DEPARTAMENTO DE INGENIERÍA INDUSTRIAL
Entrenamiento 2 - IIND2103 - Principios de Optimización 2023-10 **PROFESORES:** Andrés Medaglia, Daniel Yamín, Felipe Pulido, Daniel Cuellar.**ASISTENTES:** Juan Betancourt, Ariel Rojas, Alejandro Mantilla, Laura Levy, Andrés Rueda.

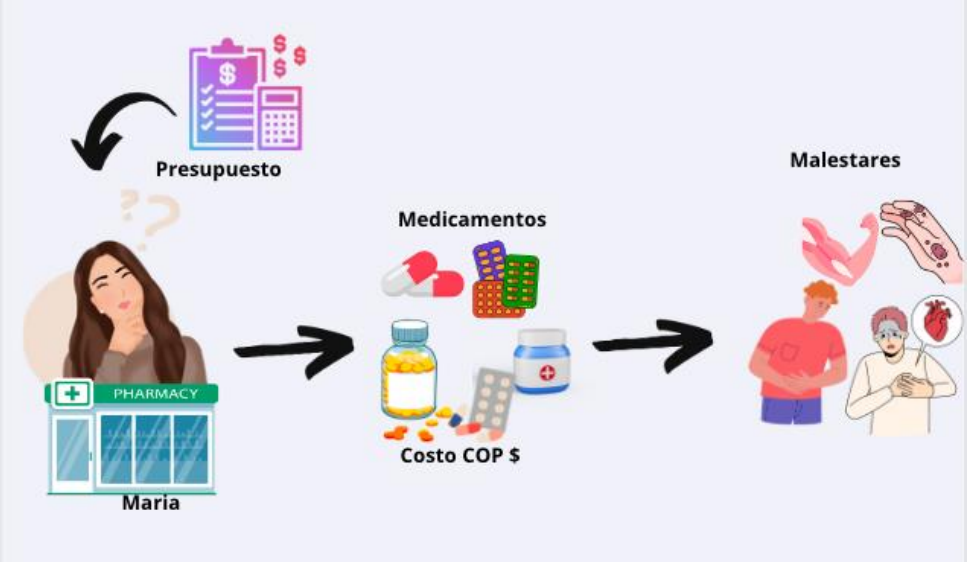
ABET Engineering Accreditation Commission

Nombre Completo	Código	Login	Sección Magistral	Sección Complementaria	Envía por Sicua Plus
Aura Lucia Martinez	202012606	Al.martinezc1	8	04	
Tomas Acosta Bernal	202011237	t.acosta	1	02	

← Inicio del texto.

Punto 1

a. Represente de forma gráfica la situación planteada. Esta representación gráfica debe ser de fácil comprensión.



b. Formule matemáticamente un modelo de optimización de forma general que represente la situación anterior. Defina clara y rigurosamente:

I. Conjuntos

$$P\text{: } Medicamentos\ i \in P$$

$$M\text{: } Malestar\ j \in M$$

II. Parámetros

$$C_i\text{: } Costo\ que\ cobra\ la\ farmacia\ por\ medicamento\ i \in P$$

$R: Restriccion\ de\ presupuesto=85000000$

III. Variables de decisión

$$x_i: \begin{cases} si\ se\ decide\ ofertar\ medicamento\ i\ \in P\ para\ el\ malestar\ j\ \in M, & 1 \\ dlc, & 0 \end{cases}$$

IV. Restricciones

1. María decidió que se deben comprar máximo dos de los medicamentos para malestares cardiovasculares

$$\sum_{i \in P} x_i \leq 2, \qquad \forall j\ \in M |\ j\ \in \{cardiovasculares\}$$

2. Se debe comprar al menos un medicamento para malestares digestivos.

$$\sum_{i \in P} xi \geq 1, \qquad \forall j\ \in M |\ M: \{Digestivos\}$$

3. En caso de comprar el Metacarbamol, se debe comprar el Ibuprofeno debido a que es un suplemento que sirve para aliviar el dolor asociado con espasmos musculares.

$$\sum_{i\ \in P|P: \{Ibuprofeno\}} x_i \geq \sum_{i\ \in P|P: \{Metacarbamol\}} x_i\ , \qquad \forall j \in M$$

4.En caso de comprar la Rifaximina o Loperamida o los dos, se debe comprar el medicamento Enterogermina, ya que este se usa cuando los trastornos intestinales causan un desequilibrio en la flora intestinal.

$$x_i(Enterogermina) \geq x_i(Rifaximina) + x_i(Loperamida), \forall j \in M$$

5.En caso de comprar el Naproxeno y Diezepam, se debe comprar Loratadina o Desloratadina (o los dos), debido a que, en la mayoría de los casos donde se indican medicamentos para músculos, el malestar está relacionado con alergias que causan estos dolores musculares.

$$x_i(Naproxeno) + x_i(Diezepam) - 1 \leq x_i(Loperamida) + x_i(Desloratdina), \forall j \in M$$

6.En caso de comprar Levocetirizina o Desloratadina, se debe comprar Naproxeno o Aspirina.

$$x_i(Levocetirizina) \leq x_i(Naproxeno) + x_i(Aspirina), \forall j \in M$$

$$(Levocetirizina) \leq x_i(Naproxeno) + x_i(Aspirina), \forall j \in M$$

7. En caso de comprar el Doxiciclina, no se debe comprar Cetirizina.

$$x(Doxiciclina) \leq 1 - x_i(Cetirizina), \forall j \in M$$

8. No puede exceder la restricción de presupuesto

$$\sum_{i \in P} xi * ci \leq R$$

9. Naturaleza de las variables

$$x_i \in \{0,1\}, \forall i \ \in P, j \in M$$

V. Función objetivo

$$Max \sum_{i \in P} xi$$

c.
El resultado de la formulación en Python nos da un estado optimo.
El objetivo debería ser de :17
Es decir,. el numero de medicamentos utilizados deben ser 17

Medicamentos que se deberían ofertar:
Metacarbamol
Ibuprofeno
Naproxeno
Diezepam
Loperamida
Losartan
Floratil
Enterogermina
Acetaminofén
Levocetirizina
Doxiciclina
Ciprofloxamina
Eritromicina
Rifaximina
Sacubitril
Albendazol
Desloratadina

El uso de esos medicamentos da un costo total de :81.300.000 lo cual respeta el presupuesto en un principio establecido de 85.000.000

Problema 2

a. Represente de forma gráfica la situación planteada. Esta representación gráfica debe ser de fácil comprensión.

b. Formule matemáticamente un modelo de optimización de forma general que represente la situación anterior.

Defina clara y rigurosamente:

I. Conjuntos

$$F: Conjunto\ de\ fincas\{1,2,3,4,5,6,7,8,9,10\}\ i\ \in F$$
$$V: Conjunto\ de\ vehiculos\{camion, carro, moto\}\ j\ \in V$$

II. Parámetros

$$k_i: Capacidad\ de\ produccion\ de\ la\ finca\ i\ \in F\ [Kg]$$
t: limite de vehículos
$$c_i: Costo\ de\ adecuacion\ de\ la\ finca\ i\ \in F\ [COP]$$
$$d_i: Distancia\ de\ finca\ i \epsilon F\ al\ centro\ de\ distribucion\ [km]$$
$$s_j: Capacidad\ de\ cada\ vehiculo\ j\ \in V\ [Kg]$$
$$a_j: Alquiler\ de\ vehiculo\ j\ \in V\ [COP]$$
$$g_j: Consumo\ de\ un\ vehiculo\ j\ \in V\ \left[\frac{L}{Km}\right]$$
$$m: monto\ maximo\ adecuacion\ fincas\ [COP]$$
$$l: Limite\ minimo\ de\ cacao\ proveniente\ de\ fincas\ productoras\ [Kg]$$
$$p: presupuesto\ de\ adecuacion\ de\ fincas = 55$$
$$e: precio\ gasolina\ \frac{COP}{L} = 2500$$

III. Variables de decisión

$$\mathbf{x_i} = \begin{cases} \mathbf{1\ si\ se\ escoge\ la\ finca\ i \in F} \\ \mathbf{0\ d.l.c} \end{cases}$$

$$y_{ij}: Cantidad\ vehiculos\ j\ \in V\ \ usados\ en\ la\ finca\ i\ \in F$$

IV. Restricciones

a. No es posible enviar más de 1 camión, 2 carros y 4 motos a una finca. Las Fincas 2,8 y 10 no cuentan con carreteras de acceso. Solo es posible enviar carros y motos.

$$y_{ij} \leq 1, \qquad \forall i\ \in F | F \notin \{2,8,10\}, j \in V | V: \{Camion\}$$
$$y_{ij} \leq 2, \qquad \forall i\ \in F, j \in V | V: \{Carros\}$$

$$y_{ij} \leq 4, \qquad \forall i \in F, j \in V | V : \{Motos\}$$

b. No superar el presupuesto

$$\sum_{i \in F} c_i xi \leq p$$

c. Se debe cumplir un límite mínimo de cacao producido desde las fincas.

$$\sum_{\boldsymbol{j \in V}} \sum_{\boldsymbol{i \in F}} \boldsymbol{s_j y_{ji}} \geq \boldsymbol{l}$$

d. No se puede transportar mas de lo producido por una fina

$$\sum_{\boldsymbol{j \in V}} \boldsymbol{s_j y_{ji}} \leq \boldsymbol{k_i x_i} \; \forall \; \boldsymbol{i} \in \; \boldsymbol{F}$$

e. enviar la cantidad determinada de vehículos a una finca

$$zij \leq (\boldsymbol{v(j)} * \boldsymbol{x(i)} \forall \; \boldsymbol{i} \in \; \boldsymbol{F}, \boldsymbol{j} \in \boldsymbol{V}$$

f. Naturaleza de las variables

$$\boldsymbol{x_i} \in \{\mathbf{0}, \mathbf{1}\} \; \forall \; \boldsymbol{i} \in \boldsymbol{F}$$
$$\boldsymbol{y_{ji}} \in \mathbf{Z}^{+} \; \forall \; \boldsymbol{i} \in \boldsymbol{F}, \boldsymbol{j} \in \boldsymbol{V}$$

V. Función objetivo

$$Min \sum_{j \in V} \sum_{i \in F} a_j y_{ij} + \sum_{i \in F} \sum_{j \in V} e(d_i g_j y_{ji})$$

Presentar en el informe PDF un gráfico, diagrama y/o tabla de la solución encontrada en el inciso c. En esta(s) deberá reportar el valor de su función objetivo, las fincas seleccionadas, la cantidad total de cacao que se recoge y el número de vehículos enviados a cada una de las fincas.

El valor optimo es de 26.317.950 lo cual representa el costo de alquiler y funcionamiento

Finca	Vehículo	Cantidad Vehículos	Cantidad cacao
1	camion	1	200
	moto	3	675
	carro	2	450
3	camion	1	200
	moto	3	550
	carro	1	325
4	camion	0	0
	moto	0	0
	carro	0	0
5	camion	1	200

	moto	1	525
	carro	2	450
6	camion	1	200
	moto	4	750
	carro	2	450
7	camion	1	200
	moto	4	750
	carro	2	450
9	camion	1	200
	moto	4	750
	Carro	2	450

Problema 3.

a. Represente de forma gráfica la situación planteada. Esta representación gráfica debe ser de fácil comprensión.
b. Formule matemáticamente un modelo de optimización de forma general que represente la situación anterior.

Defina clara y rigurosamente:

I. Conjuntos

$$G: Grupos\ de\ investigacion\{CUPA, PALO, Logistics, Biotec, Inestiobras, AlPesIA\}\ i \in G$$

$$D: Dias\ de\ la\ semana\ excepto\ domingo\ j\ \in D$$

$$H: Franja\ horaria\ k\ \in H$$

subconjunto:

I: intervalo

II. Parámetros

$$d_i: Duracion\ de\ la\ franja\ horaria\ h\ \in H\ en\ cad\ grupo\ dde\ investigacion\ i \in G$$
$$s_i: Sesiones\ a\ la\ semana\ del\ grupo\ de\ investigacion\ i\ \in G\ en\ dias\ j\ \in D$$

III. Variables de decisión

$$x_{ih}: \begin{cases} Si\ \ el\ grupo\ i\ \in G\ esta\ en\ la\ frna\ h\ \in H, & 1 \\ dlc, & 0 \end{cases}$$

$$y_{ih}: \begin{cases} Si\ \ el\ grupo\ i\ \in G\ se\ presenta\ en\ la\ frna\ h\ \in H,, & 1 \\ dlc, & 0 \end{cases}$$

$$Z_{ih}: \begin{cases} Si\ \ el\ grupo\ i\ \in G\ termina\ en\ la\ frna\ h\ \in H,, & 1 \\ dlc, & 0 \end{cases}$$

$$W_{ih}: Horario\ mayor$$

IV. Restricciones

a. No halla mas de un grupo en una franja

$$, \quad \sum_{k \in H} x_{gk} \leq 3 \, \forall \, i \in G$$
$$\sum_{k \in H} x_{gk} \leq 1 \, \forall \, i \in G, j \in D$$

b. cantidad de franjas igual a sesiones requeridas

$$\sum_{k \in H} x_{ik} = si * di, \qquad \forall i \in G$$

c. cada vez que un grupo termina una sesión debe ser igual a las sesiones que requieren

$$\sum_{k}^{t+d_i-1} y_{ik} \leq 1, \qquad \forall i \in G, k \in H$$
$$\sum_{k \in H} Z_{ik} = si * di, \qquad \forall i \in G$$

d. veces que inicia la sesión sea igual a las sesiones requeridas

$$\sum_{k \in H} z_{ik} * H - \sum_{k \in H} y_{ik} * H = si * (di - 1), \qquad \forall i \in G$$

$$\sum_{k}^{t+d_i-1} x_{ik} \geq (di * yik), \qquad \forall i \in G, k \in H \perp k + di - 1 \leq 5$$

$$\sum_{k}^{t+d_i-1} x_{ik} \geq (di * zik), \qquad \forall i \in G, k \in H \perp k + di - 1 \leq 5$$

$$wik \geq Ii * zik, \qquad \forall i \in G, k \in H$$

$$yik = 0, \qquad \forall i \in G, k \in H \perp k + di - 1 \leq 5$$

e. Naturaleza de la variables:

$$xik \in \{0,1\}, k \in H$$
$$yik \in \{0,1\}, k \in H$$
$$zik \in \{0,1\}, k \in H$$
$$w \in z + \forall \, k \in H$$

V. Función objetivo

$$\min W$$

c. valor de su función objetivo, la planeación de los horarios y el programa de cada grupo de investigación

Función objetivo: 4

Dias	CUPA	PALO	Logistics	BioTech	InvestiObras	AlpesIA
1		PALO	Logistics			
2			Logistics	BioTech		
3	CUPA	PALO				
4			Logistics		InvestiObras	
5	CUPA					AlpesIA
6	CUPA	PALO				

Cada una de las columnas es la suma de los grupos en la semana, por lo tanto: El grupo cupa se presenta 3 veces a la semana en los días de la semana 3,5,6. Esto aplica para cada uno de los otros grupos.

d. Implementación

e.

Función objetivo: 24

Para modelar la situación se debe agregar una nueva restricción al modelo para asegurarse de que las sesiones de investigación finalicen lo antes posible en la semana. Para esto, se debe agregar una nueva variable binaria que indique si la última sesión del grupo i termina en la franja horaria h. Esta variable se puede llamar "t". Luego, se debe agregar una restricción que indique que la última sesión del grupo i debe terminar en la menor franja horaria posible en la semana. Esto se puede expresar como:

$$w >= t[i,k] * franjas\_semana\_j[k]$$

Donde w es la variable que representa la mayor franja horaria en la que termina alguna sesión de investigación en la semana, t[i,k] es la variable binaria que indica si la última sesión del grupo i termina en la franja horaria k, y franjas_semana_j[k] es el número de franjas horarias en el día j de la semana.

Además, se debe asegurar que todas las sesiones de investigación finalicen antes de que los estudiantes puedan disponer de la sala de cómputo en tiempo continuo. Esto se puede expresar como:

$$lp.lpSum(z[i,k]*k \text{ for } i \text{ in } G \text{ for } k \text{ in } H) <= w$$

Donde z[i,k] es la variable binaria que indica si la sesión del grupo i termina en la franja horaria k.

Dias	CUPA	PALO	Logistics	BioTech	InvestiObras	AlpesIA
1	CUPA		Logistics			
2	CUPA		Logistics			AlpesIA
3	CUPA	PALO				
4		PALO		BioTech		
5		PALO			InvestiObras	
6						

Punto 4.

Formule matemáticamente y de forma explícita el problema de optimización planteado. Defina claramente las variables de decisión, la función objetivo y las restricciones. Tenga en cuenta que en este modelo particular NO necesariamente se cumplen todos los supuestos de optimización lineal que se tratan en el curso

Variables de decisión:

X

Y

Parametros:

$$\lambda: \boldsymbol{distancia\ que\ se\ va\ a\ recorrer\ para\ cada\ variable}$$

Restricciones:

a. La variable y debe se menor o igual a 6

$$\boldsymbol{y \leq 6}$$

b. La variable x debe ser menor o igual a 6

$$\boldsymbol{x \leq 6}$$

c. La variable y debe ser mayor o igual a -6

$$\boldsymbol{y \geq -6}$$

d. La variable x+y debe ser igual a -6

$$\boldsymbol{x + y = -6}$$

e. La variable x debe ser mayor o igual a -6

$$\boldsymbol{x \geq -6}$$

Funcion objetivo:

$$\boldsymbol{min\ x + y}$$

b.

```python
def evaluar_indice(coord: tuple)-> int:

    indice = 100 * (coord[0]**2 + coord[1] - 11) **2  + 100 *(coord[0] + coord[1]**2  -7)**2
    return indice
```

c.

```python
def generar_radar(coord:tuple, l: float) -> list:
    #Restriccion de Arriba     #Restriccion Derecha  Restriccion Izq #Restriccion Abajo

        radar = []

        #Arriba Izquierda
        AI = []
        AI.append(coord[0]-l)
        AI.append(coord[1]+l)
        radar.append(tuple(AI))

        #Arriba
        A = []
        A.append(coord[0])
        A.append(coord[1]+l)
        radar.append(tuple(A))

        #Arriba Derecha

        AD = []
        AD.append(coord[0]+l)
        AD.append(coord[1]+l)
        radar.append(tuple(AD))

        #Derecha

        D = []
        D.append
        D.append(coord[0]+l)
        D.append(coord[1])
        radar.append(tuple(D))

        #Abajo Derecha

        ABD = []
        ABD.append(coord[0]+l)
        ABD.append(coord[1]-l)
        radar.append(tuple(ABD))

        #Abajo

        AB = []
        AB.append(coord[0])
        AB.append(coord[1]-l)
        radar.append(tuple(AB))

        #Abajo Izquierda

        ABI = []
        ABI.append(coord[0]-l)
        ABI.append(coord[1]-l)
        radar.append(tuple(ABI))

        #Izquierda

        I = []
        I.append(coord[0]-l)
        I.append(coord[1])
        radar.append(tuple(I))

        return radar
```

d.

```python
def evaluar_factibilidad(coord: tuple):

    if coord[1] > 6.0 or coord[0] > 6.0 or coord[0] < -6.0 or coord[1]  < -6.0 or coord[0]+coord[1] <
-6.0 :
        return False
    else:
        return True
```

e.

```python
def encontrar_mejor_coor(dic:dict)->tuple:

    # Creo 2 listas en la lista pongo todos los valores del diccionario
    lista =[]
    #En place voy a añadir las soluciones optimas y luego lo vuelvo una tupla
    place = []
    #El primer for recorre los valores y los añade a lista
    for v in dic.values():
        lista.append(v)
        #sort los organiza de menor a mayor entonces no toca revisar uno por uno
    lista.sort()
    #Hago otro for para revisar cual llave esta asignada al valor menor del diccionario
    key = next(key for key, value in dic.items() if value == lista[0])
    # Añado a la solucion optima la llave y el indice
    place.append(key)
    place.append(lista[0])
    # y ya xd

    return tuple(place)
```

f.

La función "busqueda_local" es una función que se utiliza para mejorar la solución encontrada por el algoritmo de búsqueda de vecindario en el problema de optimización que se está resolviendo. Esta función toma como entrada una solución actual y busca otras soluciones en su vecindario, evaluándolas y seleccionando la mejor solución encontrada en el vecindario.

Los parámetros de entrada de la función "busqueda_local" son la solución actual que se desea mejorar (representada como una lista de índices) y el número de iteraciones que se realizarán para buscar soluciones en el vecindario. La salida de esta función es una tupla que contiene la mejor solución encontrada (representada como una lista de índices) y su evaluación correspondiente.

La función "busqueda_local" se relaciona con las funciones implementadas previamente de varias maneras. En primer lugar, utiliza la función "evaluar_indice" para evaluar la solución actual y las soluciones encontradas en el vecindario. En segundo lugar, utiliza la función "generar_radar" para generar soluciones en el vecindario de la solución actual. En tercer lugar, utiliza la función "evaluar_factibilidad" para verificar si las soluciones encontradas en el vecindario son factibles. Finalmente, utiliza la función "encontrar_mejor_coor" para seleccionar la mejor solución encontrada en el vecindario. De esta manera, la función "busqueda_local" aprovecha las funciones implementadas previamente para mejorar la solución encontrada por el algoritmo de búsqueda de vecindario.

g.

Para responder la pregunta, se deben invocar la función busqueda_local tres veces, con valores diferentes de la longitud de paso lbd. En particular, se debe utilizar lbd = 0.5, lbd = 0.1 y lbd = 0.01.
Al hacer esto, se puede observar que el valor de la longitud de paso afecta significativamente el resultado de la búsqueda local. Con una longitud de paso grande (lbd = 0.5), la búsqueda puede ser muy ineficiente, ya que se salta muchas posibles coordenadas óptimas. Por otro lado, con una longitud de paso demasiado pequeña (lbd = 0.01), se pueden explorar demasiadas coordenadas, lo que puede resultar en una búsqueda demasiado costosa computacionalmente.
En general, se recomienda utilizar una longitud de paso que sea lo suficientemente grande como para explorar las coordenadas vecinas relevantes de manera eficiente, pero no tan grande como para saltar posibles soluciones óptimas. Una buena práctica es experimentar con diferentes valores de longitud de paso y observar cómo afecta el rendimiento de la búsqueda.
Para ubicar el campamento del equipo de investigación, se recomienda invocar la función

busqueda_local con diferentes valores de lbd y observar la ubicación óptima encontrada. En general, se espera que la ubicación óptima se encuentre en una región cercana a la coordenada de la entrada occidental del parque (−6.0,2.0), y que cumpla todas las restricciones de factibilidad.
Por lo tanto, mi recomendación para el equipo de investigación es que utilicen una longitud de búsqueda pequeña pero no demasiado pequeña, como λ = 0.1, para obtener una ubicación más precisa del campamento y que no deba revisar demasiadas coordenadas. Esto lo acerca a un resultado cercano al mas preciso sin tener que buscar en todas las posibles ubicaciones.

Longitug	Corrdenada inicial (x,y)	Numero de coordenadas visitadas	Mejor ubicación encontrada	Indice de la mejor ubicación encontrada
0.5	-6.2	7	-3.3	200
0.1	-6.2	33	-2.8,3.1	3.97
0.01	-6.2	320	-2.81,3.13	0.08528

Sin embargo, si se cuenta con un tiempo ilimitado sin tener problema sobre revisar todas las ubicaciones posibles, les recomendaríamos la longitud de 0.01 ya que es la mas precisa.

h.

Longitud Lambda	Coordenada	Numero de Coordenadas	Mejor Ubicación	Indice Mejor Ubicación
0.01	(-6.0, 2.0)	320	(-2.81, 3.13)	0.08528
0.01	(3.5, 6.0)	401	(3.0, 2.0)	0.0
0.01	(6.0, 0.0)	243	(3.58, -1.85)	0.11352
0.01	(0.0, -6.0)	334	(-3.32, -2.67)	1719.2905

Coordenadas:

(0,-6)

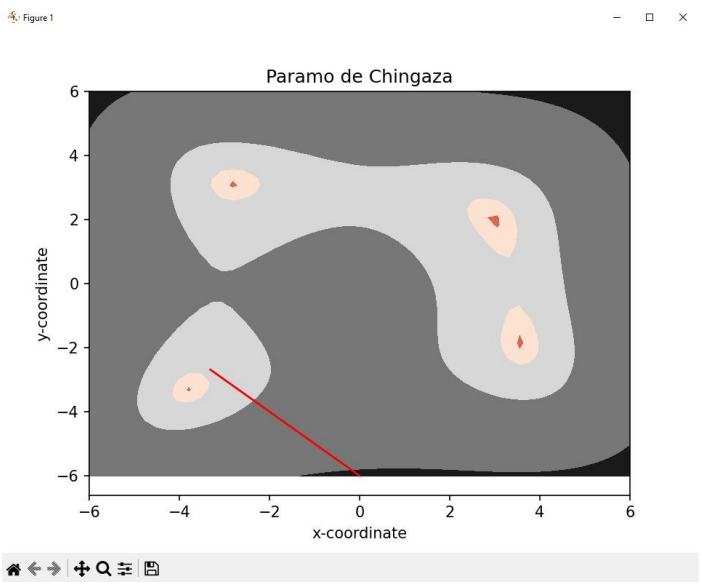


Figure 1
Paramo de Chingaza
y-coordinate
x-coordinate
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6

(-6,2):

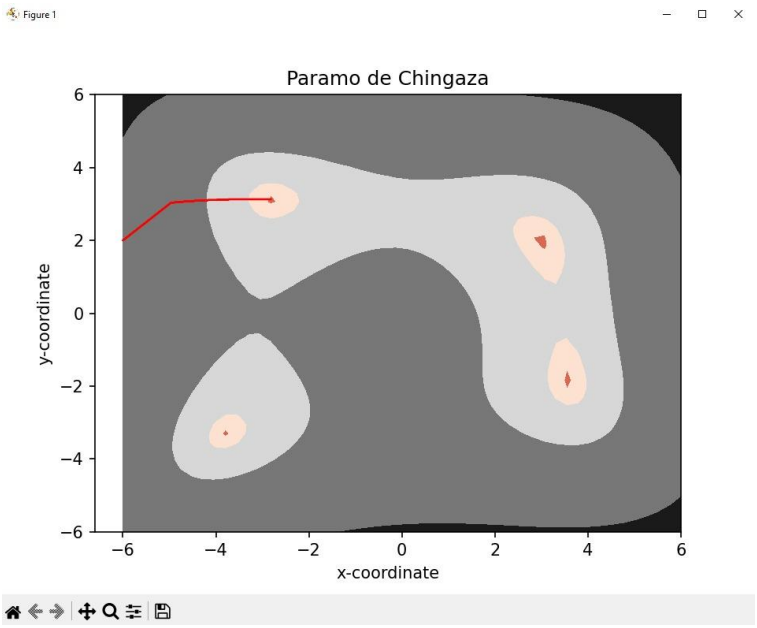


Figure 1
Paramo de Chingaza
y-coordinate
x-coordinate
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6

(3.5,6.0)

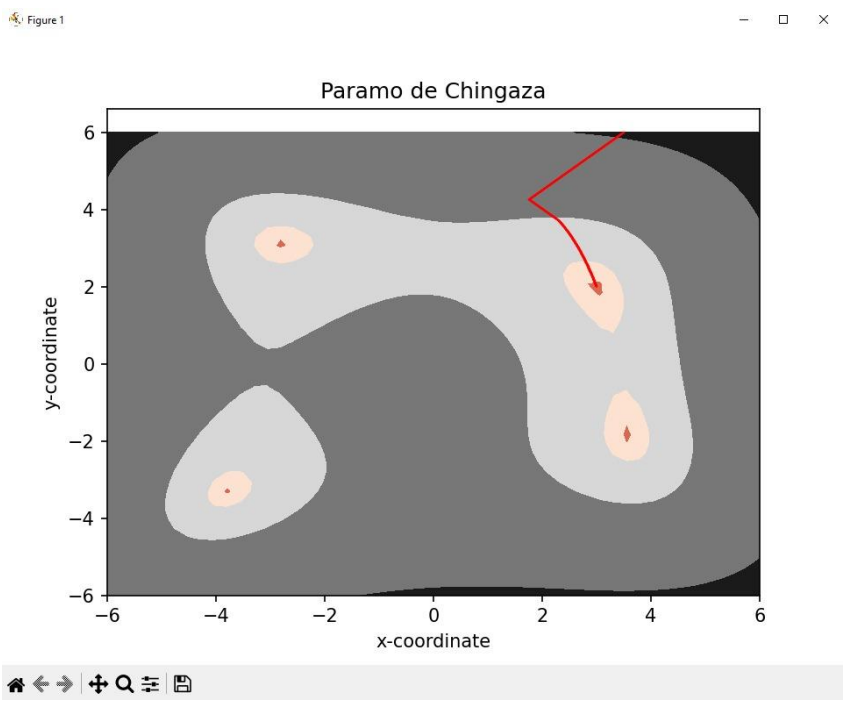


Figure 1
Paramo de Chingaza
y-coordinate
x-coordinate
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6

(6.0):

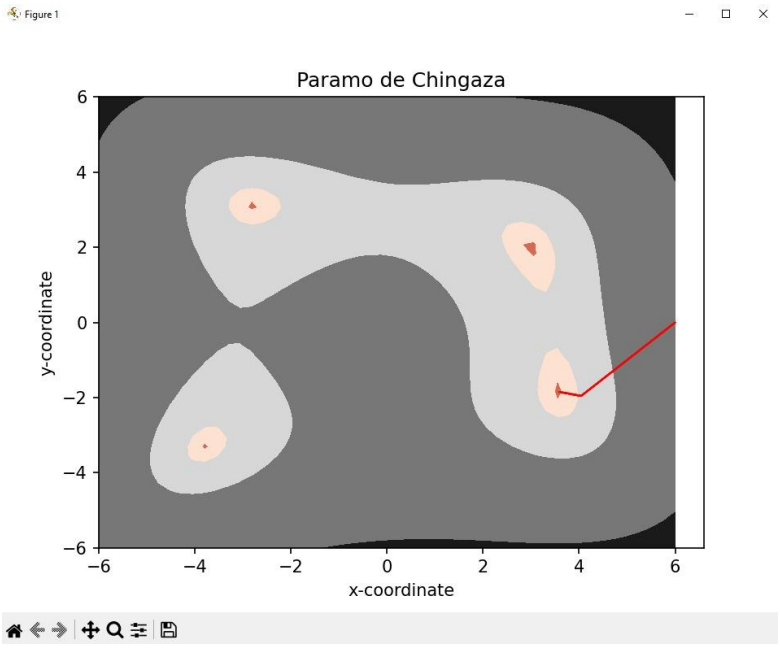


Figure 1
Paramo de Chingaza
y-coordinate
x-coordinate
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6