



Área Académica de Ingeniería Mecatrónica

MT 8008 – Inteligencia Artificial

Tarea 2: Algoritmo evolutivo simple

Elaborado por:

Diana Cerdas Vargas

2020023718

Jeaustin Calderón Quesada

2020027413

Juan Luis Crespo Mariño, Dr. Ing

Noviembre 2023

Tabla de contenidos

| | |
|---|----|
| Introducción y exposición del problema general..... | 3 |
| Análisis estático..... | 6 |
| Análisis del peso de las vigas | 6 |
| Análisis de la fuerza externa..... | 12 |
| Cálculo de esfuerzos | 17 |
| Implementación | 19 |
| Justificación de distintas decisiones de diseño | 19 |
| Estudio de hiperparámetros | 22 |
| Caso 1 | 23 |
| Caso 2 | 24 |
| Caso 3 | 25 |
| Caso 4 | 26 |
| Caso 5 | 27 |
| Caso 6 | 28 |
| Caso 7 | 29 |
| Caso 8 | 30 |
| Caso 9 | 31 |
| Caso 10 | 32 |
| Análisis de resultados | 33 |
| Conclusiones..... | 36 |
| Bibliografía..... | 37 |
| Anexos | 38 |
| Anexo 1 | 38 |

Introducción y exposición del problema general

Los algoritmos evolutivos son una clase de algoritmos de optimización inspirados en la evolución natural que se utilizan para resolver problemas complejos en una amplia gama de campos. Estos algoritmos imitan el proceso de selección natural, reproducción y mutación observado en la evolución biológica para buscar soluciones óptimas o cercanas a óptimas en un espacio de búsqueda [1].

En esencia, los algoritmos evolutivos operan en una población de posibles soluciones, representadas como "individuos", y aplican operadores genéticos como la selección, el cruce (crossover) y la mutación para iterativamente mejorar las soluciones a lo largo de generaciones [1].

Los algoritmos evolutivos son versátiles y se pueden adaptar a diferentes problemas y dominios mediante la modificación de sus operadores genéticos y parámetros. Se han aplicado con éxito en la optimización de funciones matemáticas, diseño de ingeniería, problemas de programación, aprendizaje automático, entre otros, ofreciendo soluciones aproximadas en casos complejos donde otras técnicas pueden no ser efectivas [1].

El concepto principal detrás de los algoritmos evolutivos consiste en una población dada de individuos en un ambiente en el que hay recursos limitados, la competencia por los recursos resulta en selección natural. La selección natural implica un aumento en la aptitud de la población [1].

Si se tiene una función de calidad que se desea maximizar o minimizar, se puede crear una población aleatoria de individuos que según su aptitud (evaluada por la función de calidad) continua para engendrar nuevos individuos. La creación de nuevos individuos se da mediante recombinación y mutación. Seguidamente, los individuos recién generados compiten con los padres para asegurar un espacio en la siguiente generación. Esto se itera hasta que se encuentre un individuo que satisfaga un criterio de calidad determinado o hasta que se alcance un límite de computación definido con anticipación.

En el diagrama mostrado en la figura 1 se puede observar la lógica detrás del algoritmo evolutivo.

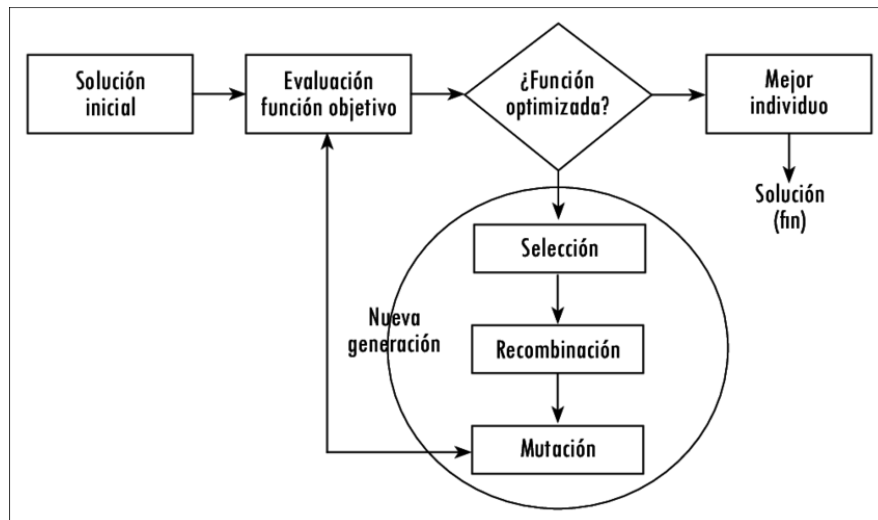


Figura 1. Representación general de un algoritmo evolutivo [2].

A continuación, se va a discutir brevemente cada una de las etapas del diagrama anterior [1].

Solución inicial:

Se genera una población inicial con elementos aleatorios. Es importante realizar una codificación acertada que permita realizar recombinación sin afectar la coherencia en las soluciones. Asimismo, hay que definir un espacio de alelos para cada uno de los genes del individuo, esto según el conocimiento previo que se tenga del problema.

Evaluación:

Se evalúa cada individuo en la población utilizando una función de calidad. Esta función evalúa que tan buena es una solución y es completamente dependiente del problema, es decir, se debe plantear según el problema a solucionar.

Selección:

Es posible seleccionar únicamente a los individuos más aptos, sin embargo, esto significa usar un criterio meramente elitista que podrían descartar genes ideales que se encuentran en individuos con baja evaluación. Para evitar esto se pueden realizar distintas dinámicas que a pesar de que premian la aptitud, también dan posibilidad de que individuos no tan aptos sean seleccionados. Ejemplos de métodos de selección son el torneo y la ruleta.

Recombinación:

Los individuos seleccionados se combinan aleatoriamente utilizando operadores de cruce para crear descendencia.

Mutación:

Permite realizar leves cambios esporádicos en algunos individuos, introduciendo una componente de variabilidad en el pool genético. Esto ayuda a prevenir la convergencia prematura.

Reemplazo:

La nueva descendencia (junto con algunos individuos de la generación actual, a menudo los más aptos) reemplaza a la población anterior, formando una nueva generación.

Iteración:

Estos pasos se repiten durante varias generaciones hasta que se alcance un criterio de parada predefinido, como un número máximo de generaciones o la convergencia hacia una solución que cumpla con la calidad deseada.

Análisis estático

En la figura 2 se muestra la configuración de armadura que se desea optimizar. La estrategia de resolución consiste en primero realizar un análisis de la afectación del peso de las vigas en la estructura para seguidamente hacer el análisis del impacto de la carga externa en la estructura. Es importante resaltar algunas convenciones tomadas para los cálculos: el momento es positivo si está dirigido en contra de las manecillas del reloj, la fuerza vertical es positiva si se dirige hacia arriba, la horizontal es positiva a la derecha y todas las vigas se asumen en tensión, por lo que un valor de fuerza negativo significa que realmente la viga se encuentra en compresión.

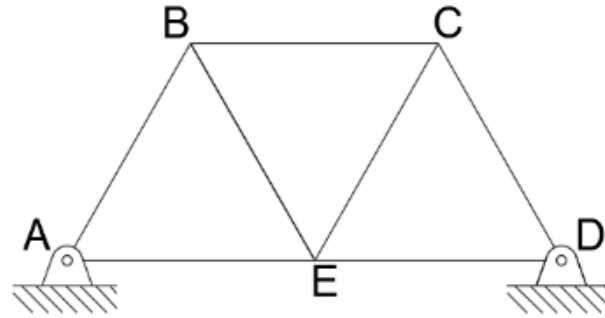


Figura 2. Armadura *Warren* a optimizar.

Análisis del peso de las vigas

En primer lugar, se va a obtener el efecto del peso de las barras en las fuerzas que experimentan cada uno de los elementos de la armadura.

Se tiene que la ecuación para calcular el volumen de una barra corresponde a:

$$V = \frac{\pi}{4} d^2 l$$

Para calcular la masa se usa la siguiente ecuación.

$$m = V\rho$$

Finalmente, se puede calcular el peso de la barra de la siguiente manera.

$$W = mg = V\rho g = \frac{\pi}{4} d^2 l \rho g$$

Se encierran varias constantes en una nueva llamada K_1 con el fin de mantener el orden.

$$K_1 = \frac{\pi}{4} l \rho g$$

$$K_1 = \frac{\pi}{4} \cdot 5m \cdot 2870 \frac{kg}{m^3} \cdot 9.81 \frac{m}{s^2} = 110563.25 \frac{kg}{m \cdot s^2}$$

Se analizan los nodos uno por uno, se toma que cada uno de los nodos soporta la mitad del peso de cada una de las barras que se encuentran conectadas a dicho nodo [3].

Nodo A

En el nodo A únicamente se tienen conectadas las vigas AB y AE.

$$F_A = \frac{W_{AB} + W_{AE}}{2}$$

$$F_A = \frac{K_1}{2} (d_{AB}^2 + d_{AE}^2)$$

Nodo B

En el nodo B se tienen unidas las vigas AB, BC y BE.

$$F_B = \frac{W_{AB} + W_{BC} + W_{BE}}{2}$$

$$F_B = \frac{K_1}{2} (d_{AB}^2 + d_{BC}^2 + d_{BE}^2)$$

Nodo C

En el nodo C se tienen unidas las vigas BC, EC y CD.

$$F_C = \frac{W_{BC} + W_{EC} + W_{CD}}{2}$$

$$F_C = \frac{K_1}{2} (d_{BC}^2 + d_{EC}^2 + d_{CD}^2)$$

Nodo D

En el nodo D únicamente se tienen conectadas las vigas CD y ED.

$$F_D = \frac{W_{CD} + W_{ED}}{2}$$

$$F_D = \frac{K_1}{2} (d_{CD}^2 + d_{ED}^2)$$

Nodo E

Finalmente, en el nodo E se tienen unidas las vigas AE, BE, EC y ED.

$$F_E = \frac{W_{AE} + W_{BE} + W_{EC} + W_{ED}}{2}$$

$$F_E = \frac{K_1}{2} (d_{AE}^2 + d_{BE}^2 + d_{EC}^2 + d_{ED}^2)$$

A continuación, se procede a calcular las fuerzas de reacción en los puntos de soporte. Se inicia con la reacción en el punto D. Se realiza una sumatoria de momentos en el punto A. Con apoyo de la figura 3 se puede observar las distancias necesarias para realizar la sumatoria de momentos desde el punto A.

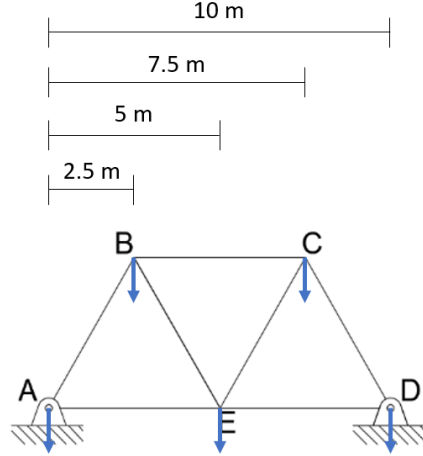


Figura 3. Distancias entre nodos de la armadura a optimizar.

$$\sum M_A = 0$$

$$-2.5F_B - 5F_E - 7.5F_C - 10F_D + 10N_D = 0$$

Se sustituyen los valores por las expresiones encontradas anteriormente y se agrupan.

$$N_D = \frac{K_1}{20} [2.5(d_{AB}^2 + d_{BC}^2 + d_{BE}^2) + 5(d_{AE}^2 + d_{BE}^2 + d_{EC}^2 + d_{ED}^2) + 7.5(d_{BC}^2 + d_{EC}^2 + d_{CD}^2) + 10(d_{CD}^2 + d_{ED}^2)]$$

$$N_D = \frac{K_1}{20} [2.5d_{AB}^2 + 10d_{BC}^2 + 7.5d_{BE}^2 + 5d_{AE}^2 + 12.5d_{EC}^2 + 15d_{ED}^2 + 17.5d_{CD}^2]$$

Para encontrar la otra reacción en el punto A, se procede a realizar una sumatoria de fuerzas en el eje vertical tomando en cuenta todo el sistema y sustituir por las expresiones encontradas en términos de los diámetros de las vigas.

$$\sum F_y = 0$$

$$N_A + N_D - F_B - F_C - F_E = 0$$

$$N_A = -N_D + F_B + F_C + F_E$$

$$\begin{aligned} N_A = & -\frac{K_1}{20} [2.5d_{AB}^2 + 10d_{BC}^2 + 7.5d_{BE}^2 + 5d_{AE}^2 + 12.5d_{EC}^2 + 15d_{ED}^2 + 17.5d_{CD}^2] \\ & + \frac{K_1}{20} (10d_{AB}^2 + 10d_{BC}^2 + 10d_{BE}^2) + \frac{K_1}{20} (10d_{BC}^2 + 10d_{EC}^2 + 10d_{CD}^2) \\ & + \frac{K_1}{20} (10d_{AE}^2 + 10d_{BE}^2 + 10d_{EC}^2 + 10d_{ED}^2) \end{aligned}$$

$$N_A = -\frac{K_1}{20} [-7.5d_{AB}^2 - 10d_{BC}^2 - 12.5d_{BE}^2 - 5d_{AE}^2 - 7.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2]$$

A continuación, se van a analizar los distintos nodos que permiten obtener los valores de las fuerzas que soportan las barras y que generan un esfuerzo axial en las mismas. Se asume tensión para todas las fuerzas de los elementos que son desconocidas.

Nodo C

En la figura 4 se muestran las fuerzas que actúan en el nodo D, usando el diagrama de cuerpo libre (DGL). Se tienen 3 fuerzas que provienen de las vigas y la fuerza resultante C.

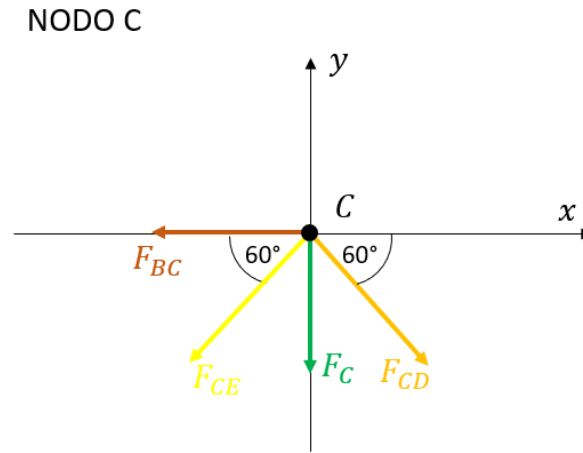


Figura 4. DCL del nodo C, tomando en cuenta únicamente el peso de las vigas.

Se hace sumatoria de fuerzas en el eje vertical, se despeja la fuerza F_{CE} y en términos de diámetros.

$$\sum F_y = 0$$

$$-F_C - F_{CE} \sin(60^\circ) - F_{CD} \sin(60^\circ) = 0$$

$$F_{CE} = -F_{CD} - \frac{F_C}{\sin(60^\circ)}$$

$$F_{CE} = \frac{K_1}{20 \sin(60^\circ)} [2.5d_{AB}^2 + 10d_{BC}^2 + 7.5d_{BE}^2 + 5d_{AE}^2 + 12.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2] - \frac{K_1}{20 \sin(60^\circ)} (10d_{BC}^2 + 10d_{EC}^2 + 10d_{CD}^2)$$

$$F_{CE} = \frac{K_1}{20 \sin(60^\circ)} [2.5d_{AB}^2 + 7.5d_{BE}^2 + 5d_{AE}^2 + 2.5d_{EC}^2 + 5d_{ED}^2 - 2.5d_{CD}^2]$$

Se procede a despejar F_{BC} en términos de los diámetros al realizar una sumatoria de fuerzas en x.

$$\sum F_x = 0$$

$$F_{BC} = -F_{CE} \cos(60^\circ) + F_{CD} \cos(60^\circ)$$

$$F_{BC} = -\frac{K_1 \cos(60^\circ)}{20 \sin(60^\circ)} [2.5d_{AB}^2 + 7.5d_{BE}^2 + 5d_{AE}^2 + 2.5d_{EC}^2 + 5d_{ED}^2 - 2.5d_{CD}^2]$$

$$-\frac{K_1 \cos(60^\circ)}{20 \sin(60^\circ)} [2.5d_{AB}^2 + 10d_{BC}^2 + 7.5d_{BE}^2 + 5d_{AE}^2 + 12.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2]$$

$$F_{BC} = -\frac{K_1 \cos(60^\circ)}{20 \sin(60^\circ)} [5d_{AB}^2 + 10d_{BC}^2 + 15d_{BE}^2 + 10d_{AE}^2 + 15d_{EC}^2 + 10d_{ED}^2 + 5d_{CD}^2]$$

Nodo A

En la figura 5 se muestran las fuerzas que actúan en el nodo A. Se tienen 2 fuerzas que provienen de las vigas, la fuerza de reacción del soporte y la fuerza resultante A.

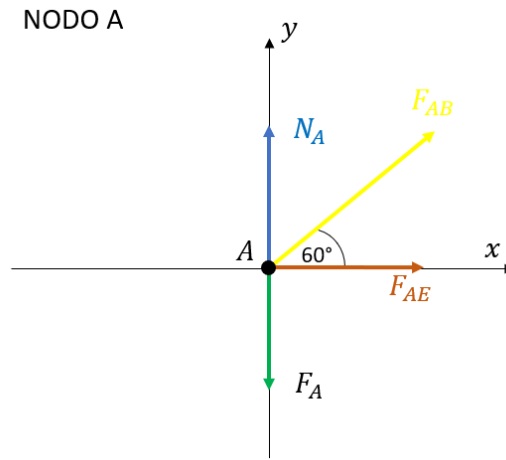


Figura 5. DCL del nodo A, tomando en cuenta únicamente el peso de las vigas

Se realiza una sumatoria de fuerzas en el eje vertical, se despeja la fuerza F_{AB} y se pone en términos de los diámetros.

$$\sum F_y = 0$$

$$N_A - F_A + F_{AB} \sin(60^\circ) = 0$$

$$F_{AB} = \frac{-N_A + F_A}{\sin(60^\circ)}$$

$$F_{AB} = \frac{K_1}{20 \sin(60^\circ)} [-7.5d_{AB}^2 - 10d_{BC}^2 - 12.5d_{BE}^2 - 5d_{AE}^2 - 7.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2] + \frac{K_1}{20 \sin(60^\circ)} (10d_{AB}^2 + 10d_{AE}^2)$$

$$F_{AB} = \frac{K_1}{20 \sin(60^\circ)} [2.5d_{AB}^2 - 10d_{BC}^2 - 12.5d_{BE}^2 + 5d_{AE}^2 - 7.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2]$$

Se procede a hacer una sumatoria de fuerzas horizontales, con esta se logra despejar F_{AE} en términos de diámetros.

$$\sum F_x = 0$$

$$F_{AE} + F_{AB} \cos(60^\circ) = 0$$

$$F_{AE} = -F_{AB} \cos(60^\circ)$$

$$F_{AE} = -\frac{K_1 \cos(60^\circ)}{20 \sin(60^\circ)} [2.5d_{AB}^2 - 10d_{BC}^2 - 12.5d_{BE}^2 + 5d_{AE}^2 - 7.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2]$$

Nodo B

En la figura 6 se muestran las fuerzas que actúan en el nodo b. Se tienen 3 fuerzas que provienen de las vigas y la fuerza resultante B.

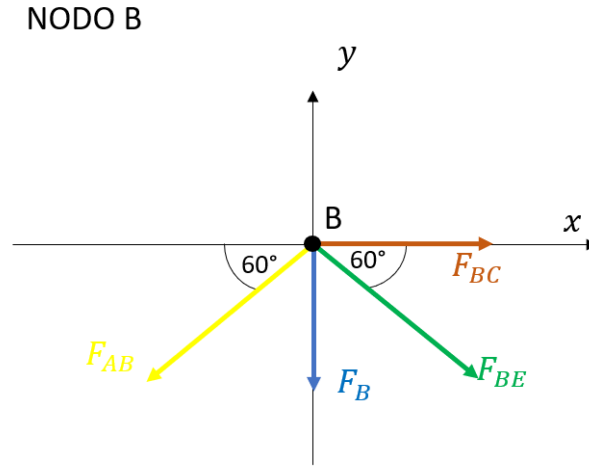


Figura 6. DCL del nodo B, tomando en cuenta únicamente el peso de las vigas.

Se realiza una sumatoria de fuerzas en el eje vertical, se despeja la fuerza F_{BE} y se pone en términos de los diámetros.

$$\sum F_y = 0$$

$$-F_{AB} \sin(60^\circ) - F_{BE} \sin(60^\circ) - F_B = 0$$

$$F_{BE} = -F_{AB} - \frac{F_B}{\sin(60^\circ)}$$

$$F_{AB} = \frac{K_1}{20 \sin(60^\circ)} [2.5d_{AB}^2 - 10d_{BC}^2 - 12.5d_{BE}^2 + 5d_{AE}^2 - 7.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2]$$

$$F_B = \frac{K_1}{2} (d_{AB}^2 + d_{BC}^2 + d_{BE}^2)$$

$$F_{BE} = -\frac{K_1}{20 \sin(60^\circ)} [2.5d_{AB}^2 - 10d_{BC}^2 - 12.5d_{BE}^2 + 5d_{AE}^2 - 7.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2] - \frac{K_1}{20 \sin(60^\circ)} (10d_{AB}^2 + 10d_{BC}^2 + 10d_{BE}^2)$$

$$F_{BE} = -\frac{K_1}{20 \sin(60^\circ)} [12.5d_{AB}^2 - 2.5d_{BE}^2 + 5d_{AE}^2 - 7.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2]$$

Análisis de la fuerza externa

Ahora se obtiene el efecto de la carga externa en las fuerzas que experimentan los elementos. Para esto, primero se va a obtener la reacción en el nodo D y luego a partir de esto, se va analizando nodo por nodo para obtener las fuerzas de cada elemento.

Para esto, se va a tratar toda la armadura como un cuerpo rígido. En la figura 7 se muestra un planteamiento general de las distancias necesarias para hacer sumatoria de momentos tomando en cuenta únicamente la carga externa.

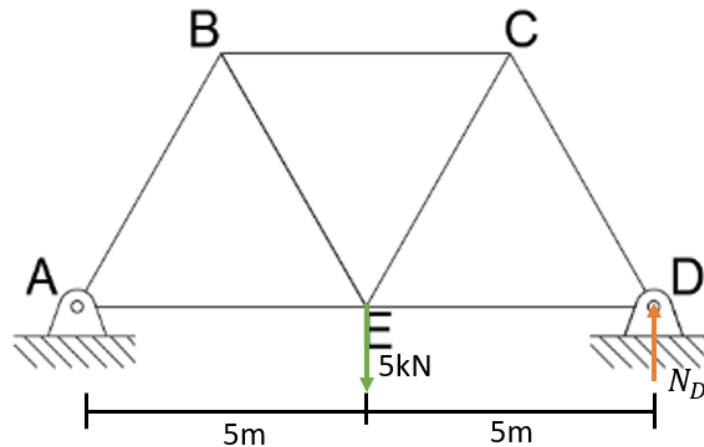


Figura 7. Distancias entre nodos de la armadura a optimizar.

Cálculo de reacción en D

Sumatoria de momentos en el nodo A:

$$\begin{aligned}\sum M_A &= 0 \\ \Rightarrow -5000 \cdot 5 + N_D \cdot 10 &= 0 \\ \Rightarrow N_D &= \frac{5000 \cdot 5}{10} \\ \Rightarrow N_D &= 2500 \text{ N}\end{aligned}$$

Seguidamente, se analizan cada uno de los nodos para obtener las fuerzas en los elementos, es importante recalcar que se van a asumir que todas las fuerzas están en tensión y posteriormente el signo va a indicar si están en compresión.

Nodo D

En la figura 8 se muestra el diagrama de cuerpo libre en el nodo D.

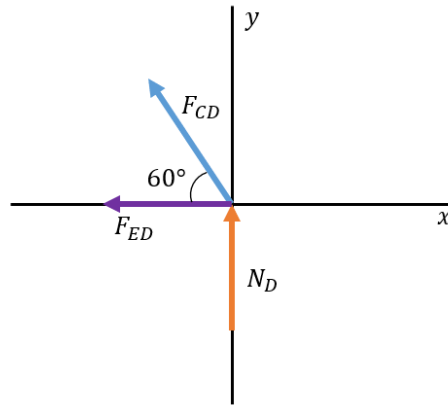


Figura 8. DCL del nodo D, tomando en cuenta únicamente la carga externa.

A continuación, se muestra una sumatoria en el eje vertical que permite encontrar el valor numérico de F_{CD} .

$$\begin{aligned}\sum F_y &= 0 \\ \Rightarrow N_D + F_{CD} \cdot \text{sen}(60) &= 0 \\ \Rightarrow F_{CD} &= \frac{-N_D}{\text{sen}(60)} \\ \Rightarrow F_{CD} &= -\frac{2500}{\text{sen}(60)} \\ \Rightarrow F_{CD} &= -2886.75 \text{ N}\end{aligned}$$

Se puede interpretar que esta viga se encuentra en compresión.

Seguidamente, se muestra una sumatoria en el eje horizontal que permite encontrar el valor numérico de F_{DE} .

$$\begin{aligned}\sum F_x &= 0 \\ \Rightarrow -F_{CD} \cdot \cos(60) - F_{ED} &= 0 \\ \Rightarrow F_{ED} &= -F_{CD} \cdot \cos(60) \\ \Rightarrow F_{DE} &= 1443.38 \text{ N}\end{aligned}$$

Nodo C

En la figura 9 se muestra el diagrama de cuerpo libre en el nodo C.

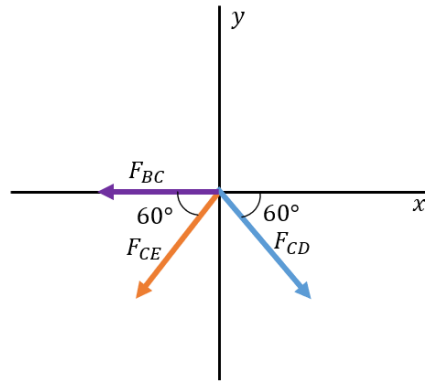


Figura 9. DCL del nodo C, tomando en cuenta únicamente la carga externa.

A continuación, se muestra una sumatoria en el eje vertical que permite encontrar el valor numérico de F_{CE} .

$$\begin{aligned}\sum F_y &= 0 \\ \Rightarrow -F_{CD} \cdot \text{sen}(60) - F_{CE} \cdot \text{sen}(60) &= 0 \\ \Rightarrow F_{CE} &= -F_{CD} \\ \Rightarrow F_{CE} &= 2886.75 \text{ N}\end{aligned}$$

Asimismo, se muestra una sumatoria en el eje horizontal que permite encontrar el valor numérico de F_{BC} .

$$\begin{aligned}\sum F_x &= 0 \\ \Rightarrow F_{CD} \cdot \cos(60) - F_{CE} \cdot \cos(60) - F_{BC} &= 0 \\ \Rightarrow F_{BC} &= F_{CD} \cdot \cos(60) - F_{CE} \cdot \cos(60) \\ \Rightarrow F_{BC} &= -2886.75 \cdot \cos(60) - 2886.75 \cdot \cos(60) \\ \Rightarrow F_{BC} &= -2886.75 \text{ N}\end{aligned}$$

Es oportuno entonces recalcar que F_{BC} se encuentra en compresión, puesto que su valor numérico tuvo signo opuesto al asumido inicialmente.

Nodo B

En la figura 10 se muestra el diagrama de cuerpo libre en el nodo B.

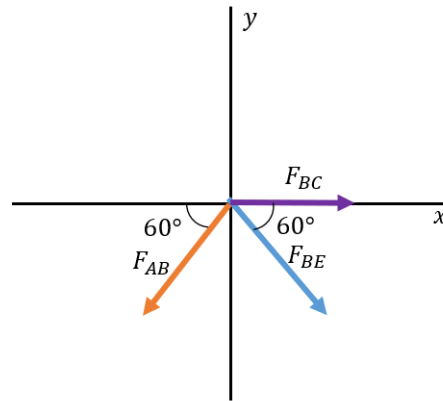


Figura 10. DCL del nodo B, tomando en cuenta únicamente la carga externa.

Seguidamente, se muestra una sumatoria en el eje horizontal que permite encontrar una expresión de F_{BE} en términos de F_{AB} .

$$\begin{aligned}\sum F_y &= 0 \\ \Rightarrow -F_{BE} \cdot \text{sen}(60) - F_{AB} \cdot \text{sen}(60) &= 0 \\ \Rightarrow F_{BE} &= -F_{AB} \quad (1)\end{aligned}$$

Asimismo, se realiza una sumatoria en el eje horizontal que permite encontrar la segunda ecuación necesaria para encontrar los valores numéricos de F_{BE} y F_{AB} .

$$\begin{aligned}\sum F_x &= 0 \\ \Rightarrow F_{BE} \cdot \cos(60) - F_{AB} \cdot \cos(60) + F_{BC} &= 0\end{aligned}$$

Sustituyendo (1) en la ecuación encontrada:

$$\begin{aligned}\Rightarrow 2F_{AB} \cdot \cos(60) &= F_{BC} \\ \Rightarrow F_{AB} &= \frac{F_{BC}}{2\cos(60)} \\ \Rightarrow F_{AB} &= -2886.75 \text{ N}\end{aligned}$$

Regresando a la ecuación (1)

$$\Rightarrow F_{BE} = 2886.75 \text{ N}$$

Se puede concluir que F_{AB} se encuentra en compresión.

Nodo E

En la figura 11 se muestra el diagrama de cuerpo libre en el nodo E.

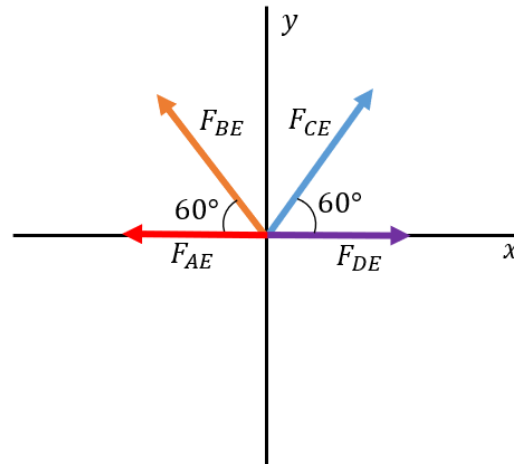


Figura 11. DCL del nodo E, tomando en cuenta únicamente la carga externa.

Únicamente es necesario realizar una sumatoria de fuerzas horizontales con el fin de encontrar el valor numérico de F_{AE} .

$$\begin{aligned}\sum F_x &= 0 \\ \Rightarrow F_{DE} + F_{CE} \cdot \cos(60) - F_{BE} \cdot \cos(60) - F_{AE} &= 0 \\ \Rightarrow F_{AE} &= F_{DE} + F_{CE} \cdot \cos(60) - F_{BE} \cdot \cos(60) \\ \Rightarrow F_{AE} &= 1443.38 + 2886.75 \cdot \cos(60) - 2886.75 \cdot \cos(60) \\ \Rightarrow F_{AE} &= 1443.38 \text{ N}\end{aligned}$$

Cálculo de esfuerzos

Se sabe por definición que el esfuerzo normal corresponde a la razón entre la fuerza y el área trasversal [3].

$$\sigma = \frac{F}{A}$$

En este caso el área trasversal corresponde a un círculo, su fórmula se muestra a continuación.

$$A = \frac{\pi}{4} d^2$$

Uniendo ambas expresiones se tiene entonces.

$$\sigma = \frac{4}{\pi} \frac{F}{d^2}$$

Se procede a calcular cada uno de los esfuerzos sustituyendo las expresiones de fuerza en términos de diámetros definidas anteriormente y divididas entre el área trasversal de un círculo.

Esfuerzo normal en la barra AB

$$\sigma = \frac{4}{\pi} \frac{F_{AB}}{d_{AB}^2}$$

$$\sigma_{AB} = \frac{4}{\pi} \frac{\left[\left(\frac{K_1}{20 \sin(60^\circ)} \right) [2.5d_{AB}^2 - 10d_{BC}^2 - 12.5d_{BE}^2 + 5d_{AE}^2 - 7.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2] \right] - 2886.75}{d_{AB}^2}$$

Esfuerzo normal en la barra AE

$$\sigma = \frac{4}{\pi} \frac{F_{AE}}{d_{AE}^2}$$

$$\sigma_{AE} = \frac{4}{\pi} \frac{\left[\left(-\frac{K_1 \cos(60^\circ)}{20 \sin(60^\circ)} \right) [2.5d_{AB}^2 - 10d_{BC}^2 - 12.5d_{BE}^2 + 5d_{AE}^2 - 7.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2] \right] + 1443.38}{d_{AE}^2}$$

Esfuerzo normal en la barra BE

$$\sigma = \frac{4}{\pi} \frac{F_{BE}}{d_{BE}^2}$$

$$\sigma_{BE} = \frac{4}{\pi} \frac{\left[\left(-\frac{K_1}{20 \sin(60^\circ)} \right) [12.5d_{AB}^2 - 2.5d_{BE}^2 + 5d_{AE}^2 - 7.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2] \right] + 2886.75}{d_{BE}^2}$$

Esfuerzo normal en la barra BC

$$\sigma = \frac{4}{\pi} \frac{F_{BC}}{d_{BC}^2}$$

$$\sigma_{BC} = \frac{4}{\pi} \frac{\left[\left(-\frac{K_1 \cos(60^\circ)}{20 \sin(60^\circ)} [5d_{AB}^2 + 10d_{BC}^2 + 15d_{BE}^2 + 10d_{AE}^2 + 15d_{EC}^2 + 10d_{ED}^2 + 5d_{CD}^2] \right) - 2886.75 \right]}{d_{BC}^2}$$

Esfuerzo normal en la barra EC

$$\sigma = \frac{4 F_{EC}}{\pi d_{EC}^2}$$

$$\sigma_{EC} = \frac{4}{\pi} \frac{\left[\left(\frac{K_1}{20 \sin(60^\circ)} [2.5d_{AB}^2 + 7.5d_{BE}^2 + 5d_{AE}^2 + 2.5d_{EC}^2 + 5d_{ED}^2 - 2.5d_{CD}^2] \right) + 2886.75 \right]}{d_{EC}^2}$$

Esfuerzo normal en la barra CD

$$\sigma = \frac{4 F_{CD}}{\pi d_{CD}^2}$$

$$\sigma_{CD} = \frac{4}{\pi} \frac{\left[\left(-\frac{K_1}{20 \sin(60^\circ)} [2.5d_{AB}^2 + 10d_{BC}^2 + 7.5d_{BE}^2 + 5d_{AE}^2 + 12.5d_{EC}^2 + 5d_{ED}^2 + 7.5d_{CD}^2] \right) - 2886.75 \right]}{d_{CD}^2}$$

Esfuerzo normal en la barra ED

$$\sigma = \frac{4 F_{ED}}{\pi d_{ED}^2}$$

$$\sigma_{ED} = \frac{4}{\pi} \frac{\left[\left(\frac{K_1 0.5}{20 \sqrt{3}/2} [2.5d_{AB}^2 + 10d_{BC}^2 + 7.5d_{BE}^2 + 5d_{AE}^2 + 12.5d_{EC}^2 + 5d_{DE}^2 + 7.5d_{CD}^2] \right) + 1443.38 \right]}{d_{ED}^2}$$

Finalmente, para calcular el factor de seguridad de cada una de las vigas, se utiliza la siguiente expresión. Es importante recalcar que el enunciado da el esfuerzo de fluencia y este corresponde a 205 MPa. No se escriben puntualmente cada una de las expresiones puesto que son muy repetitivas.

$$FS = \frac{\sigma_y}{\sigma}$$

Implementación

A continuación, se hace referencia a recursos que se consultaron para la implementación del algoritmo evolutivo.

- El algoritmo evolutivo como tal se implementó mediante la librería *deap* que permite escoger entre una variedad de operaciones de cruzamiento, mutación y selección. Tiene herramientas como *hall of fame* que almacena las mejores soluciones en general. Asimismo, se usó la función *eaSimple* que trae incorporado en su funcionamiento un algoritmo evolutivo como tal [4].
- El reporte de cifras relevante se calculó usando la librería *numpy*, precisamente las funciones que obtienen el máximo, mínimo, desviación estándar y promedio [5].
- La visualización de las curvas pertinentes se realizó mediante la librería *matplotlib* [6].

Justificación de distintas decisiones de diseño

Función de evaluación

Se tiene un problema en el que se desean optimizar los diámetros de una armadura *Warren*. Dentro de los requerimientos del problema se plantea que el factor de seguridad del sistema sea 1,5. Entonces, en palabras más sencillas, lo que se desea es que los diámetros sean lo más pequeños posibles, pero que ninguna viga tenga un factor de seguridad menor a 1,5. Por lo tanto, lo que se planteó fue una función definida a trozos.

$$\sum d + 1 \quad \text{si} \quad \min(FS_{AB}, FS_{AE}, \dots) < 1.5$$
$$\sum d \quad \text{si} \quad \min(FS_{AB}, FS_{AE}, \dots) \geq 1.5$$

Esta función se puede comprender como una sumatoria de los diámetros, que en el caso de que se tenga un resultado inválido, o sea, un factor de seguridad del sistema menor a 1,5, entonces se castiga a la solución sumando 1. Por la manera en que se plantea esta función de evaluación, se tiene entonces un problema de minimización.

Mutación

El criterio de selección principal para el operador de mutación fue la funcionalidad. Por la manera en que están planteados los individuos, los genes y el espacio de alelos, es fácil descartar algunas soluciones. Un ejemplo de función descartada es el operador de mutación gaussiana, este método utiliza una distribución de Gauss con una desviación estándar y promedio definida por el usuario, sin embargo, es posible que, con los valores tan pequeños correspondientes a las posibles soluciones, las mutaciones resulten en valores negativos que son incompatibles con la problemática planteada. Otra función descartada consiste en la uniforme, esto porque únicamente genera números enteros en un rango delimitado por el usuario, sin embargo, por la naturaleza del problema es necesario que los genes correspondan

a valores decimales. Finalmente, se seleccionó el método de mutación llamado *ShuffleIndex*, este consiste en intercambiar valores de ciertos genes de posición, como en este problema todos los genes tienen el mismo espacio de alelos, entonces este método es válido.

Espacio de alelos de los genes

Este criterio se definió utilizando la experimentación. Se corrió el algoritmo genético con un espacio de alelos definido aleatoriamente, y se pudo observar una tendencia de las soluciones a converger a valores relativamente pequeños (cerca de los 40 mm). Entonces se definió el espacio de alelos de 0 m a 20 cm, esto con el fin de procurar variabilidad, pero en un rango de soluciones válidas.

Métodos de selección

Este consiste en uno de los hiperparámetros del algoritmo, por lo que se seleccionaron dos métodos diferentes. Para escoger estos métodos se basó principalmente en que ambos tienen que funcionar para un problema de minimización, ya que así fue como se planteó la función de calidad del algoritmo. Además de esto, se quería incentivar la exploración y la explotación de forma balanceada, es por ello por lo que inicialmente, se seleccionaron el torneo y la ruleta. No obstante, al revisar la documentación de la librería DEAD, se notó que la ruleta no se puede utilizar en problemas de minimización, por tanto, al final se descartó. Entonces, fue acá donde se decidió probar con un método de selección elitista (se seleccionan los individuos de mejor calidad), de modo que se pueda comprobar el comportamiento del algoritmo al utilizar dos métodos de selección opuestos.

Método de cruce

Se seleccionó un único método para todas las iteraciones. Dentro de los parámetros de selección se destaca la naturaleza del problema y su compatibilidad con el método de cruce, esto porque existen métodos que varían las dimensiones del problema o podrían resultar en genes que se encuentran fuera del espacio de alelos previamente definido. Finalmente, se decidió utilizar el método uniforme, alguna de las ventajas de utilizar este método consiste en mantener una diversidad genética alta que brinda la oportunidad de mezclar los genes de los padres de manera óptima, evitando así convergencias prematuras y brinda la posibilidad de eludir estancamientos en mínimos locales que sin una mezcla sin sesgos entre padres podrían parecer ser la mejor solución.

Criterio de selección de mejores soluciones

Para esto se completó la tabla 1 que permitió saber el valor de calidad máxima obtenida en cada una de las combinaciones, para seguidamente seleccionar una muestra menor de combinaciones de hiperparámetros para estudiarlas con mayor profundidad. Para esto se observaron las gráficas y se planteó una discusión para cada uno de los casos.

Número de generaciones

Para definir el número de generaciones se plantearon tres opciones, en las que se plantea abordar la posibilidad de que se tengan casos en el que las generaciones sean pocas,

intermedias y muchas. Con esto se propone sacar conclusiones con respecto a la necesidad de incrementar el número de generaciones o si se llega a la convergencia con una cantidad reducida de generaciones.

Cantidad de población inicial

La población inicial se vio limitada, al igual que el número de generaciones, a tres posibilidades en las que se tienen dos casos extremos con pocos y muchos individuos, así como un caso intermedio. Con estas variaciones se espera poder analizar el impacto que tiene la cantidad de población inicial con la variabilidad y por ende con los resultados finales de las iteraciones.

Probabilidad de mutación

Finalmente, la probabilidad de mutación permite compensar situaciones de poca variación genética, como podría ser el caso de una población inicial reducida.

Estudio de hiperparámetros

En la tabla 1 se muestran los valores de calidad del mejor individuo de la iteración, así como la desviación estándar en la última generación. La selección de rangos de cada uno de los hiperparámetros que se varían en la tabla se argumenta en la sección anterior.

Tabla 1. Calidad y desviación de distintas variaciones de hiperparámetros usadas para evaluar un algoritmo evolutivo.

| | | | Número de generaciones | | | | | | | | |
|---------------------|--------------------------|------------|------------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| | | | 50 | | | 100 | | | 200 | | |
| | | | Población inicial | | | | | | | | |
| Método de selección | Probabilidad de mutación | | 200 | 5000 | 12000 | 200 | 5000 | 12000 | 200 | 5000 | 12000 |
| Torneo | 0.2 | Calidad | 0.04223 | 0.03519 | 0.03469 | 0.04487 | 0.03503 | 0.03480 | 0.04179 | 0.03536 | 0.03449 |
| | | Desviación | 0.00000 | 0.34319 | 0.32573 | 0.00000 | 0.34761 | 0.37752 | 0.00000 | 0.35250 | 0.35592 |
| | 0.1 | Calidad | 0.03801 | 0.03655 | 0.03476 | 0.04001 | 0.03451 | 0.03476 | 0.04659 | 0.03581 | 0.03440 |
| | | Desviación | 0.17059 | 0.22837 | 0.23825 | 0.00000 | 0.24758 | 0.25402 | 0.00000 | 0.24688 | 0.24697 |
| | 0.05 | Calidad | 0.04239 | 0.03528 | 0.03488 | 0.04099 | 0.03506 | 0.03442 | 0.04304 | 0.03580 | 0.03452 |
| | | Desviación | 0.00000 | 0.17759 | 0.16257 | 0.00000 | 0.16892 | 0.18272 | 0.00000 | 0.17968 | 0.17399 |
| Mejor calidad | 0.02 | Calidad | 0.08933 | 0.05268 | 0.05148 | 0.06094 | 0.04633 | 0.04308 | 0.05239 | 0.03984 | 0.04063 |
| | | Desviación | 0.37706 | 0.33495 | 0.33385 | 0.36962 | 0.29596 | 0.29263 | 0.31876 | 0.26307 | 0.25787 |
| | 0.01 | Calidad | 0.09415 | 0.05112 | 0.04549 | 0.05217 | 0.04209 | 0.04117 | 0.05610 | 0.03812 | 0.03742 |
| | | Desviación | 0.36989 | 0.33130 | 0.33598 | 0.34511 | 0.28877 | 0.28801 | 0.32628 | 0.26119 | 0.26039 |
| | 0.05 | Calidad | 0.10019 | 0.04797 | 0.04988 | 0.07040 | 0.04382 | 0.04114 | 0.05603 | 0.03931 | 0.04054 |
| | | Desviación | 0.36295 | 0.33263 | 0.33127 | 0.33099 | 0.28907 | 0.28955 | 0.36201 | 0.26245 | 0.25768 |

En la tabla anterior, se señalaron con color amarillo los casos donde se obtuvo los menores valores de calidad y una desviación baja, pero diferente de cero. Ahora, para cada uno de estos, se van a mostrar: las gráficas de calidad y desviación, los valores del mejor individuo y los factores de seguridad obtenidos. Esto con el objetivo de poder observar la influencia que tienen los diferentes hiperparámetros en los resultados del algoritmo.

Caso 1

En la tabla 2 se puede observar la combinación de hiperparámetros ejecutada en este caso.

Tabla 2. Parámetros del algoritmo evolutivo para el caso 1.

| Método de selección | Población inicial | Número de generaciones | Probabilidad de mutación |
|---------------------|-------------------|------------------------|--------------------------|
| Torneo | 5000 | 50 | 0.02 |

En la figura 12 se puede observar la gráfica que describe la evolución de la calidad y la desviación estándar en términos de la generación para el caso 1.

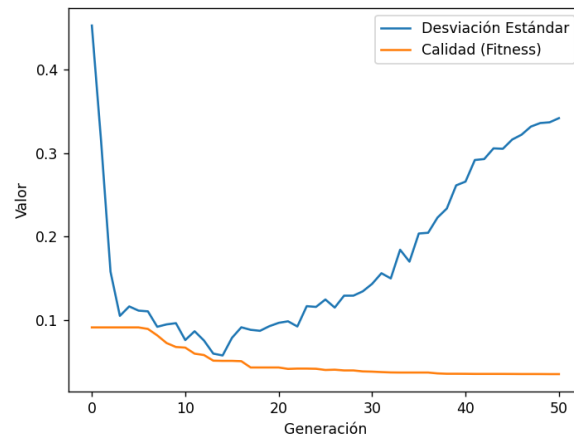


Figura 12. Gráfico de desviación estándar y calidad del mejor individuo vs generación para el caso 1.

La mejor calidad de esta combinación corresponde a 0.03522. Mientras que la desviación estándar de la última iteración fue 0.33217. Asimismo, en la tabla 3 se pueden observar los diámetros y factores obtenidos para el caso 1.

Tabla 3. Valores de diámetros y factores de seguridad para el caso 1.

| Diámetro | Valor (m) | Factor de seguridad |
|----------|-----------|---------------------|
| d_{ab} | 0.005495 | 1.516612 |
| d_{ae} | 0.004191 | 1.764731 |
| d_{bc} | 0.005336 | 1.589199 |
| d_{be} | 0.005336 | 1.954333 |
| d_{cd} | 0.005336 | 1.583311 |
| d_{ce} | 0.005336 | 1.586299 |
| d_{de} | 0.004191 | 1.584804 |

Caso 2

En la tabla 4 se puede observar la combinación de hiperparámetros ejecutada en este caso.

Tabla 4. Parámetros del algoritmo evolutivo para el caso 2.

| Método de selección | Población inicial | Número de generaciones | Probabilidad de mutación |
|---------------------|-------------------|------------------------|--------------------------|
| Torneo | 5000 | 50 | 0.05 |

En la figura 13 se puede observar la gráfica que describe la evolución de la calidad y la desviación estándar en términos de la generación para el caso 2

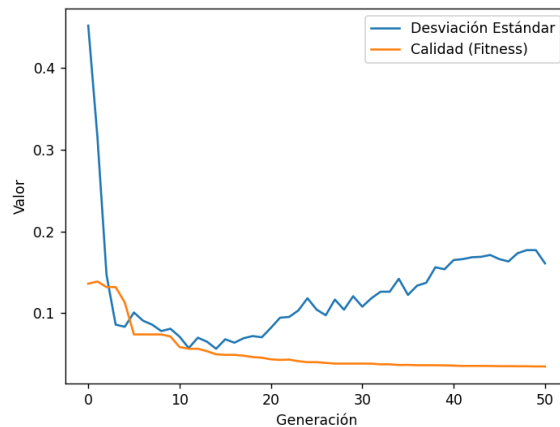


Figura 13. Gráfico de desviación estándar y calidad del mejor individuo vs generación para el caso 2.

La mejor calidad de esta combinación corresponde a 0.03528. Mientras que la desviación estándar de la última iteración fue 0.16092 Asimismo, en la tabla 5 se pueden observar los diámetros y factores obtenidos para el caso 2.

Tabla 5. Valores de diámetros y factores de seguridad para el caso 2.

| Diámetro | Valor (m) | Factor de seguridad |
|----------|-----------|---------------------|
| d_{ab} | 0.005721 | 1.644658 |
| d_{ae} | 0.004015 | 1.620599 |
| d_{bc} | 0.005282 | 1.566301 |
| d_{be} | 0.005282 | 1.793531 |
| d_{cd} | 0.005282 | 1.560542 |
| d_{ce} | 0.005282 | 1.554276 |
| d_{de} | 0.004015 | 1.552838 |

Caso 3

En la tabla 6 se puede observar la combinación de hiperparámetros ejecutada en este caso.

Tabla 6. Parámetros del algoritmo evolutivo para el caso 3.

| Método de selección | Población inicial | Número de generaciones | Probabilidad de mutación |
|---------------------|-------------------|------------------------|--------------------------|
| Torneo | 12000 | 50 | 0.2 |

En la figura 14 se puede observar la gráfica que describe la evolución de la calidad y la desviación estándar en términos de la generación para el caso 31.

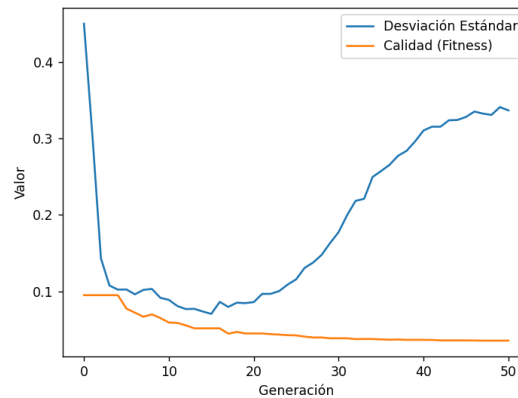


Figura 14. Gráfico de desviación estándar y calidad del mejor individuo vs generación para el caso 3.

La mejor calidad de esta combinación corresponde a 0.03530. Mientras que la desviación estándar de la última iteración fue 0.33693. Asimismo, en la tabla 7 se pueden observar los diámetros y factores obtenidos para el caso 3.

Tabla 7. Valores de diámetros y factores de seguridad para el caso 3.

| Diámetro | Valor (m) | Factor de seguridad |
|----------|-----------|---------------------|
| d_{ab} | 0.005542 | 1.538848 |
| d_{ae} | 0.003932 | 1.549416 |
| d_{bc} | 0.005302 | 1.666301 |
| d_{be} | 0.005468 | 2.078134 |
| d_{cd} | 0.005302 | 1.563181 |
| d_{ce} | 0.005435 | 1.645856 |
| d_{de} | 0.004322 | 1.564661 |

Caso 4

En la tabla 8 se puede observar la combinación de hiperparámetros ejecutada en este caso.

Tabla 8. Parámetros del algoritmo evolutivo para el caso 4.

| Método de selección | Población inicial | Número de generaciones | Probabilidad de mutación |
|---------------------|-------------------|------------------------|--------------------------|
| Torneo | 12000 | 50 | 0.1 |

En la figura 15 se puede observar la gráfica que describe la evolución de la calidad y la desviación estándar en términos de la generación para el caso 4.

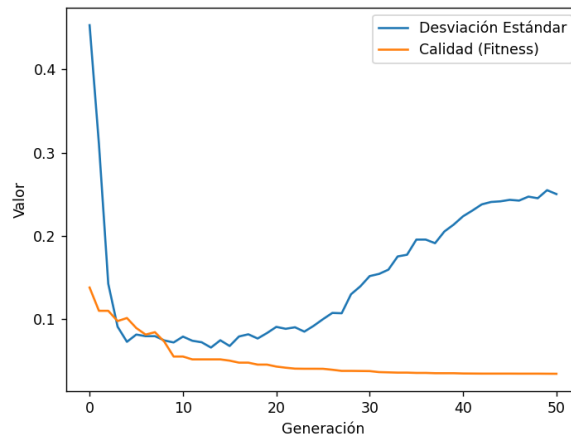


Figura 15. Gráfico de desviación estándar y calidad del mejor individuo vs generación para el caso 4.

La mejor calidad de esta combinación corresponde a 0.03473. Mientras que la desviación estándar de la última iteración fue 0.25031. Asimismo, en la tabla 9 se pueden observar los diámetros y factores obtenidos para el caso 4.

Tabla 9. Valores de diámetros y factores de seguridad para el caso 4.

| Diámetro | Valor (m) | Factor de seguridad |
|----------|-----------|---------------------|
| d_{ab} | 0.005487 | 1.509871 |
| d_{ae} | 0.003873 | 1.504638 |
| d_{bc} | 0.005356 | 1.633021 |
| d_{be} | 0.005409 | 1.668682 |
| d_{cd} | 0.005375 | 1.606642 |
| d_{ce} | 0.005356 | 1.598300 |
| d_{de} | 0.003873 | 1.596779 |

Caso 5

En la tabla 10 se puede observar la combinación de hiperparámetros ejecutada en este caso.

Tabla 10. Parámetros del algoritmo evolutivo para el caso 5.

| Método de selección | Población inicial | Número de generaciones | Probabilidad de mutación |
|---------------------|-------------------|------------------------|--------------------------|
| Torneo | 5000 | 100 | 0.1 |

En la figura 16 se puede observar la gráfica que describe la evolución de la calidad y la desviación estándar en términos de la generación para el caso 5.

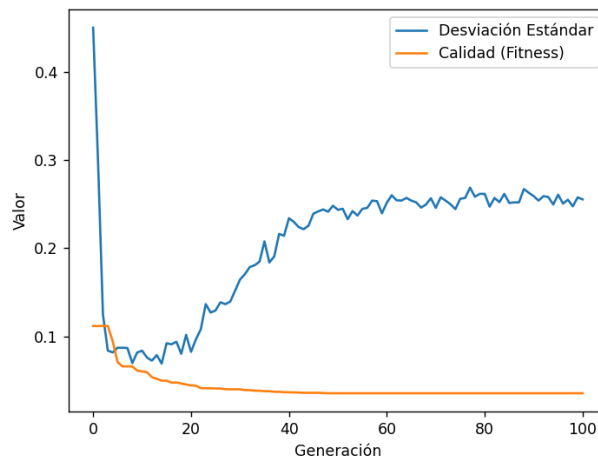


Figura 16. Gráfico de desviación estándar y calidad del mejor individuo vs generación para el caso 5.

La mejor calidad de esta combinación corresponde a 0.035230. Mientras que la desviación estándar de la última iteración fue 0.25548. Asimismo, en la tabla 11 se pueden observar los diámetros y factores obtenidos para el caso 5.

Tabla 11. Valores de diámetros y factores de seguridad para el caso 5.

| Diámetro | Valor (m) | Factor de seguridad |
|----------|-----------|---------------------|
| d_{ab} | 0.005580 | 1.564498 |
| d_{ae} | 0.004621 | 2.146189 |
| d_{bc} | 0.005309 | 1.573214 |
| d_{be} | 0.005309 | 1.598351 |
| d_{cd} | 0.005309 | 1.567364 |
| d_{ce} | 0.005309 | 1.570291 |
| d_{de} | 0.003791 | 1.568826 |

Caso 6

En la tabla 12 se puede observar la combinación de hiperparámetros ejecutada en este caso.

Tabla 12. Parámetros del algoritmo evolutivo para el caso 6.

| Método de selección | Población inicial | Número de generaciones | Probabilidad de mutación |
|---------------------|-------------------|------------------------|--------------------------|
| Torneo | 12000 | 100 | 0.05 |

En la figura 17 se puede observar la gráfica que describe la evolución de la calidad y la desviación estándar en términos de la generación para el caso 6.

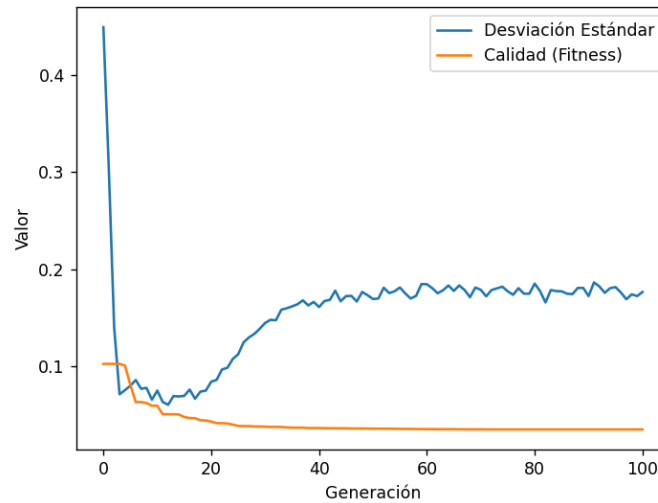


Figura 17. Gráfico de desviación estándar y calidad del mejor individuo vs generación para el caso 6.

La mejor calidad de esta combinación corresponde a 0.03471. Mientras que la desviación estándar de la última iteración fue 0.17644. Asimismo, en la tabla 13 se pueden observar los diámetros y factores obtenidos para el caso 6.

Tabla 13. Valores de diámetros y factores de seguridad para el caso 6..

| Diámetro | Valor (m) | Factor de seguridad |
|----------|-----------|---------------------|
| d_{ab} | 0.005531 | 1.538299 |
| d_{ae} | 0.004073 | 1.668730 |
| d_{bc} | 0.005259 | 1.543783 |
| d_{be} | 0.005259 | 1.845531 |
| d_{cd} | 0.005259 | 1.538204 |
| d_{ce} | 0.005259 | 1.541024 |
| d_{de} | 0.004072 | 1.539612 |

Caso 7

En la tabla 14 se puede observar la combinación de hiperparámetros ejecutada en este caso.

Tabla 14. Parámetros del algoritmo evolutivo para el caso 7.

| Método de selección | Población inicial | Número de generaciones | Probabilidad de mutación |
|---------------------|-------------------|------------------------|--------------------------|
| Torneo | 1200 | 200 | 0.2 |

En la figura 18 se puede observar la gráfica que describe la evolución de la calidad y la desviación estándar en términos de la generación para el caso 7.

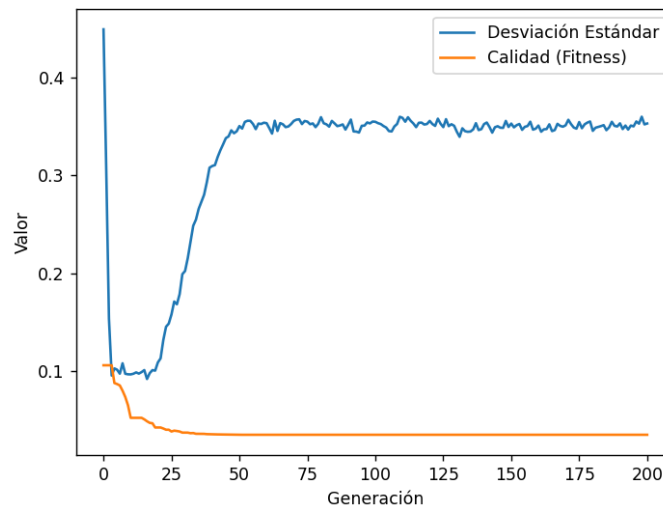


Figura 18. Gráfico de desviación estándar y calidad del mejor individuo vs generación para el caso 7.

La mejor calidad de esta combinación corresponde a 0.03469. Mientras que la desviación estándar de la última iteración fue 0.35310. Asimismo, en la tabla 15 se pueden observar los diámetros y factores obtenidos para el caso 7.

Tabla 15. Valores de diámetros y factores de seguridad para el caso 7.

| Diámetro | Valor (m) | Factor de seguridad |
|----------|-----------|---------------------|
| d_{ab} | 0.005475 | 1.508298 |
| d_{ae} | 0.004165 | 1.746373 |
| d_{bc} | 0.005222 | 1.522345 |
| d_{be} | 0.005222 | 1.930084 |
| d_{cd} | 0.005222 | 1.516870 |
| d_{ce} | 0.005222 | 1.519611 |
| d_{de} | 0.004165 | 1.518239 |

Caso 8

En la tabla 16 se puede observar la combinación de hiperparámetros ejecutada en este caso.

Tabla 16. Parámetros del algoritmo evolutivo para el caso 8.

| Método de selección | Población inicial | Número de generaciones | Probabilidad de mutación |
|---------------------|-------------------|------------------------|--------------------------|
| Mejor calidad | 12000 | 200 | 0.1 |

En la figura 19 se puede observar la gráfica que describe la evolución de la calidad y la desviación estándar en términos de la generación para el caso 8.

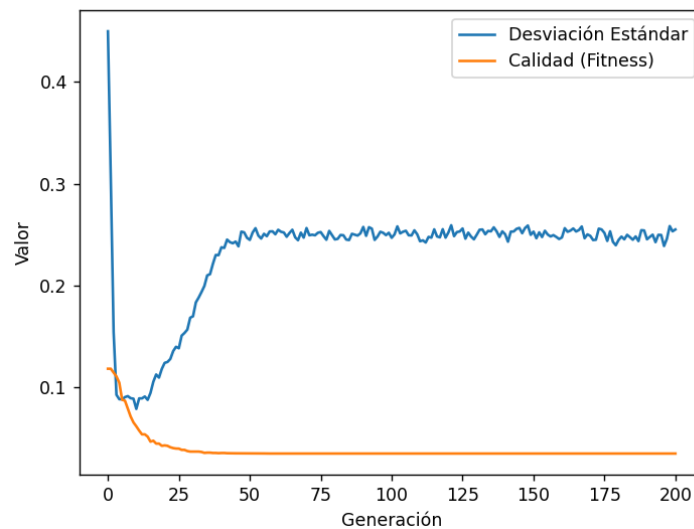


Figura 19. Gráfico de desviación estándar y calidad del mejor individuo vs generación para el caso 8.

La mejor calidad de esta combinación corresponde a 0.03428. Mientras que la desviación estándar de la última iteración fue 0.25486. Asimismo, en la tabla 17 se pueden observar los diámetros y factores obtenidos para el caso 8.

Tabla 17. Valores de diámetros y factores de seguridad para el caso 8.

| Diámetro | Valor (m) | Factor de seguridad |
|----------|-----------|---------------------|
| d_{ab} | 0.005477 | 1.510260 |
| d_{ae} | 0.003991 | 1.603437 |
| d_{bc} | 0.005205 | 1.512354 |
| d_{be} | 0.005205 | 1.771659 |
| d_{cd} | 0.005205 | 1.507018 |
| d_{ce} | 0.005205 | 1.509724 |
| d_{de} | 0.003990 | 1.508370 |

Caso 9

En la tabla 18 se puede observar la combinación de hiperparámetros ejecutada en este caso.

Tabla 18. Parámetros del algoritmo evolutivo para el caso 9.

| Método de selección | Población inicial | Número de generaciones | Probabilidad de mutación |
|---------------------|-------------------|------------------------|--------------------------|
| Mejor calidad | 5000 | 200 | 0.1 |

En la figura 20 se puede observar la gráfica que describe la evolución de la calidad y la desviación estándar en términos de la generación para el caso 9.

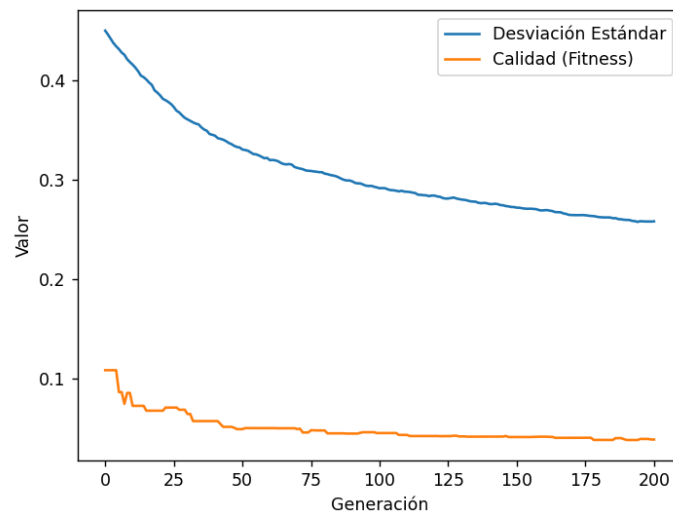


Figura 20. Gráfico de desviación estándar y calidad del mejor individuo vs generación para el caso 9.

La mejor calidad de esta combinación corresponde a 0.03856. Mientras que la desviación estándar de la última iteración fue 0.25861. Asimismo, en la tabla 19 se pueden observar los diámetros y factores obtenidos para el caso 9.

Tabla 19. Valores de diámetros y factores de seguridad para el caso 9.

| Diámetro | Valor (m) | Factor de seguridad |
|----------|-----------|---------------------|
| d_{ab} | 0.005858 | 1.701785 |
| d_{ae} | 0.004419 | 1.936688 |
| d_{bc} | 0.005587 | 2.023119 |
| d_{be} | 0.006020 | 3.325034 |
| d_{cd} | 0.005817 | 1.880947 |
| d_{ce} | 0.005387 | 1.616370 |
| d_{de} | 0.005469 | 1.737022 |

Caso 10

En la tabla 20 se puede observar la combinación de hiperparámetros ejecutada en este caso.

Tabla 20. Parámetros del algoritmo evolutivo para el caso 10.

| Método de selección | Población inicial | Número de generaciones | Probabilidad de mutación |
|---------------------|-------------------|------------------------|--------------------------|
| Torneo | 12000 | 200 | 0.1 |

En la figura 21 se puede observar la gráfica que describe la evolución de la calidad y la desviación estándar en términos de la generación para el caso 10.

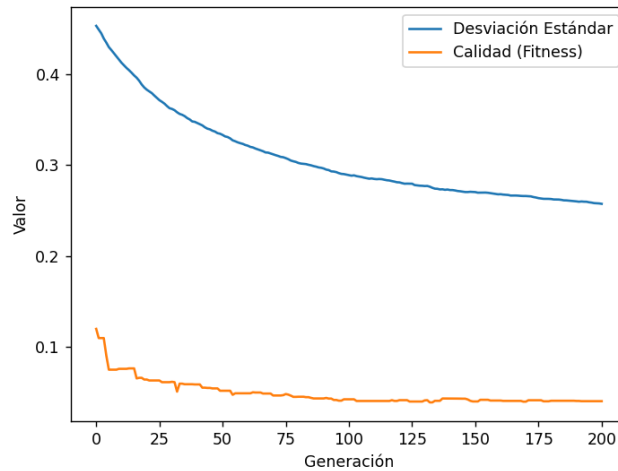


Figura 21. Gráfico de desviación estándar y calidad del mejor individuo vs generación para el caso 10.

La mejor calidad de esta combinación corresponde a 0.03872. Mientras que la desviación estándar de la última iteración fue 0.25748. Asimismo, en la tabla 21 se pueden observar los diámetros y factores obtenidos para el caso 10.

Tabla 21. Valores de diámetros y factores de seguridad para el caso 10.

| Diámetro | Valor (m) | Factor de seguridad |
|----------|-----------|---------------------|
| d_{ab} | 0.005703 | 1.634294 |
| d_{ae} | 0.004751 | 2.268320 |
| d_{bc} | 0.006438 | 1.574034 |
| d_{be} | 0.005310 | 3.345110 |
| d_{cd} | 0.005475 | 1.665810 |
| d_{ce} | 0.005554 | 1.718582 |
| d_{de} | 0.005486 | 2.306423 |

Análisis de resultados

A partir de los resultados de la tabla 1, lo primero que se puede notar es que existe una diferencia bastante marcada en la calidad de los individuos de acuerdo con el método de selección utilizado. Como es evidente en esta tabla, los resultados de calidad y desviación obtenidos con el torneo fueron muy similares entre sí, sin importar tanto los demás hiperparámetros. Mientras que, para el método de selección del mejor individuo, se obtuvo una mayor variabilidad en ambos valores.

Otro hiperparámetro que generó cambios significativos en la calidad del mejor individuo fue la población, pues en todos los casos, al incrementar su valor, se minimiza el valor de la calidad (se tiene un problema de minimización). De hecho, se puede notar que los mejores resultados se obtuvieron con poblaciones de 5000 y 12000 individuos. La razón de esto es que, al aumentar la población inicial, se incrementa la variabilidad genética en el algoritmo evolutivo, lo cual a su vez hace que durante el proceso de recombinación se puedan generar soluciones nuevas que optimicen el problema. En pocas palabras, se incentiva la exploración en el algoritmo. También, con el aumento de la población vino el aumento de generaciones, como se puede notar, las iteraciones resaltadas de amarillo tienen un mayor número de generaciones y esto se hizo con el fin de dejar al algoritmo explorar más recombinaciones y soluciones al problema, dando más posibilidad de que converja a una solución mejor.

Es importante resaltar que al reducir la población se obtuvo una desviación estándar de 0, lo cual indica que el algoritmo converge prematuramente. Esto es una señal de que no hay variabilidad en el pool genético y por ende las soluciones al problema van a ser muy similares entre ellas, además, como se puede notar en los valores de calidad obtenidos, tampoco se consideran soluciones óptimas.

Ahora, se va a profundizar más en el análisis de los casos que fueron considerados con mejores resultados. Lo primero que se nota en las gráficas es que, para el método de selección de torneo, la gráfica de variabilidad al inicio decrece rápidamente y luego comienza a subir y tiende a estabilizarse alrededor de un valor, este comportamiento es porque con este método de selección todos los individuos tienen la oportunidad de ser seleccionados para ser padres, aunque su calidad sea mala. Por otro lado, con el método de selección de mejor calidad, la curva decrece de manera más lenta y al final sigue una tendencia hacia un valor. Analizando ambos comportamientos, se puede interpretar que en cada iteración se crean individuos con buena y mala calidad, por esto es que, con el método de selección elitista la convergencia se da de manera más paulatina, pues siempre se escogen individuos con buena calidad para ser padres y por ende la tendencia es generar hijos con contenido genético similar a los mejores individuos iniciales.

Es importante resaltar porque existe una tendencia a un valor en específico para la desviación estándar. Esto se debe a la manera en que se planteó la función de calidad (por trozos) en la que se castigan las soluciones que, a pesar de estar cerca del factor de seguridad deseado, se encuentran por debajo del factor de seguridad mínimo del sistema. Por lo que van a existir

soluciones buenas (las más cercanas a cero) y soluciones castigadas (mayores a uno) resultando en una desviación estándar de un valor intermedio entre estas dos categorías.

No obstante, el hecho de que todos los individuos tengan una buena calidad no significa que este método de selección es mejor, ya que luego de unas generaciones, todos los individuos de la población van a ser muy similares entre sí, por lo que, al hacer la recombinación, no se generan soluciones nuevas y por ende al algoritmo le cuesta más encontrar una mejor solución. Esto se ve reflejado en los valores de calidad obtenidos, así como en los factores de seguridad. Como se puede apreciar en las dos mejores combinaciones en las que se utilizó el método de selección elitista, los valores de calidad obtenidos fueron significativamente superiores al peor caso obtenido con el torneo. En el mejor caso con el método elitista tenía calidad de 0.03856 (caso 10), mientras que la peor calidad obtenida con torneo fue de 0.03530 (caso 3), como se puede notar, la diferencia es 0.00326 m, tomando en cuenta que los diámetros están en el rango de los milímetros, esta diferencia sí se considera significativa en los diámetros. En cuanto a los factores de seguridad, se puede notar que en el mejor caso del método elitista genera factores de seguridad bastante alejados del objetivo (1,5), lo cual es un indicativo que las vigas están sobredimensionadas y que por ende esos no son los diámetros óptimos.

Por otro lado, en cuanto a los casos del 1 al 8, todos presentaron valores de calidad bastante similares y por ende diámetros de las vigas muy similares. Entre estos se destacan los casos 6, 7 y 8, que fueron los que generaron mejores calidades (0.03471, 0.0346 y 0.03428). Ahora, si también se toman en cuenta los factores de seguridad, sumando todos los factores de seguridad para cada caso se obtiene: 11.21, 11.26 y 10.92. Como se puede notar, entonces la combinación de hiperparámetros que mejor optimiza tanto los diámetros como los factores de seguridad es el caso 8.

Además, algo que se notó en la mayoría de los casos es que hay algunos genes (diámetros) que dieron iguales entre ellos. Esto puede deberse a varios factores, el primero de ellos es la naturaleza del problema, pues, en una armadura, se busca que la distribución de esfuerzos en los elementos sea uniforme, por eso los esfuerzos en los elementos van a ser muy similares entre sí y por ende los diámetros también. Otro factor que puede influir en esto es el tipo y la probabilidad de mutación seleccionadas, ya que como se mencionó anteriormente, el método *ShuffleIndex* lo que hace es intercambiar los diámetros en un individuo. Entonces, al tener una alta probabilidad de mutación, varios individuos pueden tener este intercambio de genes y luego con el paso de las generaciones esto puede hacer que varios genes lleguen a tener el mismo valor.

A continuación, se van a presentar los resultados de tres iteraciones con los hiperparámetros del mejor caso obtenido para observar si existen diferencias en las soluciones brindadas.

Tabla 22. Diámetros obtenidos para el mejor individuo utilizando una población de 12000 individuos, 200 generaciones, 10% de probabilidad de mutación y selección por torneo.

| Diámetro | Iteración 1 (m) | Iteración 2 (m) | Iteración 3 (m) |
|----------|-----------------|-----------------|-----------------|
| d_{ab} | 0.005514 | 0.005522 | 0.005578 |
| d_{ae} | 0.003971 | 0.004135 | 0.004028 |
| d_{bc} | 0.005209 | 0.005242 | 0.005252 |
| d_{be} | 0.005210 | 0.005242 | 0.005252 |
| d_{cd} | 0.005210 | 0.005242 | 0.005252 |
| d_{ce} | 0.005209 | 0.005242 | 0.005252 |
| d_{de} | 0.003971 | 0.004135 | 0.004028 |

Como se puede notar, en la tabla 22, los diámetros suelen tener valores muy similares entre cada iteración, además, no existe ningún elemento que sea significativamente más pequeño que los demás y que por ende se pueda tomar como diámetro 0. Esto tiene sentido desde el punto de vista físico del problema, ya que, a partir del análisis estático realizado al inicio de la solución del problema, se obtuvo, que todos los elementos de la armadura soportaban una parte de la carga externa de 5 kN y por ende no existen elementos de fuerza cero en la armadura. Los “elementos de fuerza cero” en estática son aquellos que no soportan cargas que se colocan con el objetivo de aumentar la estabilidad de la armadura o bien para agregar apoyo en caso de que la condición de cargas en la armadura cambie [3]. En conclusión, para este problema en particular, no tiene sentido que se eliminen elementos de la armadura, ya que las cargas se distribuyen en todos ellos y al quitar un elemento se cambiaría la condición de carga y por ende sería un problema diferente.

Conclusiones

1. Para obtener mejores resultados en el algoritmo evolutivo desarrollado para este problema específico se debe promover de manera balanceada la exploración y explotación, de modo que exista cierta variabilidad en el pool genético, pero procurando llegar a cierta convergencia con los recursos computacionales definidos.
2. Para este problema en específico se determinó que un método de selección elitista no brinda buenos resultados, por lo que se concluye que el torneo es un mejor método de selección.
3. Al incrementar la población inicial y el número de generaciones en el algoritmo genético se aumenta la diversidad en la población y por ende se obtienen soluciones con una calidad más optimizada.
4. En todas las iteraciones se tuvieron desviaciones estándar intermedias debido la forma en que se planteó la función de calidad del algoritmo.
5. La combinación de hiperparámetros que optimizó de mejor manera la armadura fue: una población inicial de 12000 individuos, 200 generaciones, una probabilidad de mutación del 10% y método de selección de torneo.
6. Para la armadura Warren que se quería optimizar con la condición de carga propuesta se considera que no tiene sentido eliminar elementos de la armadura.

Bibliografía

- [1] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Second. Amsterdam: Springer, 2015.
- [2] F. Vargas Nieto and J. Montoya-Torres, “Figura 2. Seudocódigo del Algoritmo Evolutivo. - researchgate,” Implementación de un procedimiento basado en algoritmos evolutivos para programar la producción de marquillas estampadas por transferencia térmica, https://www.researchgate.net/figure/Figura-2-Seudocodigo-del-algoritmo-evolutivo_fig2_220136684 [Accessed Nov. 14, 2023].
- [3] R. C. Hibbeler, *Engineering Mechanics: Statics & Dynamics*, 14th ed. Upper Saddle River: Pearson Prentice Hall, 2016.
- [4] DEAP Project, “DEAP documentation” DEAP documentation - DEAP 1.4.1 documentation, <https://deap.readthedocs.io/en/master> [Accessed: Nov. 07, 2023].
- [5] “NumPy documentation,” NumPy documentation - NumPy v1.26 Manual. [Online]. Available: <https://numpy.org/doc/stable/>. [Accessed: Nov. 07, 2023]
- [6] C. C. Madrigal Sánchez, «Manual de trabajo #1: Redes neuronales simples utilizando Keras/TensorFlow», 2023.

Anexos

Anexo 1

Código utilizado para generar el algoritmo evolutivo.

```
from deap import base, creator, tools, algorithms
import random
import numpy
import array
import matplotlib.pyplot as plt
import math

K1 = 110563.25
sin_60 = math.sqrt(3)/2
SIGMA_Y = 205e6 # 205 MPa

def Prueba(individuo):
    # Función utilizada para obtener los factores de seguridad del mejor
    # individuo del algoritmo
    # Se extraen los valores de diámetros desde el individuo.
    d_ab = individuo[0]
    d_ae = individuo[1]
    d_bc = individuo[2]
    d_be = individuo[3]
    d_cd = individuo[4]
    d_ce = individuo[5]
    d_de = individuo[6]
    # Se calcula el esfuerzo en cada una de las vigas.
    e_ab = (4/numpy.pi)*((((K1/20*sin_60)*((2.5*d_ab**2) - (10*d_bc**2) -
    (12.5*d_be) + (5*d_ae**2) - (7.5*d_ce**2) + (5*d_de**2) + (7.5*d_cd**2)))-
    2886.75)/d_ab**2)
    e_ae = (4/numpy.pi)*((((-(K1*0.5)/20*sin_60)*(2.5*d_ab**2 - 10*d_bc**2
    - 12.5*d_be + 5*d_ae**2 - 7.5*d_ce**2 + 5*d_de**2 +
    7.5*d_cd**2))+1443.38)/d_ae**2)
    e_be = (4/numpy.pi)*((((-(K1/20*sin_60)*(12.5*d_ab**2 - 2.5*d_be**2 +
    5*d_ae**2 - 7.5*d_ce**2 + 5*d_de**2 + 7.5*d_cd**2))+2886.75)/d_be**2)
    e_ed =
    (4/numpy.pi)*((K1*(0.5)/(20*sin_60))*((2.5*d_ab**2)+(10*d_bc**2)+(7.5*d_be
    **2)+(5*d_ae**2)+(12.5*d_ce**2)+(5*d_de**2)+(7.5*d_cd**2))+1443.38)/(d_de*
    *2)
    e_cd = (4/numpy.pi)*((-
    K1/(20*sin_60))*((2.5*d_ab**2)+(10*d_bc**2)+(7.5*d_be**2)+(5*d_ae**2)+(12.
    5*d_ce**2)+(5*d_de**2)+(7.5*d_cd**2))-2886.75)/(d_cd**2)
```

```

e_ec =
(4/numpy.pi)*((K1/(20*sin_60))*((2.5*d_ab**2)+(7.5*d_be**2)+(5*d_ae**2)+(2
.5*d_ce**2)+(5*d_de**2)-(2.5*d_cd**2))+2886.75)/(d_ce**2)
e_bc = (4/numpy.pi)*((-
K1*(0.5)/(20*sin_60))*((5*d_ab**2)+(10*d_bc**2)+(15*d_be**2)+(10*d_ae**2)+
(15*d_ce**2)+(10*d_de**2)+(5*d_cd**2))-2886.75)/(d_bc**2)

# Se calcula el Factor de Seguridad en cada una de las vigas.
FS_ab = SIGMA_Y/abs(e_ab)
FS_ae = SIGMA_Y/abs(e_ae)
FS_be = SIGMA_Y/abs(e_be)
FS_ed = SIGMA_Y/abs(e_ed)
FS_cd = SIGMA_Y/abs(e_cd)
FS_ec = SIGMA_Y/abs(e_ec)
FS_bc = SIGMA_Y/abs(e_bc)

FS_array = [FS_ab, FS_ae, FS_be, FS_ed, FS_cd, FS_ec, FS_bc]

print(FS_array)

def funcionEval(individuo):
    # Función que calcula la calidad de un individuo y los factores de
    # seguridad de los elementos con base en las ecuaciones estáticas
    obtenidas
    # Se extraen los valores de diámetros desde el individuo.
    d_ab = individuo[0]
    d_ae = individuo[1]
    d_bc = individuo[2]
    d_be = individuo[3]
    d_cd = individuo[4]
    d_ce = individuo[5]
    d_de = individuo[6]
    # Se calcula el esfuerzo en cada una de las vigas.
    e_ab = (4/numpy.pi)*((((K1/20*sin_60))*((2.5*d_ab**2) - (10*d_bc**2) -
(12.5*d_be) + (5*d_ae**2) - (7.5*d_ce**2) + (5*d_de**2) + (7.5*d_cd**2)))-
2886.75)/d_ab**2)
    e_ae = (4/numpy.pi)*((((-(K1*0.5)/20*sin_60))*((2.5*d_ab**2 - 10*d_bc**2
- 12.5*d_be + 5*d_ae**2 - 7.5*d_ce**2 + 5*d_de**2 +
7.5*d_cd**2))+1443.38)/d_ae**2)
    e_be = (4/numpy.pi)*((((-(K1/20*sin_60))*((12.5*d_ab**2 - 2.5*d_be**2 +
5*d_ae**2 - 7.5*d_ce**2 + 5*d_de**2 + 7.5*d_cd**2))+2886.75)/d_be**2)
    e_ed =
(4/numpy.pi)*((K1*(0.5)/(20*sin_60))*((2.5*d_ab**2)+(10*d_bc**2)+(7.5*d_be
**2)+(5*d_ae**2)+(12.5*d_ce**2)+(5*d_de**2)+(7.5*d_cd**2))+1443.38)/(d_de*
*2)

```

```

    e_cd = (4/numpy.pi)*((-
K1/(20*sin_60))*((2.5*d_ab**2)+(10*d_bc**2)+(7.5*d_be**2)+(5*d_ae**2)+(12.
5*d_ce**2)+(5*d_de**2)+(7.5*d_cd**2))-2886.75)/(d_cd**2)
    e_ec =
(4/numpy.pi)*((K1/(20*sin_60))*((2.5*d_ab**2)+(7.5*d_be**2)+(5*d_ae**2)+(2
.5*d_ce**2)+(5*d_de**2)-(2.5*d_cd**2))+2886.75)/(d_ce**2)
    e_bc = (4/numpy.pi)*((-
K1*(0.5)/(20*sin_60))*((5*d_ab**2)+(10*d_bc**2)+(15*d_be**2)+(10*d_ae**2)+
(15*d_ce**2)+(10*d_de**2)+(5*d_cd**2))-2886.75)/(d_bc**2)

    # Se calcula el Factor de Seguridad en cada una de las vigas.
    FS_ab = SIGMA_Y/abs(e_ab)
    FS_ae = SIGMA_Y/abs(e_ae)
    FS_be = SIGMA_Y/abs(e_be)
    FS_ed = SIGMA_Y/abs(e_ed)
    FS_cd = SIGMA_Y/abs(e_cd)
    FS_ec = SIGMA_Y/abs(e_ec)
    FS_bc = SIGMA_Y/abs(e_bc)

    FS_array = [FS_ab, FS_ae, FS_be, FS_ed, FS_cd, FS_ec, FS_bc]
    if not all(i >= 1.5 for i in FS_array) or not all(j >= 0 for j in
individuo):
        error = 1
    else:
        error = sum(individuo)
    return [error, ]

if __name__ == "__main__":

    lim_inferior = 0
    lim_superior = 0.1
    poblacion = 5000
    n_generaciones = 200
    probabilidad_mutacion_ind = 0.2 # Es particular de la mutación
Gaussiana
    probabilidad_mutacion_gen = 0.05 # Este es el hiperparámetro
    # Definición del toolbox
    toolbox = base.Toolbox()
    # El fitness que manejará el toolbox será una función con el peso 1
    # (maximización con peso unitario para cada atributo)
    creator.create('Fitness_func', base.Fitness, weights=(-1.0,))
    # -El individuo que manejará el toolbox será un array de floats
    creator.create('Individuo', array.array,
                    fitness=creator.Fitness_func, typecode='f')

```



```

# Registro de la función de evaluación, usando lo definido previamente
en el código
toolbox.register('evaluate', funcionEval)
# Registro del método de cómo generar un atributo para este caso:
floats al azar
toolbox.register('atributo', random.uniform, a=lim_inferior,
                b=lim_superior)
# Se genera un atributo de float al azar 3 veces, y se guarda en un
Individuo
toolbox.register('individuo_gen', tools.initRepeat, creator.Individuo,
                toolbox.atributo, n=7)
# Luego, se registra en toolbox una operación para crear la población
toolbox.register('Poblacion', tools.initRepeat, list,
                toolbox.individuo_gen, n=poblacion)
# Para ello, llama unas 30 veces a la función 'individuo_gen', de
manera
# que queda generada una población de 'Individuo's.
# Se utiliza la función registrada para generar una población
popu = toolbox.Poblacion()
# Método de cruce de dos puntos
toolbox.register('mate', tools.cxUniform, indpb=0.5)

# Para la mutación, se utiliza el método de mutación Gaussiana
toolbox.register('mutate', tools.mutShuffleIndexes,
                indpb=probabilidad_mutacion_ind)

# Para la selección, se utiliza el método de torneo y el de mejor
calidad
toolbox.register('select', tools.selTournament, tournsize=2)
#toolbox.register('select', tools.selBest)

# Hall of Fame: presentación de los mejores 10 individuos
hof = tools.HallOfFame(10)

# Estadísticas del fitness general de la población
stats = tools.Statistics(lambda indiv: indiv.fitness.values)
stats.register('avg', numpy.mean) # Promedio de la gen
stats.register('std', numpy.std) # Desviación estándar de los
individuos
stats.register('min', numpy.min) # Fitness mínimo de la gen
stats.register('max', numpy.max) # Fitness máximo de la gen
# Una vez que todo está registrado y establecido, ya se puede comenzar
# a correr el algoritmo evolutivo.
popu, logbook = algorithms.eaSimple(popu, toolbox, cxpb=0.2,
                                   mutpb=probabilidad_mutacion_gen,

```

```

nngen=n_generaciones, stats=stats,
halloffame=hof, verbose=True)

print(popu)
print('-----')
print(logbook)

# Gráficas de los resultados
# X axis parameter:
xaxis = logbook.select("gen")
# Y axis parameter:
quality = logbook.select("min")
std = logbook.select("std")
plt.plot(xaxis, std, label='Desviación Estándar')
plt.plot(xaxis, quality, label='Calidad (Fitness)')
plt.xlabel('Generación')
plt.ylabel('Valor')
plt.legend()
plt.show()

# Se imprime el último valor de calidad y desviación estándar
print('{0:.5f}'.format(funcionEval(hof[0])[0]))
print('{0:.5f}'.format(std[-1]))

```