Centro de Investigación en Matemáticas (CIMAT).
Unidad Monterrey

Algoritmo Genérico:
Problema P-Mediana

Marcelo A. Sanches Zaragoza
Karla M. Reyes Maya

16 de noviembre de 2021

1. Problema P-Mediana

1.1. Parámetros

- I : conjunto de instalaciones
- J : conjunto de clientes
- C_ij : costo de asignación del cliente j a la instalación i, $\forall i \in I$, $j \in J$
- f_ij costo de localizar la instalación i, $\forall i \in I$
- p : número de instalaciones que se deben abrir

$$x_{ij} = \begin{cases} 1 \; si \; el \; cliente \; j \; se \; asigna \; a \; la \; instalacion \; i \\ 0 \; otro \; caso \end{cases}$$

$$y_i = \begin{cases} 1 \; si \; la \; instalacion \; abre \\ 0 \; otro \; caso \end{cases}$$

1.2. Modelo

Minimizar:

$$z = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i y_i$$

Sujeto a:

$$\sum_{i \in I} y_i = p$$

$$\sum_{j \in J} x_{ij} = 1 \; \forall i \in J$$

$$x_{ij} \leq y_i \; \forall i \in I, j \in J$$

$$y_i \in \{0, 1\}, x_{ij} \in 0, 1 \; \forall i \in I, j \in J$$

1.3. Restricciones

En la ecuación (1) la función objetivo es minimizar el costo de asignación y el costo de apertura, la restricción (2) asegura que sólo se abra la cantidad de instalaciones indicada, la restricción (3) garantiza que cada cliente sea asignado a una sola instalación, la restricción (4) asegura que cada cliente sea asignado solo a una instalación abierta y finalmente la restricción (5) es la restricción de signo de las variables de decisión.

2. Pseudocodigo

Algorithm 1 Algoritmo Genético para problema P-Mediana

```
Require: I, J, C_ij, f_i, p, N_gen, K, prob_cruza
  Generación Población Aleatoria
  P's ⟵ random[P_1, ··· , P_p] tomada del conjunto I con cardinalidad p.
  P_0 ⟵ binario[P's] codificación binaria de P's para las instalaciones
  for 0 hasta N_gen do
      function Torneo de padres(P_0,P's)
          Evaluar Fitness(P's)
          Seleccionar 2 mejores Fitness
          Seleccionar los P_0 con los mejores Fitness(P's)
          return Padres_P_0 ⟵ 2 Cromosomas P_0 con los mejores Fitness
      end function
      function Generacion Hijos(P_0,prob_cruza,p)
          p_muta = 1 − p_cruza
          if random probabilidad ≤ p_muta then
              function Operador Muta(Padres_P_0)
                  Seleccionar un genes random del cromosoma P_0
                  return Hijo ⟵ cambiar los genes
              end function
          else
              function Operador Cruza(Padres : P_0)
                  Seleccionar aleatoriamente genes de los Padres : P_0
                  return Hijo ⟵ combinación de los genes
              end function
          end if
      end function
      function Abortos(p)
          if suma(Hijo(binario)) == p then
              Hijo se agrega a la población P_0
          else
              De nuevo GENERACION HIJOS(Padres_P_0)
          end if
      end function
      function Remplazo(P_0,prob_cruza,p)
          Evaluamos el Fitness del la nueva población
          Remplazamiento del peor Fitness
      end function
  end for
  return Mínimo del Fitness (F.O.), Solución Factible(y_i^*), Solución Asignación de los clientes
```

3. Generación de la Población

Para generar la población inicial, los cromosomas se generaron aleatoriamente. En el algoritmo propuesto se crea un vector de cardinalidad p donde cada entrada indica qué instalación es la que abre, ver el Cuadro 3.1. pero este es sólo para crear la población de soluciones iniciales pero sobre este vector no se pueden aplicar los operadores genéticos.

	Vector inicial		
Instalación.	2	4	1

Cuadro 3.1: En el supuesto de que $p = 3$, el individuo indica que se abren 3 instalaciones: la instalación 1, 2 y 4.

A pesar de que el vector anterior indica que instalaciones están abiertas no hay que perder de vista que la representación del conjunto de las instalaciones I está representado como el vector binario y_i que nos indica la localización, ver el Cuadro 3.2.

	Cromosoma				
Genes	1	1	0	1	0

Cuadro 3.2: Con base en el vector inicial anterior, con un conjunto de $I = \{1, 2, 3, 4, 5\}$ la codificación de las instalaciones abiertas está representada por el vector binario.

Finalmente, respecto a la representación de la solución, lo correcto es que represente la localización de las instalaciones porque los clientes se deben asignar a la instalación más cercana (que sería la más barata). Así al saber qué instalaciones están abiertas (localización) en el vector indicador y_i se puede saber la asignación de los clientes en la matriz indicadora X_{ij} según la matriz de costos C_{ij}. Por lo tanto, la codificación de la solución debe de ser del proceso de localización (debe indicar qué instalaciones están abiertas) y esta es binaria que es y_i y será el cromosoma sobre los que se aplicaran los operadores de cruza y muta.

4. Selección de padres

La selección de padre se realizo mediante un torneo de padres en el que se elige una vecindad de K individuos de la población de soluciones y se eligen dos individuos con los mejores Fitness (mejor valor en la Función Objetivo) para ser los padres.

5. Generación de hijos

Respecto a los operadores genéticos deben aplicarse a las soluciones de la población inicial. Para ello se usaron probabilidades complementarias conde la probabilidad de cruza es p_{cruza} y la de muta es $p_{muta} = 1 - p_{cruza}$.

5.1. Operador cruza

Para la cruza se elijen genes al azar de ambos padres y estos se combinan para crear un nuevo cromosoma. Si los hijos no son solución factibles entonces son abortos y entra a reparación.

5.2. Operador Muta

Para la muta se elige un gen al azar y se modifica.

6. Método de Reemplazamiento

Una vez evaluado el Fitness de la población se selecciona la solución con el peor resultado y se reemplaza por el hijo.

7. Datos de la Implementación

Los datos para la implementación se proporcionan en un archivo txt. Este Archivo contiene los valores de la matriz de costos para los clientes C_{ij}. Mientras que el parametro f_i costo de las instalaciones, en este caso, se definen en el notebook.