

Recocido Simulado

->20 por grupo

- ☒ ~~función objetivo: suma del beneficio de cada grupo~~
donde el beneficio de cada grupo es la suma por pares
- ☒ ~~transición: swapear elementos~~
- ☒ ~~representación de la solución:~~

***Primer approach:**

Al terminar de leer el problema, me percaté que básicamente estamos hablando del problema de la mochila, ya que se encuentra una gran similitud al intentar maximizar la suma de “beneficios” que vendría siendo el peso por objeto en el caso de la mochila.

***Mecanismo de perturbación:**

El funcionamiento de dicho mecanismo consiste en primero asegurar que todos los grupos tengan los elementos necesarios para que el peso viva dentro de los límites establecidos, y una vez alcanzada esa meta, nos disponemos a asignar a cada elemento un grupo, para que quede un acomodo completo.

Posteriormente, el mecanismo se encargará de pasar elementos de un grupo a otro con esperanza de mejorar la solución.

***Explicación de la representación de la solución:**

Para representar la solución, me dispuse a utilizar vector de vectores en el caso de la primera parte del output (grupos y sus elementos), así formando la siguiente estructura

-vector en el que cada casilla coloco el número de grupo al que pertenece su elemento

Además utilicé un vector de pares que almacena la suma de pesos y los beneficios de cada grupo, así formando la siguiente estructura:

-vector de pares: [[suma de pesos, beneficios]]
g [suma, beneficio]
r ...
u ...
p ...
o [suma, beneficio]

***Evaluación de la función objetivo:**

Para la evaluación de la función objetivo, iteré sobre cada grupo, y por cada par de elementos (A y B) sumé sus beneficios, para después sumar cada uno resultante de las sumas locales, así consiguiendo una suma global por grupo. En el siguiente diagramita lo explico con mayor detalle.

Grupo 1:

A = [1,2,3,4,5,6]

B = [2,3,4,5,1,2]

sumas = [3,5,7,9,6,8]

sumG = 38

“sumG” representa la suma global del grupo 1, en este caso con valor de 38; mientras que “sumas” es la suma local entre $A_i + B_i$.

****Conclusión:***

Después de realizar las 50 iteraciones pude darme cuenta de que el algoritmo resultaría poco estable, además de que no es capaz de explorar todo el espacio de búsqueda, sin embargo de eso se trata la metaheurística y cumple con el objetivo de acercarse con su propio ritmo y dirección (que puede variar con cada mecanismo de perturbación) a la solución óptima.