**DEPARTAMENTO DE INGENIERÍA INDUSTRIAL**
**Entrenamiento 2** - IIND2103 - Principios de Optimización 2021-10
**PROFESORES:** Andrés Medaglia, Camilo Gómez, Alfaima Solano, Daniel Cuellar.
**ASISTENTES:** Johan Camacho, Ariadna De Ávila, Alejandro Mantilla, Freddy Orozco, Felipe Pulido, Daniel Yamin.

| Nombre Completo | Código | Login | Sección Magistral | Sección Complementaria | Envía por Sicua Plus |
|---|---|---|---|---|---|
| Mariana Forero Avila | 201922249 | m.foreroa | 05 | 03 | |
| Tomás Langebaek Carrizosa | 201732744 | t.langebaek | 05 | 06 | X |

**Problema 1:**

a. Formulación

i. **Conjuntos:**

$Nadadores = N.\ n \in N \mid N = \{\text{Alejandro, Alfaima, Andrés, Ariadna, Benji, Camila, Camilo G, César, Cristian, Daniel C, Daniel Y, Diana, Felipe, Freddy, Jimmy, Johan, Juan Diego, Juan E, Juliana, Lucia, María Paulina, Santiago, Saúl, Sofía, Valentina, Vivian}\}$

ii. **Parámetros:**

$p = 85.000.000 = Presupuesto\ total\ [COP]$

$d_n = Desempeño\ de\ cada\ nadador\ n \in N.$

$c_n = Costo\ de\ cada\ nadador\ n \in N.$

iii. **Variables de decisión:**

$x_n = 1\ si\ se\ escoge\ al\ nadador\ n \in N, 0\ de\ lo\ contrario$

iv. **Restricciones:**

En caso de escoger a Ariadna, debe escoger a Felipe.

$$N_{Ariadna} = N_{Felipe}$$

En caso de escoger a Freddy o a Alejandro (o los dos), debe escoger a Daniel C.

$$\frac{(N_{Freddy} + N_{Alejandro})}{2} \leq N_{Daniel\ C}$$

Se deben seleccionar máximo dos de los nadadores estándar: Juan Diego, Juan E, Juliana, Sofía o Valentina.

$$N_{\text{Juan Diego}} + N_{\text{Juan E}} + N_{Juliana} + N_{Sofía} + N_{Valentina} < 3$$

En caso de escoger a Camila y Jimmy, se debe escoger a Diana o Alfaima (o las dos).

$$N_{\text{Camila}} + N_{\text{Jimmy}} \leq N_{\text{Diana}} + N_{\text{Alfaima}}$$

En caso de escoger a Johan o Daniel Y, se debe escoger a Andrés y a Lucia.

$$\frac{N_{\text{Johan}} + N_{\text{Daniel Y}}}{2} \leq N_{Andres}$$

$$\frac{N_{\text{Johan}} + N_{\text{Daniel Y}}}{2} \leq N_{\text{Lucia}}$$

Se debe escoger al menos un nadador especial: Vivian, Cristian, María Paulina, Saúl.

$$N_{\text{Vivian}} + N_{\text{Cristian}} + N_{Maria\ Paula} + N_{Saúl} \geq 1$$

En caso de escoger a Santiago, no se debe escoger a Benji.

$$N_{Santiago} + N_{Benji} \leq 1$$

No se debe superar el presupuesto:

$$\sum_{n \in N} (x_n * c_n) \leq p$$

Naturaleza:

$$x_n \in \{0, 1\} \ \forall \ n \in N$$

**v. Función objetivo:**

$$Max\ Z = \sum_{n \in N} (x_n * d_n)$$

b. Python

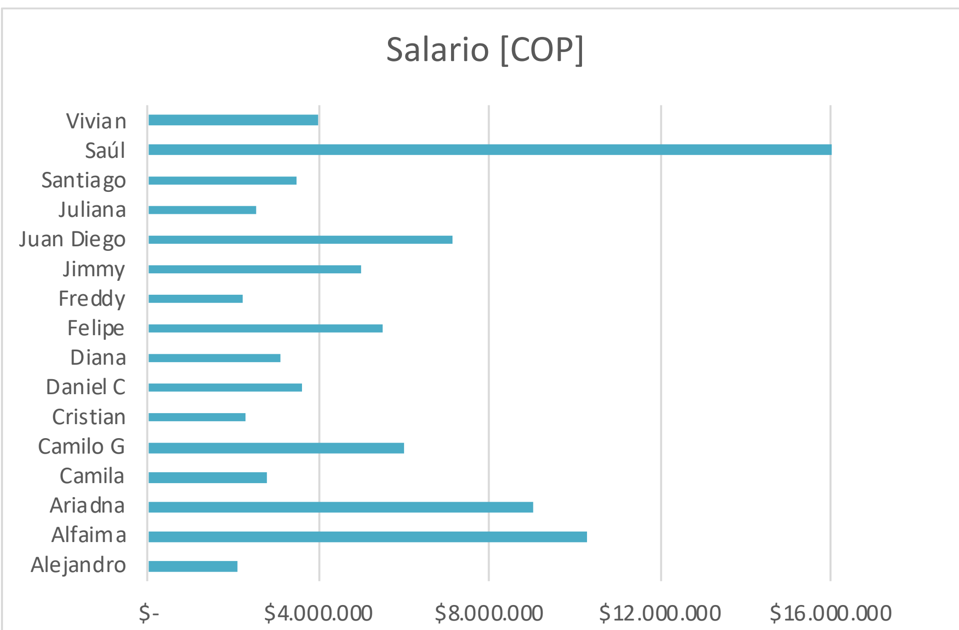
Para llegar al resultado final se implementó el modelo de optimización en Python. Esta implementación está incluida en los archivos adjuntos a este documento.

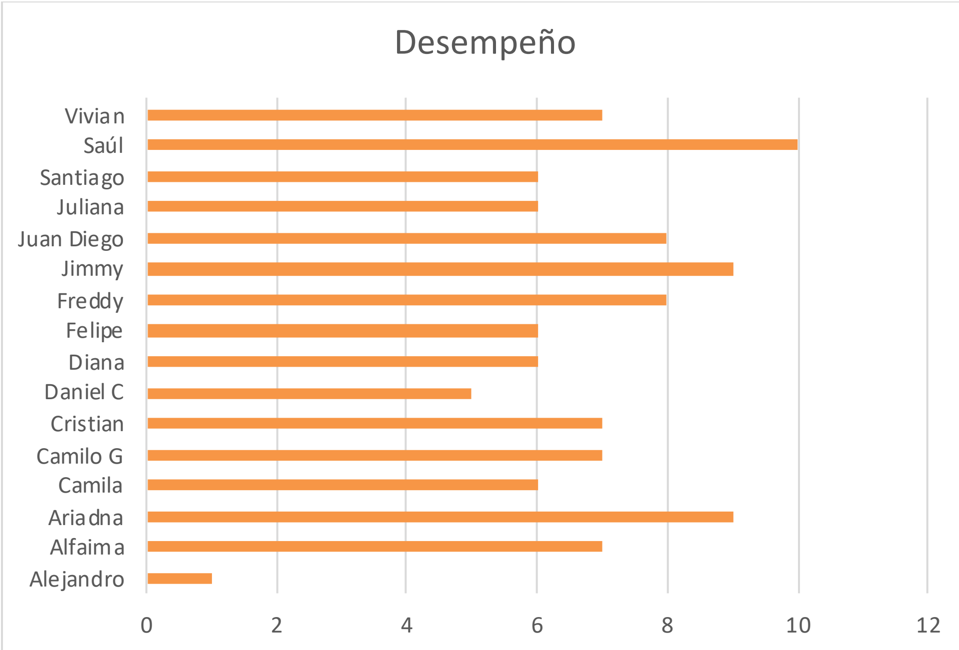
c. Resultados y Gráficas

Desempeño total: 108

Costo total: $ 84.970.000

| Equipo seleccionado | | |
|---|---|---|
| **Nadador** | **Salario [COP]** | **Desempeño** |
| Alejandro | $ 2.100.000 | 1 |
| Alfaima | $ 10.300.000 | 7 |
| Ariadna | $ 9.000.000 | 9 |
| Camila | $ 2.750.000 | 6 |
| Camilo G | $ 6.000.000 | 7 |
| Cristian | $ 2.300.000 | 7 |
| Daniel C | $ 3.600.000 | 5 |
| Diana | $ 3.100.000 | 6 |
| Felipe | $ 5.490.000 | 6 |
| Freddy | $ 2.230.000 | 8 |
| Jimmy | $ 5.000.000 | 9 |
| Juan Diego | $ 7.100.000 | 8 |
| Juliana | $ 2.550.000 | 6 |
| Santiago | $ 3.450.000 | 6 |
| Saúl | $ 16.000.000 | 10 |
| Vivian | $ 4.000.000 | 7 |






**A partir de los resultados se pudo observar que se obtuvo un desempeño de 108 entre los 16 miembros del equipo, utilizando el 99,96% del presupuesto permitido. También se pudo observar que hay una correlación entre precio y desempeño de manera que la mayoría de los nadadores que aportan más al desempeño tienden a tener un precio mas elevado.**

**Problema 2:**

a. Formulación

### i. Conjuntos:

$$Locales = L.\ l \in L \mid L = \{\text{Chapinero, La Candelaria, Av 7, Av Jiménez}\}$$

$$Universidades = U.\ u \in U \mid U = \{Javeriana, Rosario, Tadeo, Central\}$$

### ii. Parámetros:

$$p_l = Cantidad\ maxima\ de\ panaderos\ un\ local\ l \in L$$

$$m_l = Costo\ diario\ de\ mantenimiento\ [COP]\ en\ un\ local\ l \in L.$$

$$e_{l,u} = Costo\ de\ envio\ de\ un\ local\ l \in L\ a\ una\ universidad\ u \in U\ .$$

$$s = 90.000 = Salario\ diario\ de\ cada\ trabajador\ [COP].$$

$$k = 300 = Producción\ diaria\ de\ un\ panadero\ [Productos\ de\ panaderia].$$

$$d_u = Demanda\ diaria\ en\ una\ Univerisdad\ u \in U\ .$$

**Variables de decisión:**

$$x_l = 1 \; si \; se \; utiliza \; el \; local \; l \in L, 0 \; de \; lo \; contrario$$

$$y_l = numero \; de \; panaderos \; en \; el \; local \; l \in L$$

$$z_{l,u} = cantidad \; de \; productos \; enviados \; del \; local \; l \in L \; a \; la \; universidad \; u \in U$$

iii. **Restricciones:**

Cantidad máxima de empleados por local

$$y_l \leq p_l * \; x_l \; \forall \; l \in L$$

Producción diaria de un panadero

$$y_l * k \geq \sum_{u \in U} z_{l,u} \; \forall \; l \in L$$

Satisfacer demanda:

$$\sum_{l \in L} z_{l,u} \geq d_u \; \forall \; u \in U$$

Naturaleza:

$$x_l \in \{0, 1\} \; \forall \; l \in L$$

$$y_l \geq 0 \; \forall \; l \in L$$

$$z_{l,u} \geq 0 \; \forall \; l \in L, u \; \in U$$

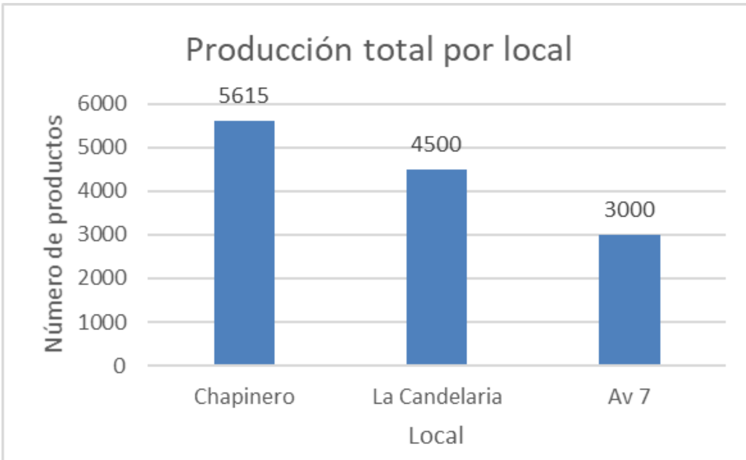
iv. **Función objetivo:**

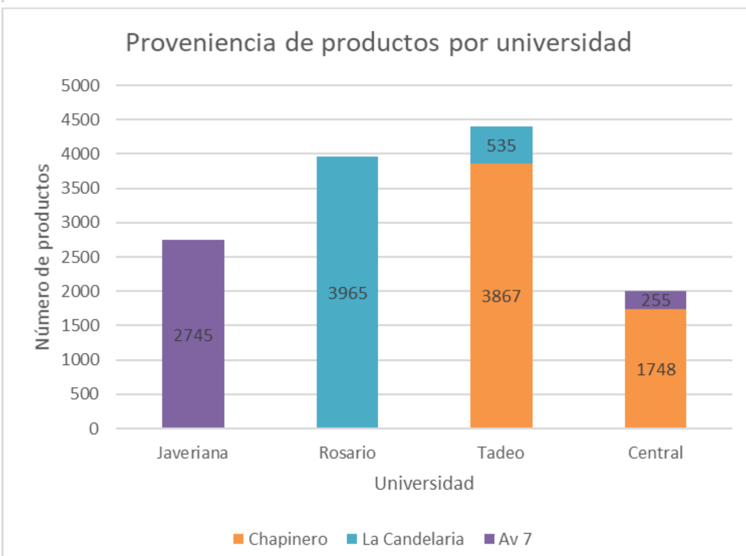
$$Min \; Z = \sum_{l \in L} (x_l * m_l) + \sum_{l \in L} \left( y_l * s \right) + \sum_{l \in L} \sum_{u \in U} \left( z_{l,u} * e_{i,u} \right)$$

b. Python
Para llegar al resultado final se implementó el modelo de optimización en Python. Esta implementación está incluida en los archivos adjuntos a este documento.

c. Resultados y Gráficas:
Costo total**: $ 4.681.198 COP**

| | | | Productos enviados a cada universidad | | | | |
|---|---|---|---|---|---|---|---|
| | **Compra** | **Panaderos** | **Javeriana** | **Rosario** | **Tadeo** | **Central** | **Producción Total** |
| **Chapinero** | Sí | 19 | - | - | 3867 | 1748 | 5615 |
| **La Candelaria** | Sí | 15 | - | 3965 | 535 | - | 4500 |
| **Av 7** | Sí | 10 | 2745 | 0 | - | 255 | 3000 |
| **Av Jiménez** | No | - | - | - | - | - | - |
| Total | | 44 | 2745 | 3965 | 4402 | 2003 | |




A partir de los resultados se puede observar que se compraron todos los locales excepto el de la Avenida Jiménez, y que la mayoría de productos provienen del local de Chapinero que suple casi por completo la demanda de la Tadeo y de la Central. El local de la avenida séptima suple toda la demanda de la Javeriana y un 12,7% de la de la Central, y el local del la Candelaria suple la demanda del Rosario en su totalidad y un 12,15% de la de la Tadeo. Por otro lado en los locales de la Candelaria y de la Av 7 se decidió contratar todos los panaderos posibles, y en el local de Chapinero uno menos de los permitidos.

**Problema 3:**

**a.**

### i. Conjuntos:

$$Peliculas = M.\ m \in M \mid M = \{\text{Titanic, Inception, Pulp Fiction, Fight club}\}$$

$$Parqueaderos = P.\ p \in P \mid P = \{Oro, Platino, Plata, Bronce, VIP\}$$

$$Franjas = F.\ f \in F \mid F = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$$

### ii. Parámetros:

$$d_m = Duración\ de\ la\ pelicula\ m \in M$$

$$n_m = Número\ minimo\ de\ proyecciones\ diarias\ de\ la\ pelicula\ m \in M.$$

$$a_{m,f} = Asistencia\ para\ la\ película\ m \in M\ en\ la\ franja\ f \in F$$

### Variables de decisión:

$$x_{m,p,f} = 1\ si\ se\ proyecta\ la\ pelicula\ m \in M\ en\ el\ parqueadero\ p \in P\ en\ la\ franja\ f \in F, 0\ de\ lo\ contrario.$$

$$y_{m,p,f} = 1\ si\ se\ inicia\ la\ proyección\ de\ la\ pelicula\ m \in M\ en\ el\ parqueadero\ p \in P\ en\ la\ franja\ f \in F, 0\ de\ lo\ contrario.$$

### iii. Restricciones:

Garantizar que en una sala solo se esté proyectando una película a la vez

$$\sum_{m \in M} x_{m,p,f} \leq 1\ \forall\ p \in P, f \in F$$

Garantizar que una película se proyecte al menos una cantidad especifica de veces durante el día

$$\sum_{p \in P} \sum_{f \in F} y_{m,p,f} \geq n_m\ \forall\ m \in M$$

Garantizar que no se inicie la proyección de una película en más

de un parqueadero al tiempo

$$\sum_{p \in P} y_{m,p,f} \leq 1 \;\forall\, f \in F, m \in M$$

Duración de una película

$$\sum_{f \in F} x_{m,p,f} = \sum_{f \in F} (d_m * y_{m,p,f}) \;\forall m \in M, p \in P$$

Garantiza que la reproducción de la pelicula sea continua

$$\sum_{i=f}^{d_m+f-1} x_{m,p,i} \geq d_m * y_{m,p,f} \;\forall\, m \in M, p \in P, f \in F | f + d_m - 1 \leq |F|$$

Garantiza que la película se empieza una vez en la franja de se reproduce

$$\sum_{i=f}^{d_m+f-1} y_{m,p,i} \leq 1 \;\forall\, m \in M, p \in P, f \in F | f + d_m - 1 \leq |F|$$

Garantiza que no inicie la reproducción de una película si no se puede terminar

$$y_{m,p,f} = 0, \forall\, m \in M, p \in P, f \in F | f + d_m - 1 > |F|$$

Naturaleza:

$$x_{m,p,f}, y_{m,p,f} \in \{0,1\} \;\forall\, l \in L$$

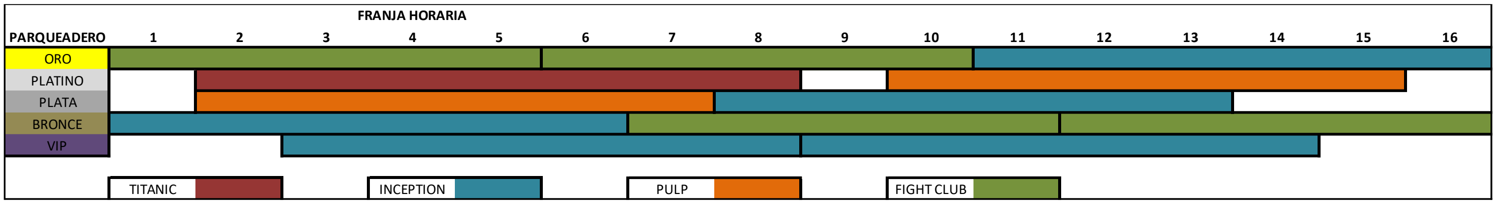
**iv. Función objetivo:**

$$Max\ Z = \sum_{m \in M} \sum_{p \in P} \sum_{f \in F} \left(y_{m,p,f} * a_{m,f}\right)$$

b. Python

Para llegar al resultado final se implementó el modelo de optimización en Python. Esta implementación está incluida en los archivos adjuntos a este documento.

c. Resultados y Gráficas




Arriba se muestra el resultado del calendario óptimo para reproducir las películas en cada una de las salas. Cada película se reprodujo exactamente el mínimo de veces requeridas. Fight Club se reprodujo 2 veces seguidas en la sala Oro y al igual que en la sala bronce, solo que en franjas diferentes para cumplir la restricción de que una misma película no puede iniciar al mismo tiempo en mas de un parqueadero. Inception se reprodujo en casi todas las salas excepto en la Platino y en la sala bronce se reprodujo 2 veces seguidas. Titanic solo se reprodujo una vez en la sala platino y Pulp se reprodujo una vez en la sala platino y otra en la sala Plata. Ni en la sala Oro ni en la sala bronce quedo ninguna franja libre, y se reprodujeron tres películas en total. En el resto de las salsas solo se reprodujeron 2 películas. Que Inception se reprodujera multiples veces en diferentes horarios señala que es la película más rentable todas.

Número total de asistentes: **1418** personas

## Problema 4:

**a)**

$$3(1-x)^2 e^{x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 + y^5\right) e^{-x^2-y^2} - \frac{1}{5} e^{-(x+1)^2-y^2}$$

**Punto 0:**

$$\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}$$

$$3(1-1)^2 e^{1^2-(0+1)^2} - 10\left(\frac{1}{5} - 1^3 + 0^5\right) e^{-1^2-0^2} - \frac{1}{5} e^{-(1+1)^2-0^2}$$

$$= 10\left(\frac{1}{5} - 1\right) e^{-1} - \frac{1}{5} e^{-(2)^2}$$

$$= 2.939$$

**Punto 1:**

$$\begin{bmatrix} \mathbf{2} \\ \mathbf{0} \end{bmatrix}$$

$$3(1-2)^2 e^{2^2-(0+1)^2} - 10\left(\frac{2}{5} - 2^3 + 0^5\right) e^{-2^2-0^2} - \frac{1}{5} e^{-(2+1)^2-0^2}$$

$$= 3(1-2)^2 e^{2^2-(1)^2} - 10\left(\frac{2}{5} - 2^3\right) e^{-2^2} - \frac{1}{5} e^{-(2+1)^2}$$

$$= 61.64857$$

**Punto 2:**

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$
$$= 3(1-0)^2 e^{0^2-(0+1)^2} - 10\left(\frac{0}{5} - 0^3 + 0^5\right) e^{-0^2-0^2} - \frac{1}{5} e^{-(0+1)^2-0^2}$$
$$= 3(1)^2 e^{-(1)^2} - \frac{1}{5} e^{-(1)^2}$$
$$= 1.03006$$

**Punto 3:**

$$\begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}$$
$$3(1-1)^2 e^{1^2-(1+1)^2} - 10\left(\frac{1}{5} - 1^3 + 1^5\right) e^{-1^2-1^2} - \frac{1}{5} e^{-(1+1)^2-1^2}$$
$$= 10\left(\frac{1}{5} - 1 + 1\right) e^{-2} - \frac{1}{5} e^{-(2)^2-1}$$
$$= -0.272$$

**Punto 4:**

$$\begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix}$$
$$3(1-1)^2 e^{1^2-(-1+1)^2} - 10\left(\frac{1}{5} - 1^3 + (-1)^5\right) e^{-1^2-(-1)^2} - \frac{1}{5} e^{-(1+1)^2-(-1)^2}$$
$$= 10\left(\frac{1}{5} - 2\right) e^{-2} - \frac{1}{5} e^{-(2)^2-1}$$
$$= 2.43468$$

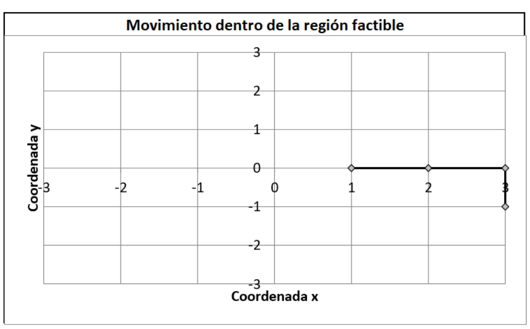
El punto central del radar de búsqueda es la coordenada (2,0). Ya que el valor en z de la función objetivo (61.64857) es el mayor entre los demás puntos de la vecindad y el punto de origen (0,1). Además, este punto se encuentra dentro de la región factible.

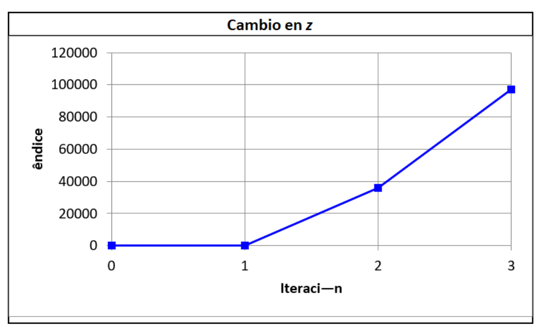
**b)** El resultado concuerda con el valor encontrado en el punto anterior.

**Solución: 97.237,0195704771**

**Reporte:**

**#Iteraciónes** = 4



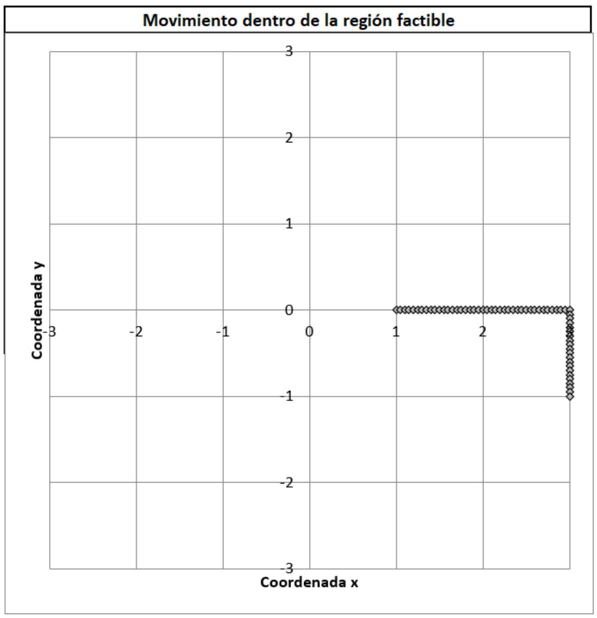



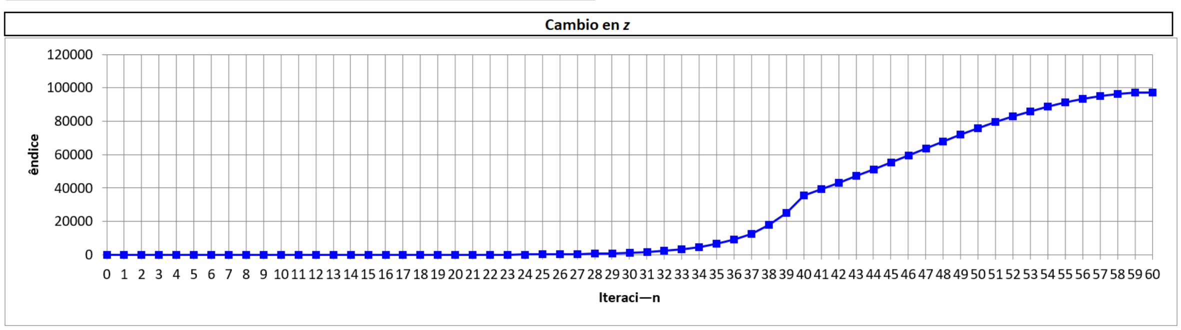

**¿Se podría afirmar que la solución encontrada en el inciso b. es el óptimo global del problema?: No.** Es posible pero improbable, dado tanto el punto inicial como las distancias de desplazamiento definen un espacio de búsqueda muy reducido que incluye pocos puntos sobre el plano. Por esta razón lo más probable es que se esté llegando a una solución buena dentro del espacio de búsqueda posible, pero no a una solución óptima.

**¿Se puede afirmar que es un óptimo local del problema? No.** Es posible pero improbable. En este caso es fácil imaginar que el resultado está cerca del óptimo local, pero no es este debido a que el óptimo local podría estar cerca de este punto, pero a una distancia que no corresponde a la escogida para realizar movimientos.

**c)**

**Reporte:**
**Mejor solución encontrada:** 97.237,0195704753
**#Iteraciónes** = 60




**d)** **D**

| | **b** | **c** |
|---|---|---|
| **Solución** | 97.237,0195704771 | 97.237,0195704753 |

| **Iteraciones** | 4 | 60 |
|---|---|---|
| **Tiempo (s)** | 2,44 | 42,72 |

En cuanto al número de iteraciones para llegar a la solución factible se puede ver que a mayor número de iteraciones la solución encontrada es mejor. En este caso al tener solo 4 como en el inciso b) los pasos fueron muy grandes (de 1 unidad) algo que provocó que el espacio de búsqueda fuera reducido a menos puntos sobre el plano. En el caso del incido c), se puede ver que al tener un paso de 0,05 (20 veces más pequeño que en el incido b) se llega a una mejor solución. Esto pasa porque los pasos en el inciso c), al ser más pequeños, tienen un mejor dominio de movimiento en el plano y se llega a una mejor solución. Esta es más cercada al óptimo local que la solución del inciso b). Es evidente que sí existe una trade off entre eficiencia y calidad de solución, ya que al usar pasos más pequeños se llega a mejores soluciones, pero se necesitan más iteraciones del algoritmo para que termine. En este caso se puede ver que al tener 4 iteraciones el resultado no es tan bueno como con 60, pero la ejecución del algoritmo demora poco. Creemos que el término correcto no es eficiencia sino tiempo de ejecución, porque en sí el algoritmo es igual de eficiente en los dos casos (es el mismo código y tiene la misma complejidad temporal), pero los parámetros si afectan el tiempo de ejecución, lo que evidencia un trade off al comparar esto con los resultados finales de cada ejecución.

$$\Delta_{\%} = \frac{\theta_b - \theta\_c}{\theta_b} \times 100$$

**Iteraciones:** $\Delta_{\%} = \frac{60-4}{60} \times 100$ = 93%

**Solución:** $\Delta_{\%} = \frac{97237{,}0195704753 - 97237{,}0195704771}{97237{,}0195704753} \times 100$ = 0,00000000000182

**Tiempo Computacional:** $\Delta_{\%} = \frac{42{,}72-2{,}44}{42{,}72} \times 100$ = 94,28

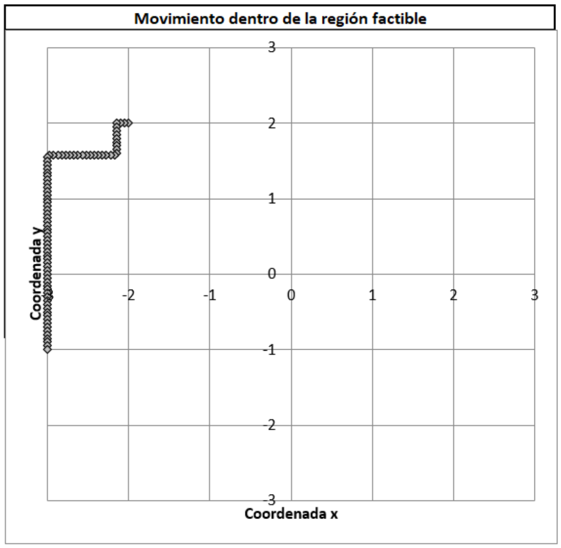
Finalmente se puede ver que en relación con el tiempo de más utilizado en la ejecución del inciso c, no se llega a una solución mucho mejor que en el inciso b a pesar de que hubo un gran numero de pasos y estos fueron más pequeños que en el incido b. Se esperaba obtener un resultado mucho mejor en este caso porque al reducir la distancia de los pasos hay un mejor acercamiento al óptimo local.

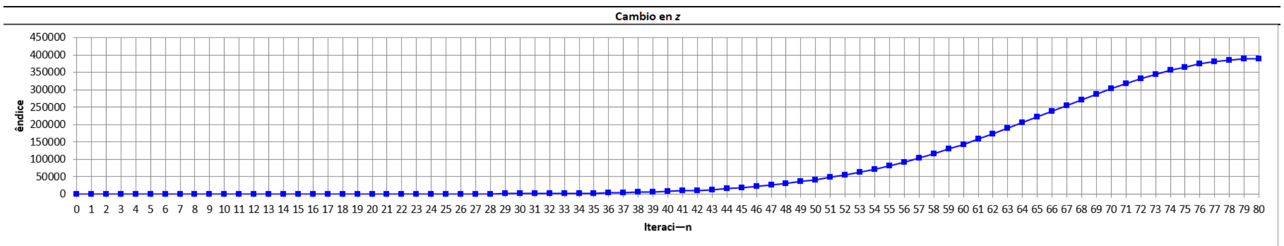
**e)**

**Reporte:**

**Mejor solución encontrada:** 388.948,015644438

**#Iteraciónes =** 80




**f)**

| | e | c |
|---|---|---|
| **Solución** | 388.948,015644438 | 97.237,0195704753 |
| **Iteraciones** | 80 | 60 |
| **Tiempo (s)** | 65,86 | 42,72 |

En cuánto a la solución obtenida se evidencia que en el incido e se obtuvo una solución mucho mejor que la del inciso c) El número de iteraciones al igual que el tiempo de ejecución fueron mayores en la ejecución del inciso e. Además, se puede ver que, dentro del movimiento en la región factible, el inciso e) converge a una solución ubicada en otro lugar diferente que la solución encontrada en el inciso c) Ambas soluciones no deberían ser iguales, porque empiezan en lugares distintos, lo cual, en este caso, conduce a que se llegue a diferentes óptimos locales. Son óptimos locales porque la búsqueda local no puede garantizar que se esté llegando un óptimo global porque no se puede saber si una solución es cercana a este. En este inciso la magnitud de los pasos fue la misma y la única variación en la ejecución del algoritmo de búsqueda local fue el punto inicial. Por lo tanto, se puede ver que la elección de un punto inicial puede afectar cual es el óptimo local al que se llega dados los parámetros de la búsqueda. Esto pasa porque si se varía el punto inicial, al tener un algoritmo avaro como el de búsqueda local (que elige la mejor solución local paso a paso sin ver un contexto más global) se puede elegir un mal camino en un inicio que lleve a un óptimo local con un valor de función objetivo bajo.

$$\Delta_{\%} = \frac{\theta_b - \theta\_c}{\theta_b} \times 100$$

**Iteraciones:** $\Delta_{\%} = \frac{80-60}{80} \times 100$ = 93%

**Solución:** $\Delta_{\%} = \frac{388.948{,}0156-97.237{,}0195704753}{388.948{,}0156} \times 100$ = 74,99

**Tiempo Computacional:** $\Delta_{\%} = \frac{65{,}86-42{,}72}{65{,}86} \times 100$ = 35,14

Finalmente se puede ver que es este caso hubo un incremento considerable en el resultado de la función objetivo en el inciso e) en comparación con el inciso c). Además, el tiempo de ejecución es mayor pero ya que el resultado es mucho mejor es justificable utilizar más tiempo. Los mismo sucede con las iteraciones.