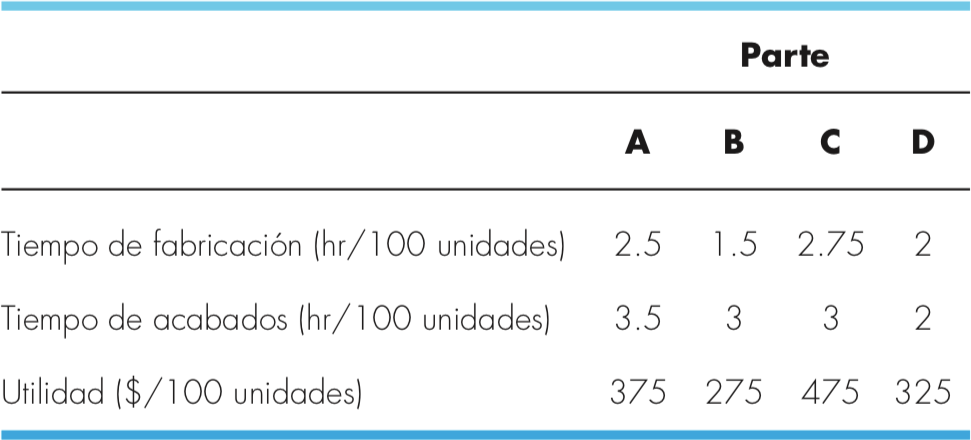
Ejercicio 1 (Valor 5 puntos)

Una fábrica produce cuatro tipos de partes automotrices. Cada una de ellas primero se fabrica y luego se le dan los acabados. Las horas de trabajador requeridas y la utilidad para cada parte son las siguientes:



Las capacidades de los talleres de fabricación y acabados para el mes siguiente son de 640 y 960 horas, respectivamente. Determinar mediante un algoritmo PSO con restricciones (sin usar bibliotecas para PSO) que cantidad de cada partes debe producirse a fin de maximizar la utilidad y resolver las siguientes consignas:

1. (Valor 0.33 puntos) Transcribir el algoritmo escrito en Python a un archivo .pdf de acuerdo a los siguientes parámetros: número de partículas = 20, máximo número de iteraciones 50, coeficientes de aceleración c1 = c2 = 1.4944, factor de inercia w = 0.6.

# Optimizacion

gbest\_fit\_iterations = []

for \_ in range(max\_iterations):

for i in range(n\_particles):

fit = f(x[i])

# Se comprueba si la nueva aptitud es mejor y si cumple las restricciones

if fit > pbest\_fit[i] and g1(x[i]) and g2(x[i]):

pbest\_fit[i] = fit

pbest[i] = x[i].copy()

if fit > gbest\_fit:

gbest\_fit = fit

gbest = x[i].copy()

# Actualizacion de la velocidad y la posicion de la particula

v[i] = w \* v[i] + c1 \* np.random.rand() \* (pbest[i] - x[i]) + c2 \* np.random.rand() \* (gbest - x[i])

x[i] += v[i]

if not (g1(x[i]) and g2(x[i])):

x[i] = pbest[i].copy()

gbest\_fit\_iterations.append(gbest\_fit)

# Se imprime la mejor solucion encontrada y también su valor optimo

print(f"Mejor solucion: [{gbest[0]:.4f}, {gbest[1]:.4f}, {gbest[2]:.4f}, {gbest[3]:.4f}]")

print(f"Valor optimo: {gbest\_fit}")

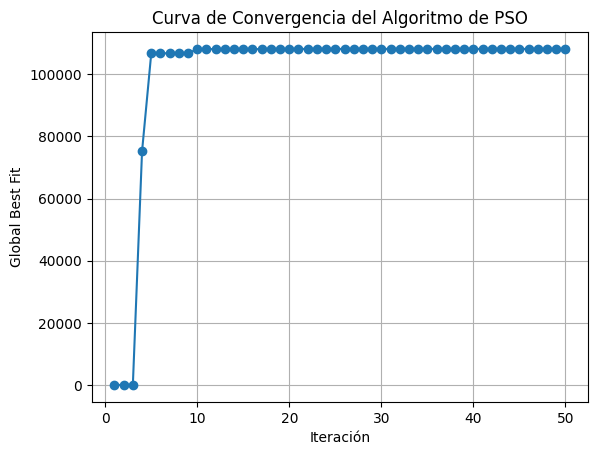
1. (Valor 0.33 puntos) Transcribir al .pdf la solución óptima encontrada (dominio) y el valor objetivo óptimo (imagen).

La solución óptima encontrada es: **(40.6335, 79.7810, 104.7745, 65.3075)**.  
El valor objetivo óptimo encontrado es: **108170.14114657712**.

1. (Valor 0.33 puntos) Indicar en el .pdf la URL del repositorio en donde se encuentra el algoritmo PSO.

<https://github.com/fede0ter0/ceia_algoritmos_evolutivos>

1. (Valor 0.5 puntos) Realizar un gráfico de línea que muestre gbest (eje de ordenadas) en función de las iteraciones realizadas (eje de abscisas). El gráfico debe contener etiquetas en los ejes, leyenda y un título. El gráfico debe ser pegado en el .pdf.



1. (Valor 0.5 puntos) Explicar (en el .pdf) y demostrar (desde el código fuente) que sucede si se reduce en 1 unidad el tiempo de acabado de la parte B.

En el código hicimos la modificación de la función de restricción *g2(X)*. El resultado es un poco mejor que en el caso anterior.

La solución óptima encontrada es: **(-3.5167, 14.9267, 249.0431, -29.2334)**. El valor objetivo óptimo encontrado es: **111580.6915472145**.

A graph with a line

Description automatically generated

Como vemos, la solución es un poco mejor que la anterior, ya que al reducir el acabado de la parte B en una unidad podemos mejorar la exploración.

1. (Valor 1 punto) Realizar 3 boxplots a partir de los gbest de 3 variantes de solución: i) El primer boxplot con el coeficiente de inercia constante w = 0.8, ii) El segundo boxplot con el coeficiente de inercia dinámico lineal y

iii) El tercer boxplot sin coeficiente de inercia pero aplicando Factor de constricción con φ a elección.

Los resultados aparecen en el notebook.

1. (Valor 1 punto) Realizar observaciones/comentarios/conclusiones en el

.pdf acerca de los resultados del ítem (f).

En este caso, el uso de un valor fijo de inercia ofrece una solución robusta y efectiva, ya que el problema es relativamente sencillo y no requiere una exploración o explotación demasiado dinámica. Además, el espacio de búsqueda no es tan complejo como para que se justifique ajustes en la inercia ni el enfoque agresivo del factor de constricción.

1. (Valor 1 punto) Realizar observaciones/comentarios/conclusiones en el

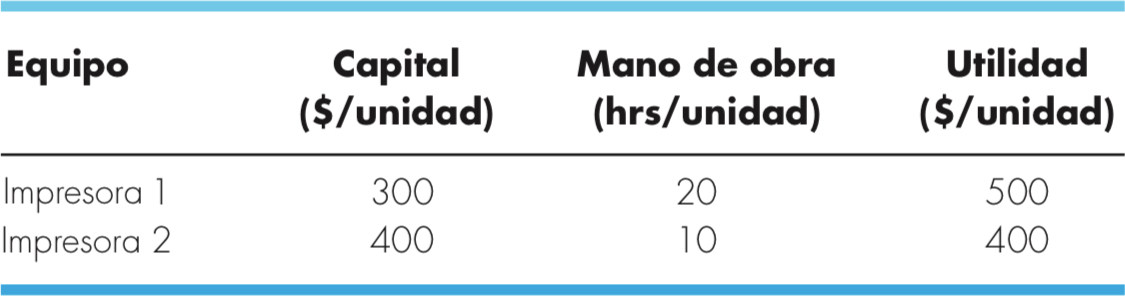
.pdf acerca de qué cantidad mínima de partículas es factible utilizar para este problema específicamente.

Dado que nuestro problema tiene una función objetivo lineal y restricciones también lineales, el espacio de búsqueda es más predecible y menos propenso a múltiples óptimos locales en comparación con problemas no lineales o altamente complejos. Esto sugiere que no necesitamos demasiadas partículas para cubrir de manera eficiente el espacio de búsqueda.

En términos generales, una regla empírica en PSO es que el número de partículas debería ser al menos 10 veces el número de dimensiones para garantizar una buena exploración del espacio de búsqueda. Sin embargo, utilizando 20 partículas ya conseguimos una solución muy buena, con lo cual consideramos que 10 partículas es una cantidad mínima aceptable.

Ejercicio 2 (Valor 5 puntos)

Un fabricante de equipos de cómputo produce dos tipos de impresoras (impresoras de tipo 1 e impresoras de tipo 2). Los recursos necesarios para producirlas así como las utilidades correspondientes son los que siguen:



Si cada día se dispone de $127000 de capital y 4270 horas de mano de obra,

¿qué cantidad de cada equipo debe producirse a diario a fin de maximizar la utilidad? Escriba el algoritmo PSO con restricciones (sin usar bibliotecas para PSO) que permita optimizar la utilidad y resolver cumpliendo con las siguientes consignas:

1. (Valor 0.33 puntos) Transcribir el algoritmo escrito en Python a un archivo .pdf de acuerdo a los siguientes parámetros: número de partículas = 10, máximo número de iteraciones 80, coeficientes de aceleración c1 = c2 = 2, factor de inercia w = 0.5.
2. (Valor 0.33 puntos) Transcribir al .pdf la solución óptima encontrada (dominio) y el valor objetivo óptimo (imagen).
3. (Valor 0.33 puntos) Indicar en el .pdf la URL del repositorio en donde se encuentra el algoritmo PSO.
4. (Valor 1 punto) Realizar un gráfico de línea que muestre gbest (eje de ordenadas) en función de las iteraciones realizadas (eje de abscisas). El gráfico debe contener etiquetas en los ejes, leyenda y un título. El gráfico debe ser pegado en el .pdf.
5. (Valor 1 puntos) Realizar 3 boxplots a partir de los gbest de 3 variantes de solución: i) El primer boxplot con el coeficiente de inercia constante w = 0.9, ii) El segundo boxplot con el coeficiente de inercia dinámico lineal y iii) El tercer boxplot sin coeficiente de inercia pero aplicando Factor de constricción con φ = 5.
6. (Valor 1 puntos) Realizar observaciones/comentarios/conclusiones en el

.pdf acerca de los resultados del ítem (f).

1. (Valor 1 puntos) Realizar observaciones/comentarios/conclusiones en el

.pdf acerca de qué cantidad mínima de partículas es factible utilizar para este problema específicamente.

 *Subir la resolución de este TP al campus en formato .pdf.*

 *Indicar en el .pdf la URL del repositorio en donde se encuentran los códigos fuentes.*

 *Fecha límite de entrega: 07/10/2024.*