

Metaheurísticas

Seminario 4. Problemas de optimización con técnicas basadas en adaptación social

1. Diseño de Componentes para Resolver un Problema con un Algoritmo de Optimización Basada en Colonias de Hormigas
2. Aspectos de Implementación: Listas de Candidatos
3. Problemas de Optimización con Trayectorias Múltiples
 - Asignación Cuadrática
 - Clustering

Diseño de Componentes para un Algoritmo de OCH

Para resolver un problema de optimización con un algoritmo de OCH hay que diseñar los siguientes componentes específicos:

- Definición del esquema de representación: los **componentes de la solución** y el **grafo de construcción** que recorren las hormigas
- Especificación de las **restricciones** del problema para tenerlas en cuenta en el proceso constructivo
- Definición de la **función objetivo**
- Diseño de los **rastros de feromona** y especificación de a qué componentes de la solución se asociarán
- Definición de la **información heurística** asociada a cada componente
- Especificación del **proceso constructivo** que seguirán la hormiga para generar soluciones

Ejemplo: OCH para el TSP

- **Esquema de representación:** Las componentes de la solución son los arcos del grafo del caso del problema a_{rs} . El grafo de construcción equivale a ese mismo grafo
- **Restricciones:** No se pueden formar ciclos, es decir, no se puede visitar un nodo más de una vez
- **Función objetivo:** Suma de las distancias del circuito
- **Rastros de feromona:** Se asocia un rastro de feromona a cada arco del grafo a_{rs} , τ_{rs} . Indica la preferencia memorística de visitar la ciudad s después de la r
- **Información heurística:** Se define como la inversa del coste del arco (la distancia entre las dos ciudades), $\eta_{rs}=1/d_{rs}$. Indica la preferencia heurística de visitar la ciudad s después de la r (los arcos más cortos tienen mayor preferencia)

Ejemplo: OCH para el TSP

■ **Proceso constructivo:**

- Se sitúa la hormiga en un nodo inicial i . Si hay bastantes hormigas, se sitúa una en cada nodo. Si no, se escoge uno aleatorio. Se inicializa la lista de nodos visitados (solución parcial): $L[1] \leftarrow i$
- En cada uno de los $n-1$ pasos necesarios para construir la solución ($k=2$ hasta n):
 - Se construye una lista de candidatos factibles con los nodos que están conectados al nodo actual r almacenado en $L[k-1]$ y **que no han sido visitados aún** (no están incluidos en L): $J(r) = \{u \mid \exists a_{ru} \text{ y } u \notin L\}$
 - Se escoge uno de esos nodos s con la regla de transición (que tiene en cuenta la feromona τ_{ru} y la heurística η_{ru}). Se almacena en la posición actual de la lista de nodos visitados: $L[k] \leftarrow s$
- **Fuerte similitud con GRASP: la diferencia principal es el uso de feromona (y no sólo heurística) para la selección del candidato**

Aspectos de Implementación: Listas de Candidatos

- Cuando los casos de los problemas son de gran tamaño, los algoritmos de OCH pueden requerir un tiempo de ejecución alto
- Por ejemplo, cuando el grafo es totalmente conexo, el Sistema de Hormigas es de orden $O(t \cdot m \cdot n^2)$ ($t = n^0$ iteraciones, $m = n^0$ hormigas, $n = n^0$ nodos)
- La mayor parte del costo computacional de la OCH proviene de las reglas de transición de estados y actualización de feromona
- Para acelerar ambas, en algunos problemas como el Viajante de Comercio, se usan las **Listas de Candidatos** (LCs)

Aspectos de Implementación: Listas de Candidatos

- Las LCs se definen antes de ejecutar del algoritmo y reducen el vecindario factible en cada paso del proceso constructivo: **aumentan la eficiencia pero reducen la eficacia**
- La LC almacena los c / nodos preferidos para visitar desde cada nodo r ordenados de mejor a peor valor heurístico
- Cuando la hormiga está en el nodo r , la regla de transición sólo considera los nodos incluidos en $LC(r)$ no visitados aún. Sólo se usa el resto cuando todos los nodos de $LC(r)$ han sido visitados
- En ese caso, existen dos posibilidades:
 - escoger directamente el nodo con mejor valor heurístico de los no incluidos en la lista
 - aplicar la regla de transición considerando los nodos restantes

Aspectos de Implementación: Listas de Candidatos

- Con la primera opción, el tiempo de ejecución necesario para que una hormiga construya una solución es lineal ($O(n \cdot cl)$)
- Además, en este caso, es posible acelerar la ejecución de la regla de actualización global de feromona en algoritmos de OCH en los que todas las hormigas actualizan (como el SH)
- Para ello, **sólo se modifica la feromona de los arcos que conectan nodos incluidos en la LC**
- **Esto no es válido en la segunda opción** que sí considera los demás rastros de feromona

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 $S(j)$

Algoritmo OCH para el QAP

- **Esquema de representación:** Las componentes de la solución L son el conjunto de unidades y el de localizaciones
El grafo de construcción G es un **grafo bipartito** compuesto por $2 \cdot n$ nodos (n localizaciones y n unidades) totalmente conexo. Cada nodo unidad u_r está conectado a los n nodos localización l_s
- **Restricciones:** Cada unidad sólo puede estar asociada a una localización y viceversa
- **Función objetivo:** Suma de los costes cuadráticos de las asignaciones unidad-localización de la solución construida L
- **Rastros de feromona:** Se asocia un rastro de feromona a cada arco del grafo a_{rs}, τ_{rs} . Indica la preferencia memorística de asignar la unidad u_r a la localización l_s

Algoritmo OCH para el QAP

- **Información heurística:** Al contrario que en otros problemas de OCH, la información heurística del QAP no es muy confiable
Por esa razón, algunos algoritmos OCH no la consideran

- La opción es usar los potenciales de flujo y distancia:

$$\hat{f}_r = \sum_{i=1}^n (f_{ri} + f_{ir}) \quad ; \quad \hat{d}_s = \sum_{j=1}^n (d_{sj} + d_{js})$$

- Los potenciales de flujo se usan para ordenar las unidades por preferencia en el proceso constructivo, las de mayor flujo primero
- Los potenciales de distancia se usan para definir la información heurística asociada a los nodos localización:

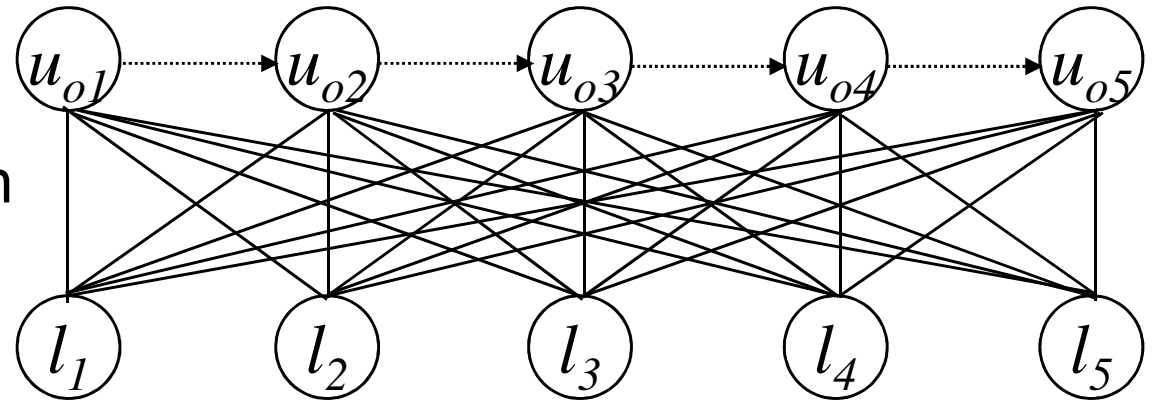
$$\eta_s = \frac{1}{\hat{d}_s}$$

- Los nodos de menor distancia potencial tienen mayor preferencia

Algoritmo OCH para el QAP

- **Proceso constructivo:**

- **Inicialización:** Se ordenan las unidades de menor a mayor potencial de flujo



- En el paso k , se toma la unidad de la posición k del orden, u_{ok}
- *También se puede escoger el orden de unidades aleatoriamente*
- La lista de candidatos factibles incluye siempre las localizaciones no asignadas aún: $J(u_{ok}) = \{j \in \{1, \dots, n\} \mid j \notin L\}$
- Se escoge una localización libre $l_s \in J(u_{ok})$ usando la regla de transición (en función de la feromona $\tau_{u_{ok}i}$ y la heurística η_i)
- Se guarda en L la asignación $u_{ok} \leftrightarrow l_s$ escogida: $L[ok] \leftarrow s$

Algoritmo OCH para la Selección de Características

- **Esquema de representación:** Las componentes de la solución L son las n características del problema. El grafo de construcción incluye n nodos (f_1, \dots, f_n) y es totalmente conexo
- **Restricciones:** No se pueden formar ciclos, es decir, no se puede visitar un nodo (no se puede escoger una característica) más de una vez
- **Función objetivo:** El porcentaje de acierto del clasificador 3-NN diseñado empleando las características seleccionadas en la solución L construida por la hormiga

Algoritmo OCH para la Selección de Características

- **Rastros de feromona:** Tendremos dos tipos distintos:

1. Se asocia un rastro a cada nodo del grafo f_i , $\tau\text{-car}_{f_i}$. Indica la preferencia memorística de seleccionar la característica correspondiente
2. Se asocia un rastro a cada valor posible del número de características nc a seleccionar por cada hormiga, $\tau\text{-num-car}_{nc}$. $nc \in \{1, \dots, n\}$

- Se almacenan en un vector de tamaño $2 \cdot n$. Las n primeras posiciones representan los rastros de la preferencia de las características y los n siguientes los de los posibles números de características:

$$\tau = (\tau\text{-car}_{f_1}, \tau\text{-car}_{f_2}, \dots, \tau\text{-car}_{f_n}, \tau\text{-num-car}_1, \tau\text{-num-car}_2, \dots, \tau\text{-num-car}_n)$$

- Las hormigas actualizan la feromona sobre ambos rastros usando las características seleccionadas en su solución para $\tau\text{-car}$ y el número de características concreto para $\tau\text{-num-car}$ ($\tau\text{-num-car}_0 = 1/N_c$)

- **Los rastros $\tau\text{-car}$ se actualizan con la regla del algoritmo OCH usado. En cambio, los rastros $\tau\text{-num-car}$ se actualizan siempre con la regla del Sistema de Hormigas (para dar diversidad)**

Algoritmo OCH para la Selección de Características

- **Información heurística:** Preferencia heurística de escoger individualmente la característica f_i , definida como la cantidad de información de la variable f_i con respecto a las clases C :

$$\eta_{f_i} = I(C, f_i) = H(C) - H(C/f_i) = \sum_{c=1}^{N_c} \sum_{f_{ij}=1}^{N_{f_i}} P(c, f_{ij}) \cdot \log_2 \left(\frac{P(c, f_{ij})}{P(c) \cdot P(f_{ij})} \right)$$

- $c = 1, \dots, N_c$ son las clases del problema
- $j = 1, \dots, N_{f_i}$ son los valores discretos de f_i . Si la característica es continua se discretiza en h intervalos (p.ej., $h = 10$)
- $P(c, f_{ij})$ es la probabilidad de que un ejemplo del conjunto de entrenamiento sea de clase c cuando la variable f_i vale f_{ij}
- $P(c)$ es la probabilidad de la clase c en el conjunto de entrenamiento
- $P(f_{ij})$ es la probabilidad del valor f_{ij} en el conjunto de entrenamiento

Algoritmo OCH para la Selección de Características

- **Proceso constructivo:**
- **Antes de iniciar el proceso constructivo**, se genera aleatoriamente el número de características que la hormiga concreta va a seleccionar en esta iteración
- Para ello, se usa la distribución de probabilidad con los segundos rastros de feromona, τ -num-car. Se construye una ruleta con ellos y se gira una vez para escoger el número concreto nc
- En cada uno de los n pasos necesarios ($k=1$ hasta nc):
 - Se construye la lista de candidatos factibles con los nodos asociados a las variables no seleccionadas: $J(k) = \{f_i \in \{1, \dots, n\} \mid f_i \notin L\}$
 - Se escoge un nodo f_j de la lista con la regla de transición (en función de la feromona τ -car $_{f_i}$ y la heurística η_{f_i}). Se almacena en L , $L[k] \leftarrow f_j$