Técnicas de Soft Computing 2024-2025

Master Ciencia de Datos

Universidad de Granada

Optimización - Problema de Diversidad Máxima

Miguel García López

Índice

Índice de figuras

Índice de cuadros

1. Introducción

El problema de diversidad máxima es un problema común de optimización combinatoria. Este problema consiste en encontrar un subconjunto M de m elementos, es decir, $|M| = m$ a partir de un conjunto inicial N con n elementos (siendo n ¿m), de forma que ese subconjunto M sea de máxima diversidad, es decir, sus elementos deben ser los más diversos entre sí.

La diversidad es una métrica a elegir, ya que puede representar multitud de operaciones. En este caso se escoge como diversidad la diferencia absoluta entre dos elementos i y j, es decir, $d_{ij}|N_i - N_j|$.

El problema se puede definir como un problema de optimización sujeto a restricciones de la siguiente manera:

$$\begin{aligned} \text{Maximizar} \quad & MD(X) = \sum_{i=1}^{n-1} \sum_{j=i}^{n} d_{ij} x_i x_j \\ \text{sujeto a} \quad & \sum_{i=1}^{n} x_i = m \\ & x_i = \{0, 1\}, i = 1, ..., n \end{aligned} \tag{1}$$

Donde x_i representa una solución al problema que consiste en un vector binario donde $x_i = 1$ representa que el elemento en la posición i ha sido seleccionado y donde d_{ij} es la diversidad entre un elemento en la posición i y otro elemento en la posición j.

2. Solución heurística

Para la solución heurística *ad-hoc*, es decir, un método diseñado específicamente para resolver un problema en particular sin seguir un enfoque generalizable o teóricamente óptimo siguiendo simplificaciones o intuiciones que funcionan bien, se propone un algoritmo **Greedy**.

2.1. Greedy

Un algoritmo **Greedy** (o voraz) es un enfoque de resolución de problemas que toma decisiones paso a paso, eligiendo en cada paso la opción que parece ser la mejor en ese momento, con la esperanza de que esta estrategia lleve a una solución óptima o cercana a ella.

Este tipo de método se llama voraz ya que no considera decisiones pasadas y solo se mueve hacia delante tomando la mejor decisión en cada momento, de forma que es bastante eficiente en tiempo. Al tomar este tipo de estrategia es posible que se acabe en óptimos locales, ya que se pierde parte de la potencia combinatoria y de exploración de otros algoritmos. Pese a ello es una solución intuitiva y fácil de implementar.

2.2. Pseudocódigo

Algorithm 1 Heurística Voraz para el Problema de Máxima Diversidad

procedure HEURÍSTICAVORAZ(N, m)
  Entrada: Conjunto $N = \{N_1, N_2, \ldots, N_n\}$ con valores numéricos, y m elementos a seleccionar
  Salida: Subconjunto M de índices tal que $|M| = m$ y maximice la diversidad
  if $m > n$ **then**
    error: "m no puede ser mayor que n"
  end if
  $i_{max} \leftarrow$ índice del máximo valor en N
  $i_{min} \leftarrow$ índice del mínimo valor en N
  $M \leftarrow \{i_{max}, i_{min}\}$
  while $|M| < m$ **do**
    $R \leftarrow \{1, 2, \ldots, n\} \setminus M$ ▷ Elementos no seleccionados
    $i_{mejor} \leftarrow \arg\max_{i \in R} \sum_{j \in M} |N_i - N_j|$
    $M \leftarrow M \cup \{i_{mejor}\}$
  end while
  return M
end procedure

El algoritmo **Greedy** comienza eligiendo dos índices, los índices de los valores máximo y mínimo del conjunto inicial. A partir de ahí se itera hasta que el subconjunto M alcance el tamaño especificado por el parámetro m de la función.

En cada iteración se escoge el índice de aquel elemento que no esté en el subconjunto M y que maximice la diversidad entre el mismo elemento i y todos aquellos elementos pertenecientes ya a M.

2.3. Resultados

Pese a no ser necesario realizar el código, tan solo con el pseudocódigo bastaba, se ha escrito una implentación del algoritmo voraz y se han corrido experimentos 30 veces con un N de tamaño 1000 y un tamaño m para el subconjunto M de 20 comparándolo con una solución aleatoria para ver la eficaz del mismo. Los resultados se pueden ver en la tabla 1

Parameters	
Number of experiments	30
Set size (n)	1000
Subset size (m)	20
Average Diversity Values	
Greedy solution	38617.90 ± 580.52
Random solution	19501.69 ± 2780.59
Improvement Ratios	
Greedy over Random	2.02×

Cuadro 1: Maximum Diversity Problem Experiment Results

3. Solución búsqueda local

La búsqueda local es un algoritmo heurístico básico que se basa en el principio de *explotación*, es decir, mejora una solución base refinándola lo máximo posible.

Como se ha dicho, mejora sobre una solución base, por lo que es necesario que esa solución sea obtenida con un algoritmo de búsqueda previo o incluso puede aplicarse sobre una solución aleatoria. Obviamente tiene más potencia cuando la solución en la que se basa es buena.

Este algoritmo evalua la solución inicial e itera sobre esta cambiando elementos de la misma (de manera aleatoria o con búsqueda exhaustiva) durante un número de iteraciones máximas definido o hasta alcanzar una métrica mínima de mejora a alcanzar. Si durante el proceso iterativo se mejora, se utiliza esa nueva solución, si no, se queda con la anterior.

3.1. Pseudocódigo

Como puede verse en el pseudocódigo, se comienza con una solución basada en **Greedy** y se itera evaluando cada nueva solución fruto de un intercambio de elementos en el vector solución final. Este cambio solo se queda en el vector si mejora la solución de la que parte.

3.2. Resultados

Los resultados obtenidos se han obtenido bajo las mismas condiciones explicadas en la sección de resultados anterior.

Algorithm 2 Búsqueda Local para Máxima Diversidad

```
procedure BúsquedaLocal(N, m, max_iter)
    M ← HeurísticaVoraz(N, m)
    V ← Evaluar(M, N)
    for iter = 1 to max_iter do
        mejor_swap ← None
        for i ∈ M, j ∈ N \ M do
            M' ← M con i reemplazado por j
            if Evaluar(M', N) > V then
                mejor_swap ← (i, j)
                V ← Evaluar(M', N)
            end if
        end for
        if mejor_swap = None then
            break
        else
            Intercambiar en M
        end if
    end for
    return M
end procedure
```

Parameters	
Number of experiments	30
Set size (n)	1000
Subset size (m)	20
Average Diversity Values	
Greedy solution	38617.90 ± 580.52
Local search solution	39565.44 ± 98.95
Random solution	19501.69 ± 2780.59
Improvement Ratios	
Greedy over Random	2.02×
Local Search over Random	2.07×
Local Search over Greedy	1.02×

Cuadro 2: Maximum Diversity Problem Experiment Results