

Metaheurísticas

Seminario 2.b. Problemas de optimización con técnicas basadas en trayectorias múltiples

3. Esquema General de los Algoritmos GRASP e ILS
4. Problemas de Optimización con Trayectorias Múltiples
 - Asignación Cuadrática
 - Selección de Características

Procedimiento GRASP

Procedimiento GRASP

Repetir Mientras (no se satisfaga el criterio de parada)

$S \leftarrow$ Construcción Solución Greedy Aleatorizada ()

$S' \leftarrow$ Búsqueda Local (S)

Actualizar (S' , *Mejor_Solución*)

Devolver (*Mejor_Solución*)

FIN-GRASP

Procedimiento GRASP

Construcción Solución Greedy Aleatorizada ()

- ✓ En cada iteración de su proceso constructivo de la solución, un algoritmo greedy básico:
 - ✓ construye una lista con los candidatos factibles (las posibles componentes a escoger de acuerdo con la solución construida hasta el momento y las restricciones del problema): **Lista de Candidatos (LC)**,
 - ✓ los evalúa de acuerdo a una función de selección (que mide su calidad/preferencia para ser escogidos), y
 - ✓ selecciona siempre el candidato de **mejor calidad** de la LC
- ✓ Los algoritmos GRASP añaden **aleatoriedad** al procedimiento anterior. La única diferencia es que en cada iteración:
 - ✓ No se consideran todos los candidatos posibles sino sólo los de mejor calidad: **Lista Restringida de Candidatos (LRC)**. El tamaño de esa lista puede ser fijo o variable en función de un umbral de calidad
 - ✓ El elemento seleccionado se escoge **aleatoriamente** de la RCL para inducir diversidad, independientemente de la calidad de los candidatos

Procedimiento ILS

Comienzo-ILS

$S_0 \leftarrow \text{Generar-Solución-Inicial}$

$S \leftarrow \text{Búsqueda Local } (S_0)$

Repetir

$S' \leftarrow \text{Modificar } (S, \text{historia})$ %Mutación

$S'' \leftarrow \text{Búsqueda Local } (S')$

$S \leftarrow \text{Criterio-Aceptación } (S, S'', \text{historia})$

Actualizar (S , *Mejor_Solución*)

Hasta (Condiciones de terminación)

Devolver *Mejor_Solución*

Fin-ILS

Problema de Asignación Cuadrática (QAP)

■ Problema de la asignación cuadrática, QAP:

Dadas n unidades y n localizaciones posibles, el problema consiste en determinar la asignación óptima de las unidades en las localizaciones conociendo el flujo existente entre las primeras y la distancia entre las segundas

$$QAP = \min_{S \in \Pi_N} \left(\sum_{i=1}^n \sum_{j=1}^n f_{ij} \cdot d_{S(i)S(j)} \right)$$

donde:

- ✓ S es una solución candidata (una posible asignación de unidades a localizaciones) representada por una permutación de n elementos
- ✓ $f_{ij} \cdot d_{S(i)S(j)}$ es el coste de la asignación de la unidad u_i a la localización $S(i)$ y u_j a $S(j)$, calculado como el coste del recorrido del flujo que circula entre esas dos unidades i y j cuando están situadas en las localizaciones $S(i)$ y $S(j)$

Procedimiento GRASP para el QAP

Variante de: Y. Li, P.M. Pardalos, M.G.C. Resende. A greedy randomized adaptive search procedure for the quadratic assignment problem. In: P.M. Pardalos and H. Wolkowicz (Eds.), Quadratic assignment and related problems, 1994, pp. 237-261

- Los candidatos son las asignaciones unidad-localización factibles. La función de selección mide su coste de interacción $f_{ij} \cdot d_{kl}$ (a menor coste, más preferencia). Se busca asociar unidades de gran flujo con localizaciones céntricas
- El procedimiento de construcción de la solución greedy aleatorizada tiene dos etapas:
 - Etapa 1: Se toma la primera decisión, en la que **se asignan conjuntamente dos unidades a dos localizaciones en un solo paso**. Se consideran los potenciales de flujo y distancia de unidades y localizaciones para escoger el par de asignaciones en una decisión *greedy* aleatorizada
 - Etapa 2: Se toman las $n-2$ decisiones restantes **asignando una unidad a una localización en cada paso**
Se escoge la asignación factible con el mínimo coste de interacción con respecto a las asignaciones realizadas hasta ahora de acuerdo a una decisión *greedy* aleatorizada

Procedimiento GRASP para el QAP

Etapa 1:

- Se dispone de dos LCs, una para las unidades LC_u y otra para las localizaciones LC_l . Ambas listas contienen inicialmente las n unidades y las n localizaciones, respectivamente
- La función de selección será el potencial de flujo/distancia de las unidades, calculado como:

$$\hat{f}_i = \sum_{j=1}^n (f_{ij} + f_{ji}) \quad ; \quad \hat{d}_k = \sum_{l=1}^n (d_{kl} + d_{lk})$$

- En el *greedy*, la decisión sería escoger el mejor candidato de cada LC, es decir, asignar la unidad de mayor potencial de flujo con la localización de menor distancia potencial

Procedimiento GRASP para el QAP

- En el GRASP, las dos LRCs, LRC_u y LRC_l , son de tamaño variable e incluyen todos los candidatos de la LC correspondiente cuyo coste iguala o supera un umbral de calidad:
 - LRC_u : Como los candidatos son mejores cuánto más alto sea el potencial de flujo, el umbral es $\mu = d_{\max} - \alpha \cdot (d_{\max} - d_{\min})$, donde d_{\max} es el mayor valor de flujo potencial de los candidatos de LC_u y d_{\min} el menor
 - LRC_l : Como los candidatos son mejores cuánto más bajo sea el potencial, el umbral es $\mu = d_{\min} + \alpha \cdot (d_{\max} - d_{\min})$, donde d_{\max} es la mayor distancia potencial de los candidatos de LC_l y d_{\min} la menor
- Se escogen aleatoriamente dos candidatos de cada LRC, se asocian el primero escogido de la LRC_u u_i con el primero de la LRC_l l_k ; y luego el segundo de la LRC_u u_j con el segundo de la LRC_l l_l
- Se guardan las dos asignaciones unidad-localización generadas de este modo en la solución parcial, $S = \{(u_i, l_k), (u_j, l_l)\}$

Procedimiento GRASP para el QAP

Etapa 2:

- En la segunda etapa, se escoge una sola asignación unidad-localización en cada paso
- La LC está formada por todas las asignaciones factibles (u_i, l_k) , es decir, todos los pares unidad-localización en los que **ninguna de ellas** esté incluida aún en la solución parcial $S = \{(u_{i1}, l_{k1}), \dots, (u_{iz}, l_{kz})\}$
- La segunda etapa comienza con $|S|=2$, las dos asignaciones realizadas en la primera etapa
- La función de selección *greedy* mide el coste de añadir la asignación candidata (u_i, l_k) a las asignaciones ya realizadas, almacenadas en la solución actual S , calculado como:

$$C_{ik} = \sum_{(j,l) \in S} f_{ij} \cdot d_{kl}$$

Procedimiento GRASP para el QAP

- Los candidatos con menor valor de la función son preferibles
- De nuevo, la LRC es de tamaño variable e incluye todas las asignaciones factibles de LC cuyo coste es menor o igual que el umbral de calidad $\mu = C_{\min} + \alpha \cdot (C_{\max} - C_{\min})$, donde C_{\min} es el menor coste de los candidatos de LC y C_{\max} el mayor
- Se escoge aleatoriamente una asignación unidad-localización candidata de la LRC y se añade a la solución parcial $S \leftarrow S \cup \{(u_i, l_k)\}$
- En cada nuevo paso del algoritmo hay que actualizar la LC, eliminando las asignaciones consideradas en el paso anterior, y construir la nueva LRC recalculando los costes para los candidatos factibles restantes y aplicando el umbral para filtrar los candidatos a emplear
- El proceso constructivo termina tras $n-2$ pasos de la segunda etapa, cuando ya se han tomado las n decisiones necesarias

ILS para el QAP

- **Representación de orden:** permutación $\pi = [\pi(1), \dots, \pi(n)]$ en el que las posiciones del vector $i=1, \dots, n$ representan las unidades y los valores $\pi(1), \dots, \pi(n)$ contenidos en ellas las localizaciones
- **Solución inicial:** aleatoria
- **Operador de mutación:** Usaremos el operador de vecino de **sublista aleatoria** de tamaño fijo t , consistente en seleccionar una cadena consecutiva de asignaciones y reasignarlas aleatoriamente
Para provocar un cambio brusco, obligaremos a que la sublista tenga un tamaño importante: $t = n/4$
- **Algoritmo de búsqueda local:** la BL-QAP de la Práctica 1
- **Criterio de aceptación:** se sigue el “criterio del mejor”, siempre se aplica la mutación sobre la mejor solución encontrada hasta ahora

Procedimiento GRASP para la Selección de Características (SC)

- Construcción de soluciones *greedy* aleatorizadas:
 - Se construye a partir del método *greedy SFS*, empleado como algoritmo de comparación en la Práctica 1
 - Los candidatos son las características no seleccionadas hasta el momento
 - En cada paso se añade la más prometedora, es decir, la que produce una mayor ganancia en porcentaje de acierto 3-NN con respecto al conjunto de características ya seleccionado
 - Para diseñar una variante *greedy* aleatorizada basta con generar una LRC con las mejores opciones y escoger aleatoriamente uno de sus elementos en cada iteración

Procedimiento GRASP para la SC

- Construcción de soluciones *greedy* aleatorizadas:
 - La LRC tendrá tamaño variable
 - En cada iteración, estará formada por aquellos atributos f_i cuya ganancia en porcentaje de clasificación g_i supere el siguiente umbral de calidad:

$$\mu = \text{cmejor} - \alpha \cdot (\text{cmejor} - \text{cpeor})$$

donde:

- cmejor es la máxima ganancia de las características de la LC,
- cpeor es la mínima ganancia (puede ser negativa), y
- $\alpha \in [0,1]$ es el grado de tolerancia definido

Procedimiento GRASP para la SC

Algoritmo SFS:

$LC \leftarrow \{f_1, \dots, f_n\}. \quad S \leftarrow \emptyset. \quad fin \leftarrow falso$

Mientras ($LC \neq \emptyset$ y $!fin$)

Evaluar la ganancia g_i de cada característica f_i en LC

$LRC \leftarrow f_i \in LC / g_i \geq \mu$

$fp \leftarrow Característica_aleatoria(LRC)$

Si $S \cup \{fp\}$ *mejora la solución actual* S ($f(S \cup \{fp\}) \geq f(S)$)

$S \leftarrow S \cup \{fp\}$

en otro caso

$fin \leftarrow verdad$

$F \leftarrow F - \{fp\}$

Devolver S

ILS para SC

- **Representación binaria**: un vector binario $s=(s_1, \dots, s_n)$ en el que cada posición i representa una característica y su valor 0/1 indica si está o no seleccionada
- **Solución inicial**: aleatoria
- **Operador de mutación**: Cada vez que se realiza una mutación, se varía la pertenencia de $t=0.1 \cdot n$ características
- **Algoritmo de búsqueda local**: la BL-SC de la Práctica 1
- **Criterio de aceptación**: se sigue el “criterio del mejor”, es decir, siempre se aplica la mutación sobre la mejor solución encontrada hasta el momento