)**

- **Aporta una solución al problema previamente comentado**
- **Se genera una muestra aleatoria de *n* puntos en el espacio de soluciones. Se evalúan los puntos y se ordenan de mejor a peor valor de función objetivo, seleccionando una proporción *q·n* del total *(0 ≤ q ≤ 1).* El procedimiento de búsqueda BL se aplica, siguiendo el orden, a los puntos seleccionados. Se descartan los puntos que:**
  1. **Estén muy cerca de otro punto al que previamente se le aplicó el método BL**
  2. **Estén muy cerca de la frontera del espacio de soluciones**
  3. **Estén muy cerca de un óptimo local obtenido anteriormente**
- **Una vez aplicado el método BL a los puntos seleccionados que cumplen estas condiciones se generan nuevamente *n* puntos o soluciones al azar y se vuelve a aplicar el mismo procedimiento al conjunto resultante de unir éstos con los generados en iteraciones anteriores. La distancia crítica para considerar que un punto es muy cercano a otro se actualiza en cada iteración disminuyendo su valor**

2. ALGORITMOS MULTIARRANQUE BÁSICOS
2.1. Generación de Soluciones

Método Random Linkage (RL) (Locatelli, Schoen, 1999)

- **El inconveniente del método MLSL es la necesidad de almacenar las soluciones generadas y obtenidas por el método BL, porque al disminuir la distancia crítica según aumentan las iteraciones, puntos previamente generados podrían ser aceptados en iteraciones posteriores.**
- ***Alternativa:*** **El método RL mantiene las propiedades de convergencia de MLSL y no requiere la revisión reiterada de las soluciones.**
- **Para RL los puntos se generan al azar uno a uno. Cada vez que se genera un punto o solución, se aplica el optimizador BL según una función de probabilidad no-decreciente *p(d)* donde *d* es la distancia al punto más cercano generado anteriormente.**
- **Otros estudios sobre la generación de soluciones se encuentran en:**
  R. Martí, J. Marcos Moreno. Métodos Multiarranque. Inteligencia Artificial 19 (2003) 49-60

2. ALGORITMOS MULTIARRANQUE BÁSICOS
2.2. Reglas de Parada

- **Se han propuesto diferentes criterios de parada en la literatura especializada. Los principales analizan 3 variables aleatorias:**
  - **valores objetivo de los mínimos locales**
  - **número de mínimos locales distintos de la función objetivo (cuando se conozcan, esto no suele ocurrir en la práctica)**
  - **número de iteraciones necesarias para alcanzar el mínimo global**

- **En muchas de las publicaciones que se encuentran en la literatura especializada, se fija un número de iteraciones de la búsqueda local**

3. MODELOS MULTIARRANQUE

- **Existen múltiples propuestas de metaheurísticas que se pueden considerar como técnicas multiarranque:**
  - **Métodos constructivos de la solución inicial**
    - **Construcción *greedy*: Algoritmos GRASP**
    - **Algoritmos Basados en Colonias de Hormigas: ACO**
  - **Métodos iterativos mediante modificación de la solución encontrada**
    - **ILS: Búsqueda Local Reiterativa**
    - **VNS: Búsqueda de Entorno Variable**
  - **Hibridaciones entre técnicas poblacionales de exploración/ combinación de soluciones y métodos de búsqueda local**
    - **Algoritmos Meméticos / Algoritmos Genéticos con BL**
    - ***Scatter Search* (Búsqueda Dispersa)**

4. ALGORITMO GRASP
4.1. Introducción

- **Un GRASP es un método multiarranque, en el que cada iteración consiste en la construcción de una solución *greedy* aleatorizada y la aplicación de una búsqueda local que toma dicha solución como punto inicial de la búsqueda**
- **Este procedimiento se repite varias veces y la mejor solución encontrada sobre todas las iteraciones GRASP se devuelve como salida del algoritmo**

PROCEDIMIENTO ITERATIVO:

GREEDY-ALEATORIZADO-ADAPTATIVO

+

BUSQUEDA LOCAL

- ***T.A. Feo, M.G.C. Resende. Greedy Randomized Adaptive Search Procedures. Journal of Global Optimization 6 (1995) 109-133***

4. ALGORITMO GRASP
4.1. Introducción

Procedimiento GRASP

Procedimiento GRASP

Repetir Mientras (no se satisfaga el criterio de parada)

S ← Construcción Solución Greedy Aleatorizada ()

S' ← Búsqueda Local (S)

Actualizar (S', *Mejor_Solución*)

Devolver (*Mejor_Solución*)

FIN-GRASP

4. ALGORITMO GRASP
4.2. Descripción

Construcción de la Solución Inicial en GRASP

- **Cuando se utiliza la función de selección para construir una solución *greedy*, se crea una lista de restringida de candidatos con un número determinado de mejores candidatos**
- **Se realiza una selección aleatoria de un candidato de la lista**
- **Se adapta la función de selección para recalcular la nueva lista de candidatos para el siguiente paso del proceso constructivo**

4. ALGORITMO GRASP
4.2. Descripción

Construcción de la Solución Inicial en GRASP

```
Procedimiento Construcción-Greedy-Aleatorizada ()
    S = {}
    Repetir Mientras (no se haya construido la solución)
        Crear Lista Restringida de Candidatos (LRC)
        s ← selección-aleatoria-elemento (LRC)
        S ← S ∪ {s}
        Adaptar la función de selección %Actualizar el conjunto de candidatos
    Devolver S
Fin-Procedimiento
```

4. ALGORITMO GRASP
4.2. Descripción

Construcción de la Solución Inicial en GRASP

- **La variante más habitual consiste en incluir en la LRC los *l* mejores candidatos de acuerdo a la función de selección. El parámetro *l* controla la diversidad de generación de soluciones. Puede ser fijo o variable**
- **Existen variantes que incluyen en la LRC todos los candidatos con un valor de selección por encima de un umbral de calidad μ = cmejor $\pm\alpha\cdot$ (cmejor- cpeor) (cmejor=coste mejor candidato; cpeor=coste peor candidato). Esto provoca que la LRC sea adaptativa y tenga un tamaño variable en cada iteración de la etapa de generación**
- **Otras variantes incluyen un sesgo en la LRC con una probabilidad mayor de selección de los mejores candidatos**
- **También se combina la construcción al azar con la construcción *greedy*, alternando ambos procedimientos aleatoriamente y en secuencias de candidatos**
- **Otro tipo de construcción es mediante perturbaciones en el costo de la función de selección**

4. ALGORITMO GRASP
4.2. Descripción

Búsqueda Local en GRASP

- **La búsqueda local desempeña un papel importante en GRASP ya que sirve para buscar soluciones localmente óptimas en regiones prometedoras del espacio de soluciones**

  S: solución generada mediante el procedimiento constructivo

  E(S): conjunto de vecinos BL: Algoritmo del mejor vecino

- **En el contexto de GRASP, se han utilizado esquemas de búsqueda local clásicos, así como otros más sofisticados (enfriamiento simulado, búsqueda tabú, etc.)**

- **Aunque los algoritmos *greedy* pueden producir soluciones iniciales razonables para búsquedas locales, su principal desventaja es su falta de diversidad, de ahí las propuestas de LRC para generar más diversidad en la primera etapa del GRASP**

4. ALGORITMO GRASP
4.2. Descripción

Ejemplo de Diversidad en las Soluciones Iniciales

Problema: Satisfacción de cláusulas (**1391 variables, 3126 cláusulas, 7025 literales**)

Distribución de las soluciones en 100000 ejecuciones de la primera etapa del GRASP

size RCL	solution values										
	3116	3117	3118	3119	3120	3121	3122	3123	3124	3125	3126
1									100000		
2							151	6053	93796		
4						75	1676	17744	80503	2	
8				1	50	750	6566	31257	61336	35	5
16				16	282	2485	13274	38329	45547	42	25
32		1	3	72	635	4196	16455	37937	40479	164	58
64		4	18	177	1213	5933	19553	37666	34832	441	163
128		4	36	269	1716	7324	21140	37186	34832	679	281
256	1	5	35	304	1980	7867	21792	36725	29027	1575	689

Greedy

Fig. 4. Sample distributions of GRASP iteration solutions.

size RCL	1	2	4	8	16	32	64	128	256
mean (3120 +)	4.00	3.94	3.79	3.53	3.27	3.14	3.00	2.91	2.89

Fig. 5. Means of sample distributions of GRASP iteration solutions.

4. ALGORITMO GRASP
4.3. Extensiones

Reencadenamiento de Trayectorias

- **Esta técnica (*path relinking*) se propuso inicialmente para explorar las trayectorias entre soluciones elite (conjunto de las mejores soluciones obtenidas en varias ejecuciones de la búsqueda tabú u otro algoritmo)**

- **Usando una o más soluciones elite, se exploran las trayectorias en el espacio de soluciones que conducen a otras soluciones elite para buscar mejores soluciones**

- **Para generar trayectorias, los movimientos se seleccionan para introducir atributos en la solución actual que estén presentes en la solución elite guía**

GRASP Híbrido

- **Se introduce una mutación (perturbación fuerte) a la solución encontrada tras la etapa de la Búsqueda Local y se repite la optimización**

Procedimiento GRASP Híbrido

Repetir Mientras (no se satisfaga el criterio de parada)

S ← Construcción Solución Greedy Aleatorizada ()

Repetir Mientras (se decida continuar)

S' ← Búsqueda Local (S)

Actualizar (S', Mejor_Solución)

S ← Mutar Solución (S')

Devolver (Mejor_Solución)

FIN-GRASP-Híbrido

4. ALGORITMO GRASP
4.4. Ejemplo: Viajante de Comercio

- Construcción de soluciones *greedy* aleatorizadas en el Viajante de Comercio:
  - Se basa en la misma regla heurística empleada en el algoritmo *greedy*: seleccionar la siguiente ciudad entre las más cercanas
  - Cada vez que se debe seleccionar una nueva ciudad en el recorrido a partir de la ciudad actual, se construye una lista LRC con las ciudades más cercanas (menor coste) a ella
  - Se selecciona aleatoriamente una ciudad de la lista y se vuelve a construir una nueva lista para la nueva ciudad

4. ALGORITMO GRASP
4.4. Ejemplo: Viajante de Comercio

- Ejemplo de ejecución de la primera etapa:
  - Instancia con 6 ciudades. Se comienza en la ciudad 1. LRC de tamaño 3.

	Distancias								
	1	2	3	4	5	6	LRC	Uniforme	Solución
1	-	23	54	17	132	41	{4, 2, 6}	6	(1 6 - - - -)
6	-	48	31	142	39	-	{3, 5, 2}	3	(1 6 3 - - -)
3	-	12	-	45	28	-	{2, 5, 4}	5	(1 6 3 5 - -)
5	-	59	-	22	-	-	{4, 2}	2	(1 6 3 5 2 -)
2	-	-	-	100	-	-	{4}	4	(1 6 3 5 2 4)

4. ALGORITMO GRASP
4.4. Ejemplo: Viajante de Comercio

- Mutación en GRASP Híbrido:
  - Debe realizarse una perturbación mayor que la que provoca el operador de vecino empleado en la búsqueda local (intercambio, 2-opt)
  - Para ello, cada vez que se realiza una mutación se aplica la modificación por sublista aleatoria de tamaño fijo ($s=n/4$) consistente en seleccionar una cadena consecutiva de elementos de tamaño s y alterar aleatoriamente sus asignaciones

(1 2 4 3 8 5 7 6)
(1 2 8 3 5 4 7 6)

5. APLICACIÓN DE GRASP
MDP

- Construcción de soluciones *greedy* aleatorizadas:
  - Se basa en la misma regla heurística empleada en el algoritmo *greedy*.
  - La lista de candidatos vendrá dada por los *k* vecinos más cercanos a la misma.

5. APLICACIÓN DE GRASP
MDP

- Mutación en GRASP Híbrido:
  - Se emplea el operador de vecino ya conocido pero generando más diversidad en el vecindario (entorno de mayor tamaño)
  - Cada vez que se realiza una mutación, se varía el estado de $0.1 \cdot n$ valores del vector.

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. Modelos Híbridos II: *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