Metaheurísticas

Seminario 6. Manejo de restricciones en metaheurísticas

Introducción: Optimización y Restricciones

- ✔ Lenguaje coloquial, **optimizar** significa más o menos **mejorar**
- ✔ Contexto científico: la **optimización** es el proceso de tratar de encontrar la mejor solución posible para un determinado problema

- Problema de optimización:

- ✔ Diferentes **soluciones**, un criterio para **discriminar** entre ellas y el **objetivo** es encontrar la mejor

  *Encontrar el valor de unas variables de decisión (**sujeto a restricciones**) para los que una determinada función objetivo alcanza su valor máximo o mínimo*

Problema de optimización:

- Encontrar el valor x^* de una serie de variables de decisión que maximiza (minimiza) una función objetivo $f(x)$: $x \in X \rightarrow R$

$$x^* \in X: f(x^*) \geq f(x), \forall x \in X$$

- sujeto a una serie de m restricciones ($m=q+p$) de:

✔ desigualdad: $g_i(x) \otimes 0$ ($\otimes \in \{<, >, \leq, \geq\}$) ; $i = 1, \ldots, q$

✔ Igualdad: $h_j(x)=0$; $j = 1, \ldots, p$

Introducción: Optimización y Restricciones

Tipos de soluciones:

- Una solución a un problema de optimización especifica los valores de las variables de decisión y, por tanto, el valor de la función objetivo
- Una solución factible satisface todas las restricciones
- Una solución óptima es factible y proporciona el mejor valor posible de función objetivo

Introducción: Optimización y Restricciones

Tipos de restricciones:

- Las restricciones pueden ser fuertes (es obligatorio satisfacerlas) o débiles (es recomendable satisfacerlas)

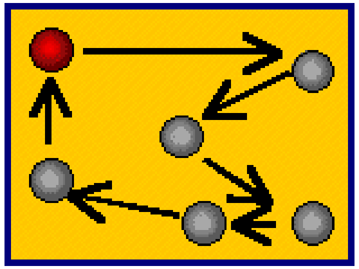
- Ejemplo: En la organización de horarios del curso, una restricción fuerte es que no se solapen las clases y una débil es que no haya clases antes de las tres de la tarde

- Las restricciones pueden ser explícitas (definidas en el problema) o implícitas (obvias para el problema). También pueden ser lineales y no lineales (más complejas)

Introducción: Optimización y Restricciones

Problema del viajante de comercio (TSP):

- Dadas las coordenadas de n ciudades, encontrar el circuito de longitud mínima que visita cada una de ellas una sola vez



- Ejemplo: En el TSP, una restricción implícita es que cada ciudad tiene que visitarse exactamente una vez

  Una restricción explícita sería que, en un entorno concreto, la ciudad A no puede visitarse después que la B

Manejo de Restricciones en Metaheurísticas

- En su diseño básico, las metaheurísticas de búsqueda local o global no están preparadas para gestionar restricciones en problemas de optimización

- En cambio, las metaheurísticas constructivas como GRASP o los algoritmos de Optimización basados en Colonias de Hormigas se adaptan fácilmente a problemas con restricciones

- El manejo de restricciones es un aspecto fundamental en el diseño de metaheurísticas eficientes debido al gran número de problemas de optimización (combinatoria y númerica) con restricciones existentes

- El manejo de restricciones suele ser una decisión genérica independiente del tipo de metaheurística concreta considerada

Manejo de Restricciones en Metaheurísticas

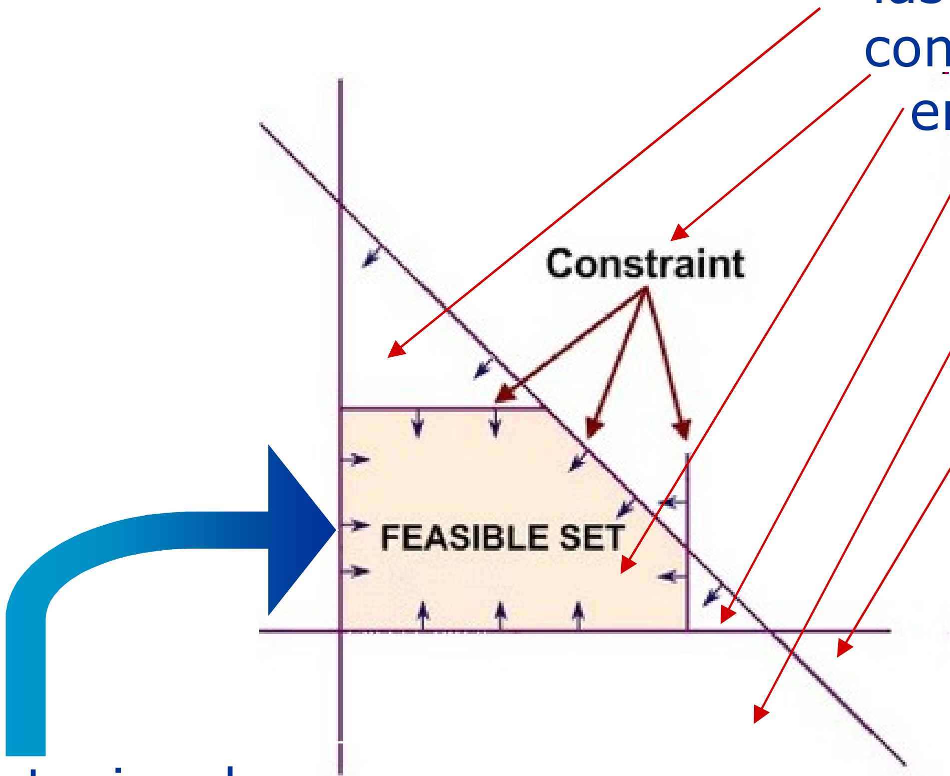
Principalmente existen dos enfoques distintos:

1. Exploración restringida de la región factible del espacio de búsqueda:
   - Ventajas: No se evalúan soluciones no factibles. Los algoritmos aseguran la obtención de una solución factible
   - Inconvenientes: La búsqueda puede ser ineficaz si se restringe solo a la región factible. Las soluciones óptimas pueden estar situadas cerca de su límite y ser difíciles de alcanzar

2. Exploración completa del espacio solución:
   - Ventajas: La exploración del espacio de búsqueda es más eficaz
   - Inconvenientes: Se malgasta tiempo evaluando soluciones no factibles. Existe la posibilidad de devolver una solución no factible como salida final del algoritmo

Manejo de Restricciones en Metaheurísticas

las estrategias de exploración completa realizan la búsqueda en todo el espacio solución



las estrategias de exploración restringida sólo buscan en la región factible

Manejo de Restricciones en Metaheurísticas

Dentro de la exploración restringida, podemos distinguir:

1. Estrategias de rechazo: cualquier solución no factible que se genera durante la búsqueda es directamente ignorada

2. Estrategias de reparación: Se aplica un operador de reparación a cada solución no factible que se genere con objeto de transformarla en otra factible. Suele basarse en un greedy

3. Estrategias de preservación: Tanto el esquema de representación como los operadores están diseñados específicamente para el problema de forma que se asegura la factibilidad de las soluciones generadas (Ej: representación y operadores de orden para el TSP)

   Es la que más esfuerzo de diseño requiere. Es específica y no es aplicable en algunos problemas

Manejo de Restricciones en Metaheurísticas

- El esquema de exploración completa del espacio más habitual son las estrategias de penalización:

- Se añade una función de penalización a la función objetivo sin restricciones original:

$$\text{Min } f'(x) = f(x) + w \cdot P(x)$$

donde $P(x)$ es una función de penalización y w es un coeficiente de ponderación (un peso)

- P toma valor 0 cuando la solución s es factible. En caso contrario, cuanto mayor sea el grado de violación de las restricciones, mayor será el valor de P

Manejo de Restricciones en Metaheurísticas

- Ventaja: Esta estrategia permite a la metaheurística moverse entre las regiones factible y no factible, mejorando así la efectividad de la búsqueda
  - Por ejemplo, permite una secuencia de soluciones del tipo (s_t, s_{t+1}, $\underline{s}_{t+2}$) en tres iteraciones consecutivas donde s_t y s_{t+2} son factibles y s_{t+1} es no factible y donde s_{t+2} es mejor que s_t

- Inconveniente: Es difícil tanto definir *P* de forma adecuada como fijar el valor de *w* para conseguir un buen equilibrio
  - Si se le da una importancia alta al término de penalización solo se consideran soluciones factibles. Por el contrario, si se le da un peso bajo se corre el riesgo de devolver una solución no factible

Manejo de Restricciones en Metaheurísticas

- Una opción sencilla para definir *P* es contar directamente el número de restricciones violadas

- Se da mayor importancia a las restricciones fuertes, las soluciones que las violan tienen un mayor valor de función de coste. Las restricciones débiles se ponderan según su importancia:

$$\text{Min } f'(s) = f(s) + \sum_{i=1}^{m} w_i \cdot \alpha_i$$

  donde α_i=1 si se viola la restricción *i* (α_i=0 en caso contrario) y w_i es el peso asociado a la misma

- Esta modalidad no funciona bien en problemas con restricciones muy fuertes o con un número pequeño de restricciones

Manejo de Restricciones en Metaheurísticas

- Una opción más avanzada implica calcular cuánto de no factible es la solución (grado de infactibilidad o coste de reparación)

- Se considera cómo de cercana está la solución a la región factible. Por ejemplo, los enfoques más eficientes consideran una distancia a la factibilidad de cada restricción individual

- En un problema con *m* restricciones, *q* de desigualdad y *p* de igualdad, se definiría como:

$$\text{Min } f'(s) = f(s) + \sum_{i=1}^{m} w_i \cdot d_i^k$$

  donde $k=\{1,2\}$, w_i es el peso asociado a la restricción *i* y d_i es una métrica de distancia para ella:

  - $d_i = |h_i(x)|$ para las *p* restricciones de igualdad
  - $d_i = \alpha_i \cdot g_i(x)$ para las *q* de desigualdad ($\alpha_i = 1$ si se viola y $\alpha_i = 0$ si no)

Manejo de Restricciones en Metaheurísticas

- Estas ponderaciones pueden mantenerse fijas durante toda la búsqueda (estáticas) o variar dinámicamente según avanza el proceso de búsqueda (dinámicas)

- Esto permite aceptar soluciones que violan las restricciones fuertes al principio de la búsqueda (diversificación) pero rechazarlas en etapas posteriores (intensificación)

- También pueden ser adaptativas y variar según cómo de eficiente esté siendo el espacio de búsqueda

- En cualquier caso, es complicado tanto definir los pesos como los mecanismos de cambio de éstos

Manejo de Restricciones en Metaheurísticas

- Otra alternativa consiste en trabajar con un espacio de búsqueda alternativo: estrategias de decodificación

- Se considera una función $R \rightarrow S$ que asocia cada representación $r \in R$ con una solución factible $s \in S$ en el espacio de búsqueda

- Se usa, por tanto, un esquema de representación indirecto que transforma la topología del espacio de búsqueda

- La función de decodificación debe cumplir una serie de propiedades:
  a) Cada solución $r \in R$ debe corresponder a una solución factible $s \in S$
  b) Cada solución factible $s \in S$ debe tener una representación $r \in R$ asociada
  c) El decodificador debe tener una complejidad baja
  d) Las soluciones factibles de S deben tener el mismo número de soluciones correspondientes en R
  e) Los dos espacios deben corresponder: las distancias entre soluciones en R deben ser parecidas a las distancias entre soluciones factibles en S

Problema de la Mochila

- **Problema de la mochila, *Knapsack Problem***:

  Dados n objetos, cada un con un peso w_j y un valor v_j, se debe seleccionar el conjunto de objetos cuyo valor total sea máximo, sin exceder un peso máximo W

  Esquema de representación: $x=(x_1, \ldots, x_n)$; $x_i=\{0,1\}$

$$KP = \begin{cases} max: \quad \sum_{i=1}^{n} v_i x_i \\ s.a., \\ \quad \sum_{i=1}^{n} w_i x_i \leq W \end{cases}$$

Problema de la Mochila

Exploración restringida, estrategia de reparación

- Se basa en extraer elementos de la mochila en la solución no factible hasta satisfacer la restricción de capacidad

- Se puede realizar aleatoriamente o basado en una heurística greedy

- Un ejemplo de un operador de reparación greedy sería:

  $x' \leftarrow x$

  Mientras x' sea no factible ($\sum w_i \cdot x'_i > W$)

  Eliminar el elemento que maximiza el ratio v_i/w_i

  Devolver x'

Problema de la Mochila

Exploración completa, estrategias de penalización

- Al ser un problema de maximización, se considera una función modificada del tipo:

$$\text{Max } f'(x) = f(x) - P(h(x))$$

donde $f(x)=\sum v_i \cdot x_i$ es la función objetivo original y P es la función de penalización definida en función del grado de violación de la restricción de capacidad $h(x)$ en la solución x:

$$\mathrm{h(x)} = \begin{cases} \sum_{i=1}^{n} w_i \cdot x_i - W, & \text{si } \sum_{i=1}^{n} w_i \cdot x_i > W \\ 0, & \text{en caso contrario} \end{cases}$$

grado de infactibilidad o coste de reparación

Problema de la Mochila

- Se pueden usar distintas funciones de penalización, con diferente grado de penalización:

$$P_{log}(x) = \log_2 (1+v_{max} \cdot h(x))$$

$$P_{lin}(x) = v_{max} \cdot h(x)$$

$$P_{cuad}(x) = [v_{max} \cdot h(x)]^2$$

donde $v_{max} = \max_{i=1, \ldots, m} v_i/w_i$

- La función de penalización logarítmica proporciona excelentes resultados para una amplia gama de casos

Problema de la Mochila

Estrategias de decodificación

- Una posibilidad es usar una representación de orden sobre los n objetos:

$$\pi=[\pi(1), \ldots, \pi(n)]$$

- Se emplean operadores de orden para obtener soluciones válidas en el espacio alternativo

- La solución se decodifica recorriéndola en orden e insertando los objetos en la mochila hasta que se supere su capacidad

- Cada solución en *R* se asocia con una única solución factible en *S* pero hay redundancia, varias soluciones de *R* (varios órdenes) representan la misma solución factible

Problema del Viajante

Exploración restringida, estrategias de preservación

- Se basa en el uso de representación y operadores de orden:

$$\pi=[\pi(1), \ldots, \pi(n)]$$

- La solución representada es directamente factible: no hay ciudades repetidas y se visitan todas
- Operadores de vecino: intercambio (2-opt), inserción, inversión, sublista aleatoria, etc.
- Operadores de cruce: OX, PMX, etc.

Problema del Clustering

Estrategias de decodificación

- Se basan en el uso de una representación de medoides y el uso de una decodificación para obtener una configuración de clusters factible (con k clusters)

- Por ejemplo, se usa una representación de orden para los patrones ($\pi=[\pi(1), \ldots, \pi(m)]$) y los k primeros patrones de la representación se consideran los centroides (medoides) iniciales

- Los m-k patrones restantes se asignan a cada cluster según la regla del prototipo más cercano

- Se van actualizando los centroides tras cada asignación (provoca menos redundancia de los genotipos, sólo en los k primeros genes)