

Metaheurísticas

Seminario 4. Técnicas basadas en trayectorias para el Problema de la Máxima Diversidad (MDP) y el Aprendizaje de Pesos en Características (APC). Evolución Diferencial para el APC

1. Trayectorias Simples

- Esquema General del Algoritmo de Enfriamiento Simulado
- Un Algoritmo de Enfriamiento Simulado para el MDP y el APC

2. Trayectorias Múltiples

- Esquema General de los Algoritmos GRASP e ILS
- Un Algoritmo GRASP para el MDP
- Un Algoritmo ILS para el MDP y el APC

3. Evolución Diferencial

- Esquema General de un Algoritmo de Evolución Diferencial
- Algoritmo de Evolución Diferencial para el APC

Algoritmo de Enfriamiento Simulado

Procedimiento Simulated Annealing (Δf para minimizar)

Start

$T \leftarrow T_0$; $s \leftarrow \text{GENERATE}()$; Best Solution $\leftarrow s$;

Repeat

For $cont = 1$ to $L(T)$ **do** /* Inner loop

Start

$s' \leftarrow \text{NEIGHBORHOOD_OP}(s)$; /* A single move

$\Delta f = f(s') - f(s)$;

If $((\Delta f < 0) \text{ or } (U(0,1) \leq \exp(-\Delta f/k \cdot T)))$ then

$s \leftarrow s'$;

If $\text{COST}(s)$ **is better than** $\text{COST}(\text{Best Solution})$

then Best Solution $\leftarrow s$;

End

$T \leftarrow g(T)$; /* Cooling scheme. The classical one is geometric: $T \leftarrow \alpha \cdot T$

until $(T \leq T_f)$; /* Outer loop

Return(Best Solution);

End

Enfriamiento Simulado para el MDP

- **Representación:** Problema de selección: un conjunto $Sel = \{s_1, \dots, s_m\}$ que almacena los m elementos seleccionados de entre los n elementos del conjunto S . Permite verificar las restricciones
- **Operador de vecino de intercambio y su entorno:** El entorno de una solución Sel está formado por las soluciones accesibles desde ella a través de un movimiento de intercambio

Dada una solución (conjunto de elementos seleccionados) se escoge un elemento y se intercambia por otro que no estuviera seleccionado ($Int(Sel, i, j)$):

$$Sel = \{s_1, \dots, i, \dots, s_m\} \quad \Rightarrow \quad Sel' = \{s_1, \dots, j, \dots, s_m\}$$

$Int(Sel, i, j)$ verifica las restricciones

Enfriamiento Simulado para el MDP

- **Exploración del vecindario**: En cada iteración del bucle interno se genera una única solución vecina, **de forma aleatoria**, y se compara con la actual. **Se usa la factorización para el cálculo del coste**
- **Esquema de enfriamiento**: esquema de Cauchy modificado
- **Condición de enfriamiento $L(T)$** : cuando se genere un número máximo de soluciones vecinas, *máx_vecinos*, o cuando se acepte un número máximo de los vecinos generados, *máx_éxitos*
- **Condición de parada**: cuando se alcance un número máximo de iteraciones o cuando el número de éxitos en el enfriamiento actual sea 0

Enfriamiento Simulado para el APC

- **Representación**: vector de números reales con pesos de características y la posibilidad de eliminarlas, igual que en prácticas anteriores
- **Operador de generación de vecinos**: mutación normal, como en la BL
- **Exploración del vecindario**: En cada iteración del bucle interno se genera una única solución vecina, de forma aleatoria, y se compara con la actual
- **Esquema de enfriamiento**: esquema de Cauchy modificado
- **Condición de enfriamiento $L(T)$** : cuando se genere un número máximo de soluciones vecinas, *máx_vecinos*, o cuando se acepte un número máximo de los vecinos generados, *máx_éxitos*
- **Condición de parada**: cuando se alcance un número máximo de iteraciones o cuando el número de éxitos en el enfriamiento actual sea 0

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

Problema de la Máxima Diversidad (MDP)

■ Problema de la Máxima Diversidad, MDP:

*seleccionar un subconjunto Sel de m elementos ($|M|=m$) de un conjunto inicial S de n elementos ($n > m$) de forma que se **maximice** la diversidad entre los elementos escogidos*

$$\text{Maximizar } z_{MS}(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} x_i x_j$$

$$\text{Sujeto a } \sum_{i=1}^n x_i = m$$

$$x_i = \{0, 1\}, \quad i = 1, \dots, n.$$

donde x es el vector binario solución al problema

Procedimiento GRASP para el MDP

Duarte, Martí, Tabu search and GRASP for the maximum diversity problem,
European Journal of Operational Research 178 (2007) 71–84

- Nuestro algoritmo GRASP estará basado en el greedy usado como algoritmo de comparación hasta ahora:

1. $Sel = \emptyset$
2. Calcular la distancia acumulada de cada elemento al resto: $DistAc(s_i) = \sum_{s_j \in S} d(s_i, s_j)$, incluir el elemento s_{i*} que la maximice en Sel : $Sel = Sel \cup \{s_{i*}\}$ y eliminarlo de S : $S = S - \{s_{i*}\}$

while ($|Sel| < m$)

3. Calcular las distancias de los elementos no seleccionados, $s_i \in S$, al conjunto de elementos seleccionados Sel : $Dist(s_i, Sel) = \min_{s_j \in Sel} d(s_i, s_j)$
4. Incluir el elemento s_{i*} que la maximice en Sel : $Sel = Sel \cup \{s_{i*}\}$
5. Eliminar el elemento s_{i*} seleccionado de S : $S = S - \{s_{i*}\}$

end while

Procedimiento GRASP para el MDP

- La LC incluye todos los elementos no seleccionados: $S-Sel$. Los candidatos con mayor distancia acumulada a los elementos seleccionados hasta el momento (Sel) son preferibles
- La LRC es de tamaño variable e incluye todos los elementos no seleccionados cuya distancia acumulada a los actualmente seleccionados es mayor o igual que el umbral de calidad $\mu = d_{\min} + \alpha \cdot (d_{\max} - d_{\min})$, donde d_{\min} es la menor distancia coste de los candidatos de LC y d_{\max} la mayor
- Se escoge aleatoriamente un elemento candidato de la LRC y se añade a la solución parcial $Sel \leftarrow Sel \cup \{s_{i*}\}$
- En cada nuevo paso del algoritmo hay que actualizar la LC , eliminando los elementos seleccionados en el paso anterior, y construir la nueva LRC recalculando las distancias para los candidatos factibles restantes y aplicando el umbral para filtrar los candidatos a emplear
- El proceso constructivo termina tras $m-1$ pasos

Procedimiento GRASP para el MDP

1. $Sel = \emptyset$
2. Calcular la distancia acumulada de cada elemento al resto: $DistAc(s_i) = \sum_{s_j \in S} d(s_i, s_j)$, incluir el elemento s_{i*} que la maximice en Sel : $Sel = Sel \cup \{s_{i*}\}$ y eliminarlo de S : $S = S - \{s_{i*}\}$

while ($|Sel| < m$)

3. Calcular las distancias de los elementos no seleccionados, $s_i \in S$, al conjunto de elementos seleccionados Sel : $Dist(s_i, Sel) = \min_{s_j \in Sel} d(s_i, s_j)$
4. Calcular la mayor y la menor distancia: $d_{max} = \max_{s_i \in S} Dist(s_i, Sel)$; $d_{min} = \min_{s_i \in S} Dist(s_i, Sel)$
5. $RCL = \{s_i \in S / Dist(s_i, Sel) \geq d_{min} + \alpha \cdot (d_{max} - d_{min})\}$
6. Escoger aleatoriamente un elemento s_{i*} de RCL
7. Incluir s_{i*} en Sel : $Sel = Sel \cup \{s_{i*}\}$
8. Eliminar s_{i*} de S : $S = S - \{s_{i*}\}$

end while

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

ILS para el MDP

- **Representación de orden**: conjunto $Sel = \{s_1, \dots, s_m\}$ que almacena los m elementos seleccionados de entre los n elementos del conjunto S
- **Solución inicial**: aleatoria
- **Operador de mutación**: Cada vez que se muta, aplicamos el operador de intercambio $Int(Sel, i, j)$ sobre $t = 0.1 \cdot m$ elementos seleccionados distintos para provocar un cambio brusco
- **Algoritmo de búsqueda local**: la BL-MDP 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

ILS para el APC

- **Representación real**: Vector real W de tamaño n , donde $w_i \in [0, 1]$, con la posibilidad de eliminar características
- **Solución inicial**: aleatoria
- **Operador de mutación**: Cada vez que se muta, se aplica la mutación normal a $t=0.1 \cdot n$ características con un mayor valor de σ
- **Algoritmo de búsqueda local**: la BL-APC 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

Evolución Diferencial

Procedure DE/rand/1 – Rec. Binomial {

t = 0;

Initialize Pop(t); /* of |Pop(t)| Individuals */

Evaluate Pop(t);

While (Not Done)

{for i = 1 to |Pop(t)| do

{parent1, parent2, parent3} = Select_3_Parents(Pop(t));

for k = 1 to n do /* n genes per Individual */

if (random < CR) /* CR is crossover constant in [0,1] */

Offspring_{ik} = parent1_{ik} + F·(parent2_{ik} – parent3_{ik});

else

Offspring_{ik} = Individual_{ik} in Pop(t);

end /* for k */

Evaluate(Offspring_i);

end /* for i */

Pop(t+1) = {j | Offspring_j is_better_than Individual_j} ∪

{k | Individual_k is_better_than Offspring_k};

t = t + 1; }

Evolución Diferencial para APC

- **Población inicial:** generada aleatoriamente.

- **Modelos a implementar:**

$$\text{DE/rand/1:} \quad \mathbf{V}_{i,G} = \mathbf{X}_{r_1,G} + F \cdot (\mathbf{X}_{r_2,G} - \mathbf{X}_{r_3,G})$$

$$\text{DE/current-to-best/1:} \quad \mathbf{V}_{i,G} = \mathbf{X}_{i,G} + F \cdot (\mathbf{X}_{best,G} - \mathbf{X}_{i,G}) + F \cdot (\mathbf{X}_{r_1,G} - \mathbf{X}_{r_2,G})$$

- **Recombinación Binomial:**

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } \text{rand}_j[0,1] \leq Cr \text{ or } j=j_{\text{rand}} \\ x_{j,i,g} & \text{otherwise} \end{cases}$$

- **Reemplazamiento:** uno a uno

- **Función objetivo:** La agregación de las tasas de clasificación y de reducción empleada hasta el momento