

	<p style="text-align: center;">UNIVERSIDAD DE LOS ANDES</p> <p style="text-align: center;">FACULTAD DE INGENIERÍA</p> <p style="text-align: center;">DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN</p> <p style="text-align: center;">Modelado, Simulación y Optimización</p> <p style="text-align: center;">Profesor</p> <p style="text-align: center;">Germán Montoya O. ga.montoya44@uniandes.edu.co Jefe de Laboratorio</p> <p style="text-align: center;">Juan Andrés Méndez ja.mendez@uniandes.edu.co</p>	
---	---	---

LABORATORIO 4

Optimización Multiobjetivo y Método Simplex

EJERCICIO 1 (30%): Método eConstraint en Pyomo

Resuelva el mismo caso presentado en “multiobjetivoHopsCosts_sumasPonderadas.py”, pero **implementando un nuevo método multiobjetivo llamado “el método multiobjetivo de eConstraint”**. Implementar este método en Pyomo.

Para la implementación de este método, siga los siguientes pasos:

-La idea general del método se puede observar en la siguiente figura:

$$\begin{array}{ll}
 \max & (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x})) \\
 \text{st} & \\
 & \mathbf{x} \in S,
 \end{array}
 \quad \longrightarrow \quad
 \begin{array}{ll}
 \max & f_1(\mathbf{x}) \\
 \text{st} & \\
 & f_2(\mathbf{x}) \geq e_2, \\
 & f_3(\mathbf{x}) \geq e_3, \\
 & \dots \\
 & f_p(\mathbf{x}) \geq e_p, \\
 & \mathbf{x} \in S.
 \end{array}$$

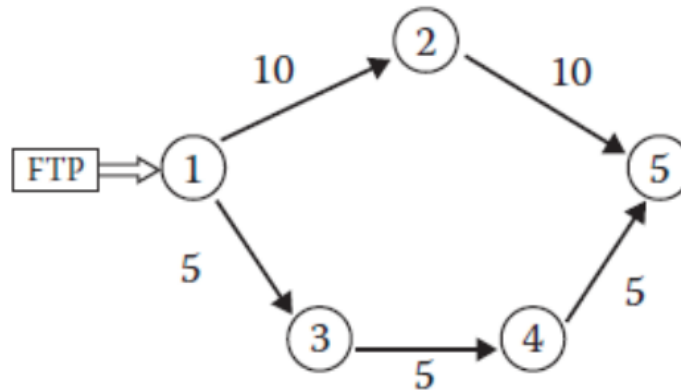
Nota: Si minimizamos, entonces el operador Relacional sería ‘menor o igual’ (<=).

-De

manera general, como se puede observar en la figura, si deseo maximizar (o minimizar) varias funciones (lado izquierdo de la figura), según este método solo debo dejar una única función como función objetivo general (lado derecho: max $f_1(\mathbf{x})$) y las demás restricciones se convierten en restricciones adicionales a las que ya tenía el modelo. Por lo tanto, la función $f_2(\mathbf{x})$ pasa a convertirse en la restricción $f_2(\mathbf{x}) \geq e_2$; la función $f_3(\mathbf{x})$ pasa a convertirse en la restricción $f_3(\mathbf{x}) \geq e_3$ y así sucesivamente. A los $e_1, e_2, e_3 \dots$ etc se les llama los epsilon de cada función y corresponden a un valor

número que debe ir cambiando para obtener el frente óptimo de Pareto. En otras palabras, cada vez que cambiemos los epsilons, debemos guardar los valores de las funciones objetivo $f_1(x)$, $f_2(x)$... $f_p(x)$ para luego obtener el frente óptimo de Pareto.

-La idea es entonces optimizar el mismo caso de “multiobjetivoHopsCosts_sumasPonderadas.py” pero **NO** usando sumas ponderadas sino el **método de e-Constraint**. Recuerde que el caso a resolver es el de la siguiente figura:

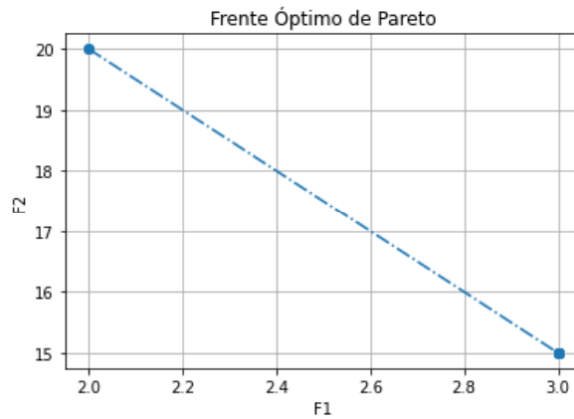


Donde deseamos encontrar una ruta que nos lleve de 1 a 5 minimizando costos y hops.

Para implementar el método de e-Constraint al caso de la figura anterior, siga los siguientes pasos:

- Deseamos minimizar dos funciones, función de costo y función de hops. Con e-Constraint solo podemos dejar una única función como función objetivo general. Por ejemplo, podríamos dejar a la función de costo como función general.
- Agregamos entonces como restricción la función de hops: $F_{\text{hops}}(x) \leq \epsilon$. El epsilon debería tener un valor que permitiera que la función de hops sea factible según el escenario que estemos resolviendo. Por ejemplo, un valor de epsilon podría ser 5, o sea $F_{\text{hops}}(x) \leq 5$. Si resolvemos el modelo con este valor de epsilon, la ruta de 1 a 5 sería la 1-3-4-5 y los valores de la funciones serían 15 para la función de costo y 3 para la de hops, obteniendo así un primer punto del frente de Pareto. Si empezamos a decrementar de a 1 el valor de epsilon, veremos que con $\epsilon=2$ la ruta de 1 a 5 sería 1-2-5, generando los valores de 20 para la función de costo y 2 para la función de hops, obteniendo así un segundo punto del frente de Pareto. Si el epsilon es igual a 1, obtendríamos infactibilidad ya que no es posible encontrar una solución en la cual el número de hops sea 1 o menor a 1. De esta manera, la idea sería poner a cambiar el epsilon hasta donde sepamos que el modelo va a ser factible.
- Dado el item anterior, ud debe implementar el método de e-Constraint de tal forma que, al ejecutarlo **UNA ÚNICA VEZ**, este solucione el modelo para varios valores de Epsilon y así arrojar la gráfica del frente óptimo de Pareto.

-El código fuente debería arrojar los mismos dos puntos (con iguales coordenadas) del frente óptimo de Pareto obtenido en “multiobjetivoHopsCosts_sumasPonderadas.py”.



Ayuda: Considerar el modelo “multiobjetivoHopsCosts_sumasPonderadas.py”, el cual itera para cambiar los pesos del método de sumas ponderadas. Usted puede inspirarse en este modelo para realizar iteraciones que cambien el Epsilon del método de eConstraint.

ENTREGABLE: El código fuente *.py con la gráfica del frente óptimo de Pareto.

EJERCICIO 2: Implementación del Algoritmo Simplex (40%)

Implemente el Algoritmo Simplex para solucionar el problema de Woodcarving:

$$\begin{aligned}
 &\max(3x_1 + 2x_2) \\
 &\text{s.a:} \\
 &2x_1 + x_2 \leq 100 \\
 &x_1 + x_2 \leq 80 \\
 &x_1 \leq 40 \\
 &x_1 \geq 0 \\
 &x_2 \geq 0
 \end{aligned}$$

Para la implementación tenga en cuenta lo siguiente:

- Los parámetros de entrada del algoritmo deberían ser las coordenadas de cada uno de los vértices del espacio de soluciones factibles. Estas coordenadas se pueden calcular manualmente para luego ser ingresadas como parámetros de entrada en la solución.
- Para realizar la prueba de optimalidad, asuma que el FEV actual inicial podría ser cualquier vértice, es decir, que se asigne aleatoriamente.
- El algoritmo debería verificar la prueba de optimalidad a partir del FEV actual inicial hasta cumplir la prueba de optimalidad, y así, ofrecer la solución del problema. En otras, palabras el algoritmo debería arrojar el valor óptimo de Z, y los valores de X1 y X2.
- Cada vez que se ejecute el algoritmo, independientemente del FEV actual inicial aleatorio, la solución arrojada **SIEMPRE** debería ser la misma.

ENTREGABLE: El código fuente en Python. Y meta código.

ENTREGABLES

Las actividades solicitadas deben ser entregadas por el estudiante teniendo en cuenta las siguientes consideraciones:

- El informe a entregar consiste en lo indicado en los entregables de cada ejercicio.
- Se puede entregar en parejas.
- Plazo de entrega: 1 semana después de la publicación de la actividad.