TAREA COMPUTACIONAL 1 2s20
INVESTIGACIÓN DE OPERACIONES - ILN250

Profesores	:	Rodrigo Mena - Rafael Favereau U. - Pablo Escalona R.
Ayudante	:	Katherine Álvarez - Ignacio Cabezas - Carla Rojas - Vincenzo Rosati Eduardo Villanueva
Fecha	:	Viernes 16 de Octubre 2020
Entrega	:	Lunes 9 de Noviembre 2020 - 09:00 am

Instrucciones

- La tarea debe ser desarrollada en parejas, permitiéndose combinaciones entre los cinco paralelos indicados en el programa del curso.
- El informe de entrega deberá incluir Resumen, Introducción, Formulación del Modelo, Enfoques de Solución (incluido pseudocódigos), y Conclusiones. Solo si es necesario se pueden incluir Anexos. La **extensión máxima** del informe es 10 páginas (sin contar apéndices ni bibliografía).
- Los supuestos utilizados para el desarrollo deben ser especificados.
- Se deberá entregar un archivo.rar con el nombre "*Tarea1OR_Apellido1_Apellido2.rar*", que contenga tanto el informe en PDF como los archivos de los programas utilizados.
- **No se aceptarán tareas fuera de plazo.**

Enunciado

Considere la siguiente formulación para el Single Source Weber Problem (Weber et al. [1929]).

$$\textbf{SSWP}: \quad \min_{\bar{x},\bar{y}} \quad \sum_{i\in I} w_i \sqrt{(x_i-\bar{x})^2+(y_i-\bar{y})^2}$$
$$\text{s.t:} \quad (\bar{x},\bar{y}) \in \mathbb{R}^2,$$

donde I ($i = 1, ..., n$) es un conjunto de puntos en el plano que representan orígenes y destinos; $w_i > 0$ es el peso asociado al punto de origen o destino i-ésimo; y (x_i, y_i) la ubicación del punto de origen o destino i-ésimo en el plano $\mathbb{R}^2$.

El objetivo en **SSWP** es minimizar la suma ponderada de las distancias entre un nuevo servidor y los puntos de origen o destino, seleccionando la mejor ubicación para un nuevo servidor en cualquier parte del plano. Sea $(\bar{x}, \bar{y})$ la localización del nuevo servidor (variables).

La formulación **SSWP** es un problema de optimización convexo irrestricto. Por lo tanto, se puede resolver usando un *solver* para problemas de optimización convexo como MINOS o aplicando CPOs, la cual es condición necesaria y suficiente para este tipo de problemas de optimización.

Aplicando CPOs, obtenemos:

$$\bar{x} = \frac{\sum_{i \in I} \frac{w_i x_i}{d_i}}{\sum_{i \in I} \frac{w_i}{d_i}}, \qquad \bar{y} = \frac{\sum_{i \in I} \frac{w_i y_i}{d_i}}{\sum_{i \in I} \frac{w_i}{d_i}},$$

donde $d_i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2}$.

Como es habitual en problemas de optimización convexos irrestrictos, no se puede encontrar una solución analítica resolviendo el sistema de ecuaciones que se derivan de las CPOs. Por lo tanto, resolvemos usando una algoritmo iterativo de descenso como *gradiente* o *Newton*, o algoritmos especialmente diseñados para este problema.

Un algoritmo especializado es el *Algoritmo de Weiszfeld* (Weiszfeld and Plastria [2009]), que se puede describir mediante el siguiente pseudocódigo:

1. Comenzar con una solución inicial $(\bar{x}^{(0)}, \bar{y}^{(0)})$.
   Calcular $d_i^{(0)} = \sqrt{\left(x_i - \bar{x}^{(0)}\right)^2 + \left(y_i - \bar{y}^{(0)}\right)^2}$ para todo $i \in I$.
   Calcular $Z^{(0)} = \sum_{i \in I} w_i d_i^{(0)}$.
   Iniciar iteración con $k = 1$.

2 Calcular la siguiente iteración $(\bar{x}^{(k)}, \bar{y}^{(k)})$ como sigue:

$$\bar{x}^{(k)} = \frac{\sum_{i \in I} \frac{w_i x_i}{d_i^{(k-1)}}}{\sum_{i \in I} \frac{w_i}{d_i^{(k-1)}}}, \qquad \bar{y}^{(k)} = \frac{\sum_{i \in I} \frac{w_i y_i}{d_i^{(k-1)}}}{\sum_{i \in I} \frac{w_i}{d_i^{(k-1)}}}.$$

   Calcular $d_i^{(k)} = \sqrt{\left(x_i - \bar{x}^{(k)}\right)^2 + \left(y_i - \bar{y}^{(k)}\right)^2}$ para todo $i \in I$.
   Calcular $Z^{(k)} = \sum_{i \in I} w_i d_i^{(k)}$.
   Si $k > k_{max}$ o $|Z^{(k)} - Z^{(k-1)}| < \varepsilon$, entonces `STOP`.
   Si no se cumple el criterio de parada, $k = k + 1$ y volver al paso 2.

Lo que se pide

En esta tarea se desea explorar y comparar diferentes algoritmos de descenso y solvers para resolver **SSWP**. Para todos los métodos de descenso se debe usar como punto de partida:

$$\bar{x}^{(0)} = \frac{\sum_{i \in I} w_i x_i}{\sum_{i \in I} w_i}, \qquad \bar{y}^{(0)} = \frac{\sum_{i \in I} w_i y_i}{\sum_{i \in I} w_i}.$$

Los métodos de resolución a comparar son los siguientes:

1. Algoritmo de Weiszfeld.

2. Método de la Gradiente con Backtracking.

3. Método de Newton con Backtracking.

4. Solver MINOS usando AMPL.

Los métodos (1), (2) y (3) pueden ser programados en C, Python o Matlab según su preferencia. Usted deberá comparar, mediante promedios y máximos, el CPU time, función objetivo y número de iteraciones para los diferentes métodos de resolución.

Instancias de testeo

En esta tarea usted elaborará tres configuraciones de testeo y para cada configuración se deben generar 100 instancias aleatorias. El detalle de cada configuración de testeo son las siguientes:

Configuración	$\|I\|$	x_i	y_i	w_i
1	100	$U \sim \{0, ..., 100\}$	$U \sim \{0, ..., 100\}$	$U \sim [10, 50]$
2	1000	$U \sim \{0, ..., 1000\}$	$U \sim \{0, ..., 1000\}$	$U \sim [100, 500]$
3	10000	$U \sim \{0, ..., 10000\}$	$U \sim \{0, ..., 10000\}$	$U \sim [1000, 5000]$

Donde x_i, y_i siguen una distribución uniforme discreta entre 0 y $|I|$, y w_i sigue una distribución uniforme continua entre los rangos indicados.

Referencias

A. Weber, C. J. Friedrich, et al. Alfred weber's theory of the location of industries. 1929.

E. Weiszfeld and F. Plastria. On the point for which the sum of the distances to n given points is minimum. *Annals of Operations Research*, 167(1):7–41, 2009.