

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA CIVIL Y AMBIENTAL**

**ESTUDIO DE OPTIMIZACIÓN APLICANDO ALGORITMOS  
GENÉTICOS EN ESTRUCTURAS PARAMÉTRICAS TIPO  
CELOSIA DE ACERO**

**TESIS DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO  
CIVIL, MENCIÓN EN ESTRUCTURAS**

**GARÓFALO GAIBOR EDISON HERNÁN**

**edisongarofalo23@gmail.com**

**DIRECTOR: ING. MBA. GERMÁN VINICIO LUNA HERMOSA**

**german.luna@epn.edu.ec**

**CODIRECTOR: ING. EDGAR DAVID MORA MARTÍNEZ Msc.**

**edgar.mora@epn.edu.ec**

**Quito, abril 2022**

## DECLARACIÓN

Yo Edison Hernán Garófalo Gaibor, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido en la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

---

**EDISON HERNÁN GARÓFALO**

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Edison Hernán Garófalo Gaibor, bajo mi supervisión.

---

**Ing. MBA. Germán Luna Hermosa**  
**DIRECTOR DE PROYECTO**

---

**Ing. Edgar Mora Martínez, Msc.**  
**CODIRECTOR DE PROYECTO**

## **DEDICATORIA**

Este trabajo dedico a mi madre que fue mi apoyo fundamental desde los inicios hasta el final de mis estudios, a mi hijo que me dio una razón más para seguir adelante, a mi esposa Jessica Naranjo por sus palabras de aliento para seguir adelante hasta el final, al ingeniero Edgar Mora por su paciencia, apoyo y guía en este trabajo y a todos los profesores de la Carrera de Ingeniería Civil que aportaron en mi formación académica.

## ÍNDICE DE CONTENIDOS

DECLARACIÓN .....	I
CERTIFICACIÓN .....	II
DEDICATORIA .....	III
ÍNDICE DE CONTENIDOS .....	IV
LISTADO DE TABLAS .....	XII
LISTADO DE FIGURAS .....	XIV
RESUMEN .....	XVI
ABSTRACT .....	XVII
<b>CAPÍTULO 1.....</b>	<b>1</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
1.1 ANTECEDENTES .....	1
1.2 OBJETIVOS .....	2
1.2.1. OBJETIVO GENERAL .....	2
1.2.2. OBJETIVOS ESPECÍFICOS .....	2
1.3 PLANTEAMIENTO DEL PROBLEMA.....	3
1.4 ALCANCE .....	5
1.5 JUSTIFICACIÓN .....	5
1.5.1. JUSTIFICACIÓN TEÓRICA .....	5
1.5.2. JUSTIFICACIÓN METODOLÓGICA.....	5

1.5.3. JUSTIFICACIÓN PRÁCTICA.....	6
<b>CAPÍTULO 2.....</b>	<b>7</b>
<b>MARCO TEÓRICO.....</b>	<b>7</b>
2.1 INTRODUCCIÓN.....	7
2.2 ESTRUCTURAS ARTICULADAS.....	7
2.3 FUNDAMENTOS DEL MÉTODO MATRICIAL DE RIGIDEZ. ....	7
2.3.1. MATRIZ DE RIGIDEZ DE UN ELEMENTO O TIPO ARMADURA.....	8
2.3.2. MATRIZ DE PASO DE UN ELEMENTO TIPO ARMADURA.....	9
2.3.3. MATRIZ DE RIGIDEZ GLOBAL DE UN ELEMENTO TIPO ARMADURA.....	10
2.3.4. MATRIZ DE RIGIDEZ AMPLIADA DE UNA ARMADURA.....	11
2.3.5. MATRIZ DE COMPATIBILIDAD POR ELEMENTO.....	11
2.3.6. MATRIZ DE COMPATIBILIDAD GLOBAL.....	12
2.3.7. MATRIZ DE RIGIDEZ GLOBAL DE LA ESTRUCTURA TIPO ARMADURA.....	12
2.3.8. DEFORMACIONES. ....	12
2.3.9. FUERZAS INTERNAS DE LOS ELEMENTOS DE LA ESTRUCTURA.....	13
2.4 MÉTODOS DE DISEÑO.....	13

2.4.1. MÉTODO DE DISEÑO POR RESISTENCIA USANDO DISEÑO EN BASE A FACTORES DE CARGA Y RESISTENCIA (LRFD).....	14
2.4.2. MÉTODO DE DISEÑO POR RESISTENCIA USANDO DISEÑO EN BASE A FACTORES DE RESISTENCIAS ADMISIBLES (ASD). ....	14
2.5 DISEÑO PARAMÉTRICO.....	15
2.6 OPTIMIZACIÓN.....	15
2.6.1. TIPOS DE OPTIMIZACIÓN EN DISEÑO DE ESTRUCTURAS.....	16
2.6.2. ALGORITMOS GENÉTICOS. ....	16
<b>CAPÍTULO 3.....</b>	<b>20</b>
<b>METODOLOGÍA DE DISEÑO DEL ALGORITMO. ....</b>	<b>20</b>
3.1 INTRODUCCIÓN.....	20
3.2 PROGRAMA PARA ANÁLISIS DE ARMADURAS EN 2D .....	20
3.2.1. USO DEL PROGRAMA DE ANALISIS MATRICIAL.....	20
3.2.2. EJEMPLO DE CALCULO POR EL MÉTODO DE ANALISIS MATRICIAL. ....	23
3.2.3. VALIDACIÓN EN SAP 2000 DEL ALGORITMO DEL MÉTODO MATRICIAL.....	30
3.3 ALGORITMO GENÉTICO DE SECCIONES.....	32
3.3.1. CROMOSOMA DE SECCIONES.....	32

3.3.2. POBLACIÓN INICIAL.....	33
3.3.3. FUNCIÓN OBJETIVO.....	33
3.3.4. MÉTODO DE SELECCIÓN.....	34
3.3.5. DIAGRAMA DE FLUJO DEL ALGORITMO GENÉTICO DE SECCIONES.....	34
3.4 PARAMETRIZACIÓN DE LA ESTRUCTURA.....	35
3.4.1. PARAMETRIZACIÓN TIPO PRATT.....	35
3.4.2. PARAMETRIZACIÓN ARMADURA CIRCULAR.....	37
3.5 AUTOMATIZACIÓN DEL PROGRAMA PARA ARMADURAS EN DOS DIMENSIONES.....	39
3.6 ALGORITMO GENÉTICO DE SECCIONES FORMA TOPOLOGÍA.....	40
3.6.1. DIAGRAMA DE FLUJO DEL ALGORITMO GENÉTICO DE SECCIONES FORMA Y TOPOLOGÍA.....	40
3.6.2. CROMOSOMA DE FORMA Y TOPOLOGÍA.....	43
<b>CAPÍTULO 4.....</b>	<b>44</b>
<b>VALIDACIÓN DEL MÉTODO MATRICIAL Y LA PARAMETRIZACIÓN DE LA ESTRUCTURA.....</b>	<b>44</b>
4.1 INTRODUCCIÓN.....	44
4.2 VALIDACIÓN DEL METODO MATRICIAL.....	44
4.2.1. SOLUCIÓN DEL EJERCICIO UNO VALIDACIÓN DEL METODO MATRICIAL.....	45

4.2.2. SOLUCIÓN DEL EJERCICIO UNO EN SAP 2000 .....	47
4.3 VALIDACIÓN DE LA PARAMETRIZACIÓN .....	47
4.3.1. SOLUCIÓN DEL EJERCICIO DOS Y TRES POR EL MÉTODO MATRICIAL.....	48
4.3.2. SOLUCIÓN DEL EJERCICIO DOS Y TRES EN SAP 2000 .....	50
<b>CAPÍTULO 5.....</b>	<b>53</b>
<b>FORMULACIÓN Y RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE ARMADURAS PLANAS.....</b>	<b>53</b>
5.1 INTRODUCCIÓN.....	53
5.2 PROBLEMAS DE OPTIMIZACIÓN DE SECCIONES EN ARMADURAS PLANAS.....	53
5.2.1 ARMADURA PLANA DE TRES ELEMENTOS.....	54
5.2.2. ARMADURA PLANA DE DIEZ ELEMENTOS .....	58
5.3 PROBLEMAS OPTIMIZACIÓN DE SECCIONES, FORMA Y TOPOLOGÍA EN ARMADURAS PLANAS. ....	63
5.3.1. ARMADURA ISOSTÁTICA DE LUZ DE 20 M ETROS ....	64
5.3.2. ARMADURA HIPERESTÁTICA DE GRADO UNO DE LUZ DE 20 METROS .....	67
<b>CAPÍTULO 6.....</b>	<b>72</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>72</b>
6.1 CONCLUSIONES.....	72

6.2 RECOMENDACIONES.....	73
<b>BIBLIOGRAFÍA .....</b>	<b>75</b>
ANEXOS.....	77
ANEXO No1 FUNCIONES DEL ANÁLISIS MATRICIAL SIN AUTOMATIZACIÓN.....	78
A1.1. Función M_KAP .....	79
A1.2. Función M_TAP .....	79
A1.3. Función Ens .....	80
A1.4. Programa validación del análisis matricial .....	80
A1.5 Función Acum2 .....	84
ANEXO No2 ALGORITMO GÉNÉTICO DE SECCIONES .....	85
A2.1. Función reproducción .....	86
A2.2. Función de mutación .....	87
A2.3. Método de selección por ruleta .....	87
A2.4. Método de selección por torneo .....	88
ANEXO No3 PARAMETRIZACIÓN.....	89
A3.1 Programa fparametPrant.....	90
A3.2 Programa fparametricaCircular.....	91
ANEXO No4 FUNCIONES DEL ANÁLISIS MATRICIAL CON AUTOMATIZACIÓN.....	94
A4.1. Función fNodos1 .....	95

A4.2. Función fCoordNodos .....	96
A4.3. Función fgradoslibertadAG.....	99
A4.4. Función fDibujar .....	100
A4.5. Función M_TAP2.....	101
A4.6. Función M_KAP2AG .....	102
<b>ANEXO No5 VALIDACIÓN DEL ANÁLISIS MATRICIAL CON AUTOMATIZACIÓN Y CON PARAMETRIZACIÓN.....</b>	<b>103</b>
A5.1. Programa validación del análisis matricial con automatización. ....	104
A5.2. Programa validación del análisis matricial estructura paramétrica .....	109
<b>ANEXO No6 ÁREAS Y PESOS DE PERFILES COMERCIALES A UTILIZAR .....</b>	<b>114</b>
A6.1. Áreas y pesos de perfiles comerciales tubulares.....	115
<b>ANEXO No7 OPTIMIZACIÓN DE SECCIONES .....</b>	<b>116</b>
A7.1. Programa optimización de secciones estructura de tres elementos.....	117
A7.2. Optimización de secciones estructura de diez elementos.....	126
<b>ANEXO No8 OPTIMIZACIÓN FORMA, TOPOLOGÍA Y SECCIONES.....</b>	<b>128</b>

A8.1. Optimización de forma, topología y secciones de estructura isostática de luz de veinte metros. ....	129
A8.2. Optimización de forma, topología y secciones de estructura de luz de veinte metros, hiperestática de grado uno.....	140

## LISTADO DE TABLAS

<b>Tabla 2.1.</b> <i>Ejemplo de un cromosoma.</i> .....	17
<b>Tabla 3.1.</b> Resultados de desplazamientos en la estructura del análisis matricial.....	28
<b>Tabla 3.2.</b> <i>Resultados de acciones y desplazamientos del análisis matricial.</i> .....	29
<b>Tabla 3.3.</b> <i>Reacciones en los apoyos, método matricial.</i> .....	29
<b>Tabla 3.4.</b> <i>Comprobación de equilibrio por el método matricial.</i> .....	30
<b>Tabla 3.5.</b> <i>Cromosoma de secciones</i> .....	32
<b>Tabla 3.6.</b> <i>Cromosoma de forma y topología</i> .....	43
<b>Tabla 4.1.</b> <i>Propiedades de la estructura</i> .....	45
<b>Tabla 4.2.</b> <i>Resultados del ejercicio uno utilizando el método matricial</i> .....	46
<b>Tabla 4.3.</b> <i>Resultados del ejercicio uno utilizando el programa SAP2000</i> .....	47
<b>Tabla 4.4.</b> <i>Parámetros de la estructura forma circular</i> .....	48
<b>Tabla 4.5.</b> <i>Resultados del ejercicio dos por el método matricial</i> .....	49
<b>Tabla 4.6.</b> <i>Resultados del ejercicio tres método matricial</i> .....	50
<b>Tabla 4.7.</b> <i>Resultados del ejercicio dos utilizando SAP 2000.</i> .....	51
<b>Tabla 5.1.</b> <i>Datos del algoritmo genético de secciones</i> .....	55
<b>Tabla 5.2.</b> <i>Resultados obtenidos por el algoritmo genético de secciones para la estructura de tres elementos, utilizando los dos métodos de selección.</i> .....	56
<b>Tabla 5.3.</b> <i>Datos del algoritmo genético de secciones</i> .....	60
<b>Tabla 5.4.</b> <i>Resultados obtenidos por el algoritmo genético de secciones para la estructura de diez elementos, utilizando el método de la ruleta.</i> .....	60
<b>Tabla 5.5.</b> <i>Resultados obtenidos por el algoritmo genético de secciones para la estructura de diez elementos, utilizando el método por torneo.</i> .....	61

<b>Tabla 5.6.</b> <i>Resultados obtenidos por el algoritmo genético de secciones para la estructura de diez elementos, utilizando el método de la ruleta y el método por torneo.</i> .....	63
<b>Tabla 5.7.</b> <i>Parámetros de la estructura isostática óptima de luz de 20 metros.</i> ...	64
<b>Tabla 5.8.</b> <i>Coeficientes estadísticos, para la optimización de forma y topología para la armadura isostática con luz de 20 metros.</i> .....	66
<b>Tabla 5.9.</b> <i>Resultados obtenidos en la optimización de forma y topología para la estructura isostática de luz de 20 metros.</i> .....	67
<b>Tabla 5.10.</b> <i>Parámetros de la estructura óptima hiperestática de primer grado de luz de 20 metros.</i> .....	68
<b>Tabla 5.11.</b> <i>Coeficientes estadísticos, para la optimización de forma y topología para la armadura hiperestática con luz de 20 metros.</i> .....	70
<b>Tabla 5.12.</b> <i>Resultados obtenidos en la optimización de forma y topología para la estructura hiperestática de primer grado con luz de 20 metros.</i> .....	71

## LISTADO DE FIGURAS

<b>Figura 2.1.</b> Desplazamientos de un elemento de una armadura .....	9
<b>Figura 2.2.</b> Transformación de coordenadas locales a coordenadas globales .....	10
<b>Figura 3.1.</b> Ejemplo de cálculo por el método matricial. ....	24
<b>Figura 3.2.</b> Numeración de nodos de la estructura .....	25
<b>Figura 3.3.</b> Numeración de grados de libertad totales .....	26
<b>Figura 3.4.</b> Resultados de la matriz de rigidez por el análisis matricial .....	27
<b>Figura 3.5.</b> Propiedades del material y tipo de sección. ....	30
<b>Figura 3.6.</b> Geometría de la estructura y carga aplicada. ....	31
<b>Figura 3.7.</b> Resultados obtenidos en el programa SAP2000. ....	31
<b>Figura 3.8.</b> Diagrama de flujo de optimización de secciones. ....	35
<b>Figura 3.9.</b> Parametrización de la armadura tipo Pratt. ....	35
<b>Figura 3.10.</b> Parametrización de la armadura tipo Pratt .....	36
<b>Figura 3.11.</b> Armadura circular. ....	37
<b>Figura 3.12.</b> Parametrización de la estructura circular. ....	38
<b>Figura 3.13.</b> Diagrama de flujo del algoritmo genético de secciones, forma y topología.....	41
<b>Figura 3.14.</b> Diagrama de flujo del algoritmo genético de forma, topología y secciones. ....	42
<b>Figura 4.1.</b> Ejercicio armadura de 33 elementos.....	45
<b>Figura 4.2.</b> Ejercicio uno armadura 33 elementos .....	46
<b>Figura 4.3.</b> Resultados ejercicio uno programa SAP 2000. ....	47
<b>Figura 4.4.</b> Ejercicio dos numeración de elementos y nodos. ....	48
<b>Figura 4.5.</b> Ejercicio tres numeración de elementos y nodos. ....	49

<b>Figura 4.6.</b> Resultados ejercicio dos programa SAP 2000 .....	50
<b>Figura 4.7.</b> Resultados ejercicio tres programa SAP 2000 .....	51
<b>Figura 5.1.</b> Armadura plana de tres elementos .....	54
<b>Figura 5.2.</b> Peso vs número de generaciones ejercicio de tres elementos. ....	57
<b>Figura 5.3.</b> Peso mínimo vs iteraciones, ejercicio de tres elementos. ....	57
<b>Figura 5.4.</b> Armadura plana de diez elementos.....	59
<b>Figura 5.5.</b> Peso vs número de generaciones ejercicio de diez elementos. ....	61
<b>Figura 5.6.</b> Peso mínimo vs número de iteraciones ejercicio de diez elementos.....	62
<b>Figura 5.7.</b> Diseño óptimo de armadura isostática de luz de 20 metros. ....	64
<b>Figura 5.8.</b> Función objetivo versus generaciones para cinco iteraciones del diseño óptimo de forma y topología de armadura isostática de luz de 20 metros.....	65
<b>Figura 5.9.</b> Función objetivo versus iteración del diseño óptimo de forma y topología de armadura isostática de luz de 20 metros. ....	66
<b>Figura 5.10.</b> Diseño óptimo de armadura hiperestático de primer grado de luz de 20 metros.....	67
<b>Figura 5.11.</b> Función objetivo versus generaciones para cinco iteraciones para el diseño óptimo de forma y topología de armadura hiperestática de luz de 20 metros.....	69
<b>Figura 5.12.</b> Función objetivo versus iteración del diseño óptimo de forma y topología de armadura hiperestática de luz de 20 metros.....	70

## RESUMEN

En el presente trabajo, se realizó un análisis matricial en estructuras de acero tipo celosía en dos dimensiones con programas en lenguaje M en la plataforma de Matlab, luego se realizó una automatización del algoritmo para así evitar procesos manuales de ingreso de datos con el fin de lograr más adelante una optimización topológica y de forma, estos algoritmos se validaron con ejercicios en el programa SAP 2000.

Luego se realizó una optimización de secciones, creando algoritmos, para este procedimiento se utilizó algoritmos genéticos para la optimización, en los cuales se utilizó los dos métodos de selección por torneo y ruleta para determinar el mejor método; También se realizó una parametrización de una estructura tipo Pratt y una circular, de esta forma se evitó ingreso de datos manuales de toda la geometría, solo ingresando los parámetros de la estructura a analizar, esto es muy importante para la optimización de forma y topología.

Finalmente, se realizó la optimización de forma, topología y secciones utilizando algoritmos genéticos con el método por torneo, esta optimización se separó en dos partes una de forma y topología, y otra de secciones esto con el fin de disminuir tiempo computacional. Todos estos algoritmos de optimización creados se validaron con estudios anteriores para determinar la efectividad del algoritmo creado.

## **ABSTRACT**

In the present work, a matrix analysis was made in two-dimensional lattice-type steel structures with programs in the M language on the MATLAB platform, then an automation of the algorithm was made to avoid manual data entry processes in order to achieve, in the future, a topological and shape optimization, these algorithms were validated with exercises in the SAP2000 program.

Afterwards, a sections optimization was made, creating algorithms for this procedure, genetic algorithms were used for the optimization in which two selection method were used, by tournament and roulette, in order to determine the best methods; A Pratt type and a circular structure parameterization was also made, this way manual data entry of all the geometry was avoided, only entering the analyze structure parameters, this is very important for the shape and topology optimization.

Finally, the shape, topology and sections optimization was made using genetic algorithms with the tournament method, this optimization was separated into two parts, one of shape and topology and another of sections in order to reduce computational time. All these optimized algorithms that were created were validated with previous studies to determine the effectiveness of the created algorithm.

# CAPÍTULO 1.

## INTRODUCCIÓN

### 1.1 ANTECEDENTES

La técnica más popular en la computación evolutiva son los algoritmos genéticos (Sivanandam & Deepa, 2008). “Los algoritmos genéticos emulan la evolución genética, las características de los individuos están expresadas con base en genotipos” (Cagnina, 2010, p. 23). Las primeras ideas fueron introducidas por John Holland a mediados de 1960 y coloco las bases de los mecanismos naturales en la publicación de su libro en 1975, sin embargo, fue su discípulo David Goldberg quien alcanzó el mayor éxito en la optimización después de aplicar los algoritmos genéticos en su tesis doctoral y posteriormente en 1989 publica un libro que recopilan una gran cantidad de aplicaciones, como el clásico problema de optimización de estructuras de 10 barras y seis nudos aunque sin la restricción de desplazamientos del problema original, por lo que este libro se convierte en un icono en el campo de optimización (Hernández, 2018).

Los algoritmos genéticos son algoritmos de optimización que se basan en la teoría de la evolución de Darwin y en mecanismos de selección natural y son ampliamente utilizados en cualquier tipo de problema debido a que tienden a acercarse al óptimo global (Gutierrez, 2014); Los algoritmos genéticos por necesitar únicamente la evaluación de una función objetivo tiene aplicación en una gran cantidad de problemas desde medicina, problemas del agente viajero y también ingeniería (eléctrica, sistemas, redes, civil, producción y aeronáutica) (Sivanandam & Deepa, 2008). El gran éxito de esta técnica de optimización es incuestionable en todo tipo de problemas e incluso supera en muchos casos a la gran mayoría de técnicas de optimización, para los algoritmos genéticos no representa problema trabajar con varios tipos de optimizaciones (Sánchez, 2012).

Desde la década de 1980 fueron aumentando la cantidad y complejidad de problemas de optimización de estructuras, por lo que estas fueron siendo resueltas con heurísticas de optimización (Hernández, 2018). En la actualidad se encuentra una gran cantidad de estudios de optimización de estructuras con algoritmos genéticos, que involucran optimización de forma y topología todos estos estudios son a nivel de postgrado y doctorado (Sánchez, 2012).

Bonelli y Begliardo en el 2016 realizaron un trabajo de optimización de armaduras planas mediante diseño paramétrico y algoritmos genéticos llegando a optimizar las secciones, forma y topología de los problemas propuestos, para esto era necesario programas tanto para el cálculo de la estructura y un programa de algoritmos genéticos para la optimización.

## **1.2 OBJETIVOS**

### **1.2.1. OBJETIVO GENERAL**

Realizar un algoritmo en el programa Matlab para el diseño paramétrico de estructuras tipo celosía de acero en dos dimensiones, utilizando un algoritmo genético para la optimización de secciones, forma y topología, el cual utilice dos métodos de selección, para encontrar la estructura de peso mínimo.

### **1.2.2. OBJETIVOS ESPECÍFICOS**

- Determinar las fuerzas internas de una estructura utilizando análisis matricial con un algoritmo en Matlab y compararlos con los resultados calculados con un programa comercial de análisis de estructuras para determinar la efectividad del programa.
- Realizar la automatización del análisis matricial de estructuras, utilizando las coordenadas de la estructura y la carga cooperante para evitar ingresos manuales de todos los datos de la estructura.

- Crear un algoritmo genético de optimización de secciones con dos métodos de selección utilizando el programa Matlab, para determinar las secciones con las cuales se obtiene el peso mínimo de la estructura, validándolo con un problema particular.
- Parametrizar la estructura de acuerdo a las variables principales de la geometría, para poder realizar la automatización de forma y topología (Esta restringida al análisis paramétrico de la armadura por lo que solo toma el número de diagonales las cuales pueden aumentar o disminuir en un rango establecido de acuerdo a la luz de la armadura).
- Realizar un algoritmo utilizando el programa Matlab que utilice algoritmos genéticos con dos métodos de selección para realizar la optimización de secciones, forma y topología (Esta restringida al análisis paramétrico de la armadura por lo que solo toma el número de diagonales las cuales pueden aumentar o disminuir en un rango establecido de acuerdo a la luz de la armadura).

### **1.3 PLANTEAMIENTO DEL PROBLEMA.**

Una armadura es un grupo de elementos estructurales que forman uno o más triángulos entre sí, conociendo que el triángulo es el único elemento estable (Weaver, 1967). Las armaduras son muy importantes para salvar grandes luces siendo éstas mucho más económicas que las vigas y se las utiliza a partir de luces de 9 metros (McCormac, 2013).

Los algoritmos genéticos son métodos que se basan en organismos biológicos los cuales van evolucionando con el pasar de las generaciones, solo los organismos más fuertes sobreviven, estos individuos más aptos de cada generación se someten a procesos de reproducción y mutación en cada generación, el resultado de este proceso generará nuevos individuos los cuales pasarán a la siguiente generación a través de un número determinado de generaciones el algoritmo convergerá encontrando la mejor solución al problema (García, 2014).

La convergencia a la solución de cualquier problema en un número determinado de generaciones dependerá en gran medida del tipo de función objetivo, el método de selección que utilice el algoritmo genético en el problema y también de realizar en porcentajes adecuados la reproducción, cruce y mutación de acuerdo a las recomendaciones de varios autores (Sánchez, 2012).

La problemática del tema se presenta en que la solución de un problema de diseño de estructuras depende en gran parte del criterio y la experiencia del ingeniero que diseña, esto genera que en muchos de los casos las configuraciones y dimensiones de los miembros estructurales no sean necesariamente las apropiadas (Ramírez, 2014). El proceso tradicional de optimización estructural es un proceso iterativo y subjetivo donde se va colocando secciones que cumplan con los esfuerzos y las solicitaciones a la cual se somete la estructura, este proceso es costoso para un cálculo manual y solo procesa una única solución (Sánchez, 2012).

A pesar de que se han realizado una gran cantidad de trabajos de optimización con algoritmos genéticos (Sánchez, 2012), esto es a nivel de postgrado y doctorado, por lo que a nivel de pregrado el tema sigue sin conocerse por la gran mayoría, con una excepción de un estudio realizado por Jesús Alberto Suárez Prado en el 2017 donde aborda a nivel de pregrado optimización estructural con un algoritmo de recocido simulado alterado, pero esto todavía no es suficiente a nivel de pregrado para entender la importancia, funcionamiento y gran utilidad de los algoritmos genéticos como mecanismo de optimización.

En este estudio se busca encontrar el peso mínimo de estructuras tipo armadura, las cuales se parametrizarán para la optimización, utilizando algoritmos genéticos como mecanismo de optimización para poder tener el mejor diseño en el menor tiempo posible. Este estudio beneficiará a eliminar la incertidumbre en la optimización de estructuras y también ayudará a comprender de mejor manera a nivel de pregrado la optimización de estructuras con algoritmos genéticos.

## **1.4 ALCANCE**

Los métodos de selección que se utilizará en el algoritmo genético únicamente serán: por torneo y por ruleta.

La optimización de secciones está restringida al uso de funciones objetivos ya establecidas por otros autores.

La optimización de forma esta únicamente establecida de acuerdo al tipo de estructura, sus parámetros principales y el rango de dimensiones asignadas.

La optimización topológica está restringida al análisis paramétrico de la armadura por lo que solo toma el número de diagonales las cuales pueden aumentar o disminuir en un rango establecido de acuerdo a la luz de la armadura.

## **1.5 JUSTIFICACIÓN**

### **1.5.1. JUSTIFICACIÓN TEÓRICA**

Siempre que se estudia la posibilidad de la introducción de algoritmos evolutivos de optimización de sistemas complejos, los algoritmos genéticos vienen avalados por un gran número de estudios y trabajos de todo tipo los cuales garantizan su efectividad (García, 2014). La optimización estructural se viene trabajando en todos sus aspectos como optimización de secciones, geométrica de dimensiones, topológica de forma y combinaciones de estas (Sánchez, 2012), por lo cual se va a utilizar los algoritmos genéticos para este estudio de optimización de armaduras reticulares debido a su gran efectividad.

### **1.5.2. JUSTIFICACIÓN METODOLÓGICA.**

En algoritmos genéticos el proceso de selección es fundamental y por lo general se realiza en forma probabilística, este mecanismo permite orientar a los individuos

con mayor adaptación observada para que se reproduzcan los cuales formarán los hijos de la siguiente generación (Rodríguez, 2010). Según el operador o método elegido en la selección de individuos permitirá que llegue al óptimo global con mayor facilidad (Sánchez, 2012).

Según Sánchez, 2012 en su estudio de optimización estructural, hay varios operadores de selección uno más adecuado que otro. En este caso se escogerá los métodos de la ruleta y el método por torneo como operadores de selección para poder entender y dejar las bases a nivel de pregrado de la efectividad de cada método y determinar que método es más eficiente.

### **1.5.3. JUSTIFICACIÓN PRÁCTICA.**

Este estudio busca realizar una optimización en la estructura para poder obtener el costo mínimo el cual está directamente relacionado con el peso mínimo de la estructura (Rodríguez, 2010). Se utilizará dos operadores de selección para encontrar la estructura más óptima y poder entender la importancia de los algoritmos genéticos como método de optimización y también encontrar la diferencia de estos dos operadores de acuerdo a sus bondades y desventajas (Sánchez, 2012).

Debido a que la gran mayoría de estudios de optimización son a nivel de postgrado y doctorado con la excepción de Suarez, J. (2017), que realizó un estudio de optimización a nivel de pregrado, este estudio busca familiarizar a nivel de pregrado la importancia de los algoritmos genéticos como mecanismo de optimización utilizando el programa Matlab para la creación de las rutinas de optimización.

## **CAPÍTULO 2.**

### **MARCO TEÓRICO**

#### **2.1 INTRODUCCIÓN**

Para realizar diseño estructural es fundamental manejar programas comerciales para el cálculo de las sollicitaciones mecánicas, pero debido a que es necesario realizar un proceso de optimización más eficiente en recursos y tiempo, es muy importante realizar el cálculo de las sollicitaciones con análisis matricial para proceder a realizar el diseño de la estructura, y la optimización de secciones con algoritmos genéticos, sin embargo, para poder realizar la optimización de forma y topología se procede a realizar la parametrización de la estructura para poder disminuir la cantidad de parámetros.

#### **2.2 ESTRUCTURAS ARTICULADAS.**

Las celosías o estructuras articuladas están formadas por barras y nudos (soldadas o enlazadas por pasadores), estos elementos forman figuras triangulares que presentan gran rigidez. Las celosías solo absorben esfuerzos axiales de tracción y compresión como esfuerzos principales por lo que los nudos se suponen libres de fricción y tiene libertad de girar. Sin embargo, no tiene libertad de girar debido a que en la construcción de estas estructuras los nudos son remachados o soldados, por lo que produce esfuerzos secundarios, pero resulta irrelevante esta magnitud en comparación de los esfuerzos principales; En este estudio solo se considerará los esfuerzos principales.

#### **2.3 FUNDAMENTOS DEL MÉTODO MATRICIAL DE RIGIDEZ.**

El Método de Rigidez o de Desplazamientos fue presentado en 1954 por Turner, Clough, Martin y Topp, sin embargo, este método no fue utilizado debido a la falta de desarrollo informático (Aguilar, 2014). Este método empezó a ser utilizado con los

avances de la informática y siendo más utilizado a partir del año de 1995, debido a la facilidad de resolver ecuaciones y matrices en programas como Matlab (Aguiar, 2014). El método de rigidez se basa en la relación de desplazamientos y de las fuerzas de los elementos de una estructura, de esta forma se obtiene un sistema de  $n$  ecuaciones la cual depende del número de elementos y del número de grados de libertad de la estructura (Suárez, 2017).

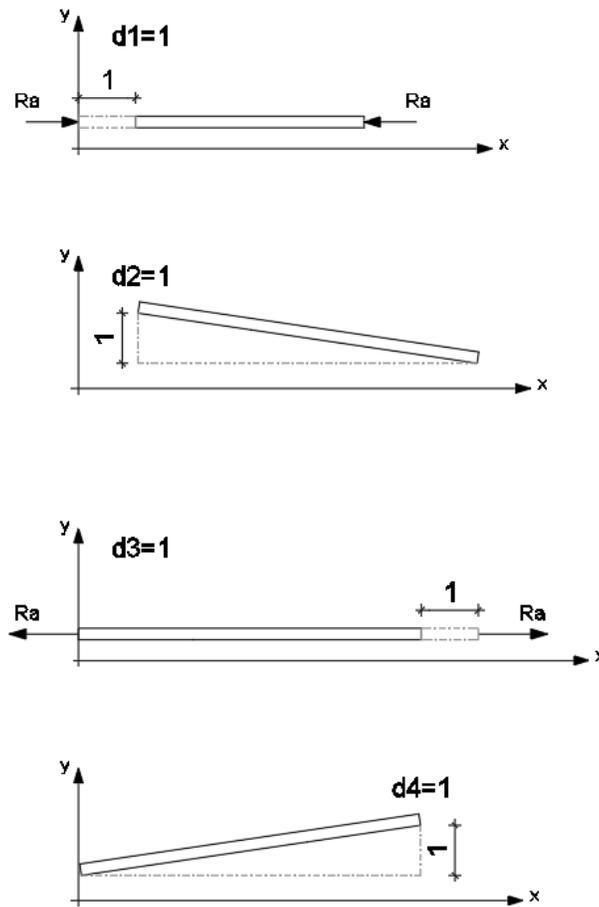
El método de rigidez parte de las siguientes hipótesis:

- El comportamiento del material es perfectamente elástico, homogéneo e isotrópico, cumple con la ley de Hooke.
- Las deformaciones son muy pequeñas, se puede utilizar la teoría de los desplazamientos infinitesimos.
- Todas las cargas se aplican gradualmente, el principio de superposición es válido.
- Las constantes de los materiales son independientes del tiempo.
- El sistema está en un estado de equilibrio estático.
- Las secciones permanecen planas antes y después de las deformaciones.

Los métodos de cálculo matricial de estructuras son un conjunto de métodos que organizan toda la información en matrices, esta estructura permite que su planteamiento y resolución pueda ser realizada de manera automática por medio de ordenadores, por lo que en la actualidad se utiliza habitualmente en la ingeniería (Blanco, Gonzáles, & García, 2012).

### **2.3.1. MATRIZ DE RIGIDEZ DE UN ELEMENT O TIPO ARMADURA.**

A un elemento de la armadura se le da un desplazamiento de una unidad para determinar la matriz de rigidez del elemento, como se observa en la figura 2.1.



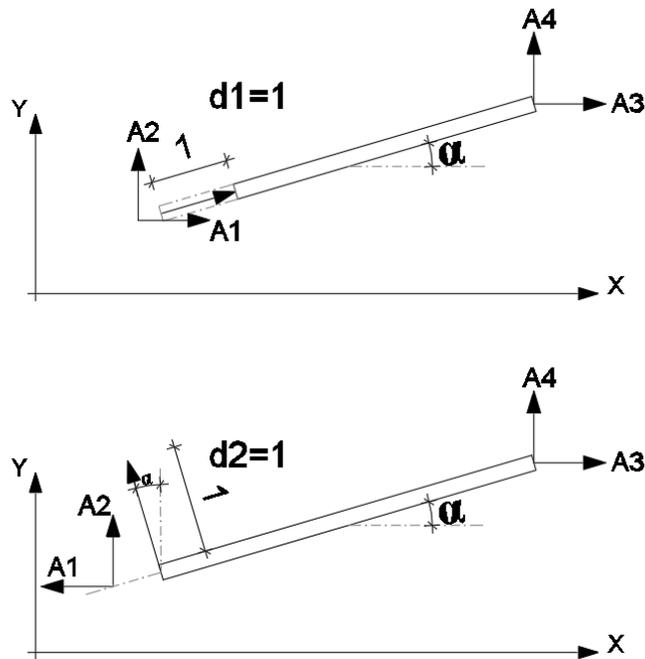
**Figura 2.1.** Desplazamientos de un elemento de una armadura

La matriz de rigidez del elemento de la armadura queda como se observa en la ecuación 2.1

$$k_i = \begin{bmatrix} \frac{E * A}{L} & 0 & -\frac{E * A}{L} & 0 \\ 0 & 0 & 0 & 0 \\ E * A & 0 & E * A & 0 \\ -\frac{L}{L} & 0 & \frac{L}{L} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.1)$$

### 2.3.2. MATRIZ DE PASO DE UN ELEMENTO TIPO ARMADURA

Esta matriz se utiliza para pasar un elemento de coordenadas locales a coordenadas globales, la deducción de esta matriz se puede observar en la figura 2.2.



**Figura 2.2.** Transformación de coordenadas locales a coordenadas globales

La matriz de transformación la cual se encuentra representada por la ecuación 2.2 se obtuvo de la figura 2.2.

$$T_i = \begin{bmatrix} \cos(\alpha) & -\text{sen}(\alpha) & 0 & 0 \\ \text{sen}(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & \cos(\alpha) & -\text{sen}(\alpha) \\ 0 & 0 & \text{sen}(\alpha) & \cos(\alpha) \end{bmatrix} \quad (2.2)$$

La matriz de transformación es muy útil para pasar de coordenadas locales a coordenadas globales.

### 2.3.3. MATRIZ DE RIGIDEZ GLOBAL DE UN ELEMENTO TIPO ARMADURA.

La matriz de rigidez global de un elemento se obtiene con la matriz de transformación y la matriz de rigidez en coordenadas locales del elemento como se observa en la ecuación 2.3.

$$K_i = T_i * k_i * T'_i \quad (2.3)$$

Utilizando este método para poder determinar la matriz de rigidez global de estructura, se debe sumar las acciones de rigidez que colaborarán a la rigidez entre elementos de acuerdo al nudo.

#### 2.3.4. MATRIZ DE RIGIDEZ AMPLIADA DE UNA ARMADURA.

Es una matriz escalonada donde se encuentra todas las matrices de rigidez de los elementos, su dimensión es  $4e \times 4e$  para estructuras tipo celosía en dos dimensiones, en donde “e” es el número de elementos de la estructura. La matriz ampliada se puede observar en la ecuación 2.4.

$$K_m = \begin{bmatrix} K_i & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & K_n \end{bmatrix} \quad (2.4)$$

#### 2.3.5. MATRIZ DE COMPATIBILIDAD POR ELEMENTO.

Es una matriz de unos y ceros la cual permite la compatibilidad de deformaciones, esta matriz relaciona las deformaciones de cada elemento en coordenadas locales con los grados de libertad cinemáticos en coordenadas globales, está representada por la matriz  $G_i$ . La dimensión de esta matriz es  $d \times 4$ , en donde “d” es el número de grados de libertad cinemáticos, en la ecuación 2.5. se muestra la traspuesta de la matriz de compatibilidad.

$$G_i^T = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} \end{bmatrix} \quad (2.5)$$

Los ceros representan que al elemento no le afecta ese grado de libertad cinemático, mientras que los uno representa que el grado de libertad cinemático le afecta al elemento. En cada fila va un uno de acuerdo a las acciones en coordenadas locales que produzca una deformación en coordenadas globales.

### 2.3.6. MATRIZ DE COMPATIBILIDAD GLOBAL

Es una matriz que contiene a todas las matrices de compatibilidad de los elementos en una sola matriz como se observa en la ecuación 2.6.

$$\mathbf{G}_m = [\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, \mathbf{G}_4 \dots \mathbf{G}_n] \quad (2.6)$$

Esta matriz es la que se encarga de sumar acciones en los nudos de acuerdo a los elementos que coinciden en el mismo nudo, esta matriz permite determinar la matriz de rigidez global de la estructura. La dimensión de la matriz de compatibilidad global de cada elemento es “d x 4e”, en donde “d” es el número de grados de libertad y “e” es el número de elementos.

### 2.3.7. MATRIZ DE RIGIDEZ GLOBAL DE LA ESTRUCTURA TIPO ARMADURA.

La matriz de rigidez global de la armadura se obtiene con la matriz de compatibilidad global y la matriz de rigidez ampliada como se observa en la ecuación 2.7.

$$\mathbf{S} = \mathbf{G}_m * \mathbf{K}_m * \mathbf{G}'_m \quad (2.7)$$

### 2.3.8. DEFORMACIONES.

Las deformaciones causadas en los elementos de la estructura por la acción de las fuerzas externas se calculan con la ecuación 2.8, 2.9 y 2.10.

$$\mathbf{Delta} = \mathbf{S}^{-1} * \mathbf{P} \quad (2.8)$$

$$\mathbf{D}_i = \mathbf{Delta}_i \quad (2.9)$$

$$\mathbf{d}_i = \mathbf{T}'_i * \mathbf{D}_i \quad (2.10)$$

Donde “Delta” son las deformaciones causadas por las fuerzas externas, son todas las deformaciones en coordenadas globales de la estructura, “S” es la matriz de

rigidez de la estructura, “P” son las cargas externas en la estructura y  $D_i$  son las deformaciones de un elemento en coordenadas globales,  $d_i$  son las deformaciones de los elementos en coordenadas locales.

### 2.3.9. FUERZAS INTERNAS DE LOS ELEMENTOS DE LA ESTRUCTURA

Las fuerzas internas ejercidas sobre los elementos se calculan con la ecuación 2.11 y 2.12.

$$A_i = K_i * D_i + A_{iep} \quad (2.11)$$

$$a_i = k_i * d_i + a_{iep} \quad (2.12)$$

Donde,  $A_i$  son las acciones internas del elemento en coordenadas globales,  $K_i$  es la matriz de rigidez en coordenadas globales del elemento,  $A_{iep}$  son las acciones de empotramiento perfecto en coordenadas globales,  $a_i$  son las acciones internas del elemento en coordenadas locales,  $k_i$  es la matriz de rigidez en coordenadas locales del elemento,  $a_{iep}$  son las acciones de empotramiento perfecto en coordenadas locales.

## 2.4 MÉTODOS DE DISEÑO.

Los métodos de diseño son requisitos que deben cumplir, los cuales no deben ser excedidos, los métodos de diseño son los siguientes:

- i. Método de diseño por resistencia usando diseño en base a factores de carga y resistencia (LRFD).
- ii. Método de diseño por resistencia usando diseño en base a resistencias admisibles (ASD).

#### **2.4.1. MÉTODO DE DISEÑO POR RESISTENCIA USANDO DISEÑO EN BASE A FACTORES DE CARGA Y RESISTENCIA (LRFD).**

Método de diseño en base a factores de carga y resistencia (LRFD), satisface los requisitos de esta especificación cuando la resistencia de diseño de cada componente estructural es mayor o igual a la resistencia requerida determinada de acuerdo con las combinaciones de carga LRFD. Se basa en el concepto de los estados límites, esto se da cuando el elemento de la estructura deja de resistir para la carga que fue diseñada (McCormac, 2013). En la ecuación 2.13 representa el criterio de diseño de este método.

$$R_u \leq \phi R_n \quad (2.13)$$

$R_u$  = resistencia requerida por las combinaciones LRFD

$R_n$  = resistencia nominal

$\phi$  = factor de resistencia ( $\phi = 0.9$  para el límite de fluencia de elementos a tracción o compresión)

$\phi R_n$  = resistencia nominal de diseño

La resistencia nominal  $R_n$  se realiza el cálculo de acuerdo a las disposiciones del AISC 360 – 16.

#### **2.4.2. MÉTODO DE DISEÑO POR RESISTENCIA USANDO DISEÑO EN BASE A RESISTENCIAS ADMISIBLES (ASD).**

El método de diseño por resistencia usando diseño en base a factores de resistencias admisibles (ASD) satisface cuando la resistencia admisible de diseño de cada componente estructural es mayor o igual que la resistencia requerida determinada de acuerdo con las combinaciones de carga ASD (Las cuales a diferencia de las cargas LRFD, estas combinaciones no tienen coeficientes de mayoración) (Ramírez, 2014). En la ecuación 2.14 representa el criterio de diseño de este método.

$$R_a \leq \frac{R_n}{\Omega} \quad (2.14)$$

$R_a$  = resistencia requerida por las combinaciones ASD

$R_n$  = resistencia nominal

$\Omega$  = factor de seguridad ( $\Omega = 1.67$  para el límite de fluencia de elementos a tracción o compresión)

$\frac{R_n}{\Omega}$  = resistencia admisible

La resistencia nominal  $R_n$  se realiza el cálculo de acuerdo a las disposiciones del AISC 360 – 16.

## 2.5 DISEÑO PARAMÉTRICO.

Es primordial optimizar tiempo y recursos con el manejo de la información por lo que el diseño paramétrico busca ser más eficiente en el diseño, en comparación a métodos tradicionales, el diseño paramétrico utiliza parámetros principales de la estructura que se evalúan. Entendiendo por parametrización un conjunto de ecuaciones que representan el problema, tales ecuaciones están gobernadas por variables independientes que representan los parámetros, estas ecuaciones representan la geometría de la estructura.

## 2.6 OPTIMIZACIÓN.

La optimización es un proceso mediante el cual se busca encontrar los recursos mínimos para satisfacer un problema en específico. La optimización es el proceso mediante el cual se escoge el mejor individuo de un conjunto de elementos posibles (Ramírez, 2014). La optimización se observa en todo tipo de problemas de ingeniería e incluso en medicina; La optimización en el ámbito estructural se viene estudiando desde el año de 1989 con los problemas de optimización de estructuras que presentó David Goldberg (Hernández, 2018). En los estudios de optimización estructural se pueden encontrar varios tipos.

### **2.6.1. TIPOS DE OPTIMIZACIÓN EN DISEÑO DE ESTRUCTURAS.**

Los tipos de optimización que se puede realizar en estructuras son las siguientes:

- i. Optimización de secciones.
- ii. Optimización de forma.
- iii. Optimización de topología.

*Optimización de secciones:* La optimización de secciones se basa en encontrar las secciones mínimas de los elementos de una estructura para que resista las cargas externas a las que puede estar sometida, todas las secciones a utilizar deben ser secciones comerciales.

*Optimización de forma:* La optimización de forma busca la forma exterior óptima de una estructura, dentro de las restricciones impuestas, para esto la estructura debe estar parametrizada (Bonelli & Begliardo, 2016).

*Optimización topológica:* La optimización topológica realiza un incremento y disminución elementos y nudos hasta tener la estructura óptima que cumpla los requerimientos de estabilidad y de resistencia.

### **2.6.2. ALGORITMOS GENÉTICOS.**

Los algoritmos genéticos son métodos que se basan en organismos biológicos los cuales van evolucionando con el pasar de las generaciones, solo los organismos más fuerte sobreviven, estos individuos más aptos de cada generación se someten a procesos de reproducción y mutación en cada generación, el resultado de este proceso generará nuevos individuos los cuales pasarán a la siguiente generación a través de un número determinado de generaciones, el algoritmo convergerá encontrando la mejor solución al problema (García, 2014). Los algoritmos genéticos representan el problema mediante parámetros los cuales se colocan en un

cromosoma, este está formado por genes y alelos; La representación del cromosoma la más utilizada es el código binario para representar todas sus componentes.

**Cromosoma:** Cadena de ADN que contiene la información genética de un individuo, esta información contiene los genes los mismos que son los parámetros a analizar; Es la representación del problema donde se encuentra todas las características o parámetros como se puede observar en la tabla 2.1.

**Tabla 2.1.**

*Ejemplo de un cromosoma.*

Pronóstico			Temperatura			Humedad			Viento			Nadar	
0	1	0	1	0	1	0	1	1	1	1	0	0	1
Gen #1			Gen #2			Gen #3			Gen #4			Gen #5	

**Gen:** Cadena de ADN que codifica una función bioquímica; contiene un parámetro específico del individuo.

**Alelo:** Cada valor posible para una cierta posición genética, en el caso de codificación binaria puede ser únicamente cero o uno.

**Población inicial:** Es la cantidad de individuos inicial, estos individuos tienen parámetros aleatorios. La cantidad de individuos de cada generación depende de la complejidad del problema, para mayor complejidad se debe utilizar mayor cantidad de individuos para evitar convergencia de soluciones en óptimos locales (Sánchez, 2012).

**Función aptitud:** Es la función con la cual se determina la calidad de un individuo a adaptarse en un medio, los individuos con mayor actitud son los que se adaptan al medio; Por tanto, solo los individuos con mejor puntuación son aptos y son seleccionados para ser padres reproducirse para la siguiente generación. La función aptitud en diseño de estructuras puede estar representado por varios parámetros, una de las funciones objetivos más sencillas puede estar representado por el peso de la estructura que está representado por la ecuación 2.15.

$$f_{obj} = W_i = \sum_i^n \rho * A_i * L_i \quad (2.15)$$

Donde,  $f_{obj}$  es la función objetivo;  $W_i$  es el peso de la estructura la unidad es el kg.;  $A_i$  es el área del elemento de la estructura, la unidad es el  $cm^2$  y  $\rho$  es la densidad del material, las unidades son los  $kg/cm^3$ .

**Operadores genéticos:** Los operadores genéticos son mecanismos mediante el cual se realiza el proceso natural de evolución, para esto se debe asignar la población inicial, el método de selección, la reproducción y mutación de los individuos, para poder pasar los descendientes de la siguiente generación.

**Operador de inicialización:** Este operador es el encargado de crear los individuos de la primera generación, los parámetros de cada individuo son creados aleatoriamente para iniciar en la búsqueda del mejor individuo.

**Selección:** El proceso de selección se realiza de acuerdo a la función de aptitud, solo los individuos que tienen mejor puntuación son aptos para reproducirse y pasar sus hijos a la siguiente generación.

Los métodos de selección hay una diversidad con sus ventajas y desventajas, pero para este estudio se ha considerado los métodos de selección por torneo y por ruleta, para ser analizados a nivel de pregrado.

**Método de selección por torneo:** Según Sánchez, 2012 este método fue propuesto por Wetzel y estudiada por Bridle en su tesis doctoral y se popularizó con los estudios de Golgberg en 1989. Este método se basa en escoger dos o más individuos de la población y de acuerdo a la función actitud compiten entre sí, el individuo ganador del torneo es seleccionado para ser padre y poder reproducirse; Esta técnica es eficiente y muy fácil de implementar (Rodríguez, 2010).

**Método de selección por ruleta:** Según Rodríguez, 2010 este método fue propuesto por De Jong en 1975, el método consiste en que un individuo puede ser seleccionado de acuerdo a su función de actitud, por lo que los individuos con mayor actitud pueden tener mayor probabilidad de ser seleccionados por lo que el procedimiento es el siguiente.

- i. Primero se debe sumar las funciones de actitud de los individuos.
- ii. Luego determinar el porcentaje que representa cada individuo.
- iii. Todos los individuos se colocan en una ruleta.
- iv. Se debe generar un número aleatorio entre 0 y 360 grados.
- v. El individuo que se encuentre entre este número aleatorio es el individuo seleccionado para ser padre.

**Cruce:** El cruce es la operación donde se transmiten los genes de los padres hacia los hijos, el punto de cruce es aleatorio de acuerdo a esto cada pareja se cruza y tiene un hijo con las características del padre y de la madre.

**Mutación:** La mutación es el proceso mediante el cual un individuo cambia, esto sucede al cambiar un alelo del cromosoma, este alelo es seleccionado aleatoriamente de acuerdo al individuo seleccionado de igual forma seleccionado aleatoriamente. El porcentaje de mutación de un gen de un individuo se recomienda entre 0.001 y 0.01 (Sánchez, 2012)

## **CAPÍTULO 3.**

### **METODOLOGÍA DE DISEÑO DEL ALGORITMO.**

#### **3.1 INTRODUCCIÓN.**

El análisis matricial es fundamental para poder realizar el algoritmo que calcule los esfuerzos en una estructura y realizar el diseño de la misma siguiendo las normas correspondientes. Sin embargo, también es importante realizar los algoritmos en Matlab para la optimización de secciones, para poder tener un entendimiento del proceso que conlleva a la optimización utilizando algoritmos genéticos.

Para la implementación del algoritmo genético de secciones, forma y topología es necesario automatizar el método matricial, para que los datos a ingresar solo dependan de la geometría de la estructura y de las cargas externas de acuerdo a áreas cooperantes. Estos cambios de la geometría están incorporados con la parametrización de la estructura, la que está encargada de crear la estructura geométrica de acuerdo a parámetros geométricos establecidos.

#### **3.2 PROGRAMA PARA ANÁLISIS DE ARMADURAS EN 2D**

Los métodos matriciales son de gran utilidad para encontrar los esfuerzos generados en una estructura, en este caso se realiza el análisis de estructura tipo celosía en dos dimensiones, para el cual se utiliza el programa Matlab, para crear las funciones y rutinas que permita determinar todas las acciones internas de la estructura.

##### **3.2.1. USO DEL PROGRAMA DE ANÁLISIS MATRICIAL**

El uso del programa se resume en los siguientes pasos:

- i. Datos de la estructura, longitud de elementos, área de secciones de elementos, inercia de los elementos y módulo de elasticidad.

- ii. Se ingresa las coordenadas de los elementos de la estructura.
- iii. Se coloca las restricciones de los apoyos, en cada apoyo, con un cero si es libre y con un uno si no está libre, como se observa en la ecuación 3.1

$$fix = \begin{bmatrix} x_i & y_i \\ 1 & 1 \\ 0 & 1 \\ \vdots & \vdots \\ n & n \end{bmatrix} \quad (3.1)$$

- iv. Enumera los nodos de cada elemento en la estructura como se observa en la ecuación 3.2, el número de filas es igual al número de elementos.

$$nodos = \begin{bmatrix} N_{i1} & N_{j1} \\ 1 & 2 \\ 3 & 4 \\ \vdots & \vdots \\ N_{in} & N_{jn} \end{bmatrix} \quad (3.2)$$

La dimensión de este vector es de nx2, donde “n” es el número de elementos.

- v. Ubica los grados de libertad totales y los cinemáticos.
- vi. Se debe ubicar el vector de ensamble, este vector representa los grados de libertad cinemáticos del nodo inicial y del nodo final del elemento.
- vii. Se calcula la matriz de rigidez de cada elemento. Para esto se utiliza la función M\_KAP el cual se encuentra en el Anexo No1 A1.1.
- viii. Se calcula la matriz de paso de cada elemento. Para esto se utiliza la función M\_TAP el cual se encuentra en el Anexo No1 A1.2
- ix. Se calcula la matriz de rigidez en coordenadas globales de cada elemento.
- x. Se calcula la matriz de rigidez global (ST) de la estructura, para esto es necesario acumular los efectos de rigidez de cada elemento de acuerdo a los grados de libertad. Para esto se utiliza la función  $Ens(leei, Ki, ST, ngdl\_rest)$  el cual se encuentra en el Anexo No1 A1.3.

- xi. Se ingresa las cargas distribuidas de la estructura como se observa en la ecuación 3.3.

$$q_i = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \quad (3.3)$$

- xii. Se ingresa los vectores de acciones por empotramiento perfecto como se observa en la ecuación 3.4.

$$a_{ep\ i} = \begin{bmatrix} m_1 \\ m_2 \\ F \end{bmatrix} \quad (3.4)$$

Donde  $m_1$  es el momento de empotramiento perfecto en el nodo 1 del elemento;  $m_2$  es el momento de empotramiento perfecto en el nodo 2 del elemento;  $F$  es la fuerza axial por empotramiento perfecto a la cual se encuentra el elemento, todas estas reacciones son inducidas por la carga externa repartida. Esto se debe ubicar de acuerdo al elemento, en este caso la ecuación 3.5, representa las acciones de empotramiento perfecto para un elemento; Se debe ubicar en un solo vector como.

$$a_{ep} = \begin{bmatrix} m_{11} \\ m_{21} \\ F_{ax\ 1} \\ m_{12} \\ m_{22} \\ F_{ax\ 2} \\ \vdots \\ \vdots \\ m_{1n} \\ m_{2n} \\ F_{ax\ n} \end{bmatrix} \quad (3.5)$$

La dimensión de este vector es de  $3n \times 1$ , donde "n" es el número de elementos.

- xiii. Se ingresa el vector de cargas externas, esta carga se debe ingresar en un vector, las cargas externas se ingresan en orden de los grados de libertad

cinemáticos que afecta la carga externa, este vector de cargas se puede observar en la ecuación 3.6.

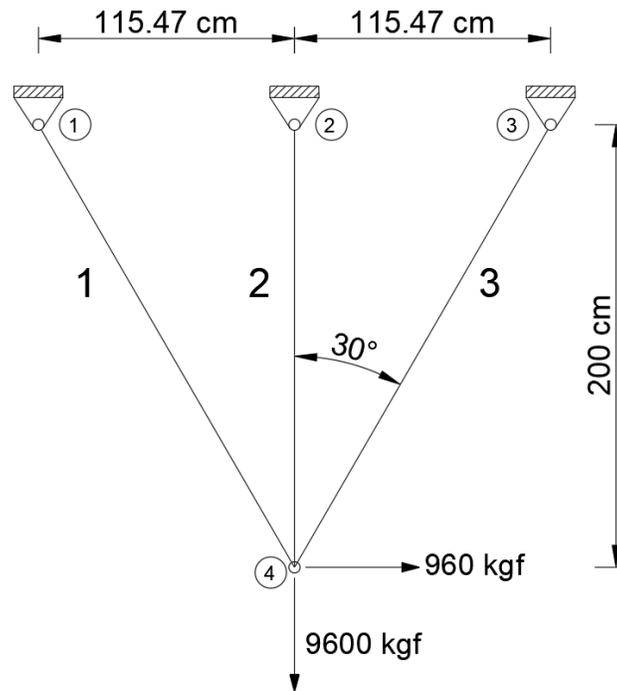
$$P = \begin{bmatrix} 0 \\ -P1 \\ -P2 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ -Pn \end{bmatrix} \quad (3.6)$$

La dimensión de este vector es  $g \times 1$ , donde “g” es el número de grados de libertad.

- xiv. Se determina los desplazamientos en coordenadas globales, coordenadas locales y las acciones en los elementos con la ecuación 2.8, 2.9, 2.10, 2.11 y 2.12.
- xv. Se determina las reacciones en los apoyos.
- xvi. Se comprueba el equilibrio de la estructura.

### **3.2.2. EJEMPLO DE CÁLCULO POR EL MÉTODO DE ANÁLISIS MATRICIAL.**

Se toma un ejercicio de 3 elementos que es ampliamente utilizado en el análisis para validación de algoritmos genéticos, la misma que se observa en la figura 3.1.



**Figura 3.1.** Ejemplo de cálculo por el método matricial.  
Fuente: (Bonelli & Begliardo, 2016)

Se ingresan los datos de la estructura.

Densidad del material  $\rho = 7850 \frac{\text{kgf}}{\text{cm}^2}$

Esfuerzo máximo permitido  $\sigma = \pm 1442 \frac{\text{kgf}}{\text{cm}^2}$

Desplazamiento máximo permitido  $\mu = \pm 0.5 \text{ cm}$

Módulo de elasticidad  $E = 1967840 \frac{\text{kgf}}{\text{cm}^2}$

Áreas  $A1=A2=A3=1.94 \text{ cm}^2$

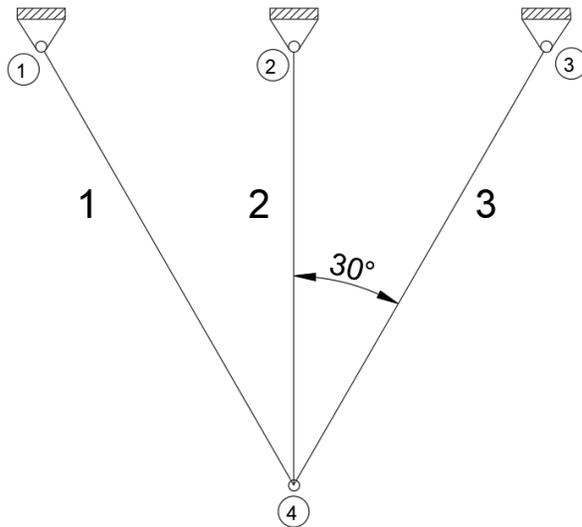
- i. Se ingresa las coordenadas de los elementos de la estructura.

$$\text{CoordXY} = \begin{bmatrix} 0 & 0 & 1.1547 & -2 \\ 1.1547 & 0 & 1.1547 & -2 \\ 2 * 1.1547 & 0 & 1.1547 & -2 \end{bmatrix}$$

- ii. Se coloca las restricciones de los apoyos, en cada apoyo, con un cero si es libre y con un uno si no está libre, como se observa a continuación.

$$fix = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

iii. Enumera los nodos en la estructura como se observa en la figura 3.2.

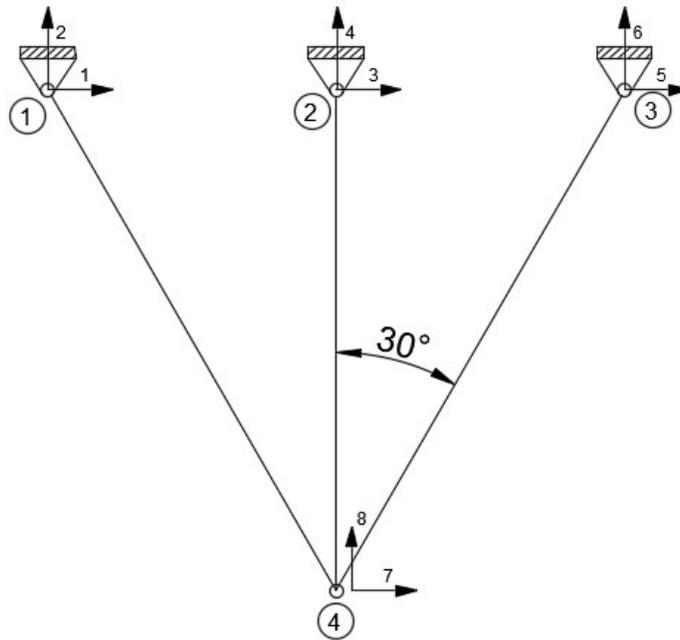


**Figura 3.2.** Numeración de nodos de la estructura

iv. Los elementos se colocan de acuerdo a los nodos como se observa a continuación:

$$nodos = \begin{bmatrix} 1 & 4 \\ 2 & 4 \\ 3 & 4 \end{bmatrix}$$

Ubicación de los grados de libertad totales se observa en la figura 3.3.



**Figura 3.3.** Numeración de grados de libertad totales

- v. Los grados de libertad totales son ocho, pero los grados de libertad cinemático son dos en este caso.
- vi. Se ubica la matriz de grados de libertad de cada nodo como se observa a continuación:

$$N_{GDL\text{MATR}} = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 3 & 4 \\ 3 & 5 & 6 \\ 4 & 7 & 8 \end{bmatrix}$$

- vii. Se ubica el vector de ensamble, este vector representa los grados de libertad cinemáticos del nodo inicial y del nodo final del elemento como se observa a continuación, esta matriz nos servirá para ensamblar la matriz de rigidez global.

$$lee_i = \begin{bmatrix} 1 & 2 & 7 & 8 \\ 3 & 4 & 7 & 8 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

- viii. Se calcula la matriz de rigidez de cada elemento de acuerdo a la ecuación 2.1.
- ix. Se calcula la matriz de paso de cada elemento de acuerdo a la ecuación 2.2.

- x. Se calcula la matriz de rigidez en coordenadas globales de cada elemento de acuerdo a la ecuación 2.3.
- xi. Se calcula la matriz de rigidez global (ST) de la estructura. Utilizando la función Ens en el programa Matlab, cuyos resultados se observan en la figura 3.4.

ST =

```

1.0e+06 *

  0.4260  -0.7379   0   0   0   0  -0.4260  0.7379
-0.7379   1.2782   0   0   0   0   0.7379 -1.2782
   0   0   0   0   0   0   0   0
   0   0   0  1.9678   0   0   0 -1.9678
   0   0   0   0  0.4260  0.7379 -0.4260 -0.7379
   0   0   0   0  0.7379  1.2782 -0.7379 -1.2782
-0.4260  0.7379   0   0 -0.4260 -0.7379  0.8521   0
  0.7379 -1.2782   0 -1.9678 -0.7379 -1.2782   0  4.5241

```

**Figura 3.4.** Resultados de la matriz de rigidez por el análisis matricial

La matriz de rigidez es 8x8 porque toma todos los grados de libertad de los nodos, la matriz de rigidez real de los grados de libertad cinemático es de 2x2, lo que representa las dos últimas filas y las dos últimas columnas.

- xii. Se ingresa las cargas distribuidas de la estructura, en este caso por ser una armadura, no hay cargas distribuidas.
- xiii. Se ingresa los vectores de acciones por empotramiento perfecto, en este caso no hay acciones de empotramiento perfecto por ser una armadura.
- xiv. Se ingresar el vector de cargas externas, esta carga se debe ingresar en un vector, las cargas externas se ingresan en orden de los grados de libertad cinemáticos que afecta la carga externa, este vector de cargas se observa a continuación.

$$P = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.96 \\ -9.6 \end{bmatrix}$$

La dimensión de este vector es 8x1 (ocho filas por una columna), debido a que 8 es el número de grados de libertad totales.

- xv. Se determinó los desplazamientos en coordenadas globales y locales y las acciones en los elementos con la ecuación 2.8, 2.9, 2.10, 2.11 y 2.12.
- xvi. El vector "Delta", contiene los desplazamientos en coordenadas globales de los elementos, el vector "di" contiene los desplazamientos en coordenadas locales de los elementos y el vector "ai" contiene las acciones de los elementos.

Los resultados calculados por el programa en Matlab se encuentran en la tabla 3.1 y 3.2. El programa de análisis matricial para armaduras en dos dimensiones se encuentra en el Anexo No1 A1.4

**Tabla 3.1.**

Resultados de desplazamientos en la estructura del analisis matricial

<i>N_GDL</i>	<i>DELTA</i> (m)
7	0,001161
8	-0,002187

**Tabla 3.2.**

Resultados de acciones y desplazamientos del análisis matricial.

Elementos	Nodos	Coordenadas	Ai	ai	Di	di
			(Ton)	(Ton)	(m)	(m)
1	1	Fxi	-2,04587	-4,09174	0	0
		Fyi	3,54360		0	0
	4	Fxf	2,04590	4,09174	0,0011614	0,002475
		Fyf	-3,54360		-0,002187	-0,000087
2	2	Fxi	0,00000	-4,17566	0	0
		Fyi	4,17570		0	0
	4	Fxf	0,00000	4,17566	0,0011614	0,0021875
		Fyf	-4,17570		-0,002187	0,0011614
3	3	Fxi	1,08590	-2,17174	0	0
		Fyi	1,08590		0	0
	4	Fxf	-1,08590	2,17174	0,0011614	0,001313
		Fyf	-1,08590		-0,002187	0,002099

- xvii. Se determina las reacciones en los apoyos, cuyos resultados se observan en la tabla 3.3.

**Tabla 3.3.**

*Reacciones en los apoyos, método matricial.*

Nodos	Fxyi	Reacciones (Ton)
1	Fx1	-2,0459
	Fy1	3,5436
2	Fx2	0,0000
	Fy2	4,1757
3	Fx3	1,0859
	Fy3	1,8808

- xviii. Se comprueba el equilibrio de la estructura, cuyos resultados se encuentran en la tabla 3.4.

**Tabla 3.4.**

*Comprobación de equilibrio por el método matricial.*

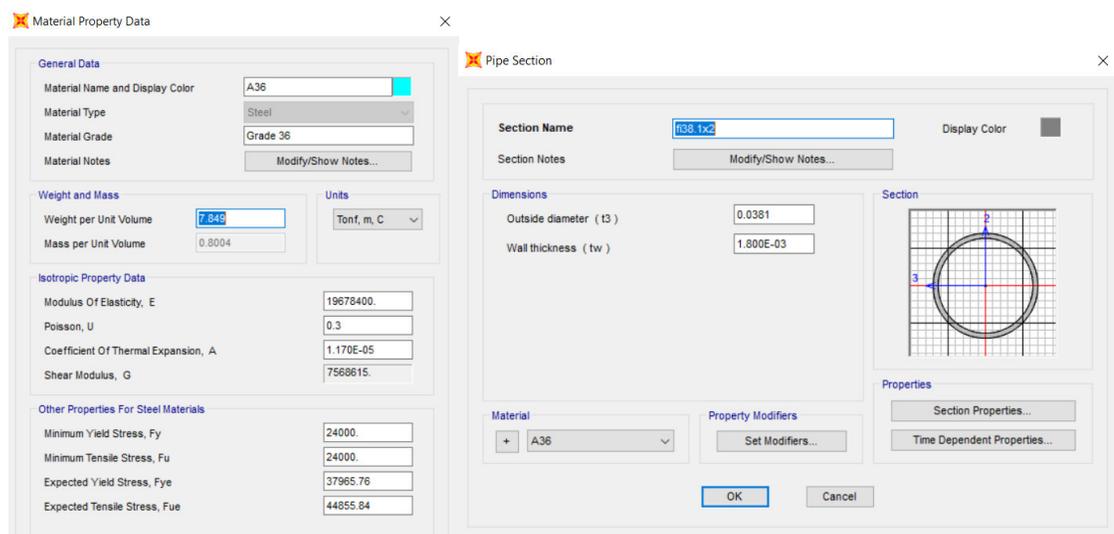
Nodos	Fxyi	Reacciones (Ton)
1	Fx1	-2,0459
	Fy1	3,5436
2	Fx2	0,0000
	Fy2	4,1757
3	Fx3	1,0859
	Fy3	1,8808
4	Px	0,96
	Py	-9,60

La tabla 3.4 muestra las reacciones en cada nodo en coordenadas globales, comprobando las fuerzas externas que se aplican en el nodo cuatro.

### 3.2.3. VALIDACIÓN EN SAP 2000 DEL ALGORITMO DEL MÉTODO MATRICIAL.

## MATERIALES Y SECCIONES

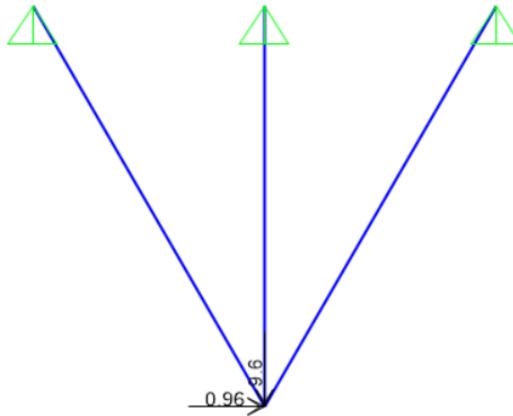
Se inserta las propiedades del material y la sección como se observa en la figura 3.5.



**Figura 3.5.** Propiedades del material y tipo de sección.

## GEOMETRÍA Y CARGAS

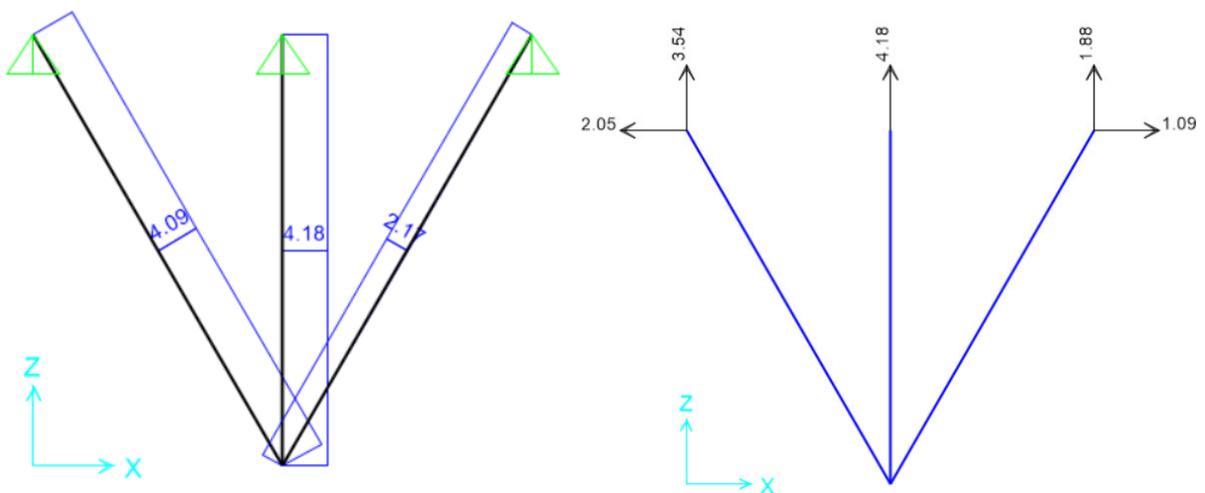
Se establece la geometría de la estructura y la carga aplicada como se observa en el figura 3.6.



**Figura 3.6.** Geometría de la estructura y carga aplicada.

## RESULTADOS.

Los resultados obtenidos en el programa SAP 2000 se observan en la figura 3.7.



**Figura 3.7.** Resultados obtenidos en el programa SAP2000.

Los resultados calculados por el método matricial son válidos debido a que arrojan los mismos resultados que el programa SAP 2000 como se puede observar en la tabla 3.2, 3.3, 3.4 y la figura 3.7.

### 3.3 ALGORITMO GENÉTICO DE SECCIONES

El algoritmo genético de secciones se elaboró con algoritmos en Matlab. Este algoritmo determina las secciones óptimas de una estructura tipo armadura con una forma definida, este algoritmo se encuentra en el Anexo No2. Para realizar este algoritmo se debe tener los siguientes parámetros:

- i. Cromosoma
- ii. Población inicial
- iii. Función objetivo
- iv. Selección
- v. Cruce el cual se encuentra en el Anexo No2. A2.1
- vi. Mutación el cual se encuentra en el Anexo No2. A2.2

#### 3.3.1. CROMOSOMA DE SECCIONES.

Se debe definir el cromosoma el cual involucra todas las secciones de todos los elementos de la estructura como se observa en la tabla 3.5.

**Tabla 3.5.**

*Cromosoma de secciones*

A1	A2	A3	.....	An
0   1   0	1   0   1	0   1   1	.....	0   0   1
Gen 1	Gen 2	Gen 3	.....	Gen n

Todas las secciones de los elementos se colocan en una matriz, las cuales se codifican en código binario para poder continuar con el procedimiento del algoritmo genético.

### 3.3.2. POBLACIÓN INICIAL

La población para optimización de secciones depende las secciones de la estructura, y la forma de obtener la población inicial es aleatoriamente de acuerdo a este método, esta se toma de una tabla de secciones comerciales, siendo asignado una sección a cada elemento de la estructura.

### 3.3.3. FUNCIÓN OBJETIVO.

La función objetivo será la optimización del peso de la estructura. La función objetivo a utilizar está definida por la ecuación 3.7.

$$f_{obj} = \frac{1}{W_i * (1 + B * C)} \quad (3.7)$$

En donde  $f_{obj}$  es la función objetivo;  $W_i$  peso de cada individuo; B es una constante la cual castiga a los individuos menos aptos, en este caso se toma el valor de 1000 de acuerdo a la recomendación de Coello; C es la penalización del individuo.

La penalización "C" se calcula de acuerdo a la fórmula 3.8 y 3.9, la cual relaciona el esfuerzo de los elementos con el esfuerzo máximo permisible. Siempre que el esfuerzo del elemento sea mayor al esfuerzo permisible, este elemento se penaliza, la penalización total es la suma de las penalizaciones de los elementos.

$$C_i = \frac{\sigma_{elemento}}{\sigma_{admisible}} < 1 \quad C_i = 0 \quad (3.8)$$

$$C = \sum_i^n C_i \quad (3.9)$$

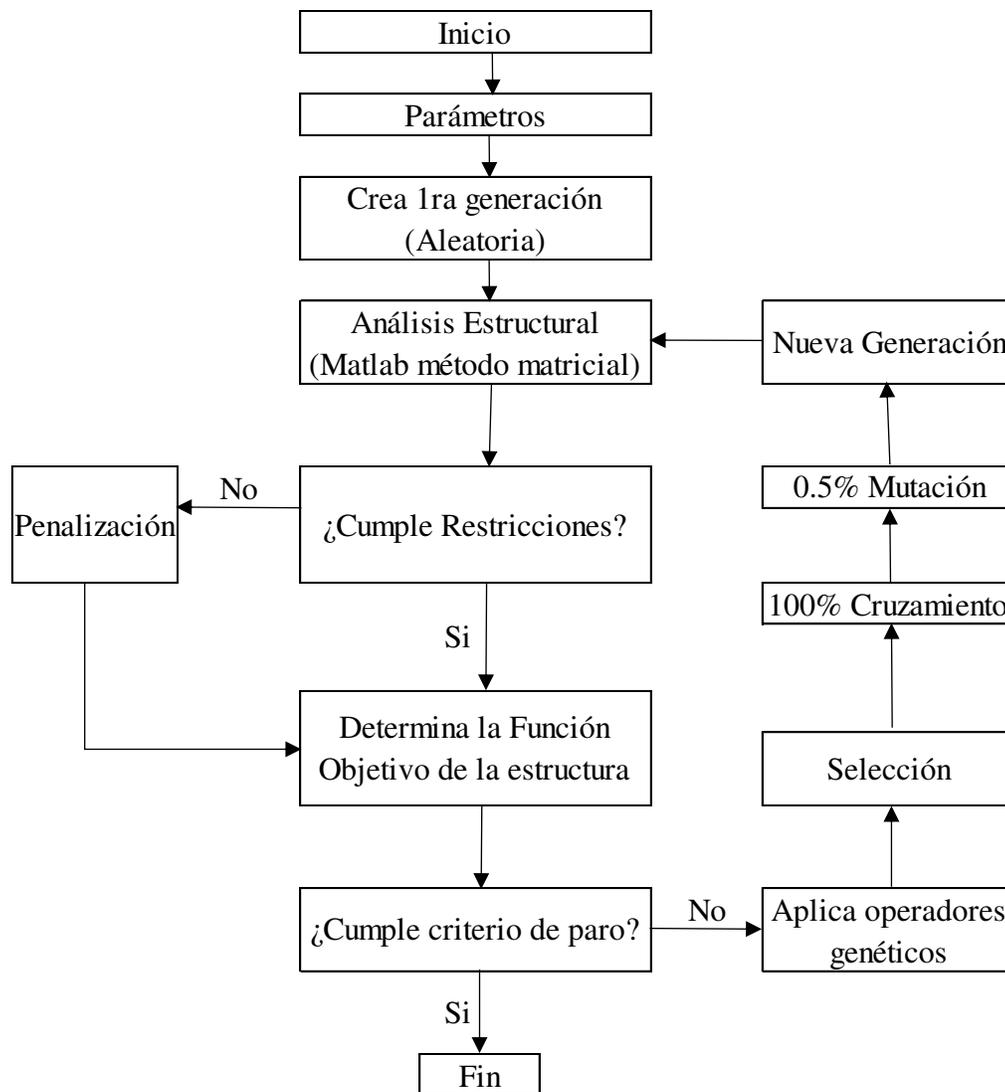
Donde  $C_i$  es la penalización de cada elemento;  $\sigma_{elemento}$  es el esfuerzo de cada elemento;  $\sigma_{admisible}$  es el esfuerzo admisible del elemento; C es la penalización de toda la armadura.

### 3.3.4. MÉTODO DE SELECCIÓN

Los métodos de selección a utilizar son los métodos de selección por ruleta y el método de selección por torneo cuyos algoritmos realizados en el programa Matlab se encuentra en el Anexo No2 A2.3. y A2.4. respectivamente.

### 3.3.5. DIAGRAMA DE FLUJO DEL ALGORITMO GENÉTICO DE SECCIONES.

El algoritmo genético de secciones sigue el diagrama de flujo de la figura 3.8.



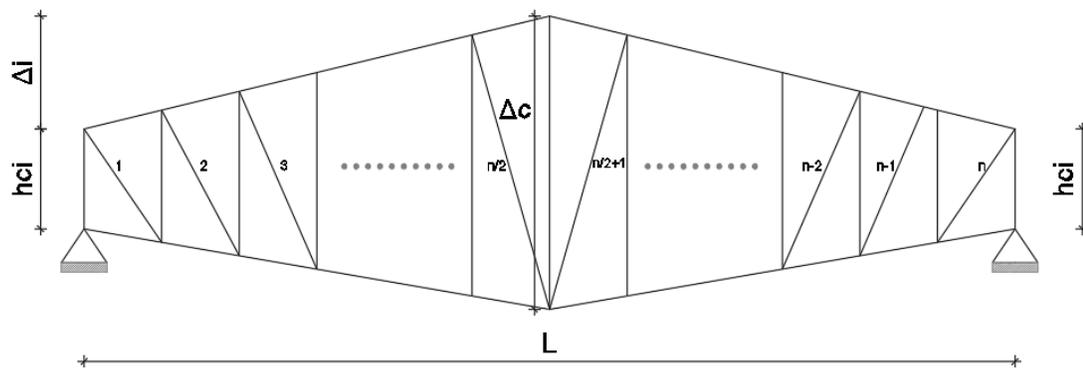
**Figura 3.8.** Diagrama de flujo de optimización de secciones.

### 3.4 PARAMETRIZACIÓN DE LA ESTRUCTURA.

La parametrización de la estructura es indispensable en la metodología a utilizar, para poder realizar la optimización de estructuras a nivel optimización de forma y topología.

#### 3.4.1. PARAMETRIZACIÓN TIPO PRATT

La parametrización de la estructura tipo Pratt se puede observar en la figura 3.9.

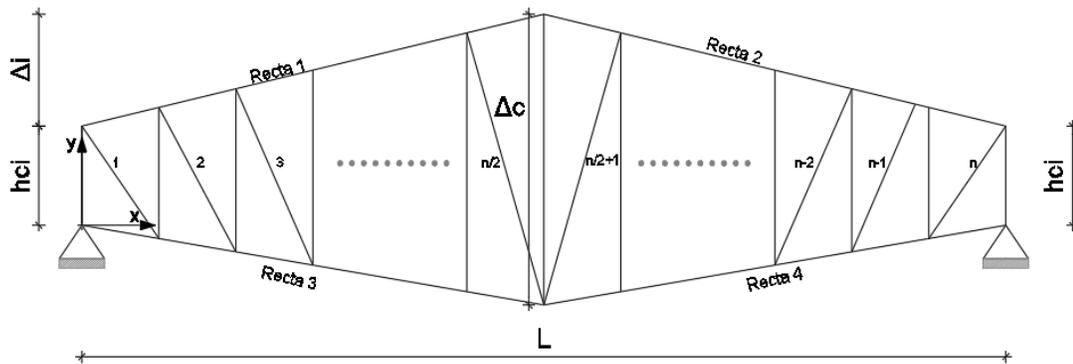


**Figura 3.9.** Parametrización de la armadura tipo Pratt.

Fuente: (Bonelli & Begliardo, 2016)

En donde  $h_{ci}$  es la altura del canto izquierdo de la armadura;  $\Delta i$  es la altura desde la altura del canto izquierdo hasta el punto más alto de la armadura;  $\Delta c$  es el ancho máximo que puede tener la armadura en la parte central;  $n$  representa el número de diagonales de la armadura;  $L$  es la luz que cubre la armadura.

En función de los cinco parámetros de la armadura se parametriza, la parametrización de la armadura observa en la figura 3.10.



**Figura 3.10.** Parametrización de la armadura tipo Pratt

Realizado por: Edison Hernán Garófalo Gaibor

Las coordenadas de los elementos de la estructura se obtienen con las ecuaciones de la recta como se observa en la ecuación 3.10.

$$(y - y_o) = m * (x - x_o) \quad (3.10)$$

En donde m es la pendiente de la recta a considerar; “x” y “y” son las coordenadas de la recta; xo y yo son las coordenadas iniciales.

Ecuación de la recta 1 se observa en la ecuación 3.11.

$$y = \left(2 * \frac{\Delta_i}{L}\right) * x + h_{ci} \quad (3.11)$$

Ecuación de la recta 2 se observa en la ecuación 3.12.

$$y = \left(-2 * \frac{\Delta_i}{L}\right) * \left(x - \frac{L}{2}\right) + (h_{ci} + \Delta_i) \quad (3.12)$$

Ecuación de la recta 3 se observa en la ecuación 3.13.

$$y = \left(-2 * \frac{(h_{ci} + \Delta_i - \Delta_c)}{L}\right) * (x) + (h_{ci} + \Delta_i) \quad (3.13)$$

Ecuación de la recta 4 se observa en la ecuación 3.14.

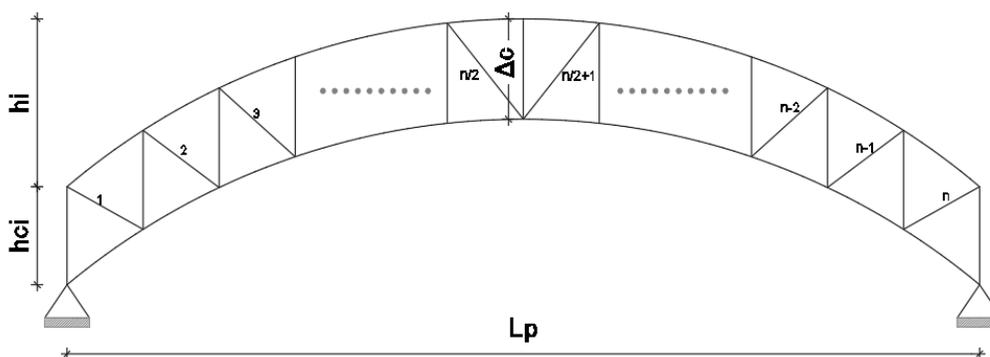
$$y = \left(2 * \frac{(h_{ci} + \Delta_i - \Delta_c)}{L}\right) * \left(x - \frac{L}{2}\right) + (h_{ci} + \Delta_i - \Delta_c) \quad (3.14)$$

Con las ecuaciones de la recta se obtiene la armadura de acuerdo a los parámetros indicados, con los cuales se determina las coordenadas de los elementos.

El programa en Matlab que parametriza la estructura tipo Pratt tiene el nombre fparametPrant el cual se encuentra en el Anexo No.3 A3.1.

### 3.4.2. PARAMETRIZACIÓN ARMADURA CIRCULAR

La armadura circular se encuentra en la figura 3.11, donde se observa los parámetros de la estructura.

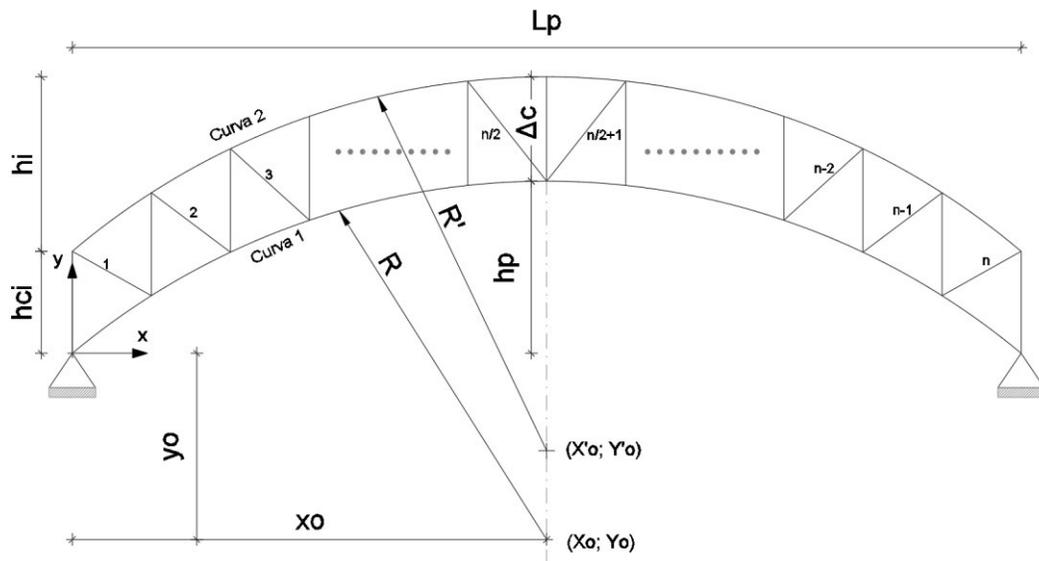


**Figura 3.11.** Armadura circular.

Realizado por: Edison Hernán Garófalo Gaibor

En donde  $h_{ci}$  es la altura del canto izquierdo;  $h_i$  es la altura del punto más alto;  $\Delta_c$  es el ancho máximo que puede tener la armadura en la parte central;  $n$  representa el número de diagonales de la armadura;  $L_p$  es la luz que cubre la armadura

La armadura se forma con los cinco parámetros de la armadura, la parametrización de la forma circular se observa en la figura 3.12.



**Figura 3.12.** Parametrización de la estructura circular.

Realizado por: Edison Hernán Garófalo Gaibor

La ecuación 3.15 representa la forma la ecuación circular de la estructura

$$(x - x_o)^2 + (y - y_o)^2 = R^2 \quad (3.15)$$

Despejando se obtiene la ecuación 3.16 que permite encontrar las coordenadas de la forma circular de la estructura.

$$y = \sqrt{R^2 - (x - x_o)^2} + y_o \quad (3.16)$$

En donde R es el radio de la circunferencia;  $x_o$  y  $y_o$  son las coordenadas del centro de la circunferencia. La ecuación 3.17 representa el radio de la circunferencia de acuerdo a los parámetros utilizados.

$$R = \frac{\frac{L_p^2}{4} + h_p^2}{2 * h_p} \quad (3.17)$$

En donde  $L_p$  que es la longitud de la armadura;  $h_p$  es la altura máxima de la circunferencia.

La coordenada del centro de la circunferencia se obtiene con las ecuaciones 3.18 y 3.19 respectivamente.

$$x_o = \frac{L_p}{2} \quad (3.18)$$

$$y_o = R - h_p \quad (3.19)$$

El programa en Matlab que parametrizar la estructura tipo circular tiene el nombre de *fparamtCircular* se puede observar en el Anexo No.3. A3.2

### 3.5 AUTOMATIZACIÓN DEL PROGRAMA PARA ARMADURAS EN DOS DIMENSIONES.

Para la automatización del análisis matricial, en este caso, se utiliza la idea de las coordenadas de la estructura, para poder encontrar automáticamente los nodos y sus grados de libertad de acuerdo a estas coordenadas. Esta matriz de coordenadas se debe representar como se observa en la ecuación 3.20.

$$CoordXY = \begin{bmatrix} x_{o1} & y_{o1} & x_{f1} & y_{f1} \\ x_{o2} & y_{o2} & x_{f2} & y_{f2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{on} & y_{on} & x_{fn} & y_{f2} \end{bmatrix} \quad (3.20)$$

Las filas representan cada elemento de la estructura. En donde la coordenada cartesiana está representada en cuatro columnas, las primeras dos columnas representan las coordenadas iniciales del elemento y las siguientes dos columnas representa las coordenadas finales del elemento.

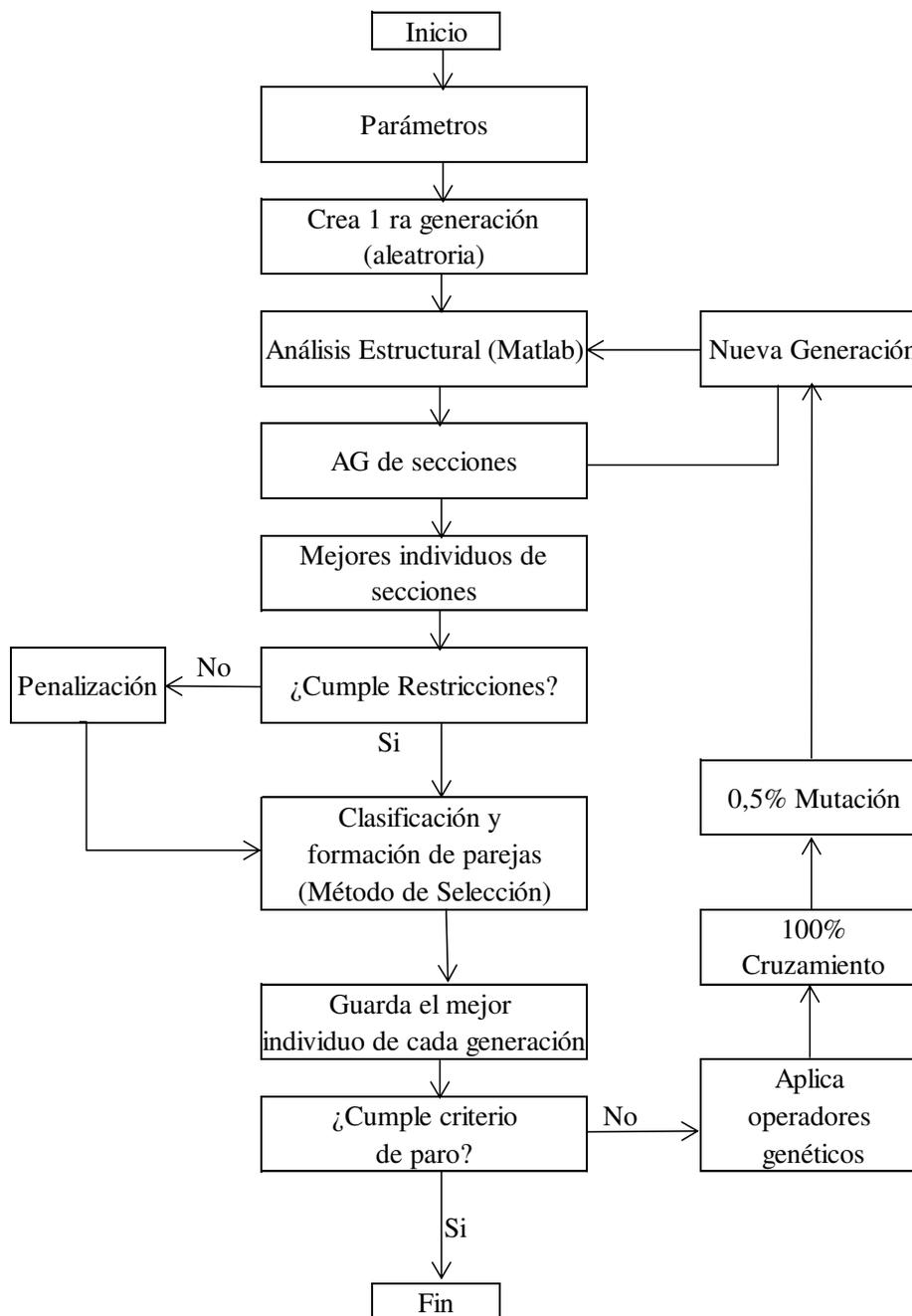
El siguiente paso es automatizar las matrices *fnodos1*, *fCoordNodos*, *fgradoslibertadAG*, *M\_TAP2*, *M\_KAP2AG*, las cargas y la matriz de ensamble. Todo esto se realizó utilizando rutinas y funciones en el programa Matlab, las mismas que se observa en el Anexo No4.

### **3.6 ALGORITMO GENÉTICO DE SECCIONES, FORMA Y TOPOLOGÍA.**

El algoritmo genético de secciones, forma y topología se caracteriza por realizar primero la optimización de secciones de todos los individuos y luego realizar la optimización de forma y topología siguiendo los parámetros de la estructura a analizar de acuerdo a (Bonelli & Begliardo, 2016). El procedimiento del algoritmo genético de forma y topología es similar al algoritmo genético de secciones con la única diferencia que cambia los parámetros del cromosoma.

#### **3.6.1. DIAGRAMA DE FLUJO DEL ALGORITMO GENÉTICO DE SECCIONES, FORMA Y TOPOLOGÍA.**

El diagrama de flujo de la figura 3.13. representa el funcionamiento del algoritmo para determinar el mejor individuo optimizando secciones, forma y topología.

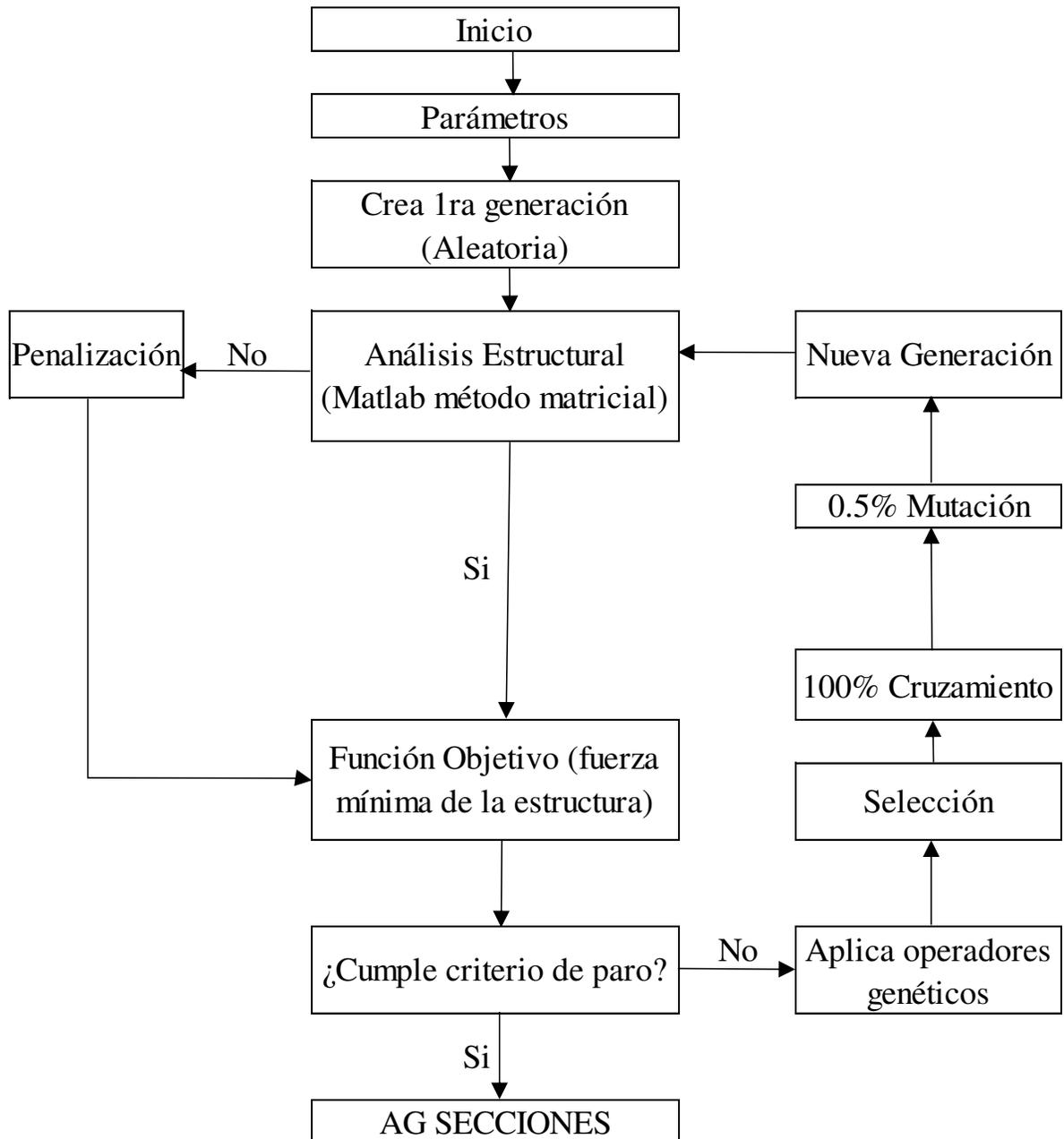


**Figura 3.13.** Diagrama de flujo del algoritmo genético de secciones, forma y topología

Realizado por: Edison Hernán Garófalo Gaibor

Realizar la operación del diagrama de flujo de la figura 3.13 el tiempo computacional sería muy extenso, y sin tener la certeza de obtener el mejor individuo, por lo que se propone dejar la optimización de secciones para el final; Por lo que es

necesario utilizar un nuevo parámetro en la función objetivo el cual se basa en la geometría que genere la fuerza mínima de los elementos de la estructura, dentro los parámetros geométricos establecidos. Utilizando esta idea el diagrama de flujo de la figura 3.14. representa la optimización de forma, topología y secciones.



**Figura 3.14.** Diagrama de flujo del algoritmo genético de forma, topología y secciones.

Realizado por: Edison Hernán Garófalo Gaibor

Es importante mencionar que inicialmente para la optimización de forma y topología todas los elementos tienen el mismo tipo de sección para poder determinar las fuerzas internas que se generan en los elementos. La penalización se basa en la fuerza máxima del elemento, la misma que toma como referencia la fuerza máxima que puede soportar la sección comercial de mayor sección, cualquier fuerza de un elemento que supere la fuerza máxima se penaliza.

### 3.6.2. CROMOSOMA DE FORMA Y TOPOLOGÍA

El cromosoma de forma y topología se utiliza los parámetros que forman la estructura que son  $h_{pi}$ ,  $\Delta_i$ ,  $\Delta_c$  y  $n$ , el parámetro topológico es el número de diagonales. El cromosoma se puede observar en la tabla 3.6.

**Tabla 3.6.**

*Cromosoma de forma y topología*

Forma y topología											
$h_{pi}$			$\Delta_i$			$\Delta_c$			n		
1	1	0	1	0	0	1	0	1	0	0	1
Gen 1			Gen 2			Gen 3			Gen 4		

## CAPÍTULO 4.

### VALIDACIÓN DEL MÉTODO MATRICIAL Y LA PARAMETRIZACIÓN DE LA ESTRUCTURA.

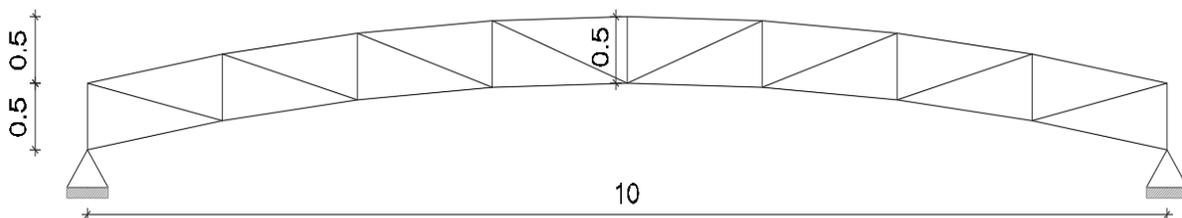
#### 4.1 INTRODUCCIÓN

La efectividad del método matricial para resolver armaduras se debe validar, por lo que se resuelve un problema en SAP 2000 y se procede a comparar con los resultados del análisis matricial. La parametrización de la armadura se válida resolviendo dos problemas de armaduras con parámetros distintos y comparando los resultados con la resolución de los mismos problemas en el programa de SAP 2000.

La importancia de la validación de los resultados del algoritmo en Matlab para resolver estructuras tipo armadura nos permite seguir al siguiente nivel en la optimización de estructuras.

#### 4.2 VALIDACIÓN DEL MÉTODO MATRICIAL

El método matricial se valida con un ejercicio de una armadura de 33 elementos como se observa en la figura 4.1



**Figura 4.1.** Ejercicio armadura de 33 elementos.

Se asume una carga distribuida de  $157.60 \text{ kg/m}^2$ , la estructura tiene un área colaborante de cinco metros por lo que la carga puntual en cada nodo de la estructura es de una tonelada, las propiedades de la estructura se encuentran en la tabla 4.1.

**Tabla 4.1.**

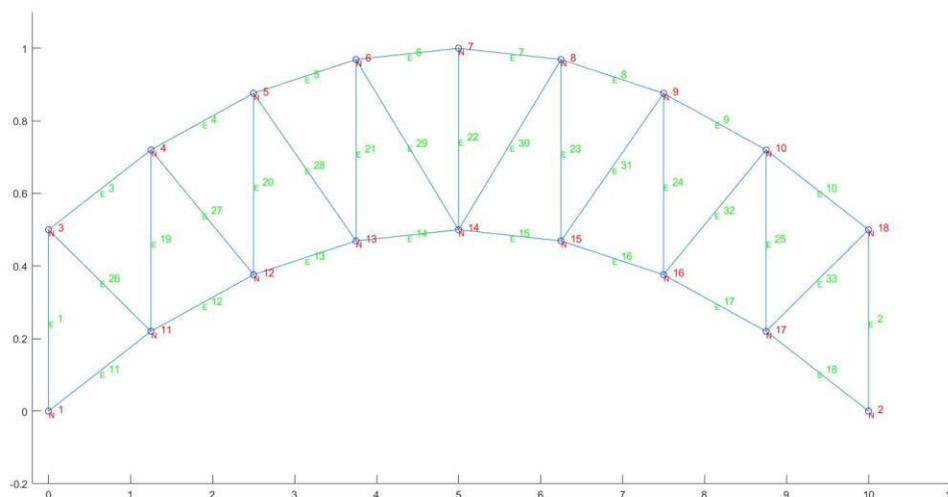
*Propiedades de la estructura*

<b>DATOS</b>	<b>EJERCICIO 1</b>
Diámetro sección tubular 50,8x2 mm	50,80 mm
Módulo de elasticidad	2030000 $\text{kg/cm}^2$
Límite de fluencia	2560 $\text{kg/cm}^2$
Coefficiente de Poisson	0,30

Realizado por: Edison Hernán Garófalo Gaibor

#### **4.2.1. SOLUCIÓN DEL EJERCICIO UNO VALIDACIÓN DEL MÉTODO MATRICIAL**

Para la solución del ejercicio uno se encuentra en el Anexo No.5 A5.1 programa Validación Método matricial el cual se procede a utilizar los algoritmos en el programa Matlab. Para la solución del ejercicio se tiene la numeración de los nodos y de los elementos como se observa en la figura 4.2.



**Figura 4.2.** Ejercicio uno armadura 33 elementos

Realizado por: Edison Hernán Garófalo Gaibor

Los resultados del algoritmo utilizando el programa en Matlab se pueden observar en la tabla 4.2.

**Tabla 4.2.**

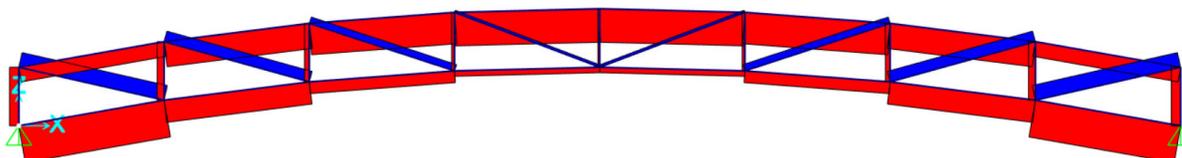
*Resultados del ejercicio uno utilizando el método matricial*

ELEM.	FUERZA (TON)						
1	-2.66	10	-4.21	19	-1.98	28	1.89
2	-2.66	11	-10.62	20	-1.36	29	0.64
3	-4.21	12	-6.37	21	-0.80	30	0.64
4	-7.19	13	-3.34	22	-0.53	31	1.89
5	-8.96	14	-1.53	23	-0.80	32	3.10
6	-9.54	15	-1.53	24	-1.36	33	4.25
7	-9.54	16	-3.34	25	-1.98		
8	-8.96	17	-6.37	26	4.25		
9	-7.19	18	-10.62	27	3.10		

Realizado por: Edison Hernán Garófalo Gaibor

#### 4.2.2. SOLUCIÓN DEL EJERCICIO UNO EN SAP 2000

El cálculo en el programa SAP 2000 se procede a realizar siguiendo la geometría las cargas externas y las propiedades de los materiales. La figura 4.3 muestra los resultados del programa SAP 2000.



**Figura 4.3.** Resultados ejercicio uno programa SAP 2000.

Realizado por: Edison Hernán Garófalo Gaibor

Los resultados calculados por el programa SAP 2000 se ubican en la tabla 4.3.

**Tabla 4.3.**

*Resultados del ejercicio uno utilizando el programa SAP2000*

ELEMENTO	FUERZA (TON)						
1	-2.67	10	-4.23	19	-1.98	28	1.88
2	-2.67	11	-10.63	20	-1.36	29	0.64
3	-4.23	12	-6.35	21	-0.79	30	0.64
4	-7.20	13	-3.34	22	-0.52	31	1.88
5	-8.95	14	-1.54	23	-0.79	32	3.09
6	-9.53	15	-1.54	24	-1.36	33	4.27
7	-9.53	16	-3.34	25	-1.98		
8	-8.95	17	-6.35	26	4.28		
9	-7.20	18	-10.63	27	3.09		

Realizado por: Edison Hernán Garófalo Gaibor

La variación de los resultados entre el método matricial y el programa SAP 2000 es del 0.1% por lo que el método matricial es válido para realizar el análisis estructural.

### 4.3 VALIDACIÓN DE LA PARAMETRIZACIÓN

Se realiza la validación de la parametrización de la estructura cambiando los parámetros de la estructura y resolviendo estos problemas tanto con el algoritmo de

análisis matricial como con el programa SAP 2000, para validar los resultados. Los parámetros a utilizar se observan en la tabla 4.4.

**Tabla 4.4.**

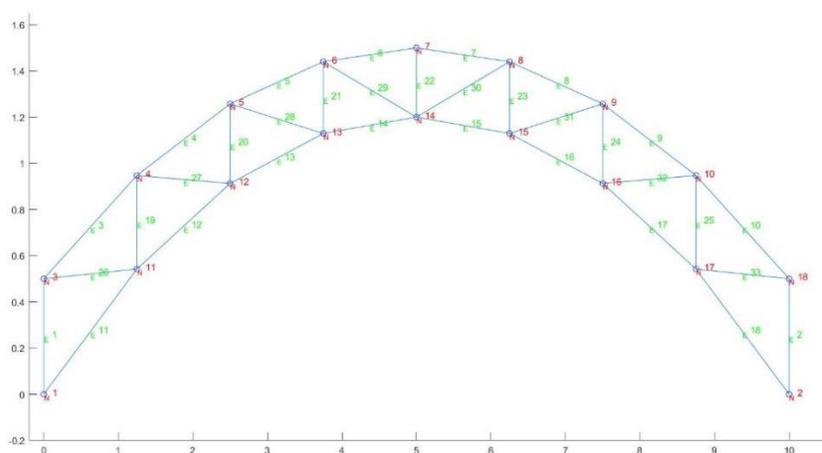
*Parámetros de la estructura forma circular*

PARAMETROS	EJERCICIO 2	EJERCICIO 3
Altura extremo izquierdo "hci"	0,50 m	0,40 m
Altura del extremo izquierdo al punto más alto "hi"	1,00 m	0,50 m
Ancho de la armadura en la parte central " $\Delta c$ "	0,30 m	0,25 m
Longitud de la armadura "Lp"	10 m	9 m
Número de diagonales "n"	8	6
Carga repartida "q"	180 kg/m <sup>2</sup>	240 kg/m <sup>2</sup>
Ancho colaborante	5 m	4 m
Carga vertical en cada nodo superior	1000,00 kg	1234,29 kg

#### 4.3.1. SOLUCIÓN DEL EJERCICIO DOS Y TRES POR EL MÉTODO MATRICIAL

##### EJERCICIO DOS

Se ingresa los datos y parámetros del ejercicio dos en el programa validación paramétrica el cual se encuentra en el Anexo No5 A5.2, donde se obtiene la figura 4.4.



**Figura 4.4.** Ejercicio dos numeración de elementos y nodos.

Realizado por: Edison Hernán Garófalo Gaibor

Los resultados del ejercicio dos por el método matricial se encuentran en la tabla 4.5.

**Tabla 4.5.**

*Resultados del ejercicio dos por el método matricial*

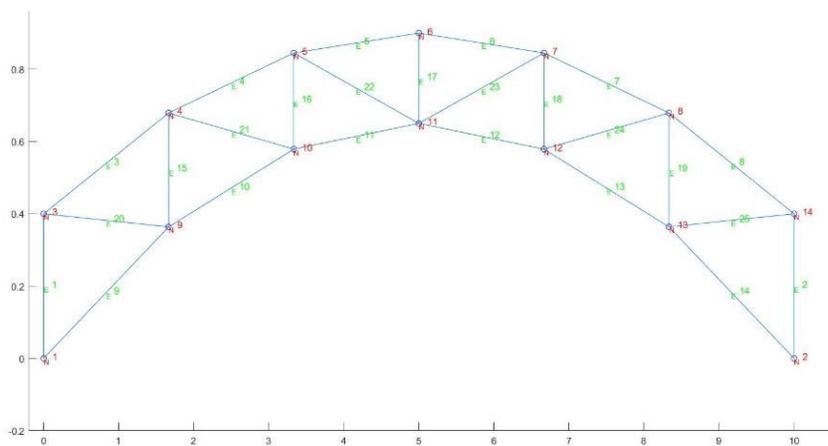
ELEMENTO	FUERZA (TON)						
1	-1.42	10	-1.38	19	-1.32	28	1.43
2	-1.42	11	-7.73	20	-1.05	29	0.57
3	-1.38	12	-6.04	21	-0.71	30	0.57
4	-3.06	13	-4.18	22	-0.52	31	1.43
5	-4.44	14	-2.71	23	-0.71	32	1.67
6	-4.96	15	-2.71	24	-1.05	33	1.30
7	-4.96	16	-4.18	25	-1.32		
8	-4.44	17	-6.04	26	1.30		
9	-3.06	18	-7.73	27	1.67		

Realizado por: Edison Hernán Garófalo Gaibor

El programa utilizado para el cálculo matricial se encuentra en el Anexo No.5 A5.2

### EJERCICIO TRES

Se ingresa los parámetros y los datos del ejercicio tres en el programa Validación Paramétrica del Anexo No5 A5.2, de donde se obtiene la figura 4.5.



**Figura 4.5.** Ejercicio tres numeración de elementos y nodos.

Realizado por: Edison Hernán Garófalo Gaibor

Los resultados del ejercicio tres por el método matricial se encuentran en la tabla 4.6.

**Tabla 4.6.**

*Resultados del ejercicio tres método matricial*

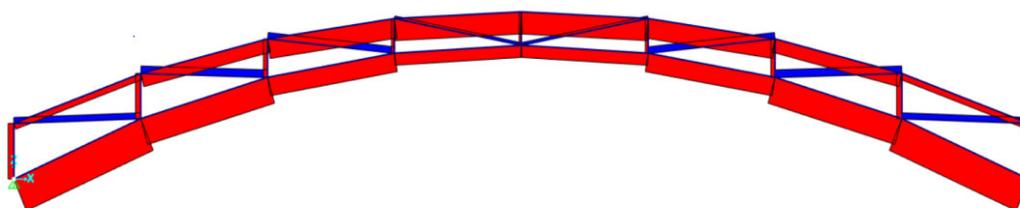
ELEMENTO	FUERZA (TON)	ELEMENTO	FUERZA (TON)	ELEMENTO	FUERZA (TON)
1	-1.85	10	-7.29	19	-1.51
2	-1.85	11	-4.43	20	2.94
3	-2.99	12	-4.43	21	2.79
4	-5.77	13	-7.29	22	1.20
5	-6.92	14	-10.45	23	1.20
6	-6.92	15	-1.51	24	2.79
7	-5.77	16	-1.01	25	2.94
8	-2.99	17	-0.73		
9	-10.45	18	-1.01		

Realizado por: Edison Hernán Garófalo Gaibor

#### 4.3.2. SOLUCIÓN DEL EJERCICIO DOS Y TRES EN SAP 2000

##### EJERCICIO DOS

El cálculo en el programa SAP 2000 se procede a realizar siguiendo la geometría las cargas externas y las propiedades de los materiales. La figura 4.6. muestra los resultados del programa SAP 2000.



**Figura 4.6.** Resultados ejercicio dos programa SAP 2000

Realizado por: Edison Hernán Garófalo Gaibor

Los resultados calculados por el programa SAP 2000 se ubican en la tabla 4.7.

**Tabla 4.7.**

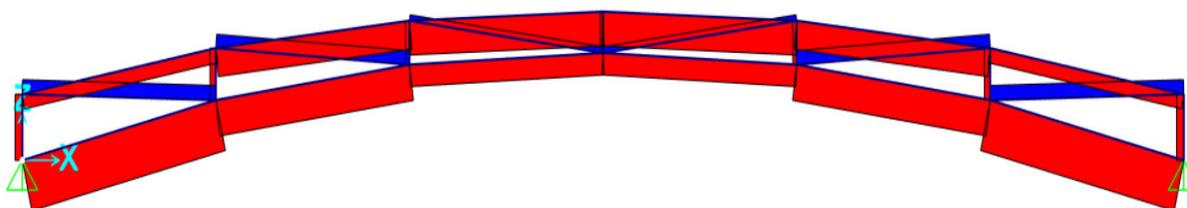
*Resultados del ejercicio dos utilizando SAP 2000.*

ELEMENTO	FUERZA (TON)						
1	-1.42	10	-1.39	19	-1.32	28	1.43
2	-1.42	11	-7.73	20	-1.05	29	0.57
3	-1.39	12	-6.04	21	-0.71	30	0.57
4	-3.06	13	-4.18	22	-0.52	31	1.43
5	-4.44	14	-2.71	23	-0.71	32	1.67
6	-4.96	15	-2.71	24	-1.05	33	1.30
7	-4.96	16	-4.18	25	-1.32		
8	-4.44	17	-6.04	26	1.30		
9	-3.06	18	-7.73	27	1.67		

La variación entre el análisis matricial y el programa SAP 2000 es de 0.01%

### EJERCICIO 3

El cálculo en el programa SAP 2000 se procede a realizar siguiendo la geometría las cargas externas y las propiedades de los materiales. La figura 4.7 muestra los resultados del programa SAP 2000.



**Figura 4.7.** Resultados ejercicio tres programa SAP 2000

Realizado por: Edison Hernán Garófalo Gaibor

Los resultados calculados por el programa SAP 2000 se ubican en la tabla 4.8

**Tabla 4.8.**

Resultados del ejercicio tres utilizando SAP 2000.

ELEMENTO	FUERZA (TON)	ELEMENTO	FUERZA (TON)	ELEMENTO	FUERZA (TON)
1	-1.85	10	-7.29	19	-1.51
2	-1.85	11	-4.43	20	2.94
3	-2.99	12	-4.43	21	2.79
4	-5.77	13	-7.29	22	1.20
5	-6.92	14	-10.45	23	1.20
6	-6.92	15	-1.51	24	2.79
7	-5.77	16	-1.01	25	2.94
8	-2.99	17	-0.73		
9	-10.45	18	-1.01		

La variación entre el análisis matricial y el programa SAP 2000 es de 0.01%

## **CAPÍTULO 5.**

### **FORMULACIÓN Y RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE ARMADURAS PLANAS.**

#### **5.1 INTRODUCCIÓN**

En el presente capítulo se validará los algoritmos realizados en Matlab para la optimización, con problemas conocidos en el transcurso del tiempo para la validación de los algoritmos genéticos. En la optimización de secciones se utilizan el problema de los tres elementos y de los diez elementos para determinar la efectividad del algoritmo de acuerdo al método de selección utilizado.

En la validación de secciones, forma y topología de armaduras paramétricas se utilizará el problema planteado por (Bonelli & Begliardo, 2016), con este problema se validará la efectividad del algoritmo de forma y topología, utilizando ambos métodos de selección, con la estructura determinada se realizará la optimización de secciones para poder determinar el peso mínimo de la estructura.

#### **5.2 PROBLEMAS DE OPTIMIZACIÓN DE SECCIONES EN ARMADURAS PLANAS.**

Para la optimización de secciones se toma los siguientes problemas de validación:

- i. Armadura plana de tres elementos.
- ii. Armadura plana de diez elementos.

### 5.2.1 ARMADURA PLANA DE TRES ELEMENTOS

#### DATOS DE LA ARMADURA DE TRES ELEMENTOS.

Datos:

Densidad del material  $\rho = 7850 \frac{\text{kgf}}{\text{cm}^2}$

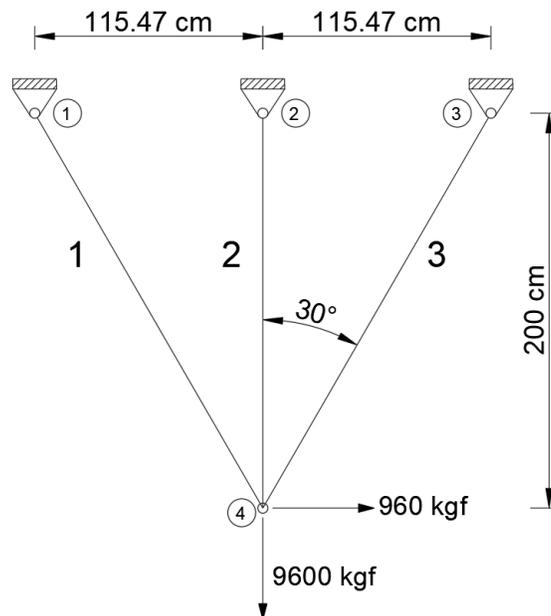
Esfuerzo máximo permitido  $\sigma = \pm 1442 \frac{\text{kgf}}{\text{cm}^2}$

Desplazamiento máximo permitido  $\mu = \pm 0.5 \text{ cm}$

Módulo de elasticidad  $E = 1967840 \frac{\text{kgf}}{\text{cm}^2}$

Archivo de áreas y pesos de perfiles comerciales, se encuentra en el Anexo No6.

La armadura plana de tres elementos es ampliamente utilizada para la validación de algoritmos genéticos, se puede observar en la figura 5.1.



**Figura 5.1.** Armadura plana de tres elementos

Realizado por: Edison Hernán Garófalo Gaibor

## PROCEDIMIENTO DE OPTIMIZACIÓN DE LA ARMADURA DE TRES ELEMENTOS.

Se ingresa los datos de la armadura, datos geométricos, datos de propiedades de los materiales y los datos del algoritmo genético de secciones.

Los datos geométricos se representan en las coordenadas de la estructura.

Los datos de las propiedades de los materiales, representa el esfuerzo permitido, el módulo de elasticidad, la densidad de los materiales, tabla de secciones de los materiales y el desplazamiento máximo permitido.

Los datos del algoritmo genético de secciones, viene dado por la cantidad de individuos por generación, cantidad de generaciones, porcentaje de mutación y método de selección, todos estos datos se los puede observar en la tabla 5.1.

**Tabla 5.1.**

*Datos del algoritmo genético de secciones*

Número de individuos por generación	10
Cantidad de generaciones	10
Porcentaje de mutación	1%
Método de selección	Ruleta y Torneo

La solución de este problema se encuentra en el Anexo No.7 A7.1, la cual se realizó utilizando el método de selección por ruleta y también por torneo. La optimización se realizó con cien iteraciones (número de optimizaciones) con cada método para obtener una comparación adecuada.

Los mejores resultados obtenidos por cada método se pueden apreciar en la tabla 5.2.

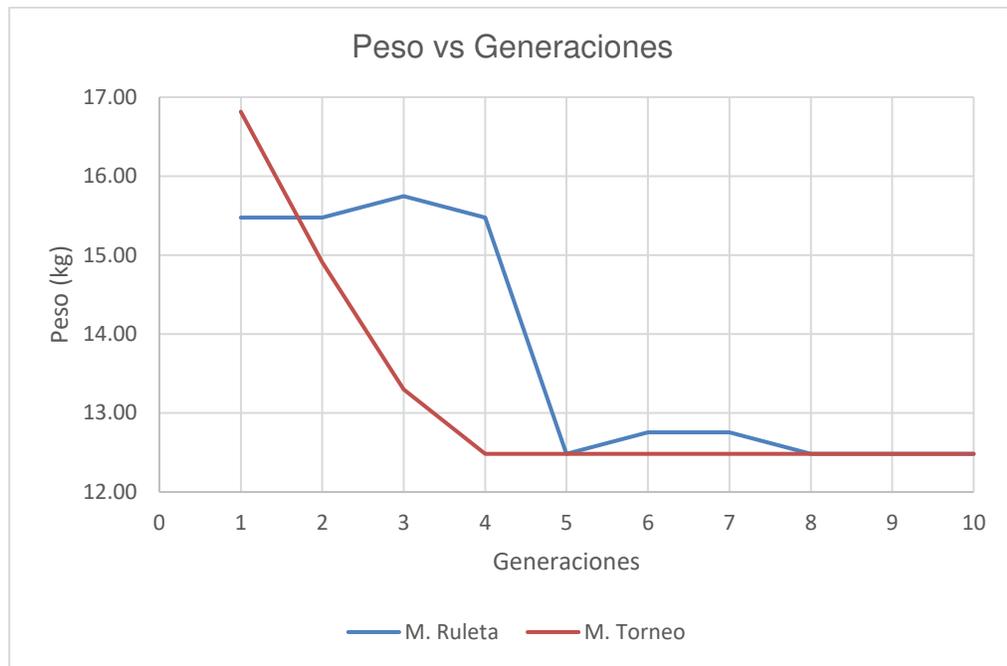
**Tabla 5.2.**

Resultados obtenidos por el algoritmo genético de secciones para la estructura de tres elementos, utilizando los dos métodos de selección.

GENERACIONES	PESO (kg) M.	PESO (kg) M.
	RULETA	TORNEO
1	15.47	16.82
2	15.47	14.91
3	15.75	13.30
4	15.47	12.48
5	12.48	12.48
6	12.75	12.48
7	12.75	12.48
8	12.48	12.48
9	12.48	12.48
10	12.48	12.48

Realizado por: Edison Hernán Garófalo Gaibor

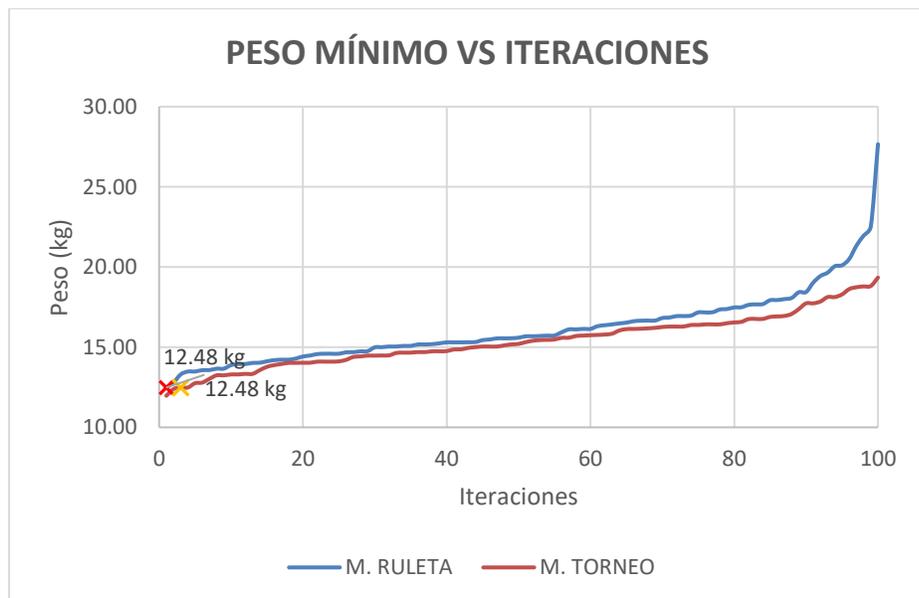
El algoritmo obtuvo las siguientes secciones en la solución de este problema:  $1.87\text{ cm}^2$ ,  $4.66\text{ cm}^2$  y  $0.98\text{ cm}^2$  para el elemento uno, dos y tres respectivamente, obteniendo un peso de  $12.48\text{ kg}$  en ambos métodos, la comparación de los dos métodos se puede observar en la figura 5.2.



**Figura 5.2.** Peso vs número de generaciones ejercicio de tres elementos.

Realizado por: Edison Hernán Garófalo Gaibor

Los resultados de la optimización de cien iteraciones ordenados de menor a mayor por cada método se observan en la figura 5.3.



**Figura 5.3.** Peso mínimo vs iteraciones, ejercicio de tres elementos.

Realizado por: Edison Hernán Garófalo Gaibor

En ambos casos se puede observar en la figura 5.3 a través de las cien iteraciones que llegan al peso mínimo, sin embargo, el método por torneo se aprecia pesos menores en todos los casos.

El peso obtenido por el algoritmo genético de secciones obtiene el peso mínimo a través de las generaciones, pero es necesario realizar iteraciones para encontrar pesos menores que el obtenido por Martínez en el 2005. El algoritmo genético de secciones es válido y nos permite encontrar óptimos globales. Sin embargo, para determinar el peso mínimo de la estructura es necesario realizar las iteraciones necesarias, que en este caso fueron cien.

### **5.2.2. ARMADURA PLANA DE DIEZ ELEMENTOS**

#### **DATOS DE LA ARMADURA DE DIEZ ELEMENTOS.**

Datos:

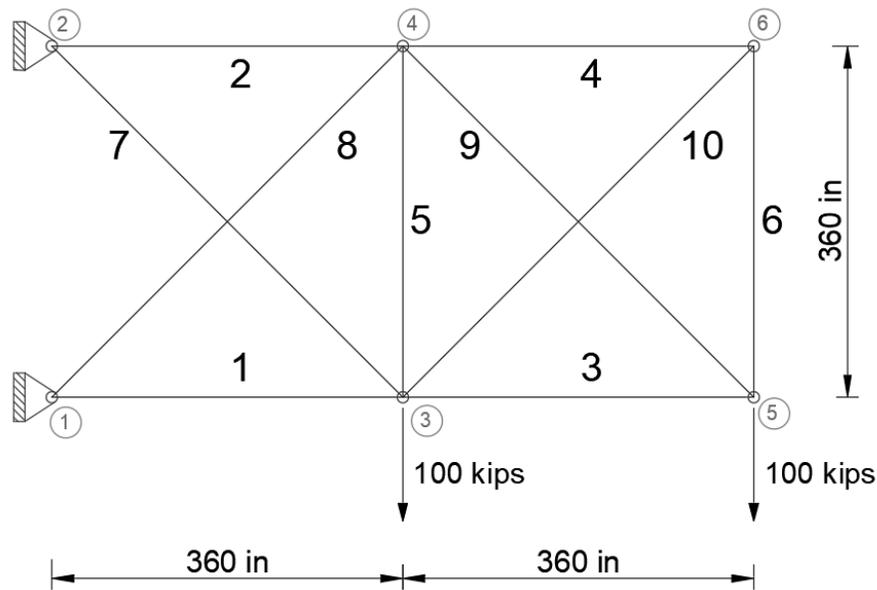
Densidad del material  $\rho = 2770 \frac{kgf}{m^3}$

Esfuerzo máximo permitido  $\sigma = \pm 25 \text{ ksi}$

Desplazamiento máximo permitido  $\mu = \pm 2 \text{ in}$

Módulo de elasticidad  $E = 10000 \text{ ksi}$

La armadura de diez elementos es ampliamente utilizada en la validación de algoritmos genéticos, se la puede observar en la figura 5.4.



**Figura 5.4.** Armadura plana de diez elementos

Realizado por: Edison Hernán Garófalo Gaibor

### PROCEDIMIENTO DE OPTIMIZACIÓN DE LA ARMADURA DE DIEZ ELEMENTOS.

Se ingresa los datos de la armadura, datos geométricos, datos de propiedades de los materiales y los datos del algoritmo genético de secciones.

Los datos geométricos se representan en las coordenadas de la estructura.

Los datos de las propiedades de los materiales, representa el esfuerzo permitido, el módulo de elasticidad, la densidad de los materiales, tabla de secciones de los materiales y el desplazamiento máximo permitido.

Los datos del algoritmo genético de secciones, viene dado por la cantidad de individuos por generación, cantidad de generaciones, porcentaje de mutación y método de selección, todos estos datos se los puede observar en la tabla 5.3.

**Tabla 5.3.***Datos del algoritmo genético de secciones*

Número de individuos por generación	10
Cantidad de generaciones	50
Porcentaje de mutación	1%
Método de selección	Ruleta y Torneo

La solución de este problema se encuentra en el Anexo No7 A7.2, la cual se realizó utilizando el método de selección por ruleta y también por torneo.

Los resultados obtenidos por los dos métodos se pueden apreciar en la tabla 5.4 y 5.5.

**Tabla 5.4.**

Resultados obtenidos por el algoritmo genético de secciones para la estructura de diez elementos, utilizando el método de la ruleta.

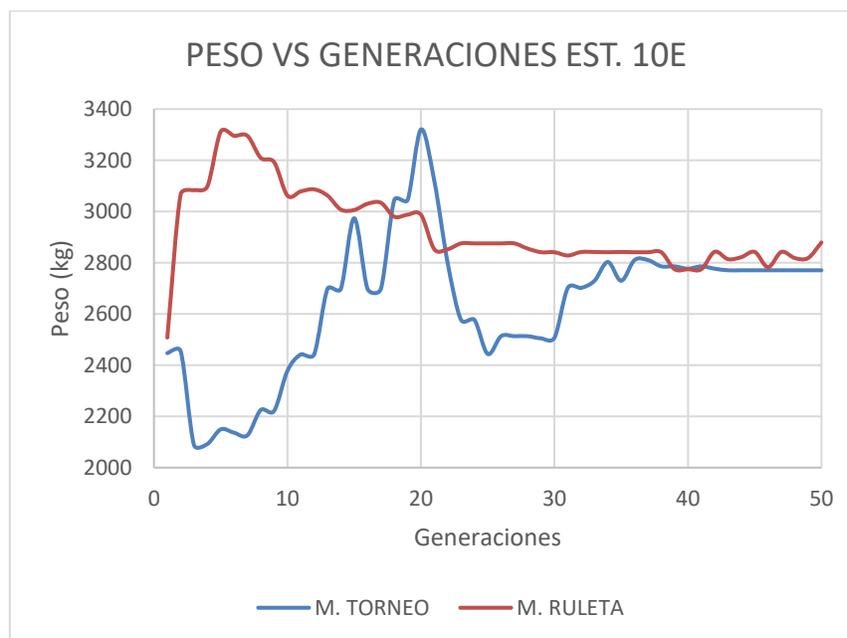
<b>M. RULETA PESO (kg)</b>					
<b>GENERACIONES</b>	<b>1-10</b>	<b>11-20</b>	<b>21-30</b>	<b>31-40</b>	<b>41-50</b>
1	2507.9	3078.6	2852.6	2828.2	2774.7
2	3066.8	3086.5	2852.6	2842.0	2842.8
3	3083.2	3062.2	2875.5	2842.0	2814.0
4	3097.9	3007.7	2875.5	2841.0	2821.8
5	3312.0	3006.0	2875.5	2842.0	2842.2
6	3295.6	3030.3	2875.5	2841.0	2782.6
7	3295.6	3034.2	2875.5	2841.0	2842.4
8	3209.3	2979.8	2855.1	2841.0	2818.4
9	3192.9	2987.6	2841.0	2774.7	2818.4
10	3062.2	2987.6	2841.0	2774.7	2879.2

**Tabla 5.5.**

Resultados obtenidos por el algoritmo genético de secciones para la estructura de diez elementos, utilizando el método por torneo.

M. TORNEO PESO (kg)					
GENERACIONES	1-10	11-20	21-30	31-40	41-50
1	2447.1	2442.2	3119.6	2701.8	2786.0
2	2455.3	2442.2	2799.3	2701.8	2776.7
3	2088.3	2698.0	2577.3	2729.6	2770.6
4	2092.2	2698.0	2577.3	2803.1	2770.6
5	2149.4	2973.7	2443.9	2729.6	2770.6
6	2136.4	2698.0	2513.2	2810.5	2770.6
7	2126.4	2698.0	2513.2	2810.5	2770.6
8	2225.3	3044.6	2513.2	2786.0	2770.6
9	2221.4	3044.6	2504.7	2786.0	2770.6
10	2378.4	3320.3	2507.0	2776.7	2770.6

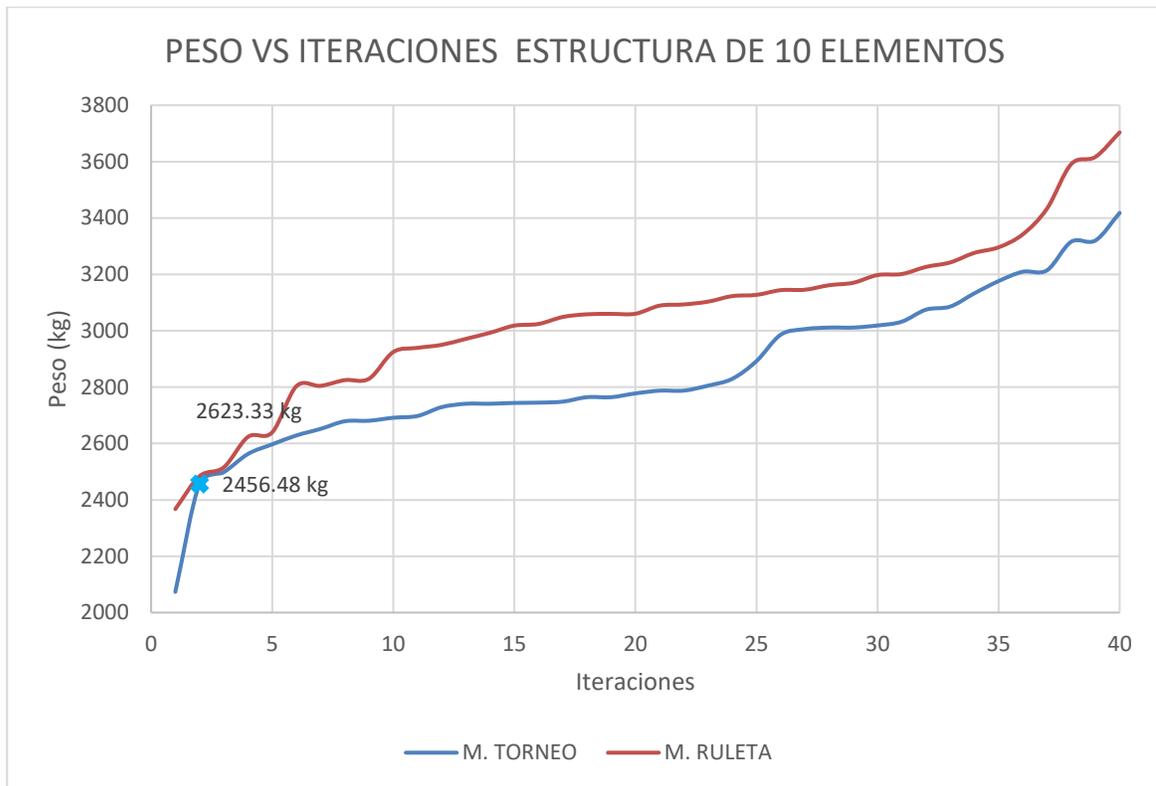
El grafico de la figura 5.5. se observa la optimización de la estructura de diez elementos por ambos métodos, obteniendo el peso mínimo por el método de la ruleta y el método por torneo, obteniendo un peso de 2879.2kg y 2770.6 kg respectivamente.



**Figura 5.5.** Peso vs número de generaciones ejercicio de diez elementos.

Realizado por: Edison Hernán Garófalo Gaibor.

En la figura 5.6 se puede observar el peso mínimo de la estructura de acuerdo al número de iteraciones (entendiendo por iteración cada proceso de optimización) que en este caso fueron cuarenta, las cuales se ordenaron de menor a mayor para la gráfica, utilizando ambos métodos de selección. Los valores menores al peso mínimo no cumplen con las condiciones del problema por tales razones se los descarta.



**Figura 5.6.** Peso mínimo vs número de iteraciones ejercicio de diez elementos.

Realizado por: Edison Hernán Garófalo Gaibor.

La estructura de secciones óptimas para la estructura de diez elementos se puede observar en la tabla 5.6. Obteniendo el menor resultado el método por torneo con un valor de 2456.48 kg.

**Tabla 5.6.**

Resultados obtenidos por el algoritmo genético de secciones para la estructura de diez elementos, utilizando el método de la ruleta y el método por torneo.

ELEMENTOS	AREA ( $cm^2$ )	
	M. RULETA	M. TORNEO
1	170.97	147.74
2	193.55	193.55
3	87.10	91.61
4	10.45	0.65
5	11.61	0.65
6	16.97	6.45
7	91.61	91.61
8	128.39	147.74
9	128.39	128.39
10	37.03	6.45
PESO (kg)	2623.33	2456.48

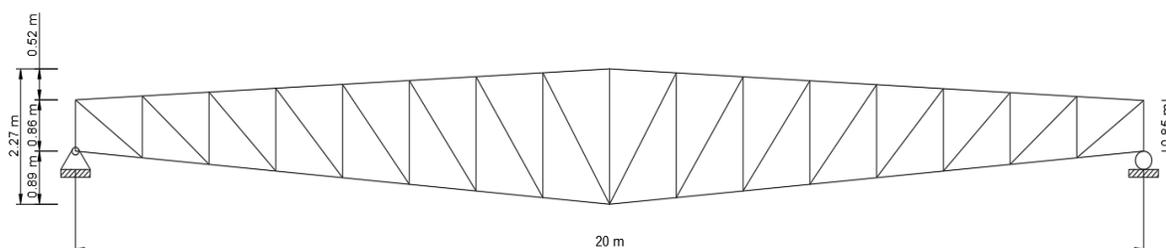
Los resultados obtenidos por el algoritmo de secciones realizado en el programa Matlab es menor que el resultado encontrado por Coello 1994, por lo tanto, es válido debido a que determina pesos mínimos a través de las generaciones, sin embargo, para encontrar el individuo más óptimo se debió realizar cuarenta iteraciones, para evitar caer en máximos locales.

### **5.3 PROBLEMAS OPTIMIZACIÓN DE SECCIONES, FORMA Y TOPOLOGÍA EN ARMADURAS PLANAS.**

Se procede a validar el algoritmo genético de secciones forma y topología con el diseño óptimo obtenido por Bonelli y Begliardo, 2016 para armaduras isostáticas y armaduras hiperestáticas de grado uno con una luz de 20 metros.

### 5.3.1. ARMADURA ISOSTÁTICA DE LUZ DE 20 METROS

Los resultados geométricos del diseño óptimo para armadura isostáticas de luz de 20 metros obtenidos por Bonelli y Begliardo, 2016, se puede observar en la figura 5.7.



**Figura 5.7.** Diseño óptimo de armadura isostática de luz de 20 metros.

Fuente: (Bonelli & Begliardo, 2016). Realizado por: Edison Hernán Garófalo Gaibor

En la tabla 5.7 se observa los parámetros de la estructura isostática óptima obtenidas por Bonelli y Begliardo 2016.

**Tabla 5.7.**

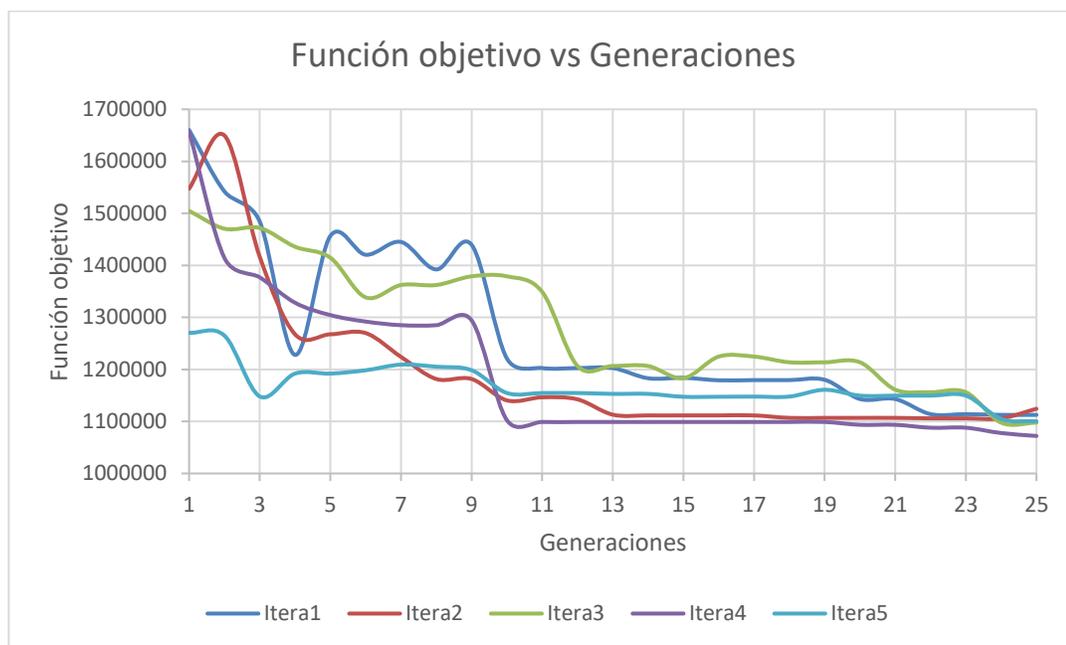
Parámetros de la estructura isostática optima de luz de 20 metros.

Variable de diseño	Pratt con SCH
Altura del canto izquierdo	0,98 m
Altura del Canto derecho	0,99 m
Dif. Altura sup. izq. – centro	0,75 m
Dif. Altura inf. der. - centro	-1,00 m
Altura del canto central	2,73 m
Cantidad de segmentos	18,00
Peso	230,13 kg

Fuente: (Bonelli & Begliardo, 2016)

Se colocó las variables de diseño en el programa para optimización de forma y topología elaborado en el programa Matlab el cual se encuentra en el Anexo No8 A8.1.

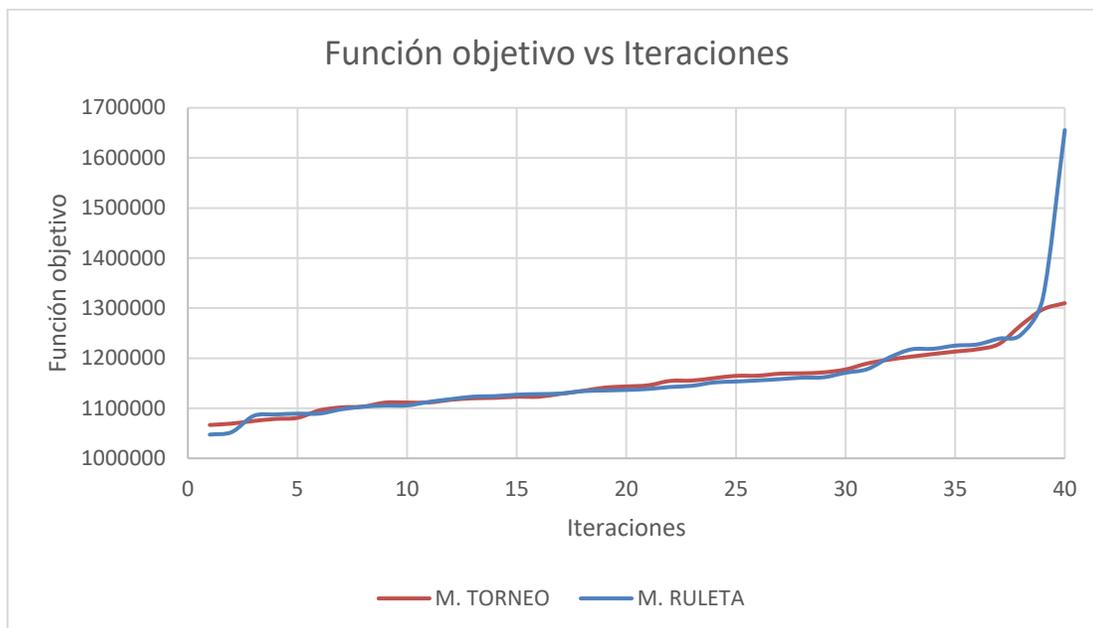
Los resultados obtenidos para cinco iteraciones se observan en la figura 5.8, la cual es el resultado de aplicar el algoritmo genético para veinticinco generaciones y diez individuos en cada generación.



**Figura 5.8.** Función objetivo versus generaciones para cinco iteraciones del diseño óptimo de forma y topología de armadura isostática de luz de 20 metros.

Realizado por: Edison Hernán Garófalo Gaibor

Los resultados mínimos de la función objetivo obtenidos por el algoritmo para forma y topología de cuarenta iteraciones, utilizando ambos métodos de selección elaborado en el programa Matlab se observan en la figura 5.9.



**Figura 5.9.** Función objetivo versus iteración del diseño óptimo de forma y topología de armadura isostática de luz de 20 metros.

Realizado por: Edison Hernán Garófalo Gaibor

El valor mínimo de la función objetivo resulta con el método de la ruleta, sin embargo, el método por torneo presenta menor coeficiente de desviación estándar y coeficiente de variación como se puede observar en la tabla 5.8.

**Tabla 5.8.**

Coeficientes estadísticos, para la optimización de forma y topología para la armadura isostática con luz de 20 metros.

COEFICIENTES ESTADÍSTICOS	M. RULETA	M. TORNEO
MÍNIMO	1052367	1066641
MÁXIMO	1655490	1309902
PROMEDIO	1160333	1153049
DESVIACIÓN ESTANDAR ( $\sigma$ )	97495	58557
COEFICIENTE DE VARIACIÓN (Cv)	8.4%	5.1%

Con el programa del Anexo No8 A8.1. se obtiene la forma y topología óptima dentro de los parámetros establecidos. La optimización de secciones se realiza con el programa del Anexo No7 A7.1, con los parámetros de forma y topología previamente obtenidos.

Los resultados de la optimización de forma y topología para la estructura isostática de luz de veinte metros, utilizando los dos métodos de selección se observa en la tabla 5.9. La optimización del peso de la estructura se obtiene con el algoritmo de optimización de secciones.

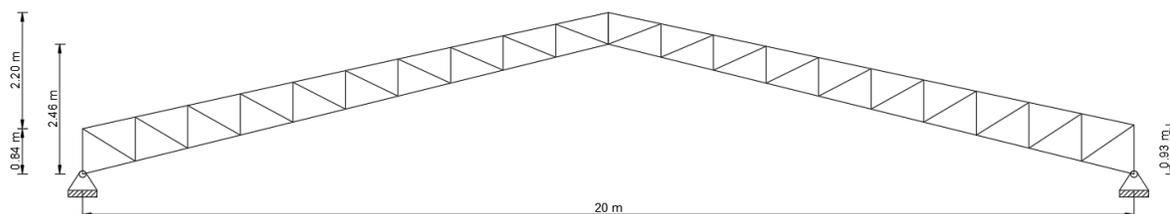
**Tabla 5.9.**

Resultados obtenidos en la optimización de forma y topología para la estructura isostática de luz de 20 metros.

Variable de diseño	M. POR TORNEO	M. RULETA
Altura del canto izquierdo	1,00 m	0,9525 m
Altura del Canto derecho	1,00 m	0,9525 m
Dif. Altura sup. izq. - centro	0,50 m	0,60 m
Dif. Altura inf. der. - centro	-1,00 m	-0,9475 m
Altura del canto central	2,50 m	2,50 m
Cantidad de segmentos	16,00	16,00
Peso	270,31 kg	286,84 kg

### 5.3.2. ARMADURA HIPERESTÁTICA DE GRADO UNO DE LUZ DE 20 METROS

Los resultados de forma y topología del diseño óptimo para armadura hiperestáticas de grado uno de luz de 20 metros obtenidos por Bonelli y Begliardo, 2016, se puede observar en la figura 5.10.



**Figura 5.10.** Diseño óptimo de armadura hiperestático de primer grado de luz de 20 metros.

Fuente: (Bonelli & Begliardo, 2016), Realizado por: Edison Hernán Garófalo Gaibor.

En la tabla 5.10 se observa los parámetros de la estructura hiperestática óptima de primer grado obtenidas por Bonelli y Begliardo, 2016.

**Tabla 5.10.**

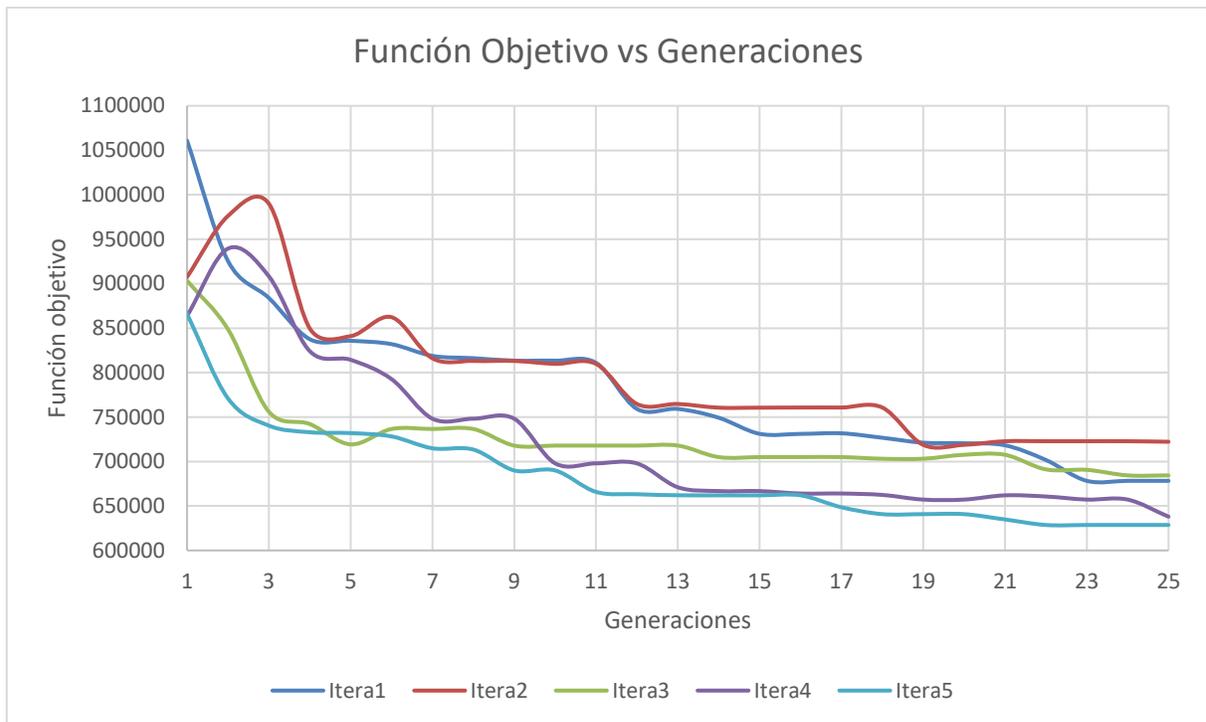
Parámetros de la estructura óptima hiperestática de primer grado de luz de 20 metros.

<b>Variable de diseño</b>	<b>Pratt con SCH</b>
Altura del canto izquierdo	0,85 m
Altura del Canto derecho	0,86 m
Dif. Altura sup. izq. – centro	2,68 m
Dif. Altura inf. der. - centro	2,67 m
Altura del canto central	0,86 m
Cantidad de segmentos	22,00
Peso	141,28 kg

Fuente: (Bonelli & Begliardo, 2016)

Se colocó las variables de diseño en el programa para optimización de forma y topología elaborado en el programa Matlab el cual es el mismo que para la estructura isostática el cual se encuentra en el Anexo A8.1, pero se debe modificar las condiciones de apoyo y los grados de libertad de la estructura el mismo que se encuentra detallado en el Anexo No8 A8.2.

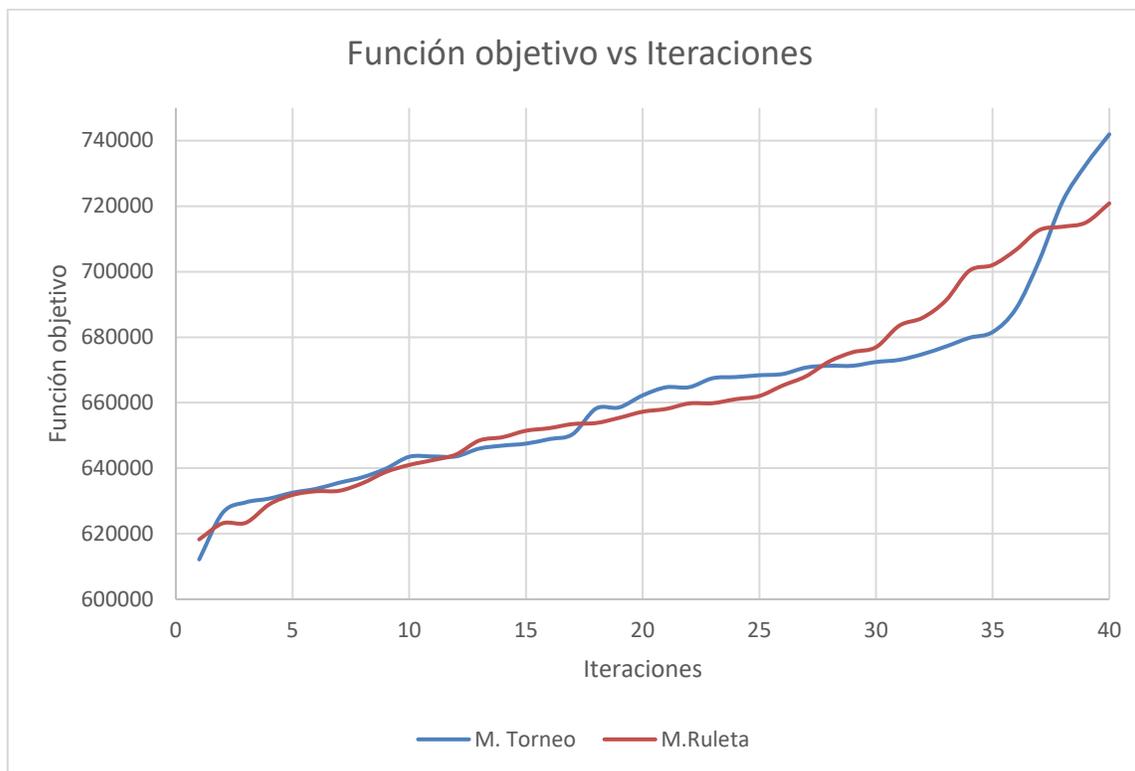
Los resultados obtenidos para cinco iteraciones se observan en la figura 5.11, la cual es el resultado de aplicar el algoritmo genético para veinticinco generaciones y diez individuos en cada generación.



**Figura 5.11.** Función objetivo versus generaciones para cinco iteraciones para el diseño óptimo de forma y topología de armadura hiperestática de luz de 20 metros.

Realizado por: Edison Hernán Garófalo Gaibor

Los resultados mínimos de la función objetivo obtenidos por el algoritmo para forma y topología de cuarenta iteraciones, utilizando ambos métodos de selección elaborado en el programa Matlab se observan en la figura 5.12



**Figura 5.12.** Función objetivo versus iteración del diseño óptimo de forma y topología de armadura hiperestática de luz de 20 metros.

Realizado por: Edison Hernán Garófalo Gaibor

La figura 5.12 se observa que el método por torneo presenta la menor función objetivo, sin embargo, para tener una mejor comparación se presenta los coeficientes estadísticos en la tabla 5.11.

**Tabla 5.11.**

Coeficientes estadísticos, para la optimización de forma y topología para la armadura hiperestática con luz de 20 metros.

COEFICIENTES ESTADÍSTICOS	M. TORNEO	M. RULETA
MÍNIMO	612157	618266
MÁXIMO	741928	720849
PROMEDIO	662235	662652
DESVIACIÓN ESTANDAR	27919	28088
COEFICIENTE DE VARIACIÓN (Cv)	4.22%	4.24%

Los coeficientes estadísticos de la tabla 5.11 se pueden observar que el método por torneo presenta coeficientes menores que el método de la ruleta.

Con el programa del Anexo No8 A8.1. se obtiene la forma y topología óptima dentro de los parámetros establecidos. La optimización de secciones se realiza con el programa del Anexo No7 A7.1, con los parámetros de forma y topología previamente obtenidos.

Los resultados de la optimización de forma y topología para la estructura isostática de luz de veinte metros, utilizando los dos métodos de selección se observa en la tabla 5.12. La optimización del peso de la estructura se obtiene con el algoritmo de optimización de secciones.

**Tabla 5.12.**

Resultados obtenidos en la optimización de forma y topología para la estructura hiperestática de primer grado con luz de 20 metros.

<b>Variable de diseño</b>	<b>M. POR TORNEO</b>	<b>M. RULETA</b>
Altura del canto izquierdo	1,00 m	0,9715 m
Altura del Canto derecho	1,00 m	0,9715 m
Dif. Altura sup. izq. - centro	2,975 m	2,95 m
Dif. Altura inf. der. - centro	3.056 m	3,0415 m
Altura del canto central	0,919 m	0,88 m
Cantidad de segmentos	16,00	16,00
Peso	137.85 kg	141.15 kg

Los resultados obtenidos reflejan resultados similares a los obtenidos por Bonelli y Begliardo en el 2016 por lo que el algoritmo genético de forma, topología y secciones entrega resultados confiables, sin embargo, el tiempo de ejecución para obtener los resultados es de 20 minutos, debido a que se realiza iteraciones para incrementar el grado de confianza.

## **CAPÍTULO 6.**

### **CONCLUSIONES Y RECOMENDACIONES.**

#### **6.1 CONCLUSIONES.**

El algoritmo en el programa Matlab utilizando el método matricial, es efectivo debido a que calcula las reacciones y fuerzas internas de las estructuras reticulares con un margen de error mínimo del 0.1%. Sin embargo, se debe tener claro el uso del algoritmo para ingresar los datos y parámetros para obtener resultados confiables.

La automatización del análisis matricial fue indispensable para poder resolver una gran cantidad de estructuras, y poder llegar a la optimización. Dependiendo este proceso únicamente de los parámetros de la geometría de la estructura, propiedades de los materiales, cargas cooperantes y tipos de apoyos.

El algoritmo genético de secciones arroja resultados mínimos a lo largo de las generaciones, pero esto depende del método de selección, dando mejores resultados al utilizar el método de selección por torneo.

La parametrización de la estructura ayudó a reducir tiempo, al ingresar los datos paramétricamente, así se evitó el ingreso manualmente de la geometría en el algoritmo de análisis matricial.

La parametrización de la estructura fue indispensable para realizar la optimización de forma y topología, debido a que los parámetros a modificar en el cromosoma fueron de acuerdo a lo parametrizado. Esto nos ayudó a restringir el tipo de estructura, así evitando estructuras inestables y reduciendo el rango de búsqueda. Sin embargo, esto restringe la búsqueda de estructuras con mejores formas que la óptima encontrada.

Es importante realizar iteraciones para la optimización, debido a que las poblaciones iniciales son aleatorias y realizar una corrida no garantiza encontrar el individuo más óptimo. De esta forma evitamos caer en máximos locales y encontramos el máximo global, de esta forma se logró encontrar el peso mínimo de la estructura.

En la optimización de secciones, forma y topología fue importante separar la optimización en dos partes; Primero en la optimización de forma y topología tomando una sección estándar para todos los elementos de la estructura, y finalmente realizar la optimización de secciones. Esto ayudó a disminuir el tiempo de optimización y también se evitó caer en máximos locales.

El método de selección por torneo y de la ruleta, en estructuras pequeñas de tres elementos presentó resultados iguales en la optimización. Si embargo el método por torneo convergió mucho más rápido que el método de selección de la ruleta.

El método de selección por torneo en estructuras mayores a tres elementos es más eficiente que el método de selección de la ruleta tanto en optimización de secciones como en optimización de forma y topología. Además, el método de selección por torneo converge siempre y mucho mejor que el método de selección de la ruleta como se pudo observar en las figura 5.2 y 5.5.

La optimización de secciones forma y topología permitió encontrar diseños más económicos en comparación con el diseño tradicional, debido a que permite encontrar estructuras más livianas, esto sucede porque el algoritmo permite explorar una gran variedad de estructuras las cuales van mejorando con el paso de las generaciones hasta encontrar la óptima.

## **6.2 RECOMENDACIONES.**

En el presente trabajo se realizó una contribución para entender los algoritmos genéticos a nivel de pregrado realizando algoritmos en Matlab y realizando una automatización del método matricial. Sin embargo, se debería realizar un entorno más

amigable para el ingreso de datos en el algoritmo para así evitar errores en el ingreso de datos y daños en el algoritmo.

## BIBLIOGRAFÍA

- Aguiar, R. (2014). *Análisis Matricial de Estructuras con CEINCI - LAB* (IV ed.). Quito.
- Blanco, J., Gonzáles, A., & Garcia, J. (2012). *Análisis estático de estructuras por el método matricial*. Malaga: PUBLIDISA.
- Bonelli, M., & Begliardo, F. (2016). Optimización de Armaduras Planas Mediante Algoritmos Genéticos. *Asociación Argentina de Mecánica Computacional*, 34, 501-515. Córdoba. Obtenido de [https://www.researchgate.net/profile/Hugo\\_Begliardo/publication/310731033\\_OPTIMIZACION\\_DE\\_ARMADURAS\\_PLANAS\\_MEDIANTE\\_DISENO\\_PARAMETRICO\\_Y\\_ALGORITMOS\\_GENETICOS\\_EFECTOS\\_DE\\_LA\\_NO\\_CORRESPONDENCIA\\_OBJETO\\_REAL\\_-\\_MODELO\\_IDEALIZADO/links/58a60a98aca27206d99d07d9/O](https://www.researchgate.net/profile/Hugo_Begliardo/publication/310731033_OPTIMIZACION_DE_ARMADURAS_PLANAS_MEDIANTE_DISENO_PARAMETRICO_Y_ALGORITMOS_GENETICOS_EFECTOS_DE_LA_NO_CORRESPONDENCIA_OBJETO_REAL_-_MODELO_IDEALIZADO/links/58a60a98aca27206d99d07d9/O)
- Cagnina, L. (04 de 2010). Optimización Mono y Multiobjetivo A Través de una Heurística de Inteligencia Colectiva. (*Tesis Doctoral*). Universidad Nacional de San Luis, San Luis, Argentina.
- Coello, C. A. (1994). *Uso de algoritmos genéticos para el diseño óptimo de armaduras*. Congreso Nacional de Informática: Herramientas Estratégicas para los Mercados Globales.
- García, J. (2014). Análisis e Implementación de Algoritmos Evolutivos para la Optimización de Simulaciones en Ingeniería Civil. (*Tesis de Postgrado*). Universidad Católica San Antonio UCAM, Murcia, España. Obtenido de [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2852539](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2852539)
- Gutierrez, N. (2014). Algoritmos Genéticos Aplicados al Diseño Estructural de Armaduras en Tres Dimensiones. (*Tesis Doctoral*). Universidad Nacional de Queretaro, Queretaro, Mexico.
- Hernández, A. (2018). Optimización de tipologías de armaduras planas usando técnicas heurísticas. (*Tesis Postgrado*). Universidad de Matanzas, Cuba.
- McCormac. (2013). *Diseño de Estructuras de Acero, Quinta Edición*. Mexico: Alfaomega Grupo Editor, S.A. de CV.

- Mutingi, M., & Mbohwa, C. (2017). *Grouping Genetic Algorithms Advances and Applications* (Vol. 666). Switzerland: Springer International.
- Muttio, E., Botello, S., & Tapia, M. (s.f.). *Parametrización Geométrica con Aplicación en el Diseño y Análisis Estructural de Edificaciones*. Obtenido de [https://www.researchgate.net/profile/Eugenio\\_Muttio\\_Zavala/publication/335118929\\_Parametrizacion\\_Geometrica\\_con\\_Aplicacion\\_en\\_el\\_Disenio\\_y\\_Analisis\\_Estructural\\_de\\_Edificaciones/links/5d516252299bf1995b785941/Parametrizacion-Geometrica-con-Aplicacion-en-el-](https://www.researchgate.net/profile/Eugenio_Muttio_Zavala/publication/335118929_Parametrizacion_Geometrica_con_Aplicacion_en_el_Disenio_y_Analisis_Estructural_de_Edificaciones/links/5d516252299bf1995b785941/Parametrizacion-Geometrica-con-Aplicacion-en-el-)
- Ramírez, S. (2014). Metodología Basada en Algoritmos Genéticos y Programación en Paralelo para el Diseño Óptimo de Armaduras de Acero. (*Tesis de Postgrado*). Pontificia Universidad Javeriana de Colombia, Bogota, Colombia.
- Rodríguez, G. (2010). Optimización de Celosías Bidimensionales Mediante algoritmos Genéticos. (*Tesis de Postgrado*). Universidad Nacional de Ingeniería, Lima, Perú.
- Sánchez, S. (2012). Optimización Estructural y Topológica de Estructuras Morfológicamente no Definidas Mediante Algoritmos Genéticos. (*Tesis Doctoral*). Universidad Politecnica de Valencia, Valencia, España.
- Sivanandam, S., & Deepa, S. (2008). *Introduction to Genetic Algorithms*. New York: Springer.
- Suárez, J. (2017). Optimización Estructural de Armaduras en 2d Mediante un Algoritmo Recocido Simulado Alterado. (*Tesis de Pregrado*). Universidad de Sucre, Sincelejo, Colombia.
- Weaver, J. (1967). *Analisis de Estructuras Reticulares*. Florida: Continental.

## **ANEXOS**

**ANEXO No1**  
**FUNCIONES DEL ANÁLISIS MATRICIAL SIN AUTOMATIZACIÓN.**

### A1.1. Función M\_KAP

```
function [ki]=M_KAP(Tabla1,Li,Es)
% Rutina para matriz de rigidez de armadura en coordenadas locales de
% elemento
%
% [ki]=M_KAP2AG(Tabla,Li,Es,ig,is)
%
% ki=rigidez de elemento armadura en coordenadas locales de
% elemento
% Tabla1= Propiedades del elemento
% ig= grados de libertad del elemento en cada nodo
% n= número de elementos
% Es=modulo del elasticidad del material [T/m2]
% Li=longitud de elemento [m]
% Ai=área de la sección transversal [m2]
    Ra=Tabla1*Es/Li;
    ki=[Ra 0 -Ra 0
        0 0 0 0
        -Ra 0 Ra 0
        0 0 0 0];

end
```

### A1.2. Función M\_TAP

```
function [T]=M_TAP(alfa)
% Rutina para matriz de paso de coordenadas locales a coordenadas globales
% de elemento tipo armadura
%
% [T]=M_TAP(alfa)
%
% T: Matriz de paso de elemento tipo armadura
% alfa: ángulo de inclinación en grados respecto al eje x positivo
%
% y
T=[cosd(alfa) -sind(alfa) 0 0
   sind(alfa) cosd(alfa) 0 0
   0 0 cosd(alfa) -sind(alfa)
   0 0 sind(alfa) cosd(alfa)];
end
```

### A1.3. Función Ens

```
function [S]=Ens(lee,K,S,ngdl)
%ENSAMBLA LA MATRIZ DE RIGIDEZ DE UN ELEMENTO EN LA MATRIZ DE
%RIGIDEZ DE LA ESTRUCTURA
% [S]=Ens(lee,K,S,ngdl)
% lee: Vector de ensamble
% K: Matriz de Rigidez del Elemento
% S: Matriz de Rigidez del Sistema
% ngdl: Numero de grados de libertad de la estructura

ng=length(lee);
for i=1:ng
    ii=lee(i);
    if ii>0
        if ii<=ngdl
            for j=1:ng
                jj=lee(j);
                if jj>0
                    if jj<=ngdl
                        S(ii,jj)=S(ii,jj)+K(i,j);
                    end
                end
            end
        end
    end
end
end
end
```

### A1.4. Programa validación del análisis matricial

```
%% Validación Análisis Matricial
% Escuela Politécnica Nacional
% Elaborado por: Edison Garófalo
tic
clc
clear all
close all
%% Datos de las propiedades del material
Es=1967840*10; %Modulo de elasticidad %T/m2
fy=1442*10; %Resistencia a la fluencia del acero %T/m2
%% Datos restricciones de los apoyos
% fix= restricciones en los apoyos [Fx Fy] 1 es restricción y 0 es libre
% ap: número de apoyos; % nglr: Numero de las restricciones de los apoyos
fix=[1 1
    1 1
    1 1];
```

```

ap=length(fix(:,1));  nglr=sum(sum(fix));
%% Coordenadas paramétricas de la armadura
[ CoordXY ] = [0      0  1.1547 -2
              1.1547  0  1.1547 -2
              2*(1.1547) 0  1.1547 -2];

%% Añade los nodos a la estructura
% ap: número de apoyos
N=[1 4; 2 4; 3 4;]; %nodos de los elementos
E=[1;2;3];
elementos=[E N]; % elementos=[E Ni Nf]

%% Enumera los nodos en la armadura.
[nodos] = [1 4; 2 4; 3 4;]; %nodos de los elementos
%Coordenadas de los nodos para graficar
nodosxy=[1 0 0; 2 1.1547 0;3 2.3094 0;4 1.1547, -2];
%% Ubica los grados de libertad totales y los cinemáticos
ig=2; %Grados de libertad por nodo
MaxN=max(max(N));  ngdl_rest=MaxN*ig; %Todas incluidas restricciones
ngl=ngdl_rest-nglr; %Indeterminación cinemática (Todas menos las reacciones)

%% Enumera los grados de libertad por cada nodo
% N_GDL=[ Nodo GDLi GDLf]
N_GDL=fgradoslibertad1(MaxN,fix,ig);
N_GDLMATR=N_GDL;
for i=1:MaxN
    N_GDLMATR(i,2:3)=[i, (i+1)];
    if i>=2
        N_GDLMATR(i,2:3)=[2*i-1, 2*i];
    end
end

%% Ensamble
% Este vector es necesario para ensamblar la matriz de rigidez global
% lee1=[GDLi GDLj]; lee1=[1 2 5 6];
% Coloca el vector de ensamblaje
n=length(CoordXY(:,1)); %número de elementos
for i=1:n
    leei(i,1:ig) = N_GDLMATR(N(i,1),2:ig+1);
    leei(i,ig+1:2*ig) = N_GDLMATR(N(i,2),2:ig+1);
end

%% Grafica la estructura
% fDibujarMat( CoordXY,nodosxy,E), drawnow
fDibujar( CoordXY,nodosxy,E ),drawnow
%% Secciones de los elementos

```

```

Tabla1=1.94e-4; %m2
Tabla2=1.94e-4; %m2
Tabla3=1.94e-4; %m2

%% Inicio del Cálculo Estructural
%% Calcula el vector de transformación de cada elemento (T)
for i=1:n
    Li(i,1)=sqrt((CoordXY(i,1)-CoordXY(i,3))^2+((CoordXY(i,2)-CoordXY(i,4))^2));
end

    Ti1=M_TAP(300); Ti2=M_TAP(270);  Ti3=M_TAP(240);
    Ti(1:4,1:4)= Ti1;  Ti(5:8,5:8)= Ti2;  Ti(9:12,9:12)=Ti3;

%% Matrices de rigidez por elemento
% % [ ki ] = M_KAP2(Tabla,L,ig)
[ k1 ] = M_KAP(Tabla1,Li(1),Es);
[ k2 ] = M_KAP(Tabla2,Li(2),Es);
[ k3 ] = M_KAP(Tabla3,Li(3),Es);
ki(1:4,1:4)= k1; ki(5:8,5:8)= k2; ki(9:12,9:12)=k3;

%% Rigidez en coordenadas globales de cada elemento;
% % Ki=Ti*ki*Ti';
Ki=zeros(n*4,n*4);
    Ki(1:4,1:4)= Ti1*k1*Ti1';
    Ki(5:8,5:8)= Ti2*k2*Ti2';
    Ki(9:12,9:12)=Ti3*k3*Ti3';
%% Matriz de rigidez Global del Sistema ST
%% Paso 6
% [ST]=Ens(lee1,K1,ST,ngdl_rest);
[ST]=zeros(ngdl_rest); nn=length(Ki);

    for i=1:n
        if i==1
            [ST]=Ens(leei(i,:),Ki(i:i+3,i:i+3),ST,ngdl_rest);
        else
            r=4*(i-1)+1;
            [ST]=Ens(leei(i,:),Ki(r:r+3,r:r+3),ST,ngdl_rest);
        end
    end

%% Paso 7 En armaduras no se tiene empotramientos
%% Cargas Externas
% Distribución de Fuerzas verticales en los nodos superiores
[ PP ] = zeros(6,1);
PP(7:8,1)=[.96;-9.6];

%% Paso 8

```

```

% % SAEP Acumula las acciones de empotramiento perfecto
% % SAEP=zeros(ngdl_rest,1);
% % [SAEP]=Ens_SAEP(lee2,AEP2,SAEP,ngdl_rest);
%% Paso 9
%% Solución del sistema de ecuaciones;
%Indeterminación cinemática (Todas menos las reacciones)
    format long
    Delta=zeros(ngdl_rest,1);
    Delta(nglr+1:ngdl_rest,1)=ST(nglr+1:ngdl_rest,nglr+1:ngdl_rest)^-
1*(PP(nglr+1:ngdl_rest,:));
%% Paso 10
%% Reacciones
% reac=ST(1:ngl,:)*Delta+SAEP(1:ngl);
    reac=ST(1:nglr,:)*Delta;

%% Paso 11
%% Desplazamientos de los elementos en coordenadas Globales
% D1=Delta(lee1);
    nn=length(Ki); [Di]=zeros(nn,1);
    for i=1:n
        if i==1
            Di(i:i+3,:)=Delta(leei(i,:));
        else
            r=4*(i-1)+1;
            Di(r:r+3,:)=Delta(leei(i,:));
        end
    end
%% Paso 12
%% Desplazamientos de los elementos en coordenadas Locales
% di=Ti'*Di;
    nn=length(Ki); [di]=zeros(nn,1);
    for i=1:n
        if i==1
            di(i:i+3,:)=Ti(i:i+3,i:i+3)*Di(i:i+3);
        else
            r=4*(i-1)+1;
            di(r:r+3,:)=Ti(r:r+3,r:r+3)*Di(r:r+3);
        end
    end
%% Paso 13
%% Acciones en los elementos(en coordenadas Locales y Globales)
% A2=K2*D2+AEP2; Globales, AEP2 en este caso no hay debido, que no hay acciones de
% empotramiento perfecto
% a1=k1*d1+aep1;      A1=K1*D1+AEP1

[Ai]=zeros(nn,1); [ai]=zeros(nn,1);

```

```

for i=1:n
    if i==1
        Ai(i:i+3,:)=Ki(i:i+3,i:i+3)*Di(i:i+3,:);
        ai(i:i+3,:)=ki(i:i+3,i:i+3)*di(i:i+3,:);
        ai_ord(i,1:2)=[i,ai(i,1)];
        di_ord(i,1:2)=[i,di(i+3,1)];
    else
        r=4*(i-1)+1;
        Ai(r:r+3,:)=Ki(r:r+3,r:r+3)*Di(r:r+3,:);
        ai(r:r+3,:)=ki(r:r+3,r:r+3)*di(r:r+3,:);
        ai_ord(i,1:2)=[i,ai(r,1)];
        di_ord(i,1:2)=[i,di(r+3,1)];
    end
end
end
%% Paso 14
%% Comprobación de equilibrio
% [PR]=Acum2(lee1,A1,PR);
PR=zeros(ngdl_rest,1);
for i=1:n
    if i==1
        [PR]=Acum2(leei(i,:),Ai(i:i+3,:),PR);
    else
        r=4*(i-1)+1;
        [PR]=Acum2(leei(i,:),Ai(r:r+3,:),PR);
    end
end
end
toc

```

### A1.5 Función Acum2

```

function [PR]=Acum2(lee,Ae,PR);
%Acumula acciones de empotramiento para ensamblar SAEP o acciones finales
%para comprobar equilibrio
%
%[PR]=Acum2(lee,Ae,PR)
%
    ng=length(lee);
    for i=1:ng
        ii=lee(i);
        PR(ii)=PR(ii)+Ae(i);
    end
end
end

```

**ANEXO No2**  
**ALGORITMO GÉNÉTICO DE SECCIONES**

### A2.1. Función reproducción

```

function [ Pob_C,Parents,Couples,Nr] = freproduction( Pob_2,Nr,nele,decr)

%Nr=Nr-2;%to dissapear 1 couple
LC=nele;% num
NPob=(1:size(Pob_2,1))';%Linear numbering of all elements of all individuals
RR=Nr*nele;
C=LC;%Chromosomes
decr=decr;%major is better
Parents=zeros(Nr,1);
cont=1;
while cont<=Nr
    [v,id]=max(dec);
    Parents(cont:cont+v-1)=id*ones(v,1);%how many individuals this's gonna couple with?
    dec(id)=0;
    cont=v+cont;
end

t1=randperm(Nr);
t2=reshape(t1,Nr/2,2);%Change the vector t1 to a [Nr/2,2] matrix

Couples=[Parents(t2(:,1)),Parents(t2(:,2))];

a1=Couples(:,1);
a2=Couples(:,2);
aux1=zeros(Nr/2,C);
aux2=aux1;aux3=aux1;aux4=aux1;
for t=1:Nr/2
    pcross=randi([1,LC-1]);
    r3=(a1(t)-1)*nele+1;
    aux1(t,1:pcross)=NPob(r3:r3+pcross-1,1)';
    aux2(t,pcross+1:LC)=NPob(r3+pcross:r3+LC-1,1)';
    r4=(a2(t)-1)*nele+1;
    aux3(t,1:pcross)=NPob(r4:r4+pcross-1,1)';
    aux4(t,pcross+1:LC)=NPob(r4+pcross:r4+LC-1,1)';
end
Pob_C10_0=[aux1+aux4;aux3+aux2];%Cross of chromosomes Matrix[Nr,nele]
Pob_C10t=Pob_C10_0';
Pob_C10=Pob_C10t(:);%Vector [Nr*nele,1]
%Pop binary
Pob_C=Pob_2(Pob_C10,:);%Binary matrix

```

## A2.2. Función de mutación

```
function Pob_2 = fMutation(Pob_2,MutationPorc)
%porc: mutation percentage (%)

%Genes quantity
[R, C] = size(Pob_2);
Tgenes = R*C;
%Genes quantity for mutation
%MutationPorc=randi([0,MutationPorc]);
GenesMut= round(Tgenes*MutationPorc/100);

for i=1:GenesMut
    row=randi([1 R]); col=randi([1 C]);
    %Pob_2(row,col)=randi([1,Nop]);
    Pob_2(row,col)=1 - Pob_2(row,col);%To change between 0 and 1
end
```

## A2.3. Método de selección por ruleta

```
function [ decr] = fSelectionRul(fxp,Nr )
%Selection by Roulette method
Ndecr=fxp/(sum(fxp)/Nr);
decr=floor(Ndecr);
resr=Ndecr-decr;
resr=360*resr/(sum(resr));
for i=2:Nr
    resr(i)=resr(i)+resr(i-1);%Accumulate
end
while sum(decr)<Nr
    roulette=360*rand;%there are big values and small in resr
    i=2;
    while i<Nr
        if roulette>=resr(i-1) && roulette<resr(i)%there is a small probability
            decr(i-1)=decr(i-1)+1;
            i=Nr;
        end
        i=i+1;
    end
end
end
```

## A2.4. Método de selección por torneo

```

function [dect] = fselectionTor(fxp,Nr,nt)
%UNTITLED2 Summary of this function goes here
% Detailed explanation goes here
% fxp = función objetivo
% Nr = Número de individuos, padres
% nt = Número de individuos de la competencia
%%

v=zeros(nt,2); %matriz para guardar la competencia
dect=zeros(Nr,1); %matriz con los descendientes

while sum(dect)<Nr % mientras no se asignen todos los descendientes
    i=1;
    while i<=nt % repetir ciclo para escoger a los padres que entran a torneo

        a=floor((Nr-1)*rand+1); % se escoge un padre aleatoriamente

        if a~= v(:,2) % para evitar que se repita al padre en el torneo
            v(i,:)=[fxp(a,1) a]; % se guarda en un vector los padres (función de adaptación) y
            posición %del padre
            i=i+1; % si no se repite el padre entonces que se pasa a escoger un nuevo padre para el
            %torneo
        end

    end

    [r j]=max(v(:,1)); % se extrae el mejor padre del torneo

    dect(v(j,2),1)=dect(v(j,2),1)+1; %sumar un descendiente al padre ganador del torneo

end
end

```

**ANEXO No3**  
**PARAMETRIZACIÓN**

### A3.1 Programa fparametPrant

```

function [ CoordXY ] = fparametPrant(La,hi,hd,nd,ap,deltah,deltac )
%% Coordenadas de los elementos
%Total de elementos
deltad=hi+deltah-deltac; %Limite Inferior (- ó +)

m=4*nd+1;
CoordXY=zeros(m,4); %tamaño de la matriz de coordenadas
Pid=[0,0]; %Coordenada inicial
Pih=[0,hi]; %Coordenada inicial Superior
Pdd=[La,0];
Pdh=[La,hd];
CoordXY(1:ap,:)= [0 0 0 hi %1er apoyo
                  La 0 La hd]; %2do apoyo
%% Coordenada del límite superior
tangteta=2*deltah/La; %ángulo superior
% Lado Izquierdo
XYsi=zeros(nd/2+1,2);
for i=1:nd/2+1
    if i==1
        XYsi(i,:)=Pih;
    else
        XYsi(i,1)=(i-1)*La/(nd);
        XYsi(i,2)=(i-1)*La*tangteta/(nd)+hi;
    end
end

% Lado derecho
XYsd=zeros(nd/2,2);
for i=1:nd/2
    if i<nd/2
        XYsd(i,1)=(i)*La/(nd)+La/2;
        XYsd(i,2)=XYsi(nd/2+1,2)-(i)*La*tangteta/(nd);
    else
        XYsd(i,:)=Pdh;
    end
end
XYsid=[XYsi;XYsd];
%% Coordenadas del límite Inferior
tangalfa=(2*(deltad))/La;
XYii=zeros(nd/2+1,2);
for i=1:nd/2+1
    if i==1
        XYii(i,:)=Pid;
    else

```

```

    XYii(i,1)=(i-1)*La/(nd);
    XYii(i,2)=(i-1)*La*tangalfa/(nd);
    end
end

% Lado derecho
XYid=zeros(nd/2,2);
for i=1:nd/2
    XYid(i,1)=(i)*La/(nd)+La/2;
    XYid(i,2)=XYii(nd/2+1,2)-(i)*La*tangalfa/(nd);
end
XYiid=[XYii;XYid];
%% Coordenadas de los elementos superiores
for i=1:nd

    CoordXY(ap+i,:)= [XYsid(i,:) XYsid(i+1,:)];
end
%% Coordenada de los elementos Inferiores
t=ap+nd;
for i=1:nd
    CoordXY(t+i,:)= [XYiid(i,:) XYiid(i+1,:)];
end
%% Coordenadas de las verticales
tt=t+nd;
for i=1:nd-1
    CoordXY(tt+i,:)= [XYiid(i+1,:) XYsid(i+1,:)];
end
%% Coordenada de las diagonales
r=tt+nd-1;
for i=1:nd
    if i<=nd/2
        CoordXY(r+i,:)= [XYsid(i,:) XYiid(i+1,:)];
    else
        CoordXY(r+i,:)= [XYsid(i+1,:) XYiid(i,:)];
    end
end
end
end

```

### A3.2 Programa fparametricaCircular

```

function [ CoordXY ] = fparametricoCircular(La,hi,hd,nd,ap,deltah,deltac)
% Crea las coordenadas de la armadura en forma circular
% La= Luz de la armadura entre los dos apoyos
% hi= Altura desde el apoyo izquierdo hasta la parte superior en el lado izquierdo;
% hd= Altura desde el apoyo derecho hasta la parte superior en el lado derecho;

```

```

% ndd= número de espacios de la armadura en la parte izquierda=derecha
% ap= Numero de restricciones de apoyos en este caso ap=2;
% deltah=Longitud desde extremo izquierdo hacia la parte superior de la
% armadura 2D lo cual se debe dar un rango.
% deltac= Longitud de la parte central el cual se debe dar un rango

%% Coordenadas de los elementos forma Circular
%Total de elementos

% nd=2*ndd;%Número de diagonales en los dos lados
hcs=hi+deltah; %Altura Circular superior hci<hcs
hci=hi+deltah-deltac; %Altura Circular inferior

m=4*nd+1;
CoordXY=zeros(m,4); %tamaño de la matriz de coordenadas
Pid=[0,0]; %Coordenada inicial
Pih=[0,hi]; %Coordenada inicial Superior
Pdd=[La,0];
Pdh=[La,hd];
CoordXY(1:ap,:)= [0 0 0 hi %1er apoyo
                  La 0 La hd]; %2do apoyo
%Constante de la ecuación de la parábola  $A*x^2+B*x+C=0$ 
As=-4*(hcs-hi)/La^2;
AAi=-4*hci/La^2;
Bs=4*(hcs-hi)/La;
Bi=4*hci/La;
Cs=hi;
Ci=0;
%% Coordenada del límite superior
% Lado Izquierdo
XYs=zeros(nd+1,2);
for i=1:nd+1
    if i==1
        XYs(i,:)=Pih;
    elseif i<=nd
        XYs(i,1)=(i-1)*La/nd;
        XYs(i,2)=As*((i-1)*La/nd)^2+Bs*((i-1)*La/nd)+Cs;
    else
        XYs(i,:)=Pdh;
    end
end

%% Coordenadas del límite Inferior
XYi=zeros(nd+1,2);

```

```

for i=1:nd+1
    if i==1
        XYi(i,:)=Pid;
    elseif i<=nd
        XYi(i,1)=(i-1)*La/nd;
        XYi(i,2)=AAi*((i-1)*La/nd)^2+Bi*((i-1)*La/nd)+Ci;
    else
        XYi(i,:)=[La 0];
    end
end

%% Coordenadas de los elementos superiores
for i=1:nd
    CoordXY(ap+i,:)=[XYs(i,) XYs(i+1,)];
end
%% Coordenada de los elementos Inferiores
t=ap+nd;
for i=1:nd
    CoordXY(t+i,:)=[XYi(i,) XYi(i+1,)];
end
%% Coordenadas de las verticales
tt=t+nd;
for i=1:nd-1
    CoordXY(tt+i,:)=[XYi(i+1,) XYs(i+1,)];
end
%% Coordenada de las diagonales
r=tt+nd-1;
for i=1:nd
    if i<=nd/2
        CoordXY(r+i,:)=[XYs(i,) XYi(i+1,)];
    else
        CoordXY(r+i,:)=[XYi(i,) XYs(i+1,)];
    end
end
end

end

```

**ANEXO No4**  
**FUNCIONES DEL ANÁLISIS MATRICIAL CON AUTOMATIZACIÓN.**

### A4.1. Función fNodos1

```
function [N,E] = fNodos1( CoordXY,ap)

% Crea los nodos de una estructura
% Esta función crea primero los nodos iniciales de cada %elemento de una
% manera consecutiva
%CoordXY: es las coordenadas de la estructura [xi yi xj yj]
%%
n=length(CoordXY(:,1));
N=zeros(n,2);
E=zeros(n,1);
%Nodos iniciales
for i=1:n
    E(i,1)=i;
    if i<=ap
        N(i,1)=i;
    elseif i>ap
        m=i-1;
        j=1;
        while j<=m
            a=CoordXY(i,1)-CoordXY(j,1);
            b=CoordXY(i,2)-CoordXY(j,2);

            if a==0 && b==0
                N(i,1)=N(j,1);
                j=m;
            else
                N(i,1)=max(N(1:i-1,1))+1;
            end
            j=j+1;
        end
    end
end

%% %Nodos Finales
for i=1:n
    if i>=1
        m=n;
        mm=i-1;
        t=m+mm;
        j=1;
        while j<=t
            if j<=n
                a=CoordXY(j,1)-CoordXY(i,3);
                b=CoordXY(j,2)-CoordXY(i,4);
```

```

        if a==0 && b==0
            N(i,2)=N(j,1);
            j=t;
        else
            N(i,2)=max(max(N(1:j,:)))+1;
        end
    else

        ii=abs(j-n);
        aa=CoordXY(i,3)-CoordXY(ii,3);
        bb=CoordXY(i,4)-CoordXY(ii,4);

        if aa==0 && bb==0
            N(i,2)=N(ii,2);
            j=t;
        else
            N(i,2)=max([max(max(N(:,1))),max(max(N(1:ii,2)))]+1;
        end
    end

    j=j+1;
end

end
end
end

```

#### A4.2. Función fCoordNodos

```
function [nodos] = fCoordNodos( CoordXY,N,ap)
```

```

% Crea Las coordenadas de los Nodos de la estructura
% CoordXY= Coordenadas de los elementos [xi yi xj yj]
% ap= es el número de apoyos
% CoordN=[Nodoi xi yi]
% nodos =[elementoi xi yi]
n=length(CoordXY(:,1));
CoordN=zeros(2*n,3);
%% Nodos iniciales
for i=1:n
    if i<=ap
        CoordN(i,1)=i;
        CoordN(i,2)=CoordXY(i,1);
        CoordN(i,3)=CoordXY(i,2);
    end
end

```

```

elseif i>ap
    m=i-1;
    j=1;
    while j<=m
        a=CoordXY(i,1)-CoordXY(j,1);
        b=CoordXY(i,2)-CoordXY(j,2);

        if a==0 && b==0
            CoordN(i,1)=CoordN(j,1);
            CoordN(i,2)=CoordXY(i,1);
            CoordN(i,3)=CoordXY(i,2);
            j=m;
        else
            CoordN(i,1)=max(CoordN(1:i-1,1))+1;
            CoordN(i,2)=CoordXY(i,1);
            CoordN(i,3)=CoordXY(i,2);
        end
        j=j+1;
    end
end
end

%% Nodos Finales
for i=n+1:2*n
    if i>=1
        m=n;
        mm=i-1;
        t=m+mm;
        j=1;

        while j<=t
            if j<=n
                a=CoordXY(j,1)-CoordXY((i-n),3);
                b=CoordXY(j,2)-CoordXY((i-n),4);
                if a==0 && b==0
                    CoordN(i,1)=CoordN(j,1);
                    CoordN(i,2)=CoordXY(j,1);
                    CoordN(i,3)=CoordXY(j,2);
                    j=t;
                end
            else

                ii=(j-n);
                aa=CoordXY((i-n),3)-CoordXY(ii,3);
                bb=CoordXY((i-n),4)-CoordXY(ii,4);
            end
        end
    end
end

```

```

if aa==0 && bb==0
    CoordN(i,1)=CoordN(j,1);
    if CoordN(j,1)<=max(N(:,1))
        CoordN(i,2)=CoordXY(ii,1);
        CoordN(i,3)=CoordXY(ii,2);
        j=t;
    else
        CoordN(i,2)=CoordXY(ii,3);
        CoordN(i,3)=CoordXY(ii,4);
        j=t;
    end
else
    CoordN(i,1)=max([max(max(N(:,1))),max(max(N(1:ii,2)))]+1;
    CoordN(i,2)=CoordXY(ii,3);
    CoordN(i,3)=CoordXY(ii,4);
end

end

    j=j+1;
end

end

end

%% Coordenada de los Nodos
% nodos =[elementoi xi yi]
nn=length(CoordN);
n=max(max(N));
nodos=zeros(n,3);
for i=1:n
    if i==1
        nodos(1,1)=CoordN(1,1);
        nodos(1,2:3)=CoordN(1,2:3);
    else
        for j=1:nn
            a=i-CoordN(j,1);
            if a==0
                nodos(i,1)=i;
                nodos(i,2:3)=CoordN(j,2:3);
            end
        end
    end
end
end
end

end

```

### A4.3. Función fgradoslibertadAG

```

function [ N_GDL ] = fgradoslibertadAG( MaxN,fix,ig)
% Enumera los grados de libertad
% N_GDL=[ Nodo GDLi GDLf]
%MaxN= número de nodos totales
%ig= grados de libertad por nodo depende del tipo de estructura
% ig=2 Armaduras 2D
% ig=3 Marcos planos 2D
% ig=3 Armaduras 3D
% ig=6 Pórtico 3D
[A,B]=size(fix);
n=length(fix(:,1));
m=length(fix(1,:));
GDL=zeros(MaxN,2);
N_GDL=zeros(MaxN,3);
GDL=ones(MaxN,ig);
GDL(1:A,1:B)=zeros(A,B);

for i=1:n
    N_GDL(i,1)=i;
    for j=1:m
        a=fix(i,j)-1;
        if a==0
            GDL(i,j)=0;
        else
            GDL(i,j)=max(max(GDL(1:A,1:B)))+1;
        end
    end
end
end

for i=length(fix(:,1))+1:MaxN
    N_GDL(i,1)=i;
    for j=1:m
        a=GDL(i,j);
        if a==0
            GDL(i,j)=0;
        else
            GDL(i,j)=max(max(GDL))+1;
        end
    end
end
end
N_GDL(:,2:3)=GDL;
end

```

#### A4.4. Función fDibujar

```

function [ figure ] = fDibujar( CoordXY,nodos,E )
% Gráfica la estructura
% CoordXY= Coordenadas de los elementos [xi yi xj yj]
% N= Nodos inicial y final de cada elemento
% E= Elemento
% nodos= son las coordenadas de los nodos [1 3 5]

% Dibuja las líneas
n=length(CoordXY(:,1));
for i=1:n
    line([CoordXY(i,1) CoordXY(i,3)],[CoordXY(i,2) CoordXY(i,4)]);
    hold on
end
Xmin=-0.2;
Xmax=1.1*max(max([(CoordXY(:,1)),(CoordXY(:,3))]));
Ymin=-0.2+min(min([CoordXY(:,2),CoordXY(:,4)]));
Ymax=1.1*max(max([(CoordXY(:,2)),(CoordXY(:,4))]));
axis ([Xmin Xmax Ymin Ymax]);

%%
% % Dibuja los círculos de los nodos de la estructura
m=length(nodos(:,1));
for i=1:m
    plot(nodos(i,2),nodos(i,3),'ob');
end

%%
%%Coordenadas para ubicar cada elemento
CordM=zeros(length(CoordXY),2);
for i=1:n
    CordM(i,1)=(CoordXY(i,1)+CoordXY(i,3))/2;
    CordM(i,2)=(CoordXY(i,2)+CoordXY(i,4))/2;
end

% Evita la superposicion de elementos
for i=1:n
    m=i-1;
    j=1;
    while j<=m
        a=CordM(i,1)-CordM(j,1);
        b=CordM(i,2)-CordM(j,2);
        if a==0 && b==0
            CordM(j,1)=CordM(i,1)+0.15;
            CordM(j,2)=CordM(i,2)+0.15;
        end
        j=j+1;
    end
end

```

```

                CordM(i,1)=CordM(i,1)+0.15;
                CordM(i,2)=CordM(i,2)-0.15;
            end
            j=j+1;
        end
    end
end
%%
%Ubica los Nodos en el grafico

m=length(nodos);
for i=1:m
    text(nodos(i,2),nodos(i,3),sprintf('_N %d',nodos(i,1)),'color',[1 0 0]);
end
% % Coloca los elementos
for i=1:n
    text(CordM(i,1),CordM(i,2),sprintf('_E %d',E(i,1)),'color',[0 1 0]);
end

end

```

#### A4.5. Función M\_TAP2

```

function [Ti,Li,alfa]=M_TAP2(CoordXY,ig,n)
% Rutina para matriz de paso de coordenadas locales a coordenadas globales
% de elemento tipo armadura
% [Ti]=M_TAP2(CoordXY,ig)
% Ti: Matriz de paso de todos los elementos tipo armadura
% CoordXY: Coordenadas de los elementos
% ig: grados de libertad cinemático por nudo
alfa=zeros(n,1); Li=zeros(n,1);
for i=1:n
    Li(i,1)=sqrt((CoordXY(i,1)-CoordXY(i,3))^2+(CoordXY(i,2)-CoordXY(i,4))^2);
    xi=(CoordXY(i,3)-CoordXY(i,1));    yi=(CoordXY(i,4)-CoordXY(i,2));
    xi=CoordXY(i,1); xj=CoordXY(i,3); yi=CoordXY(i,2); yj=CoordXY(i,4);
    if xi<=xj && yi<yj
        alfai(i,1)=atand((CoordXY(i,4)-CoordXY(i,2))/(CoordXY(i,3)-CoordXY(i,1)))+0;
    elseif xi>xj && yi<=yj
        alfai(i,1)=atand((CoordXY(i,4)-CoordXY(i,2))/(CoordXY(i,3)-CoordXY(i,1)))+180;
    elseif xi>xj && yi>yj
        alfai(i,1)=atand((CoordXY(i,4)-CoordXY(i,2))/(CoordXY(i,3)-CoordXY(i,1)))+180;
    else
        alfai(i,1)=atand((CoordXY(i,4)-CoordXY(i,2))/(CoordXY(i,3)-CoordXY(i,1)))+360;
    end
    alfa=alfai(i);
end

```

```

Ti((2*ig*(i-1)+1):(2*ig*(i)),(2*ig*(i-1)+1):(2*ig*(i)))=[cosd(alfa) -sind(alfa) 0 0
sind(alfa) cosd(alfa) 0 0
0 0 cosd(alfa) -sind(alfa)
0 0 sind(alfa) cosd(alfa)];

end
end

```

#### A4.6. Función M\_KAP2AG

```

function [ki]=M_KAP2AG(Tabla1,Li,Es,ig,n)
% Rutina para matriz de rigidez de armadura en coordenadas locales de
% elemento
% [ki]=M_KAP2AG(Tabla,Li,Es,ig,is)
% ki=rigidez de elemento armadura en coordenadas locales de
% elemento
% Tabla1= Propiedades del elemento
% ig= grados de libertad del elemento en cada nodo
% n= número de elementos
% Es=modulo del elasticidad del material [T/m2]
% Li=longitud de elemento [m]
% Ai=area de la sección transversal [m2]
for i=1:n
Ra=Tabla1(i,1)*Es/Li(i);
ki((2*ig*(i-1)+1):(2*ig*(i)),(2*ig*(i-1)+1):(2*ig*(i)))=[Ra 0 -Ra 0
0 0 0 0
-Ra 0 Ra 0
0 0 0 0];

end
end

```

**ANEXO No5**  
**VALIDACIÓN DEL ANÁLISIS MATRICIAL CON AUTOMATIZACIÓN Y CON**  
**PARAMETRIZACIÓN.**

### A5.1. Programa validación del análisis matricial con automatización.

```

%% Validación Análisis Matricial Armadura Circular.
% Escuela Politécnica Nacional
% Elaborado por: Edison Garófalo
tic
clc
clear all
close all
%% Datos de las propiedades del material
Es=2030000*10; %Modulo de elasticidad %T/m2
fy=2560*10; %Resistencia a la fluencia del acero %T/m2
%% Datos restricciones de los apoyos
% fix= restricciones en los apoyos [Fx Fy] 1 es restricción y 0 es libre
% ap: número de apoyos; % nglr: Numero de las restricciones de los apoyos
fix=[1 1
     0 1];
ap=length(fix(:,1)); nglr=sum(sum(fix));
%% Coordenadas paramétricas de la armadura
[ CoordXY ] = [0 0 0 0.5000
10.0000 0 10.0000 0.5000
0 0.5000 1.2500 0.7188
1.2500 0.7188 2.5000 0.8750
2.5000 0.8750 3.7500 0.9688
3.7500 0.9688 5.0000 1.0000
5.0000 1.0000 6.2500 0.9688
6.2500 0.9688 7.5000 0.8750
7.5000 0.8750 8.7500 0.7188
8.7500 0.7188 10.0000 0.5000
0 0 1.2500 0.2188
1.2500 0.2188 2.5000 0.3750
2.5000 0.3750 3.7500 0.4688
3.7500 0.4688 5.0000 0.5000
5.0000 0.5000 6.2500 0.4688
6.2500 0.4688 7.5000 0.3750
7.5000 0.3750 8.7500 0.2188
8.7500 0.2188 10.0000 0
1.2500 0.2188 1.2500 0.7188
2.5000 0.3750 2.5000 0.8750
3.7500 0.4688 3.7500 0.9688
5.0000 0.5000 5.0000 1.0000
6.2500 0.4688 6.2500 0.9688
7.5000 0.3750 7.5000 0.8750
8.7500 0.2188 8.7500 0.7188
0 0.5000 1.2500 0.2188
1.2500 0.7188 2.5000 0.3750

```

```

2.5000  0.8750  3.7500  0.4688
3.7500  0.9688  5.0000  0.5000
5.0000  0.5000  6.2500  0.9688
6.2500  0.4688  7.5000  0.8750
7.5000  0.3750  8.7500  0.7188
8.7500  0.2188  10.0000  0.5000];
%% Añade los nodos a la estructura
% ap: número de apoyos
[N,E]=fNodos1(CoordXY,ap);
% % elementos=[E Ni Nf]
elementos=[E N];
%% Enumera los nodos en la armadura.
%Ubica los nodos de los elementos
[nodos] = fNodos1( CoordXY,ap);
%Coloca las coordenadas de los nodos para graficar
[nodosxy] = fCoordNodos( CoordXY,N,ap);
%% Ubica los grados de libertad totales y los cinemáticos
ig=2; %Grados de libertad por nodo
MaxN=max(max(N));  ngdl_rest=MaxN*ig; %Todas incluidas restricciones
ngl=ngdl_rest-nglr; %Indeterminación cinemática (Todas menos las reacciones)

%% Enumera los grados de libertad por cada nodo
% N_GDL=[ Nodo GDLi GDLf]
% N_GDL=fgradoslibertad1(MaxN,fix,ig);
% N_GDLMATR=N_GDL;
nx=length(fix);ny=length(fix);
for ix=1:nx
    for iy=1:ny
        if fix(ix,iy)>0
            N_GDLMATR(ix,iy)=ix+iy-1;
            i=i+1;
        else
            N_GDLMATR(ix,iy)=0;
        end
    end
end
end
N_GDLMATR=[ones(length(fix()),1) N_GDLMATR];
N_GDLMATR(2,2)=4;
for i=length(fix)+1:MaxN
    N_GDLMATR(i,1)=i;
    if i>length(fix)
        N_GDLMATR(i,2:3)=[2*i-1, 2*i];
    else
        N_GDLMATR(i,2:3)=[2*i-1, 2*i];
    end
end

```

```

end
end

%% Ensamble
% Este vector es necesario para ensamblar la matriz de rigidez global
% lee1=[GDLi GDLj]; lee1=[1 2 5 6];
% Coloca el vector de ensamblaje
n=length(CoordXY(:,1)); % número de elementos
for i=1:n
leei(i,1:ig) = N_GDLMATR(N(i,1),2:ig+1);
leei(i,ig+1:2*ig) = N_GDLMATR(N(i,2),2:ig+1);
end

%% Grafica la estructura
% fDibujarMat( CoordXY,nodosxy,E), drawnow
fDibujar( CoordXY,nodosxy,E ),drawnow
%% Secciones de los elementos
    for i=1:n
        Tabla1(i,:)=3.06/10000; %Area en m2
    end

%% Inicio del Cálculo Estructural
%% Calcula el vector de transformación de cada elemento (T)
    [Ti,Li]=M_TAP2(CoordXY,ig,n);
%% Matrices de rigidez por elemento
% % [ ki ] = M_KAP2(Tabla,L,ig)
    [ ki ] = M_KAP2AG(Tabla1,Li,Es,ig,n);

%% Rigidez en coordenadas globales de cada elemento;
% % Ki=Ti*ki*Ti';
    [ Ki ] = Ti*ki*Ti';
%% Matriz de rigidez Global del Sistema ST
%% Paso 6
% [ST]=Ens(lee1,K1,ST,ngdl_rest);
    [ST]=zeros(ngdl_rest); nn=length(Ki);

    for i=1:n
        if i==1
            [ST]=Ens(leei(i,:),Ki(i:i+3,i:i+3),ST,ngdl_rest);
        else
            r=4*(i-1)+1;
            [ST]=Ens(leei(i,:),Ki(r:r+3,r:r+3),ST,ngdl_rest);
        end
    end

%% Paso 7 En armaduras no se tiene empotramientos
%% Cargas Externas
% Distribución de Fuerzas verticales en los nodos superiores

```

```

    [ PP] = zeros(ngdl_rest,1);
%   PP(7:8,1)=[0;-1];
    Pvert=-1;
    nd=8; %Número de diagonales
for i=1:nd+1
    PP(nodos(i,2)*2,1)=Pvert;
end
%% Paso 8
% % SAEP Acumula las acciones de empotramiento perfecto
% % SAEP=zeros(ngdl_rest,1);
% % [SAEP]=Ens_SAEP(lee2,AEP2,SAEP,ngdl_rest);
%% Paso 9
%% Solución del sistema de ecuaciones;
%Indeterminación cinemática (Todas menos las reacciones)
    format long
    Delta=zeros(ngdl_rest,1);

Delta(nglr+1:ngdl_rest,1)=ST(nglr+1:ngdl_rest,nglr+1:ngdl_rest)^1*(PP(nglr+1:ngdl_rest,:));
    ST1=ST(nglr:ngdl_rest,nglr:ngdl_rest);
    ST1(3,:)=[];ST1(:,3)=[];
%   Delta1=ST1^-1*(PP(nglr+1:ngdl_rest,:));
%% Paso 10
%% Reacciones
% reac=ST(1:ngl,,:)*Delta+SAEP(1:ngl);
    reac=ST(1:nglr,,:)*Delta;

%% Paso 11
%% Desplazamientos de los elementos en coordenadas Globales
% D1=Delta(lee1);
    nn=length(Ki); [Di]=zeros(nn,1);
    for i=1:n

        if i==1
            Di(i:i+3,:)=Delta(leei(i,:));
        else
            r=4*(i-1)+1;
            Di(r:r+3,:)=Delta(leei(i,:));
        end
    end

%% Paso 12
%% Desplazamientos de los elementos en coordenadas Locales
% di=Ti'*Di;
    nn=length(Ki); [di]=zeros(nn,1);
    for i=1:n
        if i==1

```

```

    di(i:i+3,:)=Ti(i:i+3,i:i+3)*Di(i:i+3);
    else
    r=4*(i-1)+1;
    di(r:r+3,:)=Ti(r:r+3,r:r+3)*Di(r:r+3);
    end
end
%% Paso 13
%% Acciones en los elementos(en coordenadas Locales y Globales)
% A2=K2*D2+AEP2; Globales, AEP2 en este caso no hay debido, que no hay acciones de
% empotramiento perfecto
% a1=k1*d1+aep1;    A1=K1*D1+AEP1
[Ai]=zeros(nn,1); [ai]=zeros(nn,1);

for i=1:n
    if i==1
        Ai(i:i+3,:)=Ki(i:i+3,i:i+3)*Di(i:i+3,:);
        ai(i:i+3,:)=ki(i:i+3,i:i+3)*di(i:i+3,:);
        ai_ord(i,1:2)=[i,ai(i,1)];
        di_ord(i,1:2)=[i,di(i+3,1)];
    else
        r=4*(i-1)+1;
        Ai(r:r+3,:)=Ki(r:r+3,r:r+3)*Di(r:r+3,:);
        ai(r:r+3,:)=ki(r:r+3,r:r+3)*di(r:r+3,:);
        ai_ord(i,1:2)=[i,ai(r,1)];
        di_ord(i,1:2)=[i,di(r+3,1)];
    end
end

%% Paso 14
%% Comprobación de equilibrio
% [PR]=Acum2(lee1,A1,PR);
PR=zeros(ngdl_rest,1);
for i=1:n
    if i==1
        [PR]=Acum2(leei(i,:),Ai(i:i+3,:),PR);
    else
        r=4*(i-1)+1;
        [PR]=Acum2(leei(i,:),Ai(r:r+3,:),PR);
    end
end
end
toc

```

## A5.2. Programa validación del análisis matricial estructura paramétrica

```

%% Análisis Matricial Estructura Paramétrica
% Escuela Politécnica Nacional
% Elaborado por: Edison Garófalo
tic
clc
clear all
close all
%% Datos de las propiedades del material
Es=2030000*10; %Modulo de elasticidad %T/m2
fy=2560*10; %Resistencia a la fluencia del acero %T/m2
%% Coordenadas paramétricas de la armadura
%% Parámetros de la geometría
    clf

    %% Geometría

    Datos=[180 5 10 0.50 1.0 0.30 8]; %Ejercicio Uno
%           Datos=[240 4 9 0.40 0.5 0.25 6]; %Ejercicio Dos
%

CM=Datos(1,1); %Carga sobre la estructura %CM=0.98*R1*R2=58 kg/m2
%58<CM<96 kg/m2
Lcolaborante=Datos(1,2); %Longitud colaborante (m)

La=Datos(3); %Luz de la estructura
hi=Datos(1,4); %Alto de la estructura lado izquierdo Este Valor no puede ser igual a Cero
    hd=hi; %Alto de la estructura lado derecho
    deltah=Datos(1,5); %Límite superior (+)
    deltac=Datos(1,6); %Limite en la parte central no puede ser igual a cero
%           hci=hc1; %Límite inferior derecho
%           % deltad=hi+deltah-deltac; %Limite Inferior (- ó +)
%           ndd=diagonales1/2 %Numero de diagonales en cada lado
    nd=Datos(1,7); %Número de diagonales en los dos lados
    hcs=hi+deltah; %Altura parábola superior hci<hcs
    hci=hi+deltah-deltac; %Altura parábola inferior
%% Datos restricciones de los apoyos
% fix= restricciones en los apoyos [Fx Fy] 1 es restricción y 0 es libre
fix=[1 1
    1 1];
% ap= número de apoyos
ap=length(fix(:,1));
nglr=sum(sum(fix)); %Numero de las restricciones de los apoyos

%% Coordenadas paramétricas de la armadura

```

```

    % Coordenadas de los Elementos
    % Las coordenadas de los elementos, siempre deben ser los primeros elementos
    % los que van en los apoyos,
        % estas coordenadas deben partir desde el apoyo como inicio
    % Paramétrico Parábola
        [ CoordXY ] = fparametricoParabola(La,hi,hd,nd,ap,deltah,deltac);

    % Paramétrico Pratt
        %[ CoordXY ] = fparametPrant(La,hi,hd,nd,ap,deltah,deltac );
% ap: número de apoyos
[N,E]=fNodos1(CoordXY,ap);
% % elementos=5 w[E Ni Nf]
elementos=[E N];
%% Enumera los nodos en la armadura.
[nodos] = fNodos1( CoordXY,ap);
%Coloca las coordenadas de los nodos para graficar
    [nodosxy] = fCoordNodos( CoordXY,N,ap);
    nodosxy(4,:)= [4 1.1547, -2];
%% Ubica los grados de libertad totales y los cinemáticos
ig=2; %Grados de libertad por nodo
MaxN=max(max(N)); ngdl_rest=MaxN*ig; %Todas incluidas restricciones
ngl=ngdl_rest-nglr; %Indeterminación cinemática (Todas menos las reacciones)

%% Enumera los grados de libertad por cada nodo
% N_GDL=[ Nodo GDLi GDLf]
N_GDL=fgradoslibertad1(MaxN,fix,ig);
N_GDLMATR=N_GDL;
for i=1:MaxN
    N_GDLMATR(i,2:3)=[i, (i+1)];
    if i>=2
        N_GDLMATR(i,2:3)=[2*i-1, 2*i];
    end
end

%% Ensamble
% Este vector es necesario para ensamblar la matriz de rigidez global
% lee1=[GDLi GDLj]; lee1=[1 2 5 6];
% Coloca el vector de ensamblaje
n=length(CoordXY(:,1)); %número de elementos
for i=1:n
    leei(i,1:ig) = N_GDLMATR(N(i,1),2:ig+1);
    leei(i,ig+1:2*ig) = N_GDLMATR(N(i,2),2:ig+1);
end

%% Gráfica la estructura
% fDibujarMat( CoordXY,nodosxy,E), drawnow

```

```

fDibujar( CoordXY,nodosxy,E ),drawnow
%% Secciones de los elementos
Tabla1=3.06*ones(n,1)/10000; %Area en m2

%% Inicio del Cálculo Estructural

%% Cálcula el vector de transformación de cada elemento (T)
[Ti,Li]=M_TAP2(CoordXY,ig,n);
%% Matrices de rigidez por elemento
% % [ ki ] = M_KAP2(Tabla,L,ig)
[ ki ] = M_KAP2AG(Tabla1,Li,Es,ig,n);

%% Rigidez en coordenadas globales de cada elemento;
% % Ki=Ti*ki*Ti';
[ Ki ] = Ti*ki*Ti';

%% Matriz de rigidez Global del Sistema ST
%% Paso 6
% [ST]=Ens(lee1,K1,ST,ngdl_rest);
[ST]=zeros(ngdl_rest); nn=length(Ki);

    for i=1:n
        if i==1
            [ST]=Ens(leei(i,:),Ki(i:i+3,i:i+3),ST,ngdl_rest);
        else
            r=4*(i-1)+1;
            [ST]=Ens(leei(i,:),Ki(r:r+3,r:r+3),ST,ngdl_rest);
        end
    end

%% Paso 7 En armaduras no se tiene empotramientos
%% Cargas Externas
% Distribución de Fuerzas verticales en los nodos superiores
[ PP ] = zeros(max(max(nodos))*ig,1);
CM=Datos(1,1); Lcolaborante=Datos(1,2); %kg/m2 %m
P=CM*Lcolaborante*Li(3,1)/1000; %Toneladas
for i=1:1:(n-5)/3
    PP(nodos(i,2)*ig,1)=-P; PP(nodos(i,2)*ig,1)=-P; %Toneladas
end

%% Paso 8
% % SAEP Acumula las acciones de empotramiento perfecto
% % SAEP=zeros(ngdl_rest,1);
% % [SAEP]=Ens_SAEP(lee2,AEP2,SAEP,ngdl_rest);
%% Paso 9
%% Solución del sistema de ecuaciones;
%Indeterminación cinemática (Todas menos las reacciones)

```

```

format long
Delta=zeros(ngdl_rest,1);
Delta(nglr+1:ngdl_rest,1)=ST(nglr+1:ngdl_rest,nglr+1:ngdl_rest)^-
1*(PP(nglr+1:ngdl_rest,:));
%% Paso 10
%% Reacciones
% reac=ST(1:ngl,:)*Delta+SAEP(1:ngl);
reac=ST(1:nglr,:)*Delta;

%% Paso 11
%% Desplazamientos de los elementos en coordenadas Globales
% D1=Delta(lee1);
nn=length(Ki); [Di]=zeros(nn,1);
for i=1:n

    if i==1
        Di(i:i+3,:)=Delta(leei(i,:));
    else
        r=4*(i-1)+1;
        Di(r:r+3,:)=Delta(leei(i,:));
    end
end

%% Paso 12
%% Desplazamientos de los elementos en coordenadas Locales
% di=Ti'*Di;
nn=length(Ki); [di]=zeros(nn,1);
for i=1:n
    if i==1
        di(i:i+3,:)=Ti(i:i+3,i:i+3)*Di(i:i+3);
    else
        r=4*(i-1)+1;
        di(r:r+3,:)=Ti(r:r+3,r:r+3)*Di(r:r+3);
    end
end

%% Paso 13
%% Acciones en los elementos(en coordenadas Locales y Globales)
% A2=K2*D2+AEP2; Globales, AEP2 en este caso no hay debido, que no hay acciones de
% empotramiento perfecto
% a1=k1*d1+aep1; A1=K1*D1+AEP1
[Ai]=zeros(nn,1); [ai]=zeros(nn,1);

for i=1:n
    if i==1
        Ai(i:i+3,:)=Ki(i:i+3,i:i+3)*Di(i:i+3,:);
        ai(i:i+3,:)=ki(i:i+3,i:i+3)*di(i:i+3,:);
    end
end

```

```

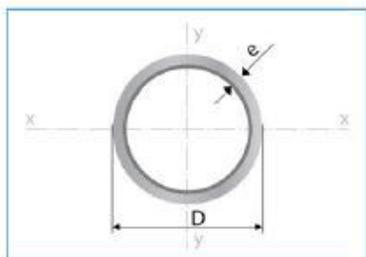
        ai_ord(i,1:2)=[i,ai(i,1)];
        di_ord(i,1:2)=[i,di(i+3,1)];
    else
        r=4*(i-1)+1;
        Ai(r:r+3,:)=Ki(r:r+3,r:r+3)*Di(r:r+3,:);
        ai(r:r+3,:)=ki(r:r+3,r:r+3)*di(r:r+3,:);
        ai_ord(i,1:2)=[i,ai(r,1)];
        di_ord(i,1:2)=[i,di(r+3,1)];
    end
end
end

%% Paso 14
%% Comprobación de equilibrio
% [PR]=Acum2(lee1,A1,PR);
PR=zeros(ngdl_rest,1);
for i=1:n
    if i==1
        [PR]=Acum2(leei(i,:),Ai(i:i+3,:),PR);
    else
        r=4*(i-1)+1;
        [PR]=Acum2(leei(i,:),Ai(r:r+3,:),PR);
    end
end
end
toc

```

**ANEXO No6**  
**ÁREAS Y PESOS DE PERFILES COMERCIALES A UTILIZAR**

## A6.1. Áreas y pesos de perfiles comerciales tubulares



Largo Normal:  
6 metros  
Recubrimiento:  
Negro o Galvanizado  
Calidad de Acero:  
JIS G 3132, ASTM A 36 y ASTM  
A572 Gr 50  
Norma de Fabricación:  
ASTM A 500/ NTE INEN 2415  
Espesores:  
Desde 1,40 a 6,00 mm  
Observaciones:  
Otras dimensiones y largos,  
previa consulta

### › Aplicaciones

- Automotriz y de autopartes: carrocería, guardachoques, tubos de escape, remolques y rodados.
- Agroganadera: maquinaria e implementos agrícolas, avícolas y ganaderos.
- Señalización y vialidad: alumbrado, soportes para señalética.
- Aparatos de gimnasia y fitness.
- Construcción: rejas, portones, columnas, andamios.
- Estructuras: portantes, galpones, invernaderos y naves industriales.

### › Beneficios

- Duradero: Al construir con acero, asegura que su producto tenga una vida útil, superior a los materiales de construcción convencionales.
- Económico: Se optimiza el uso del material produciéndose ahorro en obras, por facilidad y rapidez de uso y armado ahorra en tiempo y personal.
- Ecológico: Con el uso de acero, se minimiza la utilización de madera y su desecho, por lo que contribuye al medio ambiente.

Designaciones		Área	Peso	Propiedades Estáticas			
Diámetro Exterior	Espesor			Momento de inercia	Módulo de resistencia	Radio de giro	
D	e	A	P	I	W	i	
pulg.	mm	cm <sup>2</sup>	kg/m	cm <sup>4</sup>	cm <sup>3</sup>	cm	
3/4	19,10	1,40	0,78	0,61	0,30	0,32	0,63
		1,50	0,83	0,65	0,32	0,34	0,62
		1,80	0,98	0,77	0,37	0,39	0,61
		2,00	1,07	0,84	0,39	0,41	0,61
7/8	22,20	1,40	0,92	0,72	0,50	0,45	0,74
		1,50	0,98	0,77	0,53	0,47	0,73
		1,80	1,16	0,91	0,61	0,55	0,72
		2,00	1,27	1,00	0,66	0,59	0,72
1	25,40	1,40	1,06	0,83	0,76	0,60	0,85
		1,50	1,13	0,88	0,81	0,64	0,85
		1,80	1,33	1,05	0,93	0,74	0,84
		2,00	1,47	1,15	1,01	0,80	0,83
1 1/4	31,80	1,40	1,33	1,05	1,54	0,97	1,07
		1,50	1,43	1,12	1,63	1,03	1,07
		1,80	1,69	1,33	1,91	1,20	1,06
		2,00	1,87	1,47	2,08	1,31	1,05
1 1/2	38,10	1,40	1,61	1,27	2,72	1,43	1,30
		1,50	1,72	1,35	2,89	1,52	1,30
		1,80	2,05	1,61	3,39	1,78	1,28
		2,00	2,27	1,78	3,71	1,95	1,28
1 3/4	44,50	1,40	1,89	1,49	4,39	1,98	1,52
		1,50	2,02	1,59	4,67	2,10	1,52
		1,80	2,41	1,89	5,49	2,47	1,51
		2,00	2,67	2,09	6,02	2,71	1,50
1 7/8	47,60	1,40	2,03	1,60	5,44	2,28	1,64
		1,50	2,17	1,71	5,79	2,43	1,63
		1,80	2,59	2,03	6,81	2,86	1,62
		2,00	2,87	2,25	7,47	3,14	1,61
2	50,80	1,40	2,17	1,71	6,63	2,61	1,75
		1,50	2,32	1,82	7,06	2,78	1,74
		1,80	2,77	2,18	8,33	3,28	1,73
		2,00	3,07	2,41	9,14	3,60	1,73
2 3/8	60,30	1,50	2,77	2,18	12,00	3,98	2,08
		1,80	3,31	2,60	14,18	4,70	2,07
		2,00	3,66	2,88	15,60	5,17	2,06
		3,00	5,40	4,24	22,25	7,38	2,03
2 1/2	63,50	1,50	2,92	2,29	14,05	4,42	2,19
		1,80	3,49	2,74	16,62	5,23	2,18
		2,00	3,86	3,03	18,29	5,76	2,18
		3,00	5,70	4,48	26,15	8,24	2,14
3	76,20	1,50	3,52	2,76	24,56	6,45	2,64
		1,80	4,21	3,30	29,13	7,65	2,63
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
4	101,60	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
5	127,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
6	152,40	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
6 5/8	168,30	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
6 3/4	177,80	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
7	188,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
7 1/2	198,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
8	203,20	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
8 1/2	213,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
9	218,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
9 1/2	228,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
10	233,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
10 1/2	243,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
11	248,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
11 1/2	258,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
12	263,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
12 1/2	273,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
13	278,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
13 1/2	288,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
14	293,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
14 1/2	303,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
15	308,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
15 1/2	318,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
16	323,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
16 1/2	333,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
17	338,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
17 1/2	348,00	1,50	4,21	3,30	29,13	7,65	2,63
		1,80	4,66	3,66	32,11	8,43	2,62
		2,00	4,66	3,66	32,11	8,43	2,62
		3,00	6,90	5,42	46,29	12,15	2,59
18							

**ANEXO No7**  
**OPTIMIZACIÓN DE SECCIONES**

### A7.1. Programa optimización de secciones estructura de tres elementos.

```

%% Optimización de Secciones Armadura de Tres Elementos
% Escuela Politécnica Nacional
% Elaborado por: Edison Garófalo
tic
clc
clear all
close all
%% Datos de las propiedades del material
% Es=2070000; %kg/cm2 %Modulo de elasticidad del acero
Es=1967840*10; %T/m2
fy=1442*10; %T/m2 %Resistencia a la fluencia del acero
densidad=7850; %kg/m3 %Densidad
%% Datos restricciones de los apoyos
% fix= restricciones en los apoyos [Fx Fy] 1 es restricción y 0 es libre
fix=[1 1
     1 1
     1 1];
ap=length(fix(:,1)); %Numero apoyos
nglr=sum(sum(fix)); %Numero de las restricciones de los apoyos

%% Coordenadas paramétricas de la armadura
% Coordenadas de los Elementos
%Las coordenadas de los elementos, siempre deben ser los primeros
% elementos los que van en los apoyos, estas coordenadas deben partir desde
% el apoyo como inicio
%Paramétrico Parábola
[ CoordXY ] = [0 0 1.1547 -2
              1.1547 0 1.1547 -2
              2*(1.1547) 0 1.1547 -2];
%% Añade los nodos a la estructura
% ap= número de apoyos
% N=[Ni Nf]
% E=[nx1] Elementos
[N,E]=fNodos1(CoordXY,ap);
% % [N,E]=fNodos(CoordXY,ap);
% % elementos=[E Ni Nf]
elementos=[E N];
%% Enumera los nodos en la armadura.
[nodos] = fNodos1( CoordXY,ap);
%Coloca las coordenadas de los nodos para graficar
[nodosxy] = fCoordNodos( CoordXY,N,ap);
nodosxy(4,:)= [4 1.1547 -2];

%% Ubica los grados de libertad totales y los cinemáticos

```

```

ig=2; %Grados de libertad por nodo
MaxN=max(max(N)); %El ultimo nodo, el mayor
ngdl_rest=MaxN*ig; %Todas incluidas restricciones
ngl=ngdl_rest-nglr; %Indeterminación cinemática (Todas menos las reacciones)

%% Enumera los grados de libertad por cada nodo
% N_GDL=[ Nodo GDLi GDLf]
N_GDL=fgradoslibertad1(MaxN,fix,ig);
N_GDLMATR=N_GDL;
for i=1:MaxN
    N_GDLMATR(i,2:3)=[2*i-1, 2*i];
end
%% Ensamble
% Este vector es necesario para ensamblar la matriz de rigidez global
% % lee1=[GDLi GDLj];
% % lee1=[1 2 5 6];
% Coloca el vector de ensamblaje
% [ leei ] = fleec( CoordXY,MaxN,N_GDL,N,ig);
n=length(CoordXY(:,1)); %número de elementos
for i=1:n
    leei(i,1:ig) = N_GDLMATR(N(i,1),2:ig+1);
    leei(i,ig+1:2*ig) = N_GDLMATR(N(i,2),2:ig+1);
end

%% Grafica la estructura
fDibujar( CoordXY,nodosxy,E ),drawnow

%% Calcula la longitud de la barra
Li=zeros(n,1);
for i=1:n
    Li(i,1)=sqrt((CoordXY(i,1)-CoordXY(i,3))^2+(CoordXY(i,2)-CoordXY(i,4))^2);
end

%% Secciones de los elementos
TablaIN=[1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0 3.2 3.4 3.6 3.8 4.0 4.4]/10000; % Area
en m2

% TablaIN=[TablaIN;2*TablaIN]; %Para utilizar las áreas duplicadas
n=length(Li(:,1)); %Número de elementos
%%
for ix=1:100 %ixx puede ser máximo el número de generaciones
NumeroIndividuos=10; %Ingresar Números pares
Nr=NumeroIndividuos;
Tabla=zeros(n*Nr,1);
L=4; %Longitud del cromosoma en L=5 son 32 elementos Numero=2^L
LC=L;

```

```

%Dimencion para el de Número de Individuos

%1 Inicia desde la sección inicia y 17 porque la Tabla tiene 16 tipos de secciones
% Una tabla 16 elementos
I = [1*ones(n,1) 16*ones(n,1) ]; %Intervalos de búsqueda depende del número de filas de
%TablaIN
% Dos Tablas 34 elementos
% I = [1*ones(n,1) 34*ones(n,1) ]; %Intervalos de búsqueda depende del número de filas de
%TablaIN
%Generemos la población de individuos
Pob_2 =round(rand(Nr*n,L));%base 2
%Población en base decimal
Pob_10= fPobDec(Pob_2,n,Nr);
%Poblaciones en R
% [Pob_float] = fScale(Pob_10,I,nele,LC,Nr);
% Pob_floatModeAcc=zeros(nele,Ngeneraciones);
Pob_real = fScale(Pob_10,I,n,L,Nr);

%% Generaciones
Ngeneraciones=10;
cont=1; %Contador del algoritmo genético
fxpmin=zeros(Ngeneraciones,3); %fxpmin=[Peso Pesopenalizado Penalizacion]
ai_acum=zeros(n,Nr*Ngeneraciones);
Tabla0=zeros(Nr,size(TablaIN,1));
fagxp=zeros(Nr,3);
for cont=1:Ngeneraciones

fxp=zeros(Nr,1);
fxx=zeros(Nr,1);
    for jn=1:Nr
        for i=1:n
            r=(jn-1)*n+i;
            Tabla0(r,:)=TablaIN(Pob_real(r),1);
        end
    end
end

%% Número de Individuos
    C=zeros(Nr,1); %General Penalty
    for is=1:Nr
        Tabla1=Tabla0(n*(is-1)+1:n*(is-1)+n,1);

%% Inicio del Cálculo Estructural
%% Matrices de rigidez por elemento
% % [ ki ] = M_KAP2(Tabla,L,ig)
    [ ki ] = M_KAP2AG(Tabla1,Li,Es,ig,n);

```

```

%% Calcula el vector de transformación de cada elemento (T)
[Ti]=M_TAP2(CoordXY,ig,n);

%% Rigidez en coordenadas globales de cada elemento;
% % Ki=Ti*ki*Ti';
[ Ki ] = Ti*ki*Ti';

%% Matriz de rigidez Global del Sistema ST
%% Paso 6
% [ST]=Ens(lee1,K1,ST,ngdl_rest);
[ST]=zeros(ngdl_rest); nn=length(Ki);
m=nn/(2*ig); %Número de elementos

for i=1:n
    if i==1
        [ST]=Ens(leei(i,:),Ki(i:i+3,i:i+3),ST,ngdl_rest);
    else
        r=4*(i-1)+1;
        [ST]=Ens(leei(i,:),Ki(r:r+3,r:r+3),ST,ngdl_rest);
    end
end

%% Paso 7 En armaduras no se tiene empotramientos
%% Cargas Externas
% Distribución de Fuerzas verticales en los nodos superiores
[ PP ] = zeros(6,1);
PP(7:8,1)=[.96;-9.6]; %Toneladas

%% Paso 8
% % SAEP Acumula las acciones de empotramiento perfecto
% % SAEP=zeros(ngdl_rest,1);
% % [SAEP]=Ens_SAEP(lee2,AEP2,SAEP,ngdl_rest);
%% Paso 9
%% Solución del sistema de ecuaciones;
%Indeterminación cinemática (Todas menos las reacciones)
format long
Delta=zeros(ngdl_rest,1);
Delta(nglr+1:ngdl_rest,1)=ST(nglr+1:ngdl_rest,nglr+1:ngdl_rest)^-
1*(PP(nglr+1:ngdl_rest,:));
%% Paso 10
%% Reacciones
% reac=ST(1:ngl,:)*Delta+SAEP(1:ngl);
reac=ST(1:nglr,:)*Delta;
%% Paso 11
%% Desplazamientos de los elementos en coordenadas Globales
% D1=Delta(lee1);
nn=length(Ki); [Di]=zeros(nn,1);

```

```

for i=1:n

    if i==1
        Di(i:i+3,:)=Delta(leei(i,:));
    else
        r=4*(i-1)+1;
        Di(r:r+3,:)=Delta(leei(i,:));
    end
end

%% Paso 12
%% Desplazamientos de los elementos en coordenadas Locales
% di=Ti'*Di;
    nn=length(Ki); [di]=zeros(nn,1);
for i=1:n
    if i==1
        di(i:i+3,:)=Ti(i:i+3,i:i+3)'*Di(i:i+3);
    else
        r=4*(i-1)+1;
        di(r:r+3,:)=Ti(r:r+3,r:r+3)'*Di(r:r+3);
    end
end

%% Paso 13
%% Acciones en los elementos(en coordenadas Locales y Globales)
% A2=K2*D2+AEP2; Globales, AEP2 en este caso no hay debido, que no hay acciones de
% empotramiento perfecto
% a1=k1*d1+aep1; A1=K1*D1+AEP1;

[Ai]=zeros(nn,1); [ai]=zeros(nn,1);

for i=1:n
    if i==1
        Ai(i:i+3,:)=Ki(i:i+3,i:i+3)*Di(i:i+3,:);
        ai(i:i+3,:)=ki(i:i+3,i:i+3)*di(i:i+3,:);
        ai_ord(i,1:2)=[i,ai(i,1)];
        di_ord(i,1:2)=[i,di(i+3,1)];
    else
        r=4*(i-1)+1;
        Ai(r:r+3,:)=Ki(r:r+3,r:r+3)*Di(r:r+3,:);
        ai(r:r+3,:)=ki(r:r+3,r:r+3)*di(r:r+3,:);
        ai_ord(i,1:2)=[i,ai(r,1)];
        di_ord(i,1:2)=[i,di(r+3,1)];
    end
end

%% Paso 14

```

```

%% Comprobación de equilibrio
% [PR]=Acum2(lee1,A1,PR);
PR=zeros(ngdl_rest,1);
for i=1:n
    if i==1
        [PR]=Acum2(leei(i,:),Ai(i:i+3,:),PR);
    else
        r=4*(i-1)+1;
        [PR]=Acum2(leei(i,:),Ai(r:r+3,:),PR);
    end
end
%% Paso 15
%% Chequeos
% %% Chequeo del desplazamiento en el centro
% [ des_centro,desplaz_permitido ] = fdesplazamientoYcA2D(
La,hci,ap,Di,nodos,elementos);

%% Chequeo de Esfuerzos en las barras
%     it=(is-1)*n;
% [Fn,Ae,F,Fe,landacomp,landac] = fesfuerzosfuerzasTC(Tabla(it+1:it+n,:),Li,Es,fy,ai_ord);
    esfuerzo=zeros(n,1);
    for i=1:n
        esfuerzo(i,1)=ai_ord(i,2)/Tabla1(i,1);
    end

%% Paso 16
% Función Objetivo está en función del peso
    for i=1:n
        fx(i,1)=Li(i,1)*Tabla1(i,1)*densidad; %kg m*cm2*kg/m3
    end
    fxx(is)=sum(fx); %Total Accumulate Weight

%Coeficiente de violación (Ci)
% Esta en función de los esfuerzos del elemento
    Ci=zeros(n,1);
    for i=1:n
        Ci(i,1)=abs(esfuerzo(i,1)/(fy))-1;
        if Ci(i,1)>0
            Ci(i,1)=abs(esfuerzo(i,1)/(fy))-1;
        else
            Ci(i,1)=0;
        end
    end

end
%%
%Penalización por no cumplir la deformación máxima

```

```

%Elementos del Extremo 1, 2 y 3 Nodo de Interés es 4
%final para el elemento 6 es el nodo inicial

    C1=Di(+3*4); %Esta fórmula es para este caso en específico
    if di_ord(i,2)>=0.005 %No debe ser mayor la deformación a 5 centímetros
        C1i=0.05/C1;
    else
        C1i=0;
    end
%    C(is,1)=sum(Ci);
    C(is,1)=sum(Ci)+C1i;

%% Función objetivo penalizada fxp
%Según las pruebas realizadas por otros autores k=1000 cuando fxp=1/f
    K=1000; fxx(is,1)=sum(fx);
    fxp(is,1)=1/((fxx(is,1)*(1+K*C(is,1)))); %Se invierte para poder realizar la
minimización.

    if cont==1
        ai_acum(:,is)=ai_ord(:,2) ;
        ico(is,1)=is;
    else
        icont=Nr*(cont-1)+is;
        ai_acum(:,icont)=ai_ord(:,2);
        ico(icont,1)=icont;
    end

end

%% Nueva Generación
% Selection
nele=n;
%Metodo 1      Residuo +
[ dec ] = fselectionR(fxp,Nr);
% Metodo 2      Por Ruleta ++
[ decr] = fSelectionRul(fxp,Nr);
% Metodo 3      Por Torneo Mejor +++
nt=3; %Cantidad de individuos que entran a la competencia
[ dect] = fselectionTor(fxp,Nr,nt);
%Reproduction
[ Pob_C,Parents,Couples,Nr] = freproduction( Pob_2,Nr,nele,dect);
%LC=nele;%# of chromosomes
Pob_2= Pob_C;
MutationPorc=0.1; % menor al 1 por ciento
Pob_2 = fMutation(Pob_2,MutationPorc);
Pob_10= fPobDec(Pob_2,nele,Nr);
[Pob_float] = fScale(Pob_10,I,nele,LC,Nr);

```

```

%Results mode
Pob_floatMode=zeros(nele,1);
for ii=1:nele
    Pob_floatMode(ii)=mode(Pob_float(ii:nele:end));
end

Pob_floatModeAcc(:,cont)=Pob_floatMode;
Pob_real=Pob_float;
%%
fagxp=[fxx fpx C];
[fxpmin(cont,2),id]=(max(fagxp(:,2))); %Acumula la función penalizada 1/f
fxpmin(cont,1)=fagxp(id,1); %Acumula los pesos mínimos por generación
fxpmin(cont,3)=fagxp(id,3); %Acumula la función penalizada
Peso(:,cont)=fxx;
Penal(:,cont)=C;
% cont=cont+1;
end
    if i==1
        Pob_floatAcum(ixx:n,:)=Pob_floatModeAcc;
    else
        Pob_floatAcum((ixx-1)*n+1:ixx*n,:)=Pob_floatModeAcc;
    end

AreaRes=[TablaIN(Pob_floatModeAcc(:,end),1*ones(1))];
AreaFIN(:,ixx)=AreaRes;
PesoFin(:,ixx)=fxpmin(end,1); PesoAcum(:,ixx)=fxpmin(:,1);
Fobjetivo(:,ixx)=fxpmin(end,2); FobjAcum(:,ixx)=fxpmin(:,2);
PenaFin(:,ixx)=fxpmin(end,3); PenaAcum(:,ixx)=fxpmin(:,3);
aifin(:,ixx)=ai_ord(:,2);

%En total las iteraciones seria T=ixx*Ngeneraciones*Nr=10x10x10=1000iteraciones
PobFin(1:n,ixx)=Pob_floatModeAcc(:,end); %acumula las secciones del mejor individuo

end
%%
PesoFin22=PesoFin.*PenaFin; %Parámetro para considerar la penalización del peso mínimo
for i=1:ixx
    if PesoFin22(1,i)==0
        PesoFin2(1,i)=PesoFin(1,i);
    else
        PesoFin2(1,i)=1000*PesoFin(1,i);
    end
end
[aay,iay]=min(PesoFin2);

```

```

[aax,iax]=min(min(PesoFin2));
Place(1,1:2)=[iay(iax),iax];
% Pminf=(PesoFin((iay(iax)),iax));

iselect=Place(1,1); %put "iselect" of your needed, that depends of your iteration
ibody=Place(1,2); %put "ibody" of your individual

PminSection=TablaIN(PobFin(:,iselect),1); %Sección cm^2
PminWeight=TablaIN(PobFin(:,iselect),1)*densidad/10000; %Peso de cada perfil
kg/m
TrueWeight=sum(Li.*PminWeight);
TruePena=min(min(PenaFin));

%ended=[Iteration,Section,Weigth kg/m,Pesototal,Penalización]
ended=zeros(n,6);
[ended(2,5),uv]=min(PesoFin2);
ended(1,1)=uv; ended(1:n,2)=PobFin(1:n,iselect);
ended(:,3:4)=[PminSection PminWeight]; ended(1,5)=TrueWeight;
ended(1:2,6)=[PenaFin(1,iselect);PenaFin(1,uv)];
disp(ended)
% sum(Li.*TablaIN([10;15;1],3)) ; 12.48kg; 1.87cm2; 4.66cm2; 0.98cm2
subplot(3,1,1)
plot((1:Ngeneraciones)',fxpmin(:,1))
xlabel('N° Generaciones')
ylabel('Peso (kg)')
title(' Peso(kg)vs N° Generaciones ')

subplot(3,1,2)
plot((1:Ngeneraciones)',fxpmin(:,2))
xlabel('N° Generaciones')
ylabel('1/(fxx*(1+K*C))')
title('1/(fxx*(1+K*C)) vs N° Generaciones')

subplot(3,1,3)
plot((1:Ngeneraciones)',fxpmin(:,3))
xlabel('N° Generaciones')
ylabel('C')
title('Penalización C vs N° Generaciones')

toc

```

## A7.2. Optimización de secciones estructura de diez elementos.

Para la optimización de secciones de la estructura de diez elementos se utiliza el Anexo A7.1 con la excepción de que se debe ingresar la nueva geometría, apoyos y las cargas como se observa a continuación:

### %% Optimización de secciones Armadura de 10 Elementos

%%

% Escuela Politécnica Nacional

% Elaborado por: Edison Garófalo

tic

clc

clear all

close all

%% Datos de las propiedades del material

% Es=2070000; %kg/cm2 %Modulo de elasticidad del acero

Es=10000\*10\*1000/(2.2\*(2.54^2)); %T/m2

fy=25\*10\*1000/(2.2\*(2.54^2)); %T/m2 %Resistencia a la fluencia

densidad=2770; %kg/m3

%% Datos restricciones de los apoyos

% fix= restricciones en los apoyos [Fx Fy] 1 es restriccion y 0 es libre

fix=[1 1

1 1];

ap=length(fix(:,1)); %Numero apoyos

nglr=sum(sum(fix)); %Numero de las restricciones de los apoyos

%% Coordenadas paramétricas de la armadura

% Coordenadas de los Elementos

%Las coordenadas de los elementos, siempre deben ser los primeros

% elementos los que van en los apoyos, estas coordenadas deben partir desde

% el apoyo como inicio

[ CoordXY ] = 2.54\*[0 0 3.6 0

0 3.6 3.6 3.6

3.6 0 2\*3.6 0

3.6 3.6 2\*3.6 3.6

3.6 0 3.6 3.6

2\*3.6 0 2\*3.6 3.6

0 3.6 3.6 0

0 0 3.6 3.6

3.6 3.6 2\*3.6 0

3.6 0 2\*3.6 3.6]; %Longitud en metros

%% Añade los nodos a la estructura

% ap= número de apoyos

% N=[Ni Nf]

% E=[nx1] Elementos

```

[N,E]=fNodos1(CoordXY,ap);
% % [N,E]=fNodos(CoordXY,ap);
% % elementos=[E Ni Nf]
elementos=[E N];
%% Enumera los nodos en la armadura.
[nodos] = fNodos1( CoordXY,ap);
%Coloca las coordenadas de los nodos para graficar
[nodosxy] = fCoordNodos( CoordXY,N,ap);
%% Ubica los grados de libertad totales y los cinemáticos
ig=2; %Grados de libertad por nodo
MaxN=max(max(N)); %El ultimo nodo, el mayor
ngdl_rest=MaxN*ig; %Todas incluidas restricciones
ngl=ngdl_rest-nglr; %Indeterminación cinematica (Todas menos las reacciones)

%% Enumera los grados de libertad por cada nodo
% N_GDL=[ Nodo GDLi GDLf]
N_GDL=fgradoslibertad1(MaxN,fix,ig);
N_GDLMATR=N_GDL;
for i=1:MaxN
    N_GDLMATR(i,2:3)=[2*i-1, 2*i];
end
%% Ensamble
n=length(CoordXY(:,1)); %número de elementos
for i=1:n
    leei(i,1:ig) = N_GDLMATR(N(i,1),2:ig+1);
    leei(i,ig+1:2*ig) = N_GDLMATR(N(i,2),2:ig+1);
end
%% Grafica la estructura
fDibujar( CoordXY,nodosxy,E ),drawnow
Li=zeros(n,1);
for i=1:n
    Li(i,1)=sqrt((CoordXY(i,1)-CoordXY(i,3))^2+(CoordXY(i,2)-CoordXY(i,4))^2);
end

%%Secciones de los elementos
Tablapulg=2.54^2*[0.1;1;1.62;1.8;1.99;2.13;2.38;2.62;2.63;2.88;2.93;3.09;3.13;3.38;...
    3.47;3.55;3.63;3.84;3.87;3.88;4.18;4.22;4.49;4.59;4.80;4.97;5.12;5.74;...
    7.22;7.97;11.50;13.50;13.90;14.20;15.5;16;16.9;18.8;19.9;22;22.9;26.5;30;33.5]/10000;

TablaIN(:,1)=Tablapulg;
TablaIN=[2*TablaIN;4*TablaIN]; %Para utilizar las areas duplicadas
n=length(Li(:,1)); %Número de elementos

```

**ANEXO No8**  
**OPTIMIZACIÓN FORMA, TOPOLOGÍA Y SECCIONES**

### A8.1. Optimización de forma, topología y secciones de estructura isostática de luz de veinte metros.

```

%% Optimización de Forma y Topología de una Armadura Paramétrica.
%%
% Escuela Politécnica Nacional
% Elaborado por: Edison Garófalo
tic
clc
clear all
close all
%% Datos de las propiedades del material
Es=2.039e7; %T/m2 %Modulo de elasticidad del acero
fy=3518*10; %T/m2 %Resistencia a la fluencia

%% Datos para la Geometría
% Siempre se ingresa los elementos que van en los apoyos primero
% Siempre los elementos deben formar triángulos entre sí para que no se presente
matrices singulares
La=20; %Ancho de la armadura Dato Inicial No varia
B=6; %Ancho colaborante, Distancia entre armaduras No varia.
q=100; %Carga por metro cuadrado % kg/m2

% Datos vectores de las variables x1,x2,x3,x4 x
%Ingresar el Rango
hstart=0.05;hend=1; rango_h=hend-hstart;
deltahstart=0.5; deltahend=3; rango_deltah=deltahend-deltahstart;
deltacstart=0.1; deltacend=4; rango_deltac=deltacend-deltacstart;
hcstart=-1; hccend=4; rango_hc=hccend-hcstart;

nmin=floor(La/1.25); nmax=floor(La/0.8);
j=1;
for i=nmin:1:nmax
    res=i/2 -floor(i/2);
    if res==0
        diagonales(j,1)=i;
        j=j+1;
    end
end

NfilasM=100; %Ingresar de acuerdo al espaciamiento deseado
% Mcromosoma= [hii deltahh deltacc diagonales] Tamaño Nfilasx4
nele1=4; %3 número de variables en la forma y 1 variables topológicas
Mcromosoma=zeros((NfilasM),nele1); % (N1*N2*N3*N4,4) 4 es el número de variables
for i=1:(NfilasM+1)

```

```

    Mcromosoma(i,1)=hstart+(rango_h)*(i-1)/(NfilasM);
    Mcromosoma(i,2)=deltahstart+(rango_deltah)*(i-1)/(NfilasM);
    Mcromosoma(i,3)=deltacstart+(rango_deltac)*(i-1)/(NfilasM);
%      Mcromosoma(i,3)=hcstart+(rango_hc)*(i-1)/(NfilasM);

end

    Mcromosoma(1:length(diagonales),4)=diagonales;

%%
for ic2=1:1
for icx=1:100 %icx Es el máximo número de iteraciones
    %% AG2
    Nop1=size(Mcromosoma,1);
    nele1=4; %3 número de variables en la forma y 1 variables topológicas
    LC=ceil(log2(Nop1)); % LC Longitud del cromosoma de cada variable
    hi,deltah,deltac,diagonales
    I = [1*ones(nele1,1) Nop1*ones(nele1,1) ];%Intervalos para el escalamiento
    I(end,end)=length(diagonales);
% Ingresar valores de la generación AG2
    NgeneracionesAG=25;
    MutationPorc=1; % Ingresar valor en porcentaje, menor al 1%
    NrAG=10; % Ingresar números pares
    %% Generation of individuals of population
    Pob_2 = round(rand(NrAG*nele1,LC));%base 2
    %Poblacion en base decimal
    Pob_10= fPobDec(Pob_2,nele1,NrAG);
    %Poblaciones en R
    [Pob_float] = fScale(Pob_10,I,nele1,LC,NrAG);
    Pob_floatModeAcc=zeros(nele1,NgeneracionesAG);
%% Generations
%%
%% Inicia Algoritmo Genético de Forma
    contAG=0;
    while contAG<NgeneracionesAG
        contAG=contAG+1;

        fagxp=zeros(NrAG,3);
        fxp=zeros(NrAG,1);
        fxx=zeros(NrAG,1);

        for jn=1:NrAG
            if jn==1
                hii0(jn,1)=Mcromosoma(Pob_float(jn),1);
                deltah0(jn,1)=Mcromosoma(Pob_float(jn+1),2);

```

```

deltac0(jn,1)=Mcromosoma(Pob_float(jn+2),3);
hc0(jn,1)=Mcromosoma(Pob_float(jn+2),3);

diagonales0(jn,1)=Mcromosoma(Pob_float(4*jn),4);
else
jn1=4*(jn-1)+1;
hii0(jn,1)=Mcromosoma(Pob_float(jn1),1);
deltah0(jn,1)=Mcromosoma(Pob_float(jn1+1),2);
deltac0(jn,1)=Mcromosoma(Pob_float(jn1+2),3);
% hc0(jn,1)=Mcromosoma(Pob_float(jn+2),3);

diagonales0(jn,1)=Mcromosoma(Pob_float(jn1+3),4);
end
end

for isAG=1:NrAG %Number of individuals
%% Parámetros de la geometría
clf
hii1=hii0(isAG); deltahh1=deltah0(isAG);
diagonales1=diagonales0(isAG);deltacc1=deltac0(isAG);%hc1=hc0(isAG);
%% Geometría
%%
hi=hii1; %Alto de la estructura lado izquierdo Este Valor NO puede ser Cero
hd=hi; %Alto de la estructura lado derecho
deltah=deltahh1; %Límite superior (+)
deltac=deltacc1; %Limite en la parte central no puede ser igual a cero
% hci=hc1; %Límite inferior derecho
% deltad=hi+deltah-deltac; %Límite Inferior (- ó +)
% ndd=diagonales1/2 %Numero de diagonales en cada lado
nd=diagonales1; %Número de diagonales en los dos lados
hcs=hi+deltah; %Altura parábola superior hci<hcs
hci=hi+deltah-deltac; %Altura parábola inferior
% deltac=hi+deltah-hci; %Limite en la parte central
%% Datos restricciones de los apoyos
% fix= restricciones en los apoyos [Fx Fy] 1 es restricción y 0 es libre

fix=[1 1
0 1];
% ap= número de apoyos
ap=length(fix(:,1));
nglr=sum(sum(fix)); %Numero de las restricciones de los apoyos

%% Coordenadas paramétricas de la armadura
% Coordenadas de los Elementos

```

```

% Las coordenadas de los elementos, siempre deben ser los primeros elementos los que
% van en los apoyos,
% estas coordenadas deben partir desde el apoyo como inicio
% Paramétrico Parábola
% [ CoordXY ] = fparametricoParabola(La,hi,hd,nd,ap,deltah,deltac);

% Paramétrico Pratt
% [ CoordXY ] = fparametPrant(La,hi,hd,nd,ap,deltah,deltac );

%% Añade los nodos a la estructura
% ap= número de apoyos N=[Ni Nf] E=[nx1] Elementos
[N,E]=fNodos1(CoordXY,ap);
% [N,E]=fNodos(CoordXY,ap) % elementos=[E Ni Nf]
elementos=[E N];
%% Enumera los nodos en la armadura.
[nodos] = fNodos1( CoordXY,ap);
%Coloca las coordenadas de los nodos para graficar
[nodosxy] = fCoordNodos( CoordXY,N,ap);
%% Ubica los grados de libertad totales y los cinemáticos
ig=2; %Grados de libertad por nodo
MaxN=max(max(N)); %El ultimo nodo, el mayor
ngdl_rest=MaxN*ig; %Todas incluidas restricciones
ngl=ngdl_rest-nglr; %Indeterminación cinemática (Todas menos las reacciones)

%% Enumera los grados de libertad por cada nodo
% N_GDL=[ Nodo GDLi GDLf]
% N_GDL=[ Nodo GDLi GDLf]
N_GDL=fgradoslibertad1(MaxN,fix,ig);
N_GDLMATR=N_GDL;
for i=1:MaxN
    N_GDLMATR(i,2:3)=[2*i-1, 2*i];
end

N_GDLMATR(2,2)=4;N_GDLMATR(2,3)=3; %Para armaduras Isostáticas

%% Ensamble
% Este vector es necesario para ensamblar la matriz de rigidez global
% % lee1=[GDLi GDLj];
% % lee1=[1 2 5 6];
% Coloca el vector de ensamblaje
% [ leei ] = fleec( CoordXY,MaxN,N_GDL,N,ig);
n=length(CoordXY(:,1)); %número de elementos
for i=1:n
    leei(i,1:ig) = N_GDLMATR(N(i,1),2:ig+1);
    leei(i,ig+1:2*ig) = N_GDLMATR(N(i,2),2:ig+1);
end

```

```

%% Grafica la estructura
fDibujar( CoordXY,nodosxy,E),drawnow
%% Calcula la longitud de la barra
Li=zeros(n,1);
for i=1:n
    Li(i,1)=sqrt((CoordXY(i,1)-CoordXY(i,3))^2+(CoordXY(i,2)-CoordXY(i,4))^2);
end

%% Secciones de los elementos
Tablapulg=+2.54^2*[0.1;1;1.62;1.8;1.99;2.13;2.38;2.62;2.63;2.88;2.93;3.09;3.13;3.38;...
    3.47;3.55;3.63;3.84;3.87;3.88;4.18;4.22;4.49;4.59;4.80;4.97;5.12;5.74;...
    7.22;7.97;11.50;13.50;13.90;14.20;15.5;16;16.9;18.8;19.9;22;22.9;26.5;30;33.5]/10000;
%Area en m2

TablaIN=Tablapulg;
% TablaIN=[TablaIN;2*TablaIN]; %Para utilizar las áreas duplicadas
n=length(Li(:,1)); %Número de elementos
%%
ikx(ixx,1)=ixx;

%% Generaciones
% ai_acum=zeros(n*NrAG,NgeneracionesAG2);
Tabla0=zeros(n*NrAG,size(TablaIN,1));

% for cont=1:Ngeneraciones
    for jn=1:NrAG
        for i=1:n
            r=(jn-1)*n+i;
            Tabla0(r,:)=TablaIN(1,1);
        end
    end

%% Número de Individuos
    C=zeros(NrAG,1); %General Penalty
% for is=1:NrAG
    Tabla1=Tabla0(1*(isAG-1)+1:1*(isAG-1)+1,1); %l=n

%% Inicio del Cálculo Estructural
%% Matrices de rigidez por elemento
% % [ ki ] = M_KAP2(Tabla,L,ig,n)
% [ ki ] = M_KAP2AG(Tabla0,Li,Es,ig,n);
%% Calcula el vector de transformación de cada elemento (T)
[Ti]=M_TAP2(CoordXY,ig,n);

%% Rigidez en coordenadas globales de cada elemento;

```

```

% % Ki=Ti*ki*Ti';
    [ Ki ] = Ti*ki*Ti';

%% Matriz de rigidez Global del Sistema ST
%% Paso 6
% [ST]=Ens(lee1,K1,ST,ngdl_rest);
    [ST]=zeros(ngdl_rest); nn=length(Ki);
    m=nn/(2*ig); %Número de elementos

    for i=1:n
        if i==1
            [ST]=Ens(leei(i,:),Ki(i:i+3,i:i+3),ST,ngdl_rest);
        else
            r=4*(i-1)+1;
            [ST]=Ens(leei(i,:),Ki(r:r+3,r:r+3),ST,ngdl_rest);
        end
    end

%% Paso 7 En armaduras no se tiene empotramientos
%% Cargas Externas
% Distribución de Fuerzas verticales en los nodos superiores
    [ PP ] = zeros(max(max(nodos))*ig,1);
    CM=200; Lcolaborante=6; %kg/m2 %m
    P=CM*Lcolaborante*Li(3,1)/1000; %Toneladas
    for i=1:1:(n-5)/3
        PP(nodos(i,2)*ig,1)=-P; PP(nodos(i,2)*ig,1)=-P; %Toneladas
    end

%% Paso 8
% % SAEP Acumula las acciones de empotramiento perfecto
% % SAEP=zeros(ngdl_rest,1);
% % [SAEP]=Ens_SAEP(lee2,AEP2,SAEP,ngdl_rest);
%% Paso 9
%% Solución del sistema de ecuaciones;
%Indeterminación cinematica (Todas menos las reacciones)
    format long
    Delta=zeros(ngdl_rest,1);
    Delta(nglr+1:ngdl_rest,1)=ST(nglr+1:ngdl_rest,nglr+1:ngdl_rest)^-
1*(PP(nglr+1:ngdl_rest,:));
%% Paso 10
%% Reacciones
% reac=ST(1:ngl,:)*Delta+SAEP(1:ngl);
    reac=ST(1:nglr,:)*Delta;

%% Paso 11
%% Desplazamientos de los elementos en coordenadas Globales
% D1=Delta(lee1);

```

```

nn=length(Ki); [Di]=zeros(nn,1);
for i=1:n

    if i==1
        Di(i:i+3,:)=Delta(leei(i,:));
    else
        r=4*(i-1)+1;
        Di(r:r+3,:)=Delta(leei(i,:));
    end
end

%% Paso 12
%% Desplazamientos de los elementos en coordenadas Locales
% di=Ti'*Di;
    nn=length(Ki); [di]=zeros(nn,1);
    for i=1:n
        if i==1
            di(i:i+3,:)=Ti(i:i+3,i:i+3)*Di(i:i+3);
        else
            r=4*(i-1)+1;
            di(r:r+3,:)=Ti(r:r+3,r:r+3)*Di(r:r+3);
        end
    end
end

%% Paso 13
%% Acciones en los elementos(en coordenadas Locales y Globales)
% A2=K2*D2+AEP2; Globales, AEP2 en este caso no hay debido, que no hay acciones de
% empotramiento perfecto
% a1=k1*d1+aep1; A1=K1*D1+AEP1;

[Ai]=zeros(nn,1); [ai]=zeros(nn,1);

for i=1:n
    if i==1
        Ai(i:i+3,:)=Ki(i:i+3,i:i+3)*Di(i:i+3,:);
        ai(i:i+3,:)=ki(i:i+3,i:i+3)*di(i:i+3,:);
        ai_ord(i,1:2)=[i,ai(i,1)];
        di_ord(i,1:2)=[i,di(i+3,1)];
    else
        r=4*(i-1)+1;
        Ai(r:r+3,:)=Ki(r:r+3,r:r+3)*Di(r:r+3,:);
        ai(r:r+3,:)=ki(r:r+3,r:r+3)*di(r:r+3,:);
        ai_ord(i,1:2)=[i,ai(r,1)];
        di_ord(i,1:2)=[i,di(r+3,1)];
    end
end
end

```

```

%% Paso 14
%% Comprobación de equilibrio
% [PR]=Acum2(lee1,A1,PR);
PR=zeros(ngdl_rest,1);
for i=1:n
    if i==1
        [PR]=Acum2(leei(i,:),Ai(i:i+3,:),PR);
    else
        r=4*(i-1)+1;
        [PR]=Acum2(leei(i,:),Ai(r:r+3,:),PR);
    end
end
%% Paso 15
%% Chequeos
    esfuerzo=zeros(n,1);
    for i=1:n
        esfuerzo(i,1)=ai_ord(i,2)/Tabla0(i,1);
    end

%% Paso 16
% Función Objetivo está en función de la fuerza interna total de la estructura
    for i=1:n
        fx(i,1)=abs(ai_ord(i,2));
    end
    fxx(isAG)=sum(fx); %Fuerza interna Total Accumulate

%Coeficiente de violación (Ci)
% Esta en función del esfuerzo máximo del elemento de mayor sección
    Ci=zeros(n,1);
    for i=1:n
        Ci(i,1)=abs(ai_ord(i,2))/(fy*0.61*33.5*2.54^2))-1;
        if Ci(i,1)>0
            Ci(i,1)=abs(ai_ord(i,2))/(fy*0.61*33.5*2.54^2))-1;
        else
            Ci(i,1)=0;
        end
    end

end

%    C(is,1)=sum(Ci); %Penalización solo toma en cuenta los esfuerzos
    C(isAG,1)=sum(Ci); %Penalización toma en cuenta esfuerzos y deformaciones
máximas

%% Función objetivo penalizada fxp
%Según las pruebas realizadas por otros autores k=1000 cuando fxp=1/f
    K=1000; fxx(isAG,1)=sum(fx);

```

```

fxxx(isAG,1)=sum(fx);

fxp(isAG,1)=1/((fxxx(isAG,1)*(1+K*C(isAG,1)))); %Se invierte para poder realizar la
%minimización.

if contAG==1
    Cacum(1:length(Ci),isAG)=Ci;
    ai_acum(1:length(ai_ord),isAG)=ai_ord(:,2) ;
    ai_acumABS(1:length(ai_ord),isAG)=abs(ai_ord(:,2)) ;
    ico(isAG,1)=isAG;
else
    icont=NrAG*(contAG-1)+isAG;
    Cacum(1:length(Ci),icont)=Ci;
    ai_acum(1:length(ai_ord),icont)=ai_ord(:,2);
    ai_acumABS(1:length(ai_ord),icont)=abs(ai_ord(:,2));

    ico(icont,1)=icont;
end

end

Nr=NrAG;
Pob_2=Pob_2;
nele=4; %número de variables por estructura
% Selection
% nele=n;
%Metodo 1      Residuo +
[ dec ] = fselectionR(fxp,Nr);
% Metodo 2      Por Ruleta ++
[ decr] = fSelectionRul(fxp,Nr);
% Metodo 3      Por Torneo Mejor +++
nt=3; %Cantidad de individuos que entran a la competencia
[ dect] = fselectionTor(fxp,Nr,nt);
%Reproduction
[ Pob_C,Parents,Couples,Nr] = freproduction( Pob_2,Nr,nele,dect);
%LC=nele;%# of chromosomes
Pob_2= Pob_C;
% MutationPorc=1; % menor al 1 por ciento
Pob_2 = fMutation(Pob_2,MutationPorc);
Pob_10= fPobDec(Pob_2,nele,Nr);
[Pob_float] = fScale(Pob_10,I,nele,LC,Nr);
%Results mode
Pob_floatMode=zeros(nele,1);
for i=1:nele
    Pob_floatMode(i)=mode(Pob_float(i:nele:end));

```

```

end

Pob_floatModeAcc(:,contAG)=Pob_floatMode;
Pob(:,contAG)=Pob_float;
%%
fagxp=[fxx fxp C];
[fxpmin(contAG,2),id]=(max(fagxp(:,2))); %Acumula la función penalizada 1/f
fxpmin(contAG,1)=fagxp(id,1); %Acumula los pesos minimos por generación
fxpmin(contAG,3)=fagxp(id,3); %Acumula la función penalizada
Peso(:,contAG)=fxx;
Penal(:,contAG)=C;

ended1(1:4,contAG)=[Mcromosoma(Pob_floatModeAcc(1,contAG),1);Mcromosoma(Pob_floatModeAcc(2,contAG),2);Mcromosoma(Pob_floatModeAcc(3,contAG),3);...
                    Mcromosoma(Pob_floatModeAcc(4,contAG),4)];
% cont=cont+1;
end

Pob_floatAcum(1:4,ixx)=Pob_floatModeAcc(:,end);

AreaRes=[Mcromosoma(Pob_floatModeAcc(1,end),4*ones(1))];
AreaFIN(:,ixx)=AreaRes;
PesoFin(:,ixx)=fxpmin(end,1); PesoAcum(:,ixx)=fxpmin(:,1);
Fobjetivo(:,ixx)=fxpmin(end,2); FobjAcum(:,ixx)=fxpmin(:,2);
PenaFin(:,ixx)=fxpmin(end,3); PenaAcum(:,ixx)=fxpmin(:,3);
aifin(length(ai_ord),ixx)=ai_ord(length(ai_ord),2);

%% PobFin -->Representa el mejor individuo de cada generación ixx veces
% Esto quiere decir que son los mejores individuos de iteraciones de varias generaciones
%En total las iteraciones seria T=ixx*Ngeneraciones*Nr=10x10x10=1000iteraciones
PobFin(1:4,ixx)=Pob_floatModeAcc(:,end); %acumula las secciones del mejor individuo

%%
PesoFin22=PesoFin.*PenaFin; %Parámetro para considerar la penalización en el peso
mínimo
for i=1:ixx
    if PesoFin22(1,i)==0
        PesoFin2(1,i)=PesoFin(1,i);
    else
        PesoFin2(1,i)=1000*PesoFin(1,i);
    end
end

```

```

end
[aay,iay]=min(PesoFin2);
[aax,iax]=min(min(PesoFin2));
Place(1,1:2)=[iay(iax),iax];
% Pminf=(PesoFin((iay(iax)),iax));

iselect=Place(1,1); %put "iselect" of your needed, that depends of your iteration
ibody=Place(1,2); %put "ibody" of your individual

ended=[Mcromosoma(Pob_floatModeAcc(1,end),1);Mcromosoma(Pob_floatModeAcc(2,end)
,2);Mcromosoma(Pob_floatModeAcc(3,end),3);...
Mcromosoma(Pob_floatModeAcc(4,end),4)];
endedixx(:,ixx)=ended;
end

end

%%
subplot(3,1,1)
plot((1:NgeneracionesAG)',fxpmin(:,1))
xlabel('N° Generaciones')
ylabel('Peso (kg)')
title(' Peso(kg)vs N° Generaciones ')

subplot(3,1,2)
plot((1:NgeneracionesAG)',fxpmin(:,2))
xlabel('N° Generaciones')
ylabel('1/(fxx*(1+K*C))')
title('1/(fxx*(1+K*C)) vs N° Generaciones')

% subplot(3,1,3)
% plot((1:Ngeneraciones)',fxpmin(:,3))
% xlabel('N° Generaciones')
% ylabel('C')
% title('Penalización C vs N° Generaciones')

subplot(3,1,3)
plot((1:ikx(end,end))',sort(PesoFin(1,:)))
xlabel('N° Generaciones')
ylabel('Peso (kg)')
title('Peso vs N° Iteraciones')

toc

```

## A8.2. Optimización de forma, topología y secciones de estructura de luz de veinte metros, hiperestática de grado uno.

Se debe utilizar el programa del Anexo No8 A8.1, pero se debe realizar las siguientes modificaciones:

### ➤ RESTRICCIONES DE APOYOS.

```
%% Datos restricciones de los apoyos
% fix= restricciones en los apoyos [Fx Fy] 1 es restrccion y 0 es libre
fix=[1 1
     1 1];
% ap= número de apoyos
ap=length(fix(:,1));
nglr=sum(sum(fix)); %Numero de las restricciones de los apoyos
```

### ➤ GRADOS DE LIBERTAD.

```
%% Enumera los grados de libertad por cada nodo
% N_GDL=[ Nodo GDLi GDLf]
% N_GDL=[ Nodo GDLi GDLf]
N_GDL=fgradoslibertad1(MaxN,fix,ig);
N_GDLMATR=N_GDL;
for i=1:MaxN
    N_GDLMATR(i,2:3)=[2*i-1, 2*i];
end

%% N_GDLMATR(2,2)=4;N_GDLMATR(2,3)=3; %Para armaduras
%Isostáticas
```