**Problema de colegios (clustering) – Alejandro Uribe**

1. Escoger la metaheuristica para dar la solución al problema:

**GRASP**

En la literatura se han encontrado varios métodos como solución para el problema de clustering. Cuando se tienen pequeños datasets se puede aplicar programación dinámica o Branch and bound. Sin embargo, para grandes datasets se tiene que utilizar una metaheuristica que entregue una buena solución en un buen tiempo de procesamiento (Bagirov et al., 2020). Se pueden comparar muchos métodos para dar solución a este problema entre los cuales se encuentra búsqueda tabú, recocido simulado, GRASP, algoritmos genéticos, entre otros. En el estudio (Al-Sultan & Khan, 1996) mencionan que los mejores resultados se obtienen con los métodos mencionados y brinda mejores resultados que el k-means. Sin embargo, mencionan que el algoritmo k-means es el mas eficiente en tiempo de ejecución. Sin embargo, estudios como (Cano et al., 2002) buscan implementar un GRASP para eliminar el problema de estar atascado en un mínimo local proporcionado por k-means y permitir una exploración mayor. Entre algunos estudios donde tienen en cuenta GRASP para la solución de problemas están (Tejada et al., 2019) y (Papazek et al., 2013)

Dado el balance que se puede lograr tanto en tiempo de computo como en buenos resultados y en la facilidad de implementación dados fases configurables (construcción y local search) se elige GRASP como herramienta para abordar el problema de clustering planteado.

1. Parámetros y componentes de la heurística mencionada

* **Representación de la solución:** como voy a visualizar la solución (tanto cuales son los acopios y a que colegios cubre cada uno).
* **Cálculo de la función objetivo:** cálculo del costo de la solución
* **Evaluación de factibilidad:** verificar si la solución propuesta es factible
* **Conjuntos y datos de la instancia:** datos de demanda, ubicación y distancias entre colegios
* **Fase 1:** construcción:
  + **Método constructivo:** añadir componentes a la solución para crear una solución factible
  + **Lista restringida de candidatos:** lista con n elementos candidatos a ser el próximo nodo para seleccionar para la construcción de la solución
* **Fase 2:** Búsqueda Local:
  + **Movimiento:** cambio que se propone para iterar de una solución factible a una mejor solución
  + **Criterio de parada búsqueda local:** hasta cuando se ejecutan los movimientos en una búsqueda local
* **Criterio de parada algoritmo:** hasta cuantas soluciones se planean evaluar antes de parar y retornar la incumbente
* Incumbente y su costo: mejor solución encontrada y el costo de la F.O asociado.

1. Diseño de la implementación de la metaheuristica

**Representación de la solución:** Se define una representación de la solución con un vector Sol de tamaño k (# acopios). Este tendrá el número de los acopios. Por ejemplo, para 3 acopios, el vector puede ser (13, 3, 7).

Por otra parte, un vector de Clusters que para cada posición o nodo tiene asociado un cluster que será un numero entre 0 y k. Entonces para 5 nodos y los 3 acopios del ejemplo anterior el vector puede ser (0, 0, 1, 2, 1). En este cada numero corresponde a la posición del cluster en el vector Sol. A manera ilustrativa, para los que tienen un 1 como cluster, tienen como centro el nodo 3.

**Cálculo de la función objetivo:** se proponen como función objetivo la minimización de la suma de las distancias de cada uno de los nodos a su respectivo centro

**Evaluación de factibilidad:** verificar que el conjunto de todos los clústeres contenga cada uno de los nodos (colegios) de la instancia. Que el numero de clusters si sea igual a los definidos. También que solo se tenga un cluster por nodo.

**Conjuntos de datos:** corresponde a una instancia pequeña de 32 colegios y la instancia grande corresponde a 452 colegios. Cada uno cuenta con información de su ubicación (en términos de longitud y latitud) y su demanda, además, se tiene una matriz de distancia entre cada uno de estos colegios.

K = número de clústeres

Proponer una solución aleatoria de k clústeres y almacenarla junto a su costo como incumbente.

**Iterar 50 veces:** (Parámetro determinado empíricamente con el dataset small)

Fase constructiva:

X = solución de centros vacía

Elegir un nodo al azar como centro de clúster para comenzar la construcción

X += nodo seleccionado

**Hacer k – 1 iteraciones:**

Asignar todos los nodos al clúster de X con menor distancia y calcular el costo

Crear lista restringida de candidatos RCL con K elementos que reduzcan en mayor cantidad el costo de la F.O seleccionada sin tener en cuenta centros ya seleccionados

Seleccionar al azar uno de estos candidatos y sumar este nodo a la solución:

X += nodo aleatorio candidato

**Return:** X, costo

Fase de búsqueda local:

**Movimiento:** Intercambiar un centro con un nodo

**Hacer hasta que no se encuentren mejoras en un centro:**

Elegir aleatoriamente un nodo “centro” de clúster:

Organizar de manera aleatoria la lista de nodos que no son centros

Para cada nodo en la lista aleatoria:

Cambiar el centro por el nodo de la lista

Recalcular las distancias mínimas de los nodos a los centros

Si el costo del cambio mejora el de X (primera mejora),

reemplazar X y el costo asociado con el cambio actual y salir del loop

En otro caso

avanzar en la iteración

Si el costo de X es menor que el del incumbente se reemplaza la incumbente por la nueva solución y costo de la función objetivo

**Referencias**

Papazek, P., Raidl, G. R., Rainer-Harbach, M., & Hu, B. (2013). A PILOT/VND/GRASP hybrid for the static balancing of public bicycle sharing systems. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8111 LNCS(PART 1), 372–379. https://doi.org/10.1007/978-3-642-53856-8\_47

Cano, J. R., Cordón, O., Herrera, F., & Sánchez, L. (2002). A GRASP algorithm for clustering. Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), 2527(March 2014), 214–223. https://doi.org/10.1007/3-540-36131-6\_22

Al-Sultan, K. S., & Khan, M. M. (1996). Computational experience on four algorithms for the hard clustering problem. Pattern Recognition Letters, 17(3), 295–308. https://doi.org/10.1016/0167-8655(95)00122-0

Bagirov, A. M., Karmitsa, N., & Taheri, S. (2020). Partitional Clustering via Nonsmooth Optimization. In Unsupervised and Semi-Supervised Learning.