

*Universidad de Matanzas “Camilo Cienfuegos”  
Facultad de Ciencias Técnicas*



## **“OPTIMIZACIÓN DE UNA TORRE AUTOSOPORTADA APLICANDO ALGORITMOS GENÉTICOS”**

**Trabajo de Diploma en Ingeniería civil**

**Autor: Michel Llerena Viera**

**Tutor(es): Ing. Juan Carlos Verdecia Somoano**

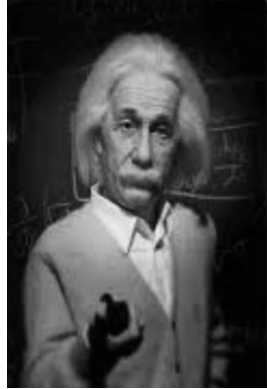
**Ing. Javier Alejandro García Miranda**

*Matanzas, 2019*

## PENSAMIENTO

*“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en lo bello y maravilloso mundo del saber.”*

Albert Einstein



## **DECLARACIÓN DE AUTORIDAD**

Por medio de la presente declare que soy el único autor de este trabajo de diploma y, en calidad de tal, autorizo a la Universidad de Matanzas a darle el uso que estime más conveniente.



## **DEDICATORIA**

*EN ESPECIAL QUISIERA DEDICARLO CON TODO EL CARIÑO DEL MUNDO A MI  
MAMA Y A MI PAPA POR SER TAN INCONDICIONAL CONMIGO POR SER LAS  
PERSONAS MÁS INFLUYENTES Y POR HABERME APOYADO Y HABER CREÍDO EN  
MÍ EN TODO MOMENTO EN LOS CINCO AÑOS.*

*SINCERAMENTE GRACIAS*

## AGRADECIMIENTOS

- Quisiera agradecer primero que todo a toda mi familia por haberme apoyado en todo y en especial a mi mamá y a mi papá.
- A mi tutor Juan Carlos por haberme aguantado tantos días en su casa haciendo posible este sueño que pensé que nunca lo terminaríamos y a toda su familia por haberme recibido y atendido en su casa.
- A mi otro tutor Javier por ayudarme también en el trabajo de diploma y atenderme sin peros.
- Al profesor Alejandro por haberme atendido las veces que me hizo falta y haberme ayudado.
- Al profesor Carlos García por recibirnos en su centro de trabajo y brindarnos una ayuda esencial para realizar esta investigación.
- Al profesor Quiza por todas las dudas que nos aclaró del tema.
- A todos los profesores del departamento de Construcciones por brindarme ayuda siempre que la necesité (Manuel, Liutmila, Adiaris, Alejandro, Carlos etc.).
- Y por último y no menos importante a nuestra piña de la universidad, mis amigos, porque son lo mejor que nos ha podido pasar en estos cinco años siempre juntos en todo lo trascendido (en estudio, en fiesta, en discusiones, en fin, en todo) siempre juntos (Yasmany, Neikys, el Gordo, Yanet, Hany y Colina) los quiero mucho en verdad.

## RESUMEN

La optimización del diseño de las torres autosoportadas de acero siempre ha sido una tarea difícil debido a un gran número de variables de diseño, estas torres son estructuras metálicas muy sensibles a los procesos de optimización debido a su compleja configuración. A pesar de los desarrollos teóricos significativos en el campo de la optimización estructural, así como la aparición de nuevas técnicas de optimización en las últimas décadas, la popularidad de la optimización estructural en la práctica de diseño de ingeniería es todavía muy limitada y escasa por lo que se tiende a realizar su diseño por la intuición del diseñador, no siendo así hoy en día donde a este campo se le ha sumado muchas técnicas de optimización aplicando métodos heurísticos ya que mediante su utilización se logran reducir costos y materiales, obteniendo una solución a un problema ingenieril cumpliendo con todas las restricciones y limitaciones impuestas. En esta investigación se pretende realizar el proceso de optimización del peso de una torre analizándola como armadura tridimensional utilizando técnicas heurísticas, para ello se realiza la parametrización de la torre definiendo las variables de diseño y restricciones de la misma. Para llevar a cabo dicho proceso se utiliza como herramienta computacional el software Matlab mediante una programación elaborada basada en el método de elementos finitos y empleando dichas técnicas de optimización. Posteriormente se realizará la comparación con el diseño un modelo de torre, pero realizando el predimensionamiento basado en la intuición del diseñador, para así demostrar la superioridad de la utilización de las técnicas heurísticas aplicadas a torres autosoportadas.

**Palabras claves:** optimización; métodos heurísticos; predimensionamiento; torres autosoportadas; armaduras tridimensionales.

## ABSTRACT

The optimization of the design of self-supporting steel towers has always been a difficult task due to a large number of design variables, these towers are metallic structures very sensitive to optimization processes due to their complex configuration. Despite the significant theoretical developments in the field of structural optimization, as well as the emergence of new optimization techniques in recent decades, the popularity of structural optimization in the practice of engineering design is still very limited and scarce. that tend to make its design by the intuition of the designer, not being so today where this field has been added many optimization techniques by applying heuristic methods because by using it they manage to reduce costs and materials, obtaining a solution to an engineering problem complying with all the restrictions and limitations imposed. This research intends to perform the process of optimizing the weight of a tower analyzing it as three-dimensional armor using heuristic techniques, for this the parameterization of the tower is done by defining the design variables and restrictions of it. To carry out this process, the *Matlab software* is used as a computational tool by means of an elaborated programming based on the finite element method and using these optimization techniques. Later, a tower model will be compared with the design, but with the predimensioning based on the intuition of the designer, in order to demonstrate the superiority of the use of the heuristic techniques applied to self-supporting towers.

**Keywords:** optimization; heuristic methods; predimensioning; self-supporting towers; three-dimensional armor



## TABLA DE CONTENIDO

Introducción .....	10
Capítulo 1 Análisis del estado del arte de las técnicas de optimización estructural y los métodos heurísticos de optimización en torres autoportadas .....	14
1.1 Tipos de optimización.....	14
1.2 Técnicas de optimización.....	16
1.2.1 Programación Matemática. ....	16
1.2.2 Programación Lineal. ....	17
1.2.3 Programación no lineal. ....	17
1.2.4 Programación cuadrática secuencial. ....	18
1.2.5 Programación lineal mixta (MILP). ....	18
1.3 Procesos Heurísticos y Metaheurísticos.....	19
1.4 Algoritmos Genéticos. ....	20
1.5 Optimización de torres empleando diferentes metodologías .....	23
1.5.1 Optimización de torres autoportadas empleando programación dinámica. .	23
1.5.2 Optimización de torres autoportadas mediante programación matemática no lineal.....	25
1.5.3 Optimización de torres autoportadas mediante programación lineal secuencial.....	25
1.5.4 Optimización de torres autoportadas empleando algoritmos genéticos. ....	26
1.5.5 Optimización de torres autoportadas mediante métodos de optimización propios de los autores.....	31
1.6 Softwares.....	37
Conclusiones parciales.....	38
Capítulo 2 Parámetros y procedimientos para la optimización .....	39
2.1 Características de la torre.....	39
2.2 Consideraciones para la modelación del diseño original.....	39
2.2.1 Cargas .....	41
2.3 Planteamiento del problema de optimización.....	41
2.3.1 Variables .....	42
2.3.2 Función objetivo .....	42
2.3.3 Parametrización de la torre según las variables .....	43
2.3.4 Consideración del modelo articulado.....	46
2.3.5 Restricciones .....	46
2.4 Parámetros de Algoritmos Genéticos.....	48
Conclusiones parciales.....	50
Capítulo 3 Análisis de los Resultados.....	51
3.1 Análisis de los resultados de la optimización al aplicar algoritmos genéticos. ....	51
3.2 Evolución de los procesos de optimización .....	52
3.3 Solución propuesta por el diseñador .....	53
3.4 Solución optimizada.....	54
Conclusiones parciales.....	55
Conclusiones .....	56
Recomendaciones .....	57
Referencias Bibliográficas .....	58

Anexos ..... 62

## INTRODUCCIÓN

Después de la Revolución Industrial, debido a la gran producción del acero en cantidades industriales se desarrolla el uso de estas torres. Las cuales tienen como objetivo soportar antenas a grandes alturas. Investigadores sobre el tema consideran que: “Las torres de celosía autoportadas son estructuras metálicas, delgadas, con características estructurales de bajo amortiguamiento y flexibles, lo que las hace realmente sensibles a la carga dinámica de viento o sismo”. (Martín and Elena, 2012).

Estas torres son estructuras comúnmente empleadas en la industria de las telecomunicaciones, aunque se tienen referencias de la existencia de estas desde tiempos muy antiguos, se consideran como estructuras relativamente jóvenes debido a que comienzan a utilizarse de forma habitual en el inicio del siglo XX con la expansión de la electrificación, el surgimiento de las grandes centrales generadoras de energía y el desarrollo explosivo de las telecomunicaciones.

Estas estructuras se caracterizan por ser altas y esbeltas. Otra característica específicamente de las torres de telecomunicaciones es lo referido a su ubicación que generalmente son sitios muy expuestos. En Cuba sus principales usos son como portadoras de antenas y como soportes de líneas de alta tensión por lo que las cargas predominantes sobre las torres de telecomunicaciones son las cargas ecológicas (viento, sismo etc.).

Son estructuras muy complejas y hay muchos desafíos para los ingenieros asociados a este campo. Pero la evolución de los métodos de análisis y de diseño, los avances en los estudios sobre las acciones de viento en las estructuras, así como el desarrollo y la disponibilidad de poderosas herramientas de cálculo como por ejemplo las computacionales y las estadísticas, comprometen a los ingenieros e investigadores de la rama, a llevar a cabo la revisión y el estudio del comportamiento de las torres bajo condiciones de trabajo extremas con el fin de contribuir a su perfeccionamiento.

Los procesos de diseños de una torre autosoportada están basados tradicionalmente en la experiencia del diseñador donde este predimensiona inicialmente cada uno de los elementos que la componen y se realiza una comprobación de la resistencia bajo las cargas, así de esta forma se repite este proceso hasta que obtenga un diseño que a juicio del diseñador es el más óptimo tanto en resistencia como en economía.

En nuestro país no está muy desarrollado el uso de las herramientas computacionales para el diseño no siendo así a nivel mundial donde se quiere incrementar el uso de esas técnicas cada día para realizar diseños más óptimos lo que por consiguiente reduce el costo de la estructura.

El desarrollo de las técnicas heurísticas a nivel mundial ha evolucionado constantemente donde los Algoritmos Genéticos juegan un papel importante en el mundo de la optimización tanto de armaduras planas como tridimensionales.

El Algoritmo Genético (AG) aplicado a estructuras metálicas permite alcanzar soluciones extremas que satisfagan criterios prescritos (cumplimiento de coeficientes de seguridad, pesos mínimos de elementos estructurales, etc.) realizándose de una forma racional sin acudir a una intuición o a las habilidades del diseñador.

### **Problema Científico**

¿Cuáles son los parámetros incidentes en el proceso de optimización estructural de una torre autosoportada utilizando algoritmos genéticos?

### **Hipótesis**

Una optimización del diseño estructural de la torre autosoportada utilizando algoritmos genéticos, permitirá determinar una variante más eficiente para el diseño de estas torres.

### **Objetivo General**

Optimizar el diseño de una torre autosoportada aplicando Algoritmos Genéticos.

## **Objetivos específicos**

- 1.-Definir el estado del conocimiento sobre la optimización estructural de torres autosoportadas.
2. Definir los métodos y parámetros para la optimización estructural de torres autosoportadas utilizando Algoritmos Genéticos según la bibliografía consultada.
3. Optimizar una torre autosoportada aplicando Algoritmos Genéticos.

En esta investigación se tiene como **objeto** la optimización estructural de torres autosoportadas.

Como **campo de acción** se tiene la optimización estructural de una torre autosoportada utilizando algoritmos genéticos.

El cumplimiento de los objetivos trazados conlleva a que se espere como resultado obtener un diseño estructural de una torre autosoportada, que sea racional y cumpla con la resistencia necesaria sin tener que acudir a la experiencia del diseñador, ahorrando además tiempo en la elaboración de los proyectos.

Esta investigación tendrá un **valor económico**, pues la optimización del diseño de los elementos estructurales traerá consigo un ahorro de las materias primas utilizadas en la elaboración de estos, además de la implicación con la conservación del medio ambiente que está implícita en el ahorro de los productos naturales.

Posee además un **valor metodológico**, pues muestra de forma clara el algoritmo y características fundamentales para el diseño y optimización de las torres autosoportadas.

También tendrá un **valor práctico**, pues el resultado final podrá ser utilizado por especialistas de entidades proyectistas en primera instancia.

Para poder cumplir con los objetivos trazados y poder lograr las metas se plantean como **tareas principales de la investigación:**

- » Actualización del estado del conocimiento sobre la optimización estructural de las torres autosoportadas
- » Descripción de la técnica de los Algoritmos Genéticos
- » Descripción de los parámetros utilizados para la optimización estructural
- » Aplicación de Algoritmos Genéticos a la torre autosoportada

Los **métodos de investigación** utilizados dependen del objeto de investigación y de sus objetivos. En esta se utilizaron del tipo teórico los siguientes:

**Análisis y síntesis:** permitirá como método, analizar la situación problemática y buscar la información necesaria para establecer el problema y la hipótesis de la investigación.

**Hipotético–deductivo:** Permitirá, a partir de la hipótesis asumida, elaborar conclusiones partiendo del análisis del conjunto de datos y de principios más generales, los resultados de la investigación.

# **CAPÍTULO 1 ANÁLISIS DEL ESTADO DEL ARTE DE LAS TÉCNICAS DE OPTIMIZACIÓN ESTRUCTURAL Y LOS MÉTODOS HEURÍSTICOS DE OPTIMIZACIÓN EN TORRES AUTOSOPORTADAS**

En los últimos años, la importancia de los aspectos económicos en el campo de las estructuras ha motivado a muchos investigadores a emplear nuevos métodos para minimizar el peso de éstas. El objetivo principal de la optimización estructural (diseño óptimo) es minimizar el peso de las estructuras al tiempo que se satisfacen todos los requerimientos impuestos por los códigos de diseño. " (Millán-Páramo, 2017)

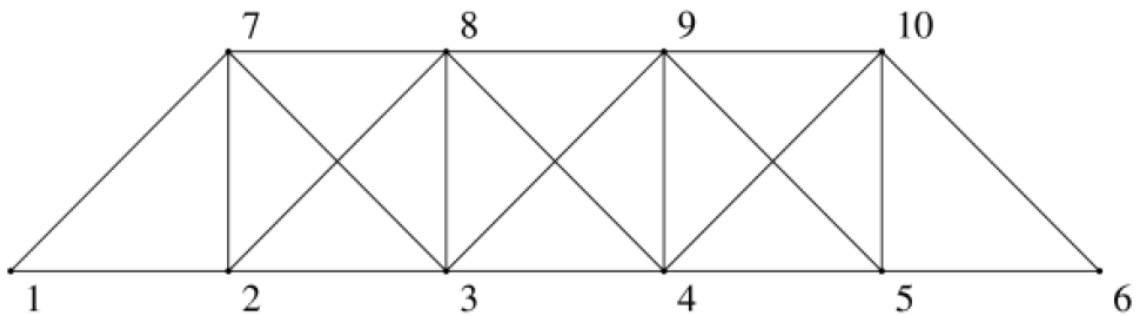
Las armaduras son tradicionalmente una de las estructuras más utilizadas en ingeniería, ya que cubren una amplia gama de soluciones para diferentes desafíos de ingeniería. También hay una gran cantidad de configuraciones de armaduras diferentes para cada propósito. Por lo tanto, la optimización del diseño o tamaño de las estructuras de armadura ha sido ampliamente estudiada por muchos autores, aplicando varios métodos diferentes, en un esfuerzo por diseñar la estructura óptima bajo ciertas condiciones.(Makris and Provatidis, 2002, Camp and Farshchin, 2014, Couceiro et al., 2016)

Las torres de acero autosoportadas son estructuras que, debido a su compleja configuración, son objeto de procesos de optimización de su diseño estructural. La mayoría de los trabajos de optimización estructural de torres autosoportadas tienen como objetivo la reducción del peso de la estructura. Para lograrlo existen diferentes parámetros y variables que pueden ser modificados por los diseñadores. El principal objetivo de este capítulo es caracterizar algunas de las técnicas de optimización utilizadas en el mundo, haciendo énfasis en aquellas que han sido empleadas en torres autosoportadas donde mediante una revisión bibliográfica por la literatura disponible sobre el tema, se definirán cuáles son los métodos y parámetros para la optimización estructural de torres autosoportadas.

## **1.1 Tipos de optimización**

### **Optimización del tamaño.**

La optimización del tamaño (ver figura 1.1), también se conoce como optimización paramétrica, en esta se tiene un diseño preestablecido y como objetivo se define encontrar las dimensiones óptimas de la estructura, como lo son secciones transversales, espesores de materiales, longitudes y cantidades de elementos, entre otros, de tal manera que se mejore el comportamiento del sistema, este tipo de optimización es la forma más simple de optimización estructural. (Sigmund, 2000)



*Figura 1.1 Optimización del tamaño. (Sanchez-Caballero et al., 2012)*

### **Optimización de material.**

Se tiene como objetivo encontrar una composición de material que optimice la estructuras, donde las variables de diseño pueden ser, la orientación de las fibras y el espesor de las capas individuales de diferentes materiales (Sigmund, 2000)

### **Optimización de la geometría.**

La optimización se logra cambiando la conectividad entre nodos (ver figura 1.2), manteniendo fijas las coordenadas nodales y las secciones rectas. (Sanchez-Caballero et al., 2012)



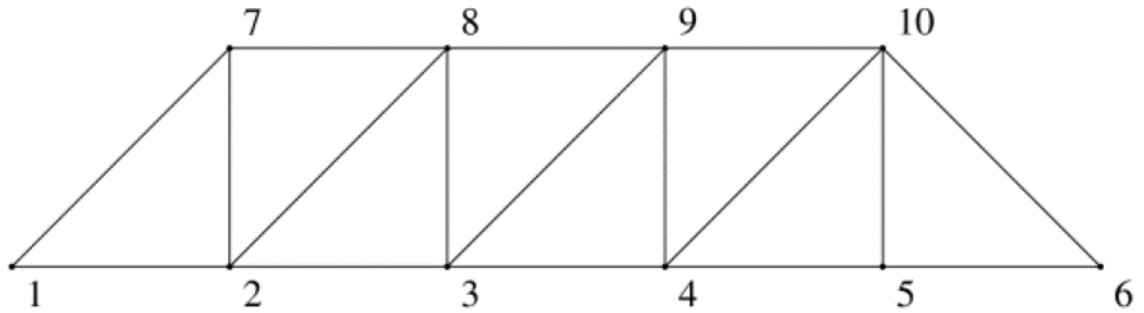


Figura 1.2. Optimización de la geometría. (Sanchez-Caballero et al., 2012)

### Optimización topológica.

La optimización se logra cambiando las coordenadas y conectividad nodales (ver figura 1.3), manteniendo fijas las secciones rectas.

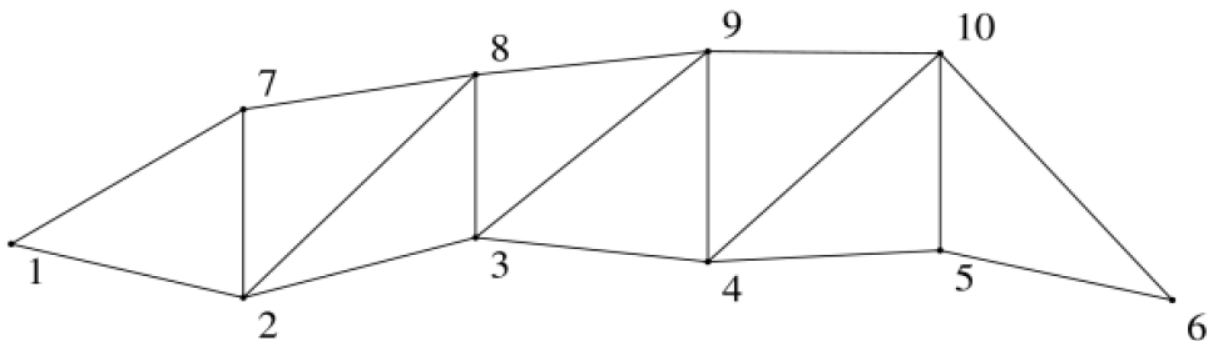


Figura 1.3. Optimización topológica (Sanchez-Caballero et al., 2012)

## 1.2 Técnicas de optimización

### 1.2.1 Programación Matemática.

Los problemas de optimización se pueden, en primer lugar, clasificar en problemas que involucran solamente variables continuas y en problemas con variables continuas y discretas (o solamente discretas). Los principales problemas de optimización con variables continuas incluyen la Programación Lineal (LP) y la Programación no lineal (NLP). Una importante subclase de la programación lineal es el problema lineal complementario (LCP)

y subproblemas importantes de programación no lineal incluyen la programación cuadrática (QP). En el caso de programación no lineal hay dos distinciones importantes a tener en cuenta. La primera es si el problema no lineal es convexo y la segunda es si el problema es diferenciable. Si el problema incluye variables discretas se puede nuevamente distinguir entre problemas lineales con variables enteras (MILP) y problemas no lineales con variables enteras (MINLP) ambas siglas del inglés *Mixed Integer (non) Linear Programming*. (Caballero and Grossman, 2007)

### 1.2.2 Programación Lineal.

Si tanto la función objetivo y las restricciones son lineales, el problema de optimización toma la siguiente forma:

$$\begin{aligned}
 \min : z &= c^T x \\
 \text{s.a. } Ax &\leq b \\
 x &\geq 0
 \end{aligned} \tag{1.2}$$

El método estándar para resolver un problema lineal es el método simplex, desarrollado por Dantzing en los años cuarenta. Véase (Caballero and Grossmann, 2007)

### 1.2.3 Programación no lineal.

La programación no lineal es un tipo de programación matemática en la que el problema de optimización toma la siguiente forma:

$$\begin{aligned}
 \min : f(x) \\
 \text{s.a. } h(x) &= 0 \\
 g(x) &\leq 0
 \end{aligned} \tag{1.3}$$

Dentro de los algoritmos para resolver NLPs aquellos que requieren un menor número de evaluaciones de la función objetivo son los basados en la programación cuadrática sucesiva (SQP –*successive quadratic programming*–) que ha dado lugar a una serie de algoritmos

que se adaptan muy bien a un gran número de problemas con diferentes estructuras. (Caballero and Grossman, 2007)

#### **1.2.4 Programación cuadrática secuencial.**

La programación cuadrática secuencial (*Sequential Quadratic Programming- SQP*) consiste en la aproximación secuencial del problema de programación no lineal como un problema de programación cuadrática. Al contrario de otros métodos que intentan convertir el problema en una sucesión de subproblemas de optimización sin restricciones, el SQP intenta resolver el problema de optimización iterativamente, donde el objetivo ( $f(x)$ ) es sustituido por una aproximación cuadrática y las restricciones no lineales ( $h(x)$  y  $g(x)$ ) son sustituidas por aproximaciones lineales. (Teles and Gomes, 2010)

#### **1.2.5 Programación lineal mixta (MILP).**

Los métodos para resolver MILPs están fundamentalmente basados en los métodos de ramificación y acotamiento (BB del inglés *Branch and Bound*) y sus variantes, donde cada subproblema lineal se resuelve utilizando el método simplex. Este método consiste en una enumeración en árbol en el cual el espacio de variables enteras se divide de forma sucesiva dando lugar a subproblemas lineales que se resuelven en cada nodo del árbol. (Caballero and Grossmann, 2007)

Aunque en un principio la Programación Lineal fue la técnica más extendida, ha sido relegada debido a sus inconvenientes:

- No puede manejar variables de diseño discretas y continuas simultáneamente.
- No puede manejar restricciones no lineales.

Es por esto por lo que otro conjunto de algoritmos metaheurísticos, basada fundamentalmente en Computación Evolutiva, se ha desarrollado durante los últimos veinte años. (Sanchez-Caballero et al., 2012)

### 1.3 Procesos Heurísticos y Metaheurísticos

La optimización estructural y particularmente la optimización de la estructura de la torre son problemas complejos y no lineales que involucran varias restricciones, incluyendo pandeo, mínimo y máxima tensión, y restricciones geométricas. Debido a limitaciones de fabricación, las variables del problema deben ser considerada de tipo discreto o mixto. Por lo tanto, los algoritmos de búsqueda local son a menudo insuficientes y los algoritmos de optimización global más complejos son necesarios. Debido a la compleja interacción entre variables de los problemas, puede ser muy difícil utilizar técnicas de optimización basados en derivadas, lo que significa que los profesionales a menudo tienen que confiar en algoritmos metaheurísticos. (Gandomi and Goldman, 2018)

Los algoritmos metaheurísticos son algoritmos de optimización global que a menudo se inspiran en la naturaleza (por ejemplo, el algoritmo de hato de kril bioinspirado (Gandomi and Alavi, 2012) y simulado inspirado en la física recocido (Kirkpatrick et al., 1987), y por lo tanto, también se les llama algoritmo inspirado en la naturaleza. Estos algoritmos exploran y explotan simultáneamente o, en otras palabras, puede realizar una búsqueda global y local al mismo tiempo. Aunque el algoritmo inspirado en la naturaleza no puede ser atrapado fácilmente en los mínimos locales, no hay garantía en ningún algoritmo de optimización para encontrar el óptimo global. Por lo tanto, encontrar el algoritmo de optimización global más adecuado es desafío, particularmente para problemas a gran escala y complejos.

Los problemas altamente complejos son difíciles de resolver y se les denomina Problemas NP, se clasifican en conjuntos o clases de complejidad (L, NL, P, P Completo, NP, NP-Completo, NP Duro...). La mayor parte de los problemas en teoría de la complejidad tienen que ver con los problemas de la clase NP de decisión, y corresponden a problemas a los cuales no se puede garantizar encontrar una mejor solución en un tiempo razonable. Es aquí donde los métodos heurísticos se convierten en procedimientos eficientes para encontrar “buenas soluciones” aunque no se pueda comprobar que sean óptimas. En estos métodos, la rapidez del proceso (que siempre es menor que el tiempo de una solución por otros métodos) es tan importante como la calidad de la solución obtenida. (Suarez, 2013)

#### **1.4 Algoritmos Genéticos.**

Los algoritmos evolutivos (EA) son técnicas de búsqueda inspiradas en la evolución biológica de las especies y aparecieron a fines de la década de 1950 (Fraser, 1957). Entre varios enfoques (Fogel et al., 1966, Holland, 1962, Rechenberg, 1965), los Algoritmos Genéticos (AG) constituyen ciertamente el ejemplo más conocido, después de la publicación en 1989 del conocido libro de Goldberg (Goldberg and Holland, 1988).

Los métodos evolutivos inicialmente despertaron un interés limitado, debido a su significativo tiempo de ejecución. Pero, en los últimos diez años, han experimentado un desarrollo considerable, que puede atribuirse al aumento significativo en la potencia de cálculo de las computadoras, en particular, después de la aparición de arquitecturas masivamente paralelas, que explotan el "paralelismo intrínseco". El principio de un algoritmo evolutivo simple se puede describir como sigue: un conjunto de  $N$  puntos en un espacio de búsqueda, seleccionado a priori en el azar, constituye la población inicial; Cada individuo  $x$  de la población tiene un determinado rendimiento, que mide su grado de adaptación al objetivo que se persigue. En el caso de la minimización de una función objetivo  $f$ ,  $x$  se vuelve más poderosa a medida que  $f(x)$  se vuelve más pequeña (Siarry, 2016).

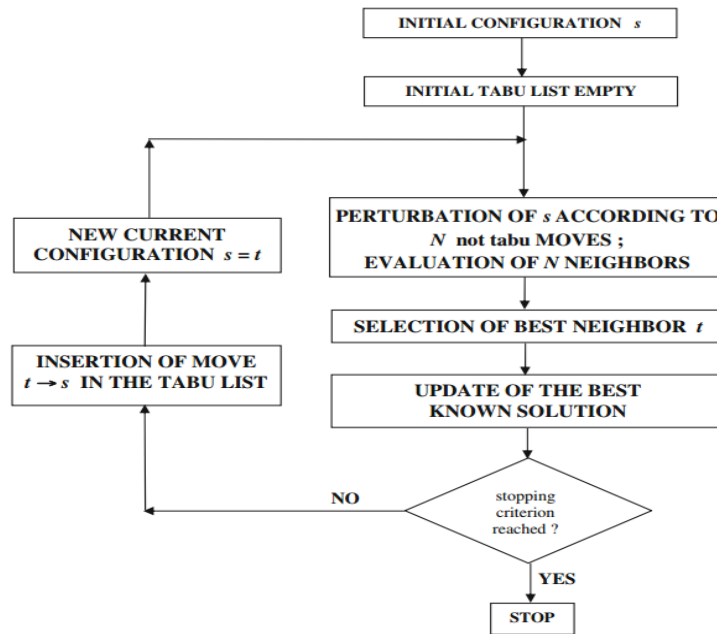
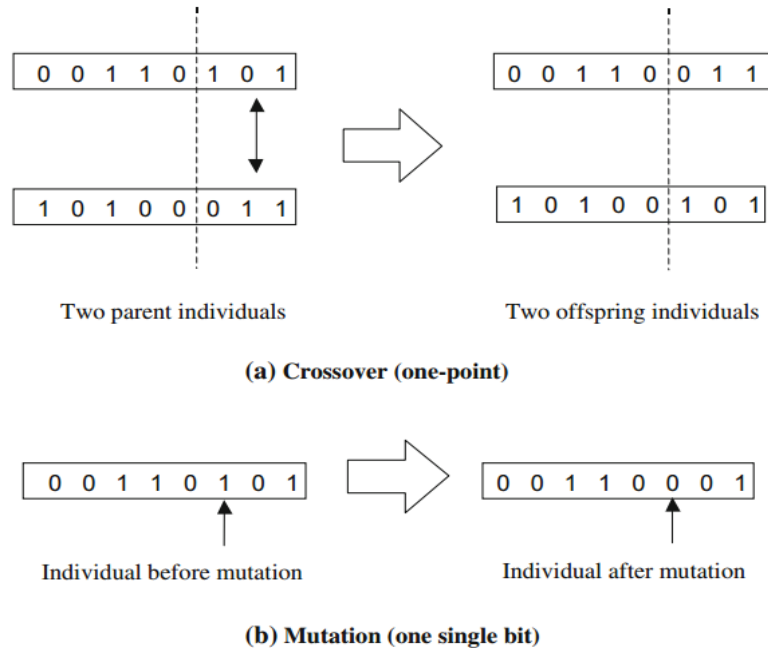


Figura 1.4 Diagrama de flujo del algoritmo simple(Siarry, 2016)

Siarry (2016) plantea que un EA consiste en evolucionar gradualmente, en sucesivas generaciones, la composición de la población, con su tamaño se mantiene constante. Durante generaciones, el objetivo es mejorar el rendimiento general de los individuos. Uno trata de obtener tal resultado imitando los dos mecanismos principales que gobiernan la evolución de los seres vivos según la teoría de Darwin:

- *selección* Lo que favorece la reproducción y supervivencia de los individuos más aptos
- *reproducción* Lo que permite la mezcla, recombinación y variación de las características hereditarias de los padres, para formar descendientes con nuevas potencialidades

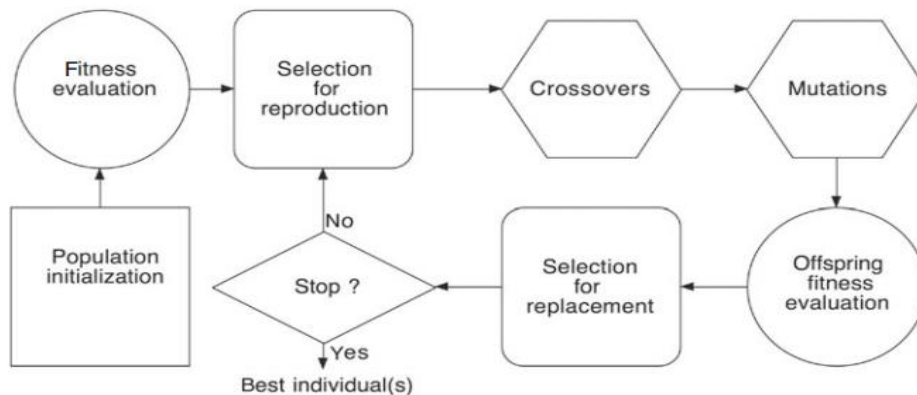


*Figura 1.5 Ejemplos de operadores de cruce y mutación, en el caso de individuos representados por cadenas binarias de ocho números(Siarry, 2016)*

En la práctica, se debe seleccionar una representación para los individuos de una población de forma clásica, un individuo puede ser una lista de enteros para un problema combinatorio, un vector de números reales para un problema numérico en un espacio continuo o una cadena de números binarios; incluso se pueden combinar estas representaciones en estructuras complejas si la necesidad se siente así. El paso de una generación a la siguiente se desarrolla en cuatro fases: una fase de selección, una fase de reproducción (o variación), una fase de evaluación de desempeño y una fase de reemplazo.

La fase de selección designa a los individuos que participan en la reproducción. Están presentes, posiblemente en varias ocasiones, a priori más a menudo los poderosos, son los individuos seleccionados que están disponibles para la fase de reproducción. Esta fase consiste en aplicar operadores de variación a las copias de los individuos seleccionados para generar nuevos individuos; los operadores más a menudo son crossover (o recombinación), que produce uno o dos descendientes a partir de dos padres, y mutación, que produce un nuevo individuo a partir de un solo individuo (ver Fig.1.8 para un ejemplo).

La estructura de los operadores de variación depende en gran medida de la representación seleccionada para los individuos. Las actuaciones de los nuevos individuos se evalúan durante la fase de evaluación, a partir de los objetivos especificados. Por último, la fase de reemplazo consiste en elegir a los miembros de la nueva generación: uno puede, por ejemplo, reemplazar a los individuos menos poderosos de la población por los mejores individuos producidos, en igual número. El algoritmo se termina después de un cierto número de generaciones, de acuerdo con un criterio de terminación que debe especificarse. La Figura 1.6 representa el principio de un algoritmo evolutivo



*Figura 1.6 Principio de un algoritmo evolutivo(Siarry, 2016).*

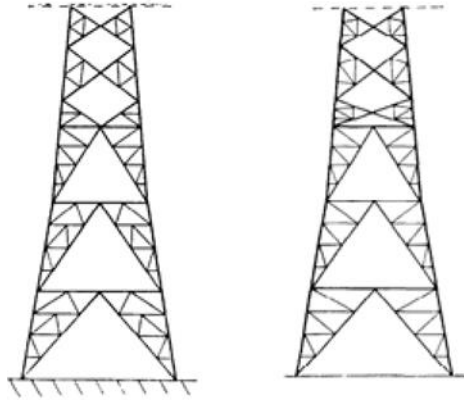
Debido a que manejan una población de instancias de soluciones, los algoritmos evolutivos son particularmente adecuados para encontrar un conjunto de soluciones diferentes cuando una función objetivo tiene varios óptimos globales. En este caso, pueden proporcionar una muestra de soluciones de compromiso cuando uno está resolviendo problemas con varios objetivos, posiblemente contradictorio.

## **1.5 Optimización de torres empleando diferentes metodologías**

### **1.5.1 Optimización de torres autosoportadas empleando programación dinámica.**

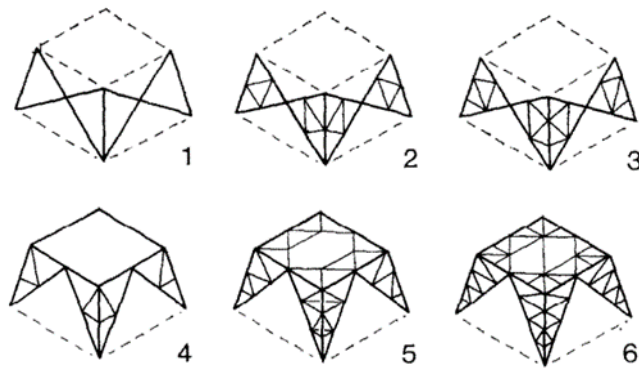
Sheppard and Palmer (1972) se plantearon la minimización del peso de la parte inferior de la estructura (la situada entre los apoyos y la primera cruceta). (Figura 1.7)





*Figura 1.7. Parte inferior de la torre objeto de optimización (izquierda), Estructura optimizada (derecha). (Sheppard and Palmer, 1972)*

Para ello fue dividida en varios segmentos (figura 1.8). El sistema de optimización tenía que determinar la altura y el diseño estructural de cada segmento (debía escoger entre seis tipos de módulos predefinidos) sin modificar la altura total y la forma externa de la torre, y dimensionar correctamente todas las barras. El problema de optimización fue resuelto mediante un algoritmo de programación dinámica, lo que exige desacoplar el problema completo en varias etapas (el diseño de cada segmento) que se habrán de resolver secuencialmente. Además, no se consideró el peso propio de la estructura y sólo se tuvo en cuenta la acción del viento de una forma muy simplificada.



*Figura 1.8. División en segmentos de la torre. (Sheppard and Palmer, 1972)*

Raj and Durrant (1976) plantearon la minimización del peso en toda la torre. Para ello dividieron la estructura en cuatro zonas. El sistema de optimización determinaba algunos parámetros geométricos y el diseño estructural de cada zona (para lo que cuenta también

con varios tipos de módulos predefinidos), y dimensionaba correctamente todas las barras. El problema de optimización se resolvía mediante un algoritmo de programación dinámica. El cálculo estructural fue realizado separadamente para cada una de las cuatro caras de la estructura, que se analizan como estructuras articuladas planas independientes.

### **1.5.2 Optimización de torres autoportadas mediante programación matemática no lineal.**

Hanssen (1981) planteó la minimización del coste (del acero estructural) de la torre. La torre se consideraba formada por varias subestructuras. El sistema de optimización debía determinar algunas variables que controlan la geometría de la torre y dimensionar correctamente todas las barras. Imponía restricciones (sencillas) sobre las variables de tipo geométrico. La función de coste se consideraba discontinua, ya que las secciones de las barras tienen que elegirse entre un cierto número de perfiles normalizados de acero laminado. El problema de optimización fue resuelto mediante varios métodos de programación matemática no lineal. Se asume que la estructura de la torre es débilmente hiperestática, por lo que los esfuerzos no dependen sensiblemente de las secciones de las barras, que se dimensionan a máxima tensión. En el cálculo estructural se tienen en cuenta los efectos no lineales debidos a la holgura existente en el atornillado.

### **1.5.3 Optimización de torres autoportadas mediante programación lineal secuencial.**

Majid and Tang (1984) analizaron la optimización de estructuras articuladas espaciales y aplicaron sus técnicas a la optimización de una torre de alta tensión de diseño muy simplificado en 1984, en el que plantearon la minimización del peso de la torre. Las variables de diseño consideradas fueron las secciones de los grupos de barras y la variación de las coordenadas horizontales de los nodos de la estructura. Se imponen restricciones en las barras para limitar la tensión a tracción y para evitar el pandeo local, y se restringen también los desplazamientos horizontales de los nodos. El problema de optimización fue resuelto mediante un algoritmo de programación lineal secuencial. Para el cálculo estructural consideraron el peso de los cables y la acción del viento.

#### **1.5.4 Optimización de torres autosoportadas empleando algoritmos genéticos.**

Adeli and Cheng (1994) Presentó un método de resolución de optimizaciones de grandes estructuras realizando simultáneamente la evaluación de la función de aptitud de cada cadena de cromosomas en la misma iteración. De la misma manera, la mutación de cada cadena y la operación de cruzamiento para cada par de cadenas (Adeli and Cheng, 1994) y lo aplicaron para la optimización de una torre autosoportada de 942 elementos y 242 nodos.

La primera estrategia consiste en paralelizar el algoritmo entre las cadenas de enteros de la población, es decir, asignar todas las evaluaciones de la función de aptitud y las operaciones de cruzamiento y mutación al número disponible de procesadores por igual. Desde que todos los cálculos de cada cadena son independientes, el algoritmo puede ser completamente paralelizado. Cada procesador procesa el mismo número de cadenas para lograr un mejor balance de cargas entre los procesadores.

La segunda estrategia consiste en dividir la población inicial en un número de subpoblaciones igual al número de procesadores. Empleando el algoritmo genético lagrangiano aumentado, cada procesador realiza la optimización de cada subpoblación independientemente. Cada subpoblación ha sido sometida a coeficientes de penalización y esos coeficientes han sido actualizados por cada procesador independientemente de otros procesadores. Después de un limitado número de iteraciones o el criterio de parada es reunido por todos los procesadores, las subpoblaciones se combinan. Entonces otro criterio de parada es chequeado para toda la población. Si este criterio reúne la solución óptima el proceso se detiene.

Los autores aplicaron su metodología a una torre de 942 elementos y 244 nodos mostrada en la figura 1.9.

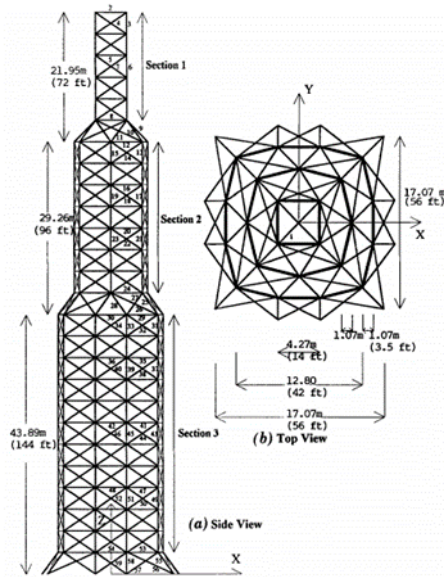


Figura 1.9: Torre autosoportada 26-story. (Adeli, 1994)

Sivakumar et al. (2004) analizaron la optimización orientada a objeto de una torre autosoportada aplicando algoritmos genéticos aplicando una metodología en la que la torre era dividida en objetos menores (figura 1.10) y compararon los resultados con la optimización de la misma torre, pero analizándola como un único objeto y llegaron a la conclusión que se obtienen mejores resultados que realizando el análisis de la torre orientada a objeto que como uno solo. La función objetivo es el peso de la estructura y las áreas de las secciones transversales de los miembros son tomados como variables discretas y son seleccionadas del catálogo de perfiles disponibles en India.

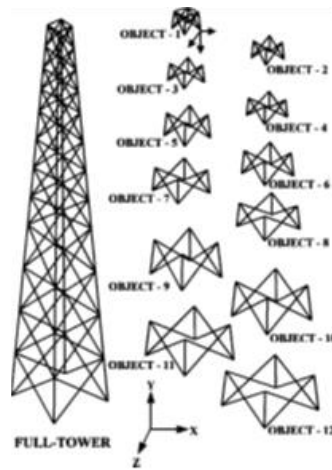


Figura 1.10 Torre completa y paneles de objetos. (Sivakumar et al., 2004)

Mohammed et al. (2012) Aplicaron algoritmos genéticos basados en FORTRAN a la optimización de torres de transmisión de electricidad utilizando distintas variables de diseño y luego compararon los resultados. En los cálculos consideraron tres sistemas de cargas aplicados a los nodos. La función objetivo a optimizar es el peso de la torre y utilizaron como variable la distribución de diferentes secciones transversales, analizando la torre como una armadura plana. (figura 1.11)

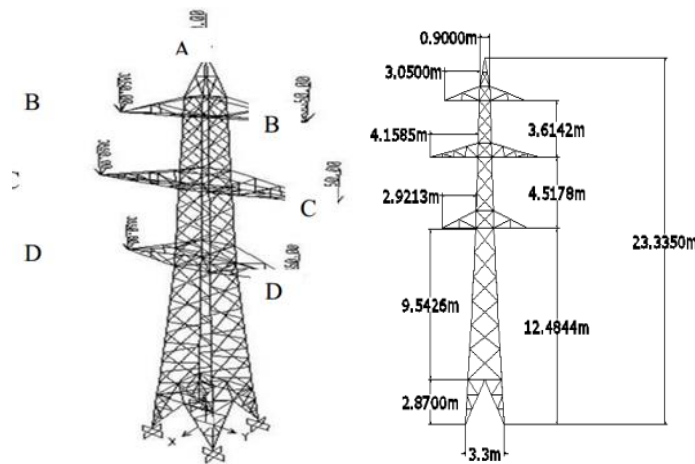
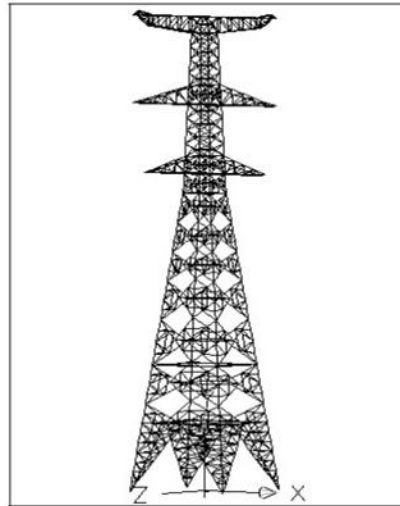


Figura 1.11 Torre. (Mohammed et al., 2012)

Chunming et al. (2012) Optimizaron una torre de alta tensión de la tipología 5D-SZC3 (figura 1.12) empleando un algoritmo genético para optimización multiobjetivo denominado *non-dominated sorting genetic algorithm* (NSGA-II).



*Figura 1.12. Torre tipología 5D-SZC3.(Chunming et al., 2012)*

Las funciones objetivo definidas por los autores fueron el costo de la torre y el desplazamiento en el eje X de las uniones con tierra de la torre. Los autores sometieron a la optimización solamente los miembros principales de la torre e ignoraron a los miembros diagonales y los miembros auxiliares. Los miembros principales fueron divididos en 28 grupos definidos por los requerimientos de diseño. Dos variables de sección transversal fueron consideradas y dos variables de tipo de material. Para el algoritmo genético fueron definidas una población de 100 y una generación límite de 500. Los autores concluyen que el uso de dos tipos de acero para la optimización de la torre puede reducir el costo entre un 4% y un 16% y que el NSGA-II puede proveer de una variedad de opciones para determinados parámetros de diseño de torres de transmisión. Acorde con las diferentes necesidades los diseñadores pueden elegir una opción apropiada.

Magalhães, et al. (2014) Optimizaron el diseño de una torre autoportada de un aerogenerador utilizando un algoritmo genético (figura 1.13). La función objetivo empleada fue el costo de la estructura y las variables de diseño consideradas fueron los costos de los materiales de la torre ( $C_m$ ); costos de los materiales de los nodos ( $C_n$ ), ensamblaje y mano de obra ( $C_b$ ); los costos de los soportes a tierra ( $C_t$ ); costos de manufactura de las diferentes secciones de la torre ( $C_{nm}$ ); el número de barras que llegan

a un nodo (costo de conectar una barra en un nodo); el ángulo de incidencia de una barra en el nodo ( $C\theta$ )

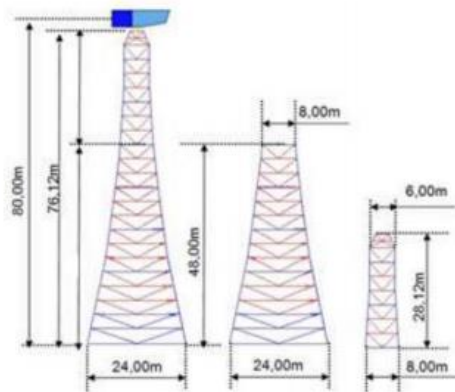


Figura 1.13. Representación de la torre de un aerogenerador. (Magalhães, et al. (2014))

Los autores aplicaron restricciones a las tensiones máximas y mínimas de las barras de la estructura, a los desplazamientos permisibles y a la frecuencia, a la que se le adicionó el costo para elegir a la solución óptima.

$$C = C_m + C_n + C_b + C_t + C_{nm} + C_\phi$$

Sony y Airin (2016) Utilizaron algoritmos genéticos para la optimización de una torre de transmisión de electricidad (figura 2.31) integrando los programas ANSYS y MATLAB.

Los autores optimizan una torre de 132 kV de doble circuito cuyas dimensiones son:

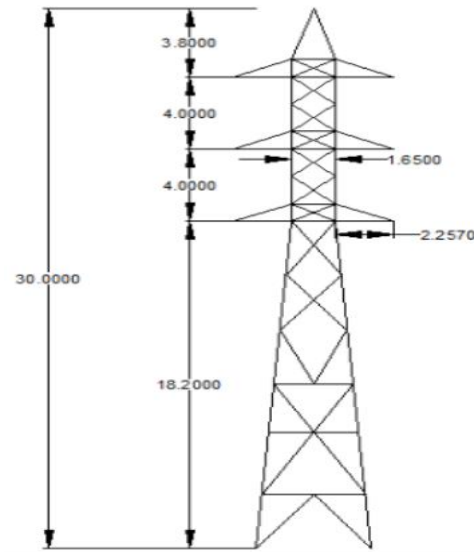
Altura total: 30 m

Cuerpo de la torre: 18.2 m

Altura de la jaula: 8.95m

Armadura de las cruces: 2.527m

Altura de la cresta 2.85 m



*Figura 1.14 torre de transmisión (Sony y Airin (2016))*

El problema de optimización es formulado como la minimización del peso de la torre. El objetivo de la optimización es la masa total bajo la restricción de la máxima tensión permisible.

La altura total de la torre, el ancho de la base y la configuración global de la torre son consideradas fijas. Los parámetros de optimización son el ancho y el espesor de las secciones angulares usadas en cada miembro. Los miembros se agrupan en 12 grupos.

### **1.5.5 Optimización de torres autoportadas mediante métodos de optimización propios de los autores.**

(Saka, 1984) Analizó la minimización del peso de una torre de alta tensión con una sola cruceta. Consideró como variables de diseño en su estudio las secciones de los veintiséis grupos de barras que componían la estructura. Las secciones de las barras son perfiles en L (angulares) de acero laminado y supone que la gama de perfiles disponibles es continua (y no discreta, como sucede en realidad). Las restricciones de las barras son impuestas para limitar las tensiones a tracción y para evitar el pandeo local de las barras. Además, impone restricciones a los desplazamientos de los nodos.



Para resolver el problema de optimización propone un método desarrollado por el autor. Este método se basa en un criterio de optimización (derivado del Lagrangiano del problema de optimización). Este planteamiento sólo se aplicó a torres de alta tensión isostáticas, porque el proceso iterativo correspondiente al criterio de optimización sólo es eficaz en la práctica cuando se aplica a este tipo de estructuras.

Felix and Vanderplaats (1987) Desarrollaron un método general para la optimización de estructuras articuladas espaciales. La función objetivo es la minimización del peso de la estructura. Consideraron como variables de diseño las secciones de las barras y la posición de los nodos e impusieron restricciones para limitar la tensión a tracción y para evitar el pandeo local, además de restringir también los desplazamientos de los nodos y el valor mínimo de la primera frecuencia fundamental de vibración de la estructura. Esta metodología fue aplicada al caso de una torre de alta tensión de una sola cruceta. El problema de optimización fue resuelto mediante un algoritmo que realiza dos fases en cada iteración: en primer lugar, se linealizan las restricciones potencialmente críticas (se ignoran las restantes) en función de las inversas de las áreas de las barras y se utiliza esta información para dimensionar las secciones; en segundo lugar, se linealiza la función objetivo y las restricciones en función de las coordenadas nodales y se utiliza esta información para mejorar la geometría de la torre. En el cálculo estructural no se tiene en cuenta el peso propio de la estructura.

Kocer and Arora (2002a) Analizaron la optimización de una torre de transmisión de energía sometida a los efectos de cargas sísmicas, además de las cargas normales. Optimizan las secciones transversales de los miembros de la torre con forma de perfiles angulares (L), no optimizan la forma. Las variables de diseño son tomadas de las secciones comerciales de un catálogo.

La torre optimizada es una torre de transmisión de 110 kV de 316 miembros, 111 nodos y 666 grados de libertad. Para la realización del diseño dividieron las secciones en nueve tipos de acuerdo con sus longitudes (figura 1.6).

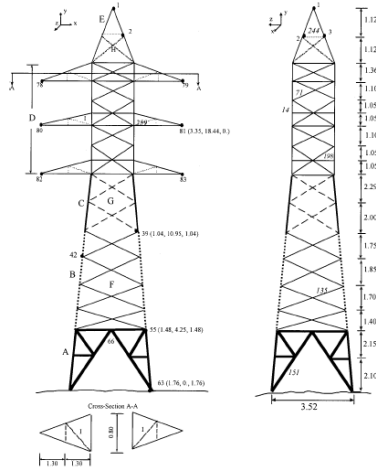


Figura 1.15 variables de diseño. (Kocer and Arora, 2002b)

El problema de optimización es formulado de dos maneras, la primera es una asignación simple de variables donde cada sección comercial es una variable de diseño de una lista disponible para el grupo. Esta formulación presenta nueve variables de diseño.

La segunda formulación emplea las dimensiones de cada sección como variables de diseño, longitudes de las alas del perfil angular y el espesor para un total de 27 variables de diseño continuas. La función objetivo empleada fue el peso de la estructura

Los autores aplicaron restricciones a los esfuerzos axiales de tracción y compresión impuestas por el manual (ANSI/ASCE 1992), además aplicaron restricciones al mínimo espesor de las alas de 3mm y al radio de giro de  $KL/r \leq 150$  para los miembros de las piernas de la torre y de  $KL/r \leq 200$  para otros miembros.

El problema de optimización es resuelto por los autores mediante dos métodos de programación matemática, un método de optimización de dos fases y el método *adaptive discrete assignment method* (ADAM)

El método de dos fases:

En la primera fase el método resuelve el problema formulado utilizando un algoritmo de optimización de variables continuas SQP, asignando tres variables de diseño a cada perfil, pero la solución óptima no está disponible en perfiles comerciales. En la segunda fase el

método usa algoritmos genéticos y variables discretas para encontrar la sección comercial óptima.

El método ADAM:

El método ADAM resuelve el problema de optimización en múltiples fases utilizando algoritmos de variables continuas. En la primera fase las secciones de la solución son comparadas con las secciones comerciales disponibles, si cualquier sección se acerca a una sección comercial, entonces es fijada.

La sección fijada es excluida y el problema es optimizado de nuevo y las secciones de la solución continua son comparadas de nuevo con las secciones comerciales. Este proceso se repite hasta que a todas las variables de diseño se les asigna una sección comercial.

También el problema es resuelto empleando algoritmos genéticos.

La solución con GA obtuvo un total de 0.359 m<sup>3</sup> de acero, la obtenida por el método de dos fases 0.375m<sup>3</sup> y la obtenida por el método ADAM 0.390m<sup>3</sup>.

El GA analizó 27,7x10<sup>6</sup> diseños por 6,18x10<sup>4</sup> el método de dos fases, pero empleó más tiempo computacional 146,606 s contra 120,710 s por lo que los autores llegan a la conclusión de que el método de GA obtiene mejores resultados, pero requiere muchos recursos informáticos.

Navarrina et al. (2005) Realizaron la optimización del diseño de una torre de la tipología FECSA/GL-110KV. Las torres de dicha tipología se dividen en bloques estructurales como muestra la figura (1.7)

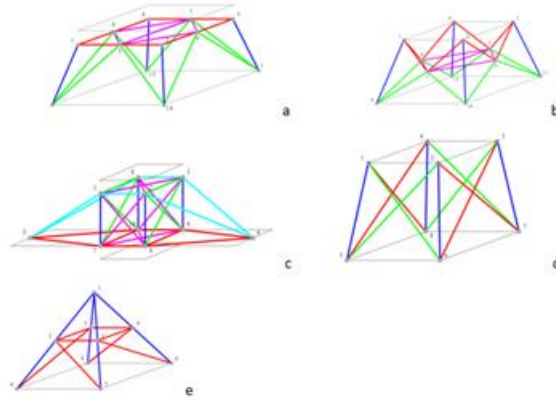


Figura 1.16 Bloques de la torre (Navarrina et al., 2005)

Además de barras de arriostre para prevenir el pandeo local que el autor no toma en cuenta por considerar su aporte estructural insignificante. Las secciones transversales de cada bloque son cuadradas y la base de un bloque inferior coincide con la cara superior del bloque inferior.

En el análisis estructural consideraron la estructura como una armadura articulada tridimensional siendo la función objetivo el peso de la estructura. Los autores toman como variables de diseño la longitud del lado de la base del bloque tipo  $k$  y del lado inferior del elemento tipo cabezal, manteniendo la altura de los elementos que componen la altura de la torre invariables y la sección de las barras. Además, incluyen algunas restricciones adicionales para evitar resultados sin sentido. El problema de optimización de los variables que definen las formas es continuo y el de selección de la sección de las barras es discreto. Para resolver el problema de optimización continua, utilizan un algoritmo de programación lineal secuencial con búsqueda unidireccional cuadrática (SLP-QLS o *Sequential Linear Programming with Quadratic Line Search*).

Beck and Verzenhassi (2008) Trataron la optimización del riesgo de diseño de una torre de telecomunicaciones sujeta a los efectos de un tornado. En este trabajo los autores se refirieron a la optimización en función del costo de la estructura en el que incluyeron:

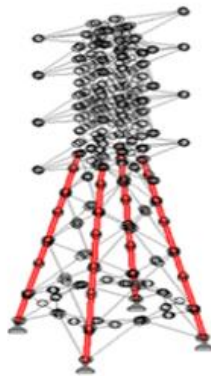
- Costo inicial de construcción
- Costo de operación
- Costos de inspección y mantenimiento

- Costo estimado si ocurre la falla

Para resolver el problema los autores emplean un programa computacional que está dividido en tres módulos independientes: un módulo de optimización, un módulo de fiabilidad estructural y un programa comercial de elementos finitos. La parte mecánica del problema es solucionada por el programa comercial de elementos finitos ANSYS, el módulo de fiabilidad estructural (StRAnD) fue desarrollado por el departamento de ingeniería de la universidad de Sao Paulo y fue escrito en FORTRAN. Para la optimización emplea interpolación cuadrática para la búsqueda lineal.

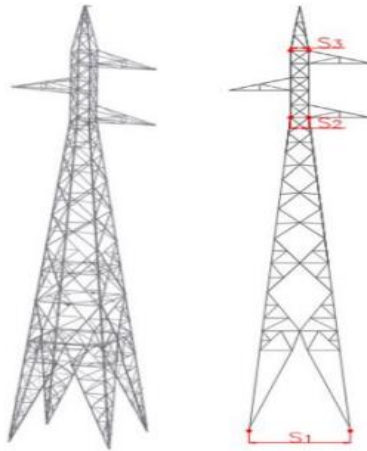
Muda et al. (2013) analizaron el diseño estructural sostenible de torres de transmisión basado en un algoritmo topográfico. La función objetivo utilizada fue el peso de la torre y la variable de diseño es el ángulo de inclinación de la torre (figura 1.8). El análisis y modelado estructural de la torre fue basado en un análisis mezclado viga armadura, todos los miembros verticales fueron modelados como elementos tipo viga y los horizontales e inclinados, incluyendo la armadura de las cruces como miembros armadura espacial.

El algoritmo empleado extrae los archivos de entrada de datos. std de STAAD Pro y altera las coordenadas automáticamente de acuerdo con el ángulo de inclinación de las patas definidas por el usuario.



*Figura 1.17 Variables de diseño. (Muda, et al., 2013)*

Tort et al. (2017) Desarrollaron un método para optimizar torres de transmisión de energía integrando un algoritmo de recocido simulado en dos fases con el software para optimizar torres autoportadas en función del peso (acorde con los requerimientos y especificaciones de diseño de la ASCE 10-97) llamado PLS-TOWER



*Figura 1.18 Variables de diseño definidas para alterar la geometría de la torre(Tort et al., 2017)*

El algoritmo SA de dos fases desarrollado por los autores es exclusivo para el diseño en torres de líneas de alta tensión. En la primera fase el método solo optimiza los parámetros de diseño y los miembros son dimensionados a esfuerzo axial basados en un acercamiento heurístico. El objetivo de la primera fase es el de proveer un diseño inicial rápidamente en un número relativamente bajo de iteraciones. En la segunda fase el mejor diseño obtenido en la fase inicial es utilizado como la solución inicial y el algoritmo SA es implementado de nuevo para el diseño y las variables del tamaño bajo un nuevo grupo de parámetros del algoritmo.

## 1.6 Softwares

El software MATLAB, de *Mathwork*, se expone como una herramienta fuerte dentro del campo de la optimización y el cálculo matemático. Las funciones de Matlab se agrupan en más de 35 cajas de herramientas y paquetes de bloques, dentro de los que se encuentra el de Matemáticas y Optimización.

Por sus características es un software utilizado ampliamente en el mundo académico y científico, dentro del cual se destaca el campo de la optimización mediante algoritmos evolutivos (Coello, et al., 2007; Gong, et al., 2015; Negrín, 2016; Rajakumar y George, 2013; Riquenes, 2010; Sağ y Çunkas, 2009; Stewart, et al., 2008; Vázquez, et al., 2017).

Por otra parte el programa SAP 2000 es uno del software líder en la ingeniería estructural: se pueden analizar cualquier tipo de estructuras con este programa e incluso diseñar elemento por elemento de manera precisa con los reglamentos más conocidos (ACI, EUROCODIGO, etc.).

Se trata de un excelente programa de cálculo estructural en tres dimensiones mediante elementos finitos. Es el descendiente directo de la familia SAP90, muy conocida hace algunos años. En este caso, el programa está totalmente renovado. Tal vez lo más visible sea su nueva interfaz, totalmente integrada en Windows y realmente sencilla de usar.

Mediante SAP2000 es posible modelar complejas geometrías, definir diversos estados de carga, generar pesos propios automáticamente, asignar secciones, materiales, así como realizar cálculos estructurales de hormigón y acero basados en variadas normativas.

### **Conclusiones parciales**

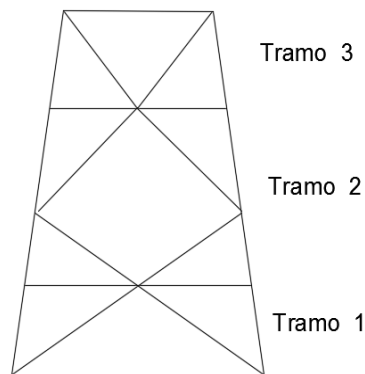
1. La tendencia mundial en la optimización del diseño estructural de torres autosoportadas apunta al empleo de técnicas de optimización metaheurísticas que buscan la reducción de la población y lograr la convergencia en un menor número de iteraciones.
2. Las técnicas de optimización modernas basadas en procesos de búsqueda estocásticos han reemplazado a las técnicas tradicionales basadas en el cálculo de derivadas para la realización del diseño óptimo de torres autosoportadas.
3. Para la realización del diseño de torres autosoportadas los Algoritmos Genéticos son de las técnicas más robustas debido a la simplicidad de su concepto. Los parámetros a optimizar son las secciones transversales de los miembros que conforman la torre, la geometría, las coordenadas nodales y los costos totales.

## CAPÍTULO 2 PARÁMETROS Y PROCEDIMIENTOS PARA LA OPTIMIZACIÓN

En el capítulo se realiza una descripción de la torre a optimizar, así como el procedimiento para su optimización, se definen las variables continuas, discretas, cargas y las restricciones tanto geométricas como de capacidad de los miembros además se establece la función objetivo a optimizar.

### 2.1 Características de la torre

La torre está compuesta por perfiles angulares de alas iguales, distribuidos en tres tramos como muestra la figura 2.1.

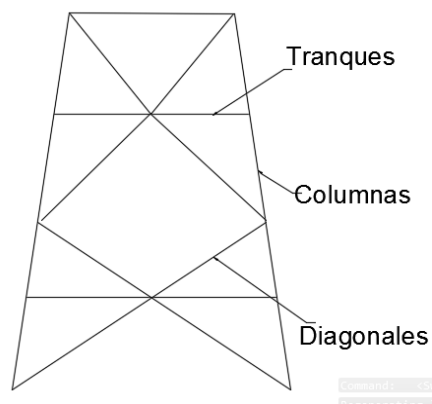


*Figura 2.1 Configuración inicial de la armadura de una cara de la torre.*

### 2.2 Consideraciones para la modelación del diseño original.

La torre objeto de estudio es de sección triangular, con una altura de 15 m. Un diseño inicial de dicha torre se creó a partir de la modelación en el programa de análisis estructural SAP 2000 (versión 16), sin utilizar técnicas de optimización. Las uniones de la torre se consideraron articuladas salvo en la unión de las columnas donde se consideraron empotradas (ver figura 2.2).





*Figura 2.2 Elementos componentes de la torre*

Los elementos de la armadura fueron modelados como barras, elementos lineales con dos nudos extremos.

La unión entre los elementos tranques y columna, se consideró articulada (ver figura 2.3) al igual que las diagonales y tranques debido a que en todos los casos la unión entre estos elementos se realiza mediante dos pernos en la misma línea de acción de la barra.



*Figura 2.3 Unión diagonal-tranque.*

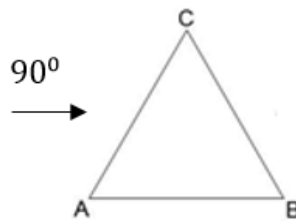
Se considera las condiciones de apoyo empotradas en la base (ver figura 2.4) debido a que las patas van a tener apoyos independientes empotrados a la cimentación por medio de planchas de acero ancladas con pernos a la masa de hormigón y unidas con pernos a las patas de la torre.



*Figura 2.4 Condiciones de apoyo.*

### **2.2.1 Cargas**

La torre fue dividida en diferentes tramos para el análisis, en cada nodo que conforma la torre se consideró una carga de 2 kN de forma concentrada que se encuentran en los vértices de la sección transversal de la torre en la dirección mostrada en la figura



*Figura. 2.5 Direcciones de aplicación de la carga de viento.*

### **2.3 Planteamiento del problema de optimización.**

El problema de optimización de una torre autoportada puede ser descrito como encontrar el diseño de torre con menor peso, variando tanto las secciones de sus elementos como su configuración y cumpliendo con las restricciones de resistencia impuestas por las normas, así como restricciones de constructibilidad. (Tort et al. (2017))

### 2.3.1 Variables

Las variables geométricas utilizadas para la optimización son, el área de las secciones transversales de los perfiles comerciales extraídos del catálogo de perfiles ACEROL (figura 2.6 a) que serían variables de tipo discretas y las variables continuas serían la alturas de los bloques (figura 2.6b), la inclinación de la torre (figura 2.6c) y su ancho.

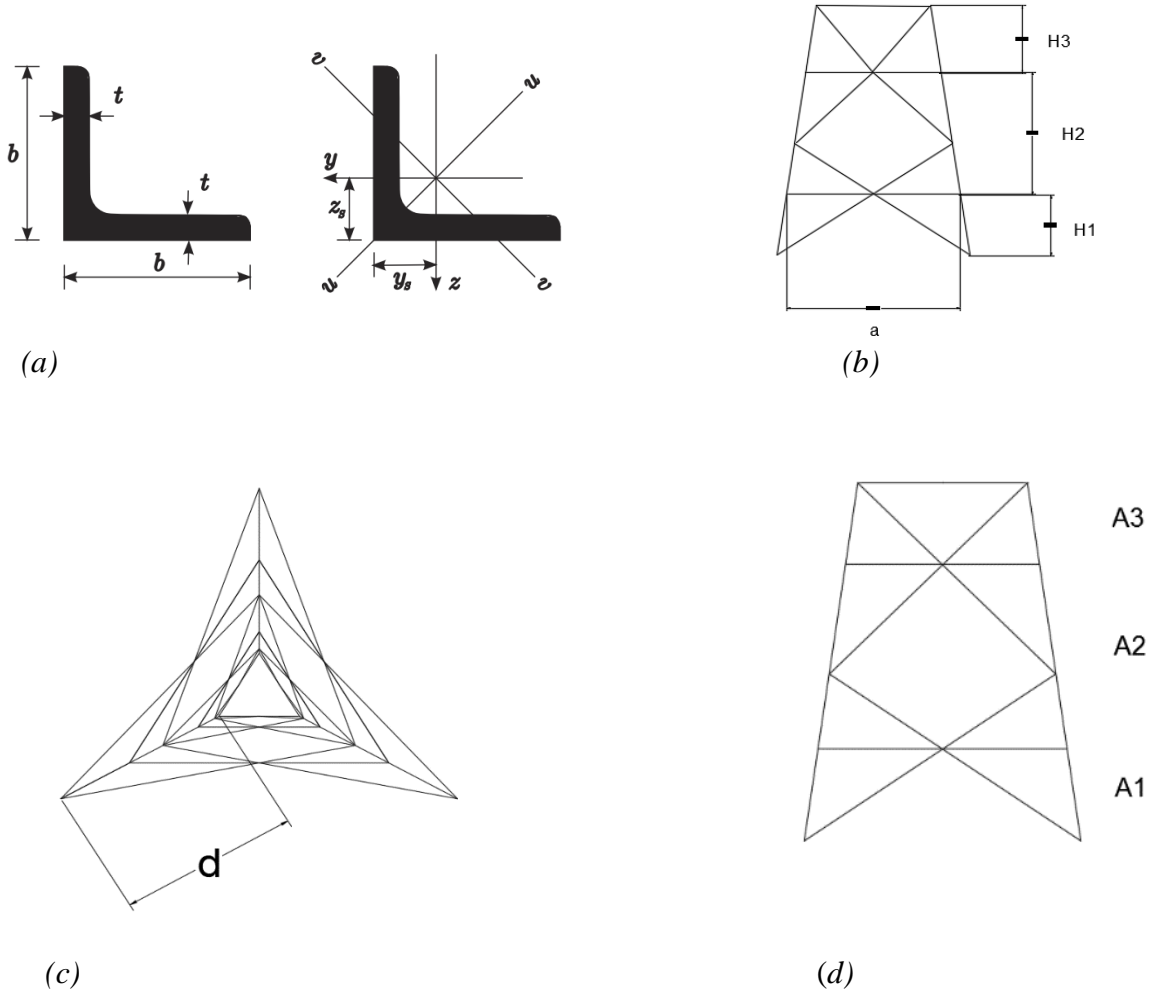


Figura 2.6 Variables de diseño (a)Sección transversales (b)Variables alturas de los bloques y ancho de la torre(c)Inclinación de la torre (d)Distribución de áreas en la torre

### 2.3.2 Función objetivo

Como se demuestra según trabajos de diversos autores como Tort et al. (2017) (Mohammed et al. (2012), Sivakumar et al. (2004), Felix and Vanderplaats (1987)) Sony y Airin (2016)

(Sheppard and Palmer (1972), Raj and Durrant (1976), Majid and Tang (1984)), la función a optimizar conocida como función objetivo, más utilizada es el peso de la torre, el cual está estrechamente relacionado con el costo de la estructura. Se formula como sigue:

$$W = \sum_i^{Nm} \rho \cdot L_i \cdot A_i \quad (2.1)$$

Donde  $\rho$  es la densidad del acero y  $L_i$  y  $A_i$  son la longitud y el área de la sección transversal de cada uno de los elementos de la torre.

Se decide penalizar la función objetivo producto al gran coste computacional que requiere imponer restricciones en Algoritmos Genéticos, es decir de aquellos perfiles que la razón entre la carga impuesta y la resistencia del mismo de un valor superior a uno se realiza la penalización de estos miembros aumentándole el peso y así el Algoritmo Genético lo desechará automáticamente y así nos evitamos poner la restricción. Otra penalización impuesta a la función objetivo fue a los miembros que su grado de esbeltez ( $\frac{KL}{r}$ ) fuese superior a 200.

### **2.3.3 Parametrización de la torre según las variables**

Para lograr la configuración de la torre en función de las variables se ubicó primeramente la posición de los nodos quedando como sigue:

#### **Triángulos**

Para los triángulos que conforman los bloques de dicha armadura se utilizó una vista en planta de los mismos para la ubicación de sus nodos en un eje de coordenadas (x,y,z) utilizando para esto las características geométricas propias de los triángulos equiláteros en función de las variables (ver figura 2.7).

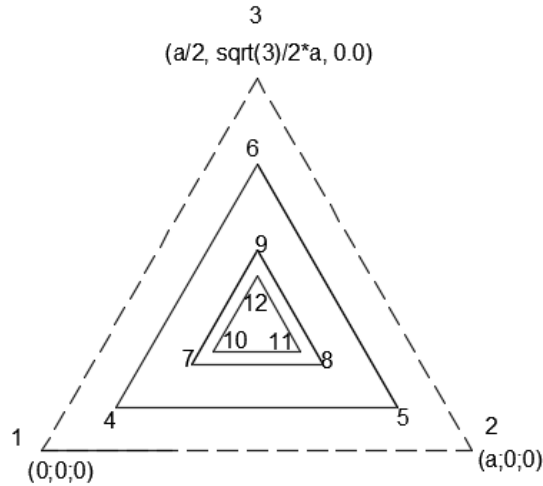


Figura 2.7 Nodos de los triángulos de la torre

Los restantes puntos se describen en el ANEXO 1

### Puntos intermedios de los triángulos

Los puntos intermedios de los triángulos mostrados en la (fig. 2.8) se parametrizaron de igual forma y se describen en el ANEXO 2.

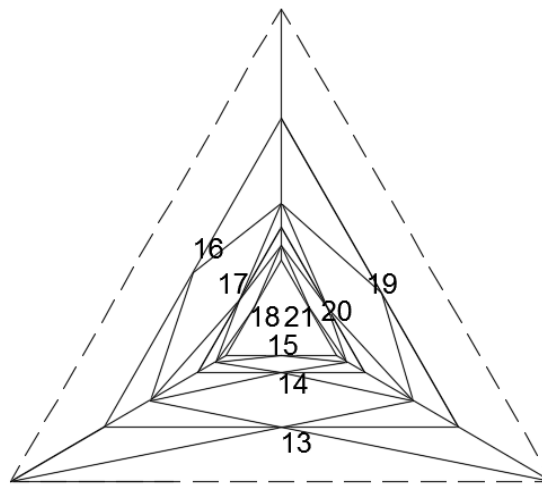


Figura 2.8 Nodos intermedios de los triángulos

### Nodos de las columnas

Los nodos de las columnas mostrados en la Figura 2.9 se parametrizaron de igual forma siendo estos nodos los que conforman las columnas como se muestra en el ANEXO 3

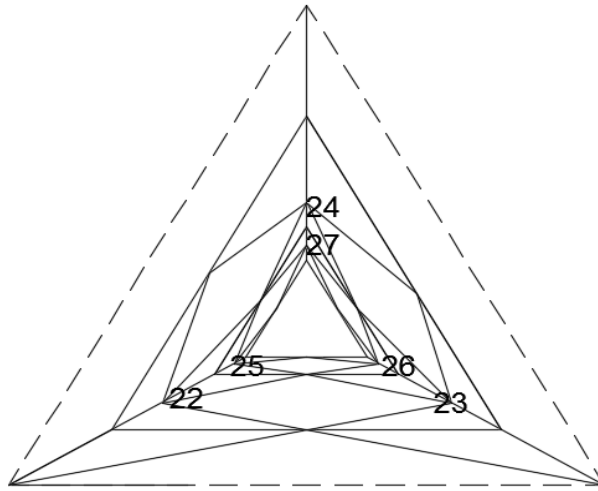
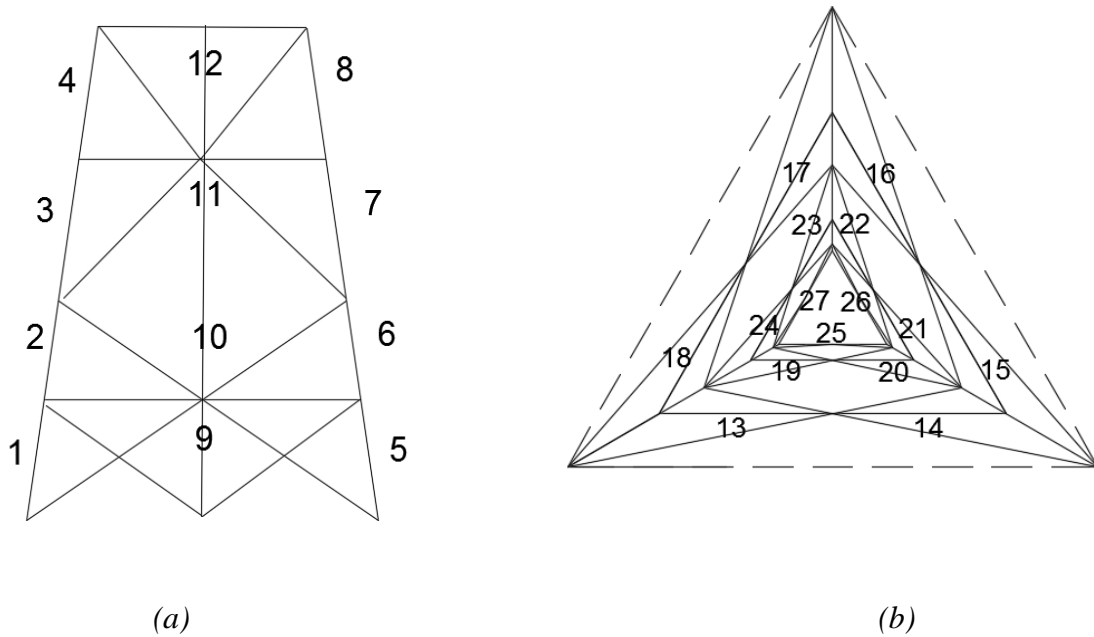
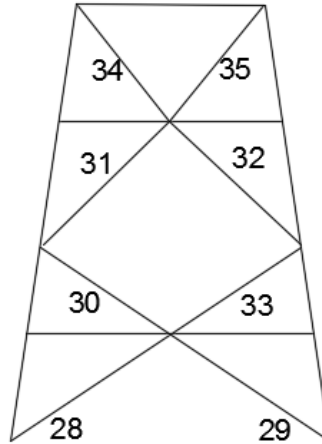


Figura 2.9 Nodos de las columnas

### Conectividad para lograr la configuración de la torre

La conectividad de los elementos de la torre se lleva a cabo por la unión de los nodos parametrizados anteriormente como se muestra en la (Fig.2.10) y el análisis más detallado se muestra en el ANEXO 4





(c)

*Figura 2.10 (a) Barras que conforman las columnas (b) Barras que conforman los triángulos (c) Barras que conforman los rombos*

### 2.3.4 Consideración del modelo articulado

Ciertamente las uniones en las torres, principalmente las uniones en las columnas, resisten cierta cantidad de momento, aunque la consideración de la estructura de la torre con un modelo de nudos articulados para la optimización según autores como (Sivakumar et al. (2004), Couceiro et al. (2016), Souza et al. (2016), Tort et al. (2017)) ..... arroja resultados satisfactorios, siendo el modelo empleado en el trabajo.

### 2.3.5 Restricciones

Los miembros de las torres autoportadas deben ser diseñados para resistir tensiones y cumplir con los límites de esbeltez según el código normativo a utilizar. Los elementos de torres autoportadas, salvo en las columnas, se consideran articulados, por tanto, los estados límites de diseño de dichas estructuras responden a resistencias por fractura y fluencia en caso de la tracción y resistencia según el pandeo en el caso de la compresión axial, además en el código AISC 360-10 se fijan varios límites de esbeltez a cumplir para condiciones de servicio.

### 2.3.5.1 Resistencia a la tracción

La resistencia de diseño en tracción  $\phi_1 P_n$  de miembros traccionados debe ser el menor valor obtenido de acuerdo con los estados límites de fluencia en tracción calculado en la sección bruta y ruptura en tracción calculado en la sección neta.

1-Para fluencia en tracción en la sección bruta:

$$P_n = \phi F_y A_g \quad \phi_t = 0.9 \quad (2.2)$$

2-Para ruptura en tracción en la sección neta:

$$P_n = \phi F_u A_e \quad \phi_t = 0.75 \quad (2.3)$$

Donde:

$A_e$  = área neta efectiva,  $\text{cm}^2(\text{mm}^2)$

$A_g$  = área bruta del miembro,  $\text{cm}^2(\text{mm}^2)$

$F_y$  = tensión de fluencia mínima especificada,  $\text{kgf}/\text{cm}^2(\text{MPa})$

$F_u$  = tensión última mínima especificada,  $\text{kgf}/\text{cm}^2(\text{MPa})$

### 2.3.5.2 Resistencia a la compresión

La resistencia de compresión nominal  $P_n$  debe ser determinada basada en el estado límite de pandeo por flexión:

$$P_n = \phi F_{cr} A_g \quad \phi_t = 0.9 \quad (2.4)$$



Para miembros de alas iguales o alas desiguales conectadas a través del ala más larga que son almas de secciones cajón o enrejados especialmente con miembros adyacentes de alma conectada al mismo lado de la plancha gusset o cordón :

$$\text{Cuando: } 0 \leq \frac{L}{r_x} \leq 75 \qquad \frac{KL}{r} = 60 + 0.8 \frac{L}{r_x} \qquad (2.5)$$

$$\text{Cuando : } \frac{L}{r_x} \geq 75 \qquad \frac{KL}{r} = 45 + \frac{L}{r_x} \leq 200 \qquad (2.6)$$

Donde:

L = Longitud del miembro entre puntos de trabajos, cm(mm)

Rx = radio de giro en torno del eje geométrico paralelo al ala conectada, cm (mm)

$$\text{Si: } \frac{KL}{r} \leq 4.71 \sqrt{\frac{E}{F_y}} \qquad F_{cr} = (0.658^{\frac{F_y}{F_e}}) F_y \qquad F_e = \frac{\pi^2 E}{\left(\frac{KL}{r}\right)^2} \qquad (2.7)$$

$$\text{Si: } \frac{KL}{r} > 4.71 \sqrt{\frac{E}{F_y}} \qquad F_{cr} = 0.877 F_e \qquad (2.8)$$

La programación de la función y los parámetros se pueden ver en el ANEXO 5 en código Matlab.

## 2.4 Parámetros de Algoritmos Genéticos

El algoritmo genético utiliza tres tipos principales de reglas en cada paso para crear la próxima generación a partir de la población actual:

*Selección:* Selecciona a los individuos, llamados padres, que contribuyen a la población en la próxima generación.

*Cruce*: Combina dos padres para formar individuos para la próxima generación.

*Mutación*: Aplicar cambios aleatorios a los padres individuales para formar individuos.

El algoritmo genético en la línea de comando, llama a la función de algoritmo genético "AG" con la sintaxis.

```
[x fval] = ga(@fitnessfun, nvars, options)
```

Donde:

*@fitnessfun*: Es un llamado a la función *fitness*.

*Nvars*: Es el número de variables independientes para la función de *fitness*.

*Options*: Es una estructura que contiene opciones para el Algoritmo Genético. Si no se especifican cambios en este argumento, "AG" utiliza su opción predeterminado.

El algoritmo genético y la ToolBox de búsqueda directa de MATLAB consiste en siguientes funciones:

Selección por algoritmo genético:

*selectionremainder*: Resto de muestras estocásticas sin reemplazo

*selectionroulette*: Elija a los padres utilizando la rueda de la ruleta

*selectionstochunif*: Elija los padres utilizando muestreo universal estocástico (SUS)

*selectiontournament*: Cada padre es el mejor de un conjunto aleatorio

*selectionuniform*: Elige a los padres al azar

*Crossover*(recombinación) función por algoritmos genéticos

*crossoverheuristic*: Pasar del peor padre al ligeramente mejor padre

*crossoverintermediate*: Promedio ponderado de los padres

*crossoverscattered*: Posicionar la función de cruce independiente

*crossoveringlepoint*: Punto de cruce de un solo punto

*crossovertwopoint*: Dos puntos de cruce

*Mutation* función por algoritmo genético

*mutationgaussian*: Mutación gaussiana

*mutationuniform*: Mutación multipunto uniforme

## **Conclusiones parciales**

1- La consideración de un modelo de nudos articulados, para la optimización de una torre autoportada es la más difundida por los autores de referencia en el tema, brindando resultados satisfactorios en todos los casos analizados.

2- Debido al gran costo computacional que requieren las restricciones en los Algoritmos Genéticos se decide penalizar la función objetivo(peso) cuando no se cumplan las restricciones y no ponerlas directamente en la función restricción.

3-El problema de optimización a analizar cuenta con dos tipos de variables, la sección transversal de los perfiles de los tramos de la torre y las variables descritas como continuas siendo las alturas de los bloques, el ancho y la inclinación.

## CAPÍTULO 3 ANÁLISIS DE LOS RESULTADOS

El presente capítulo muestra los resultados obtenidos luego de haber optimizado el peso de la torre aplicando Algoritmos Genéticos y se describe el proceso seguido para lograr la convergencia de la función objetivo a un valor óptimo global y evitar los óptimos locales, a continuación se compara con un caso de estudio diseñado por experiencia del diseñador para de esta forma demostrar la eficiencia y superioridad de la aplicación de Algoritmos Genéticos en el proceso de diseño de una torre autoportada.

### 3.1 Análisis de los resultados de la optimización al aplicar algoritmos genéticos.

Los parámetros predeterminados de Algoritmos Genéticos utilizados fueron para el caso de cruce *@crossoversinglepoint*, para el caso de la mutación *@mutationadaptfeasible* y para el caso de la selección *@selectiontournament*. En el caso de la población y generaciones se fueron variando hasta lograr la convergencia, dicho proceso se describe a continuación.

Los primeros resultados se obtuvieron variando los valores de población, comenzando con 100 hasta 330 aumentando los valores en un rango de 10 y las generaciones se decidió mantenerlas en 100 como se puede mostrar en el (Gráfico 3.1).

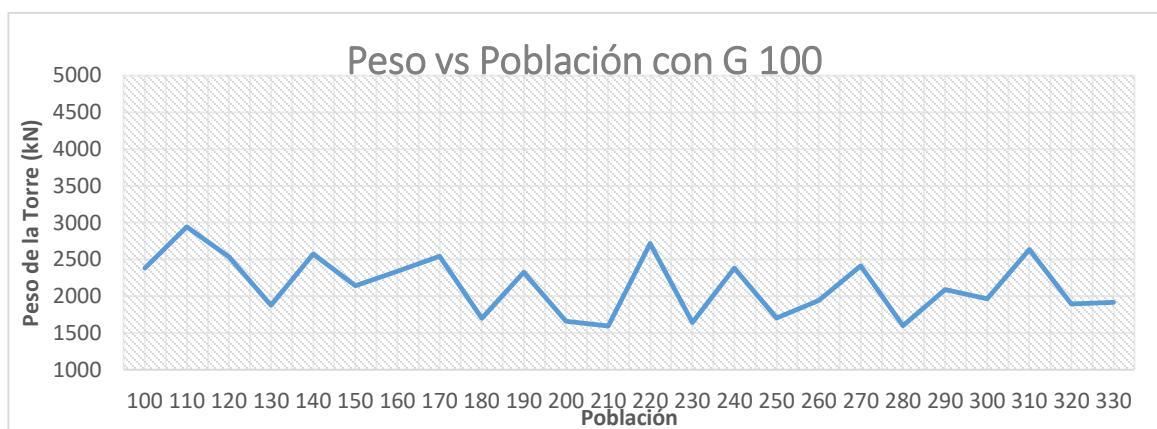
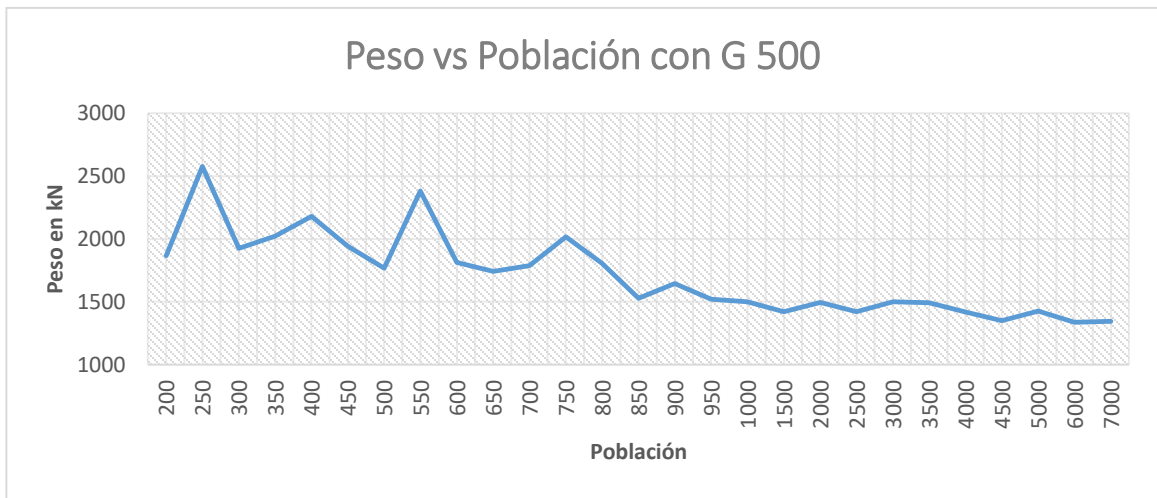


Gráfico 3.1 Resultados de los pesos obtenidos con las diferentes poblaciones

Como se muestra en el (Gráfico 3.1) los valores obtenidos no convergen a un resultado, variando notablemente los rangos del peso entre una corrida y otra comportándose entre los rangos de 1593.41 kN -2945.62 kN lo cual es lógico debido a que la cantidad de variables es superior a lo de los trabajos referenciados, por lo que se decide llevar el número de generaciones a 500 e ir aumentando el número de población en rangos mayores.

En segunda instancia los resultados se obtuvieron variando los valores de población comenzando con 200 hasta 5000, hasta la población 1000 se aumentaron los valores en rangos de 50 y a partir de 1000 se aumentaron en rangos de 500 y las generaciones se decidió mantenerla, pero ascendida en este caso a 500 como se muestra en el (Gráfico 3.2).



*Gráfico 3.2 Resultados de los pesos obtenidos al aumentar las poblaciones.*

En el Gráfico 3.2 se pudo apreciar claramente que los valores de la función objetivo en primera instancia no convergen a un valor específico, no siendo así hasta que la población tomó valores superiores a los 950 donde se puede definir la convergencia a un único valor.

### **3.2 Evolución de los procesos de optimización**

Se aprecia en el (Gráfico 3.2) mostrado anteriormente como a medida que se va aumentando la población y las generaciones se obtienen resultados satisfactorios en la (Figura 3.1) se muestra la evolución de las torres con la variación de dichos parámetros.

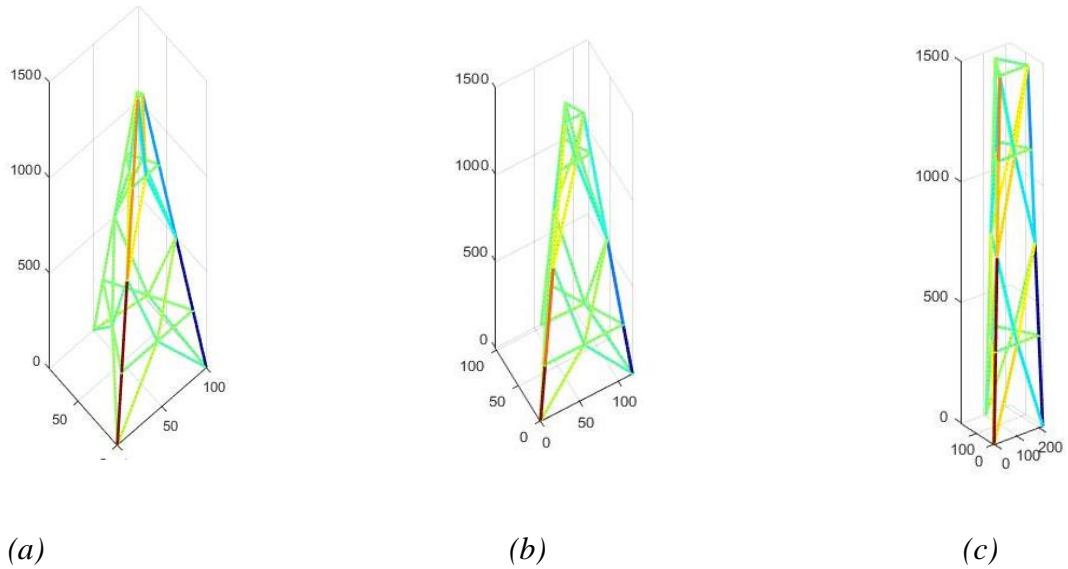


Figura 3.1 (a) Modelo con P100G100 (b) Modelo con P220G100 (c) Modelo con P6000G500

### 3.3 Solución propuesta por el diseñador

En este caso se presenta un modelo realizado por el diseñador donde el mismo a través de su intuición y experiencia predimensiona una torre autosoportada de 15 m de altura como se muestra en la (Figura 3.2) teniendo como valores de ancho de 300 cm, inclinación de 30 cm, altura del bloque 1 y del bloque 2 de 500 cm y utilizando perfiles L110x100x10 con un valor de área de  $21.2 \text{ cm}^2$  utilizando perfiles angulares de alas iguales para todos los miembros de la armadura obteniéndose un peso total de 2734 kN. La torre está sometida a una carga de 2 kN en cada nodo en la sección del eje x.

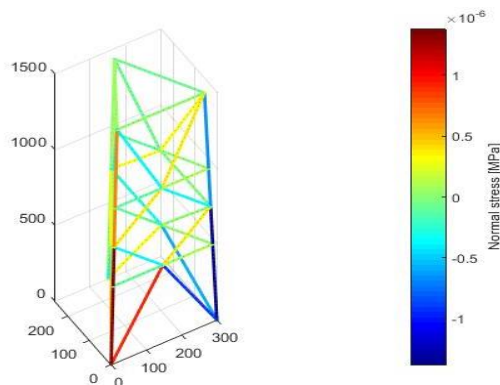


Figura 3.2 Modelo propuesto por el diseñador

### 3.4 Solución optimizada

A partir de los resultados de la función objetivo obtenidos se concluye que los valores convergen a un valor mínimo de 1335.838441 obtenido a partir de una población 4500 y generación 500 correspondiente a las variables (221.116096, 54.997264, 392.037884, 783.924106, 12.2, 12.2, 8.73) siendo el ancho, la inclinación, altura del bloque 1, altura del bloque 2, área del bloque 1, área del bloque 2, área del bloque 3 respectivamente. Los perfiles utilizados para la configuración de dicha torre son L90x90x7 para el bloque 1 y 2 y L75x75x6 para el bloque 3, siendo perfiles angulares de alas iguales. (Figura 3.3) (a) y (b)

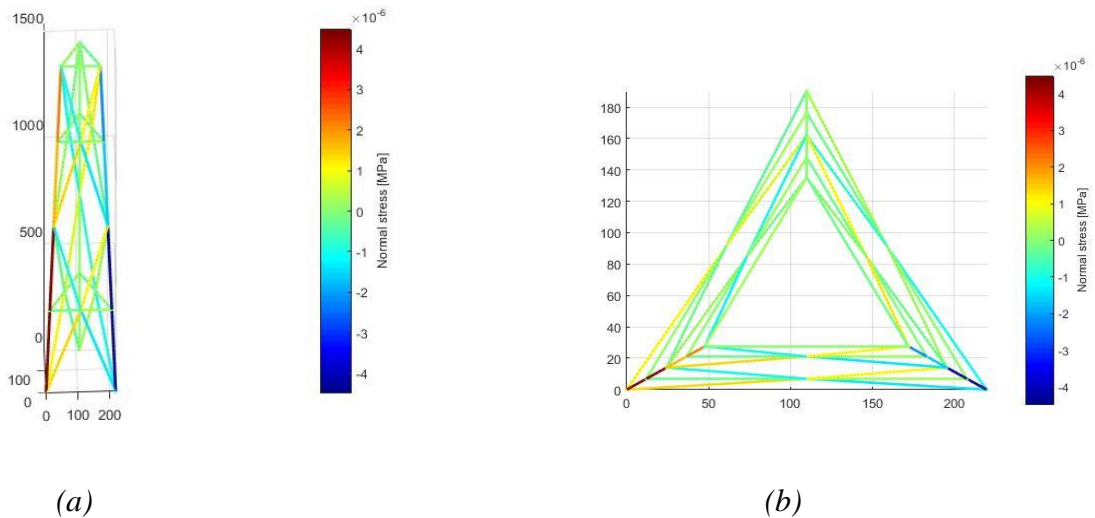
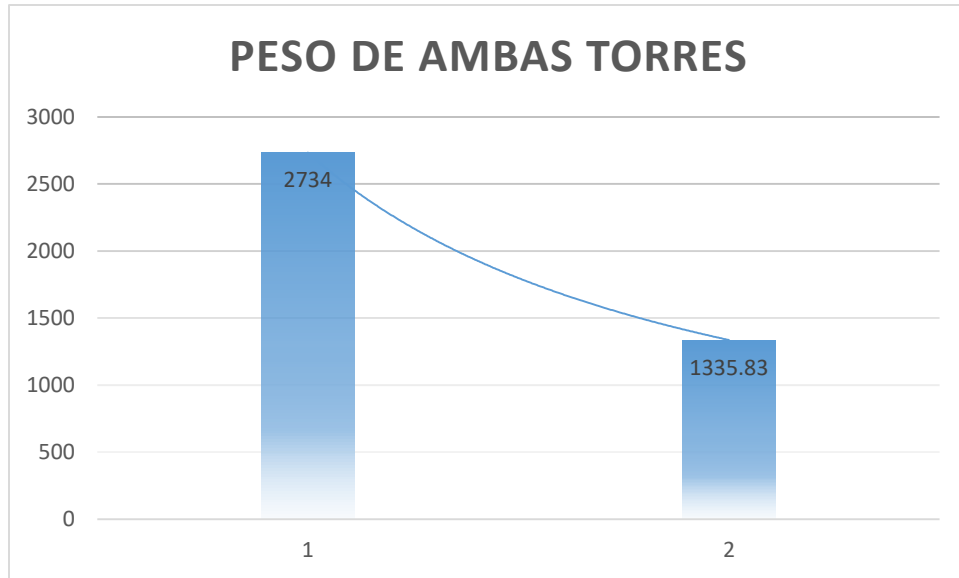


Figura 3.1 (a) Vista en elevación de la torre (b) Vista en planta

Después de comparados los valores de peso del diseño optimizado y el propuesto por el diseñador se muestra en el (Gráfico 3.3) la diferencia de los mismos, donde el peso de la torre optimizada es un 51.14 % menor, garantizando la eficiencia del uso de los Algoritmos Genéticos al proceso de diseño de este tipo de estructura.



*Gráfico 3.3 Valores de peso de ambos diseños*

### **Conclusiones parciales**

1- Se puede afirmar que tiene una estrecha relación el número de variables de la función objetivo con el número de población y generaciones, y se demuestra que con población pequeña los resultados no convergen a un valor específico mientras que con el aumento de población y generaciones se obtienen resultados más satisfactorios, en el caso de estudio la convergencia se obtuvo con población 950 y 500 generaciones.

2- Al analizar los resultados obtenidos al aplicar los Algoritmos Genéticos en el proceso de optimización y los obtenidos con la experiencia del diseñador se ve claramente la eficiencia de la utilización de las técnicas heurísticas ya que se obtiene valores de la función objetivo (peso) mucho menores que en diseño por intuición del diseñador.



## CONCLUSIONES

1. Debido a la complejidad y al carácter multifactorial del diseño de torres autoportadas, las técnicas de optimización basadas en procesos de búsquedas estocásticas, han reemplazado a las técnicas tradicionales, siendo Algoritmos Genéticos una de las más robustas.
2. La función optimizada es el peso de la estructura siendo esta penalizada al no incluir restricciones para disminuir el tiempo computacional, teniendo como variables el ancho, altura de los bloques, inclinación y las secciones transversales de los perfiles, concibiendo un modelo numérico de nudos articulado arrojando a resultados satisfactorios como demuestra la bibliografía consultada.
3. Al analizar el resultado obtenido en el caso de estudio con la aplicación de Algoritmos Genéticos y el obtenido por la experiencia del diseñador, se demuestra la eficiencia de la técnica aplicada, pues se reduce en un 51.14 % el peso total de la estructura.

## RECOMENDACIONES

Basadas en las conclusiones obtenidas y para la extensión futura del presente trabajo, se realizan las siguientes recomendaciones:

1. Aplicar la técnica empleada a casos de estudios más complejos, donde existan diferentes casos y tipos de cargas.
2. Convertir de optimización monobjetivo a multiobjetivo donde se optimice el peso de la estructura e incluya las deflexiones y desplazamientos.
3. Utilizar la interacción Sap-Matlab en la optimización para incluir los efectos dinámicos de las cargas.

## REFERENCIAS BIBLIOGRÁFICAS

- ADELI, H. & CHENG, N.-T. 1994. Concurrent genetic algorithms for optimization of large structures. *Journal of Aerospace Engineering*, 7, 276-296.
- BECK, A. T. & VERZENHASSI, C. C. 2008. Risk optimization of a steel frame communications tower subject to tornado winds. *Latin American Journal of Solids and Structures*, 5, 187-203.
- CABALLERO, J. A. & GROSSMAN, I. E. 2007. Una revision del estado del arte en optimización. *Revista Iberoamericana de Automática e Informática Industrial*, 4, 5-23.
- CABALLERO, J. A. & GROSSMANN, I. E. 2007. Una revisión del estado del arte en optimización. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 4, 5-23.
- CAMP, C. V. & FARSHCHIN, M. 2014. Design of space trusses using modified teaching–learning based optimization. *Engineering Structures*, 62-63, 87-97.
- COUCEIRO, I., PARÍS, J., MARTÍNEZ, S., COLOMINAS, I., NAVARRINA, F. & CASTELEIRO, M. 2016. Structural optimization of lattice steel transmission towers. *Engineering Structures*, 117, 274-286.
- CHUNMING, W. E., TINGTING, S., BIN, M. & JING, G. 2012. Research on the optimal layout of high-strength steel in the transmission tower. *Physics Procedia*, 33, 619-625.
- FELIX, J. & VANDERPLAATS, G. 1987. Configuration optimization of trusses subject to strength, displacement and frequency constraints. *Journal of Mechanisms, Transmissions, and Automation in Design*, 109, 233-241.
- FOGEL, L. J., OWENS, A. J. & WALSH, M. J. 1966. Artificial intelligence through simulated evolution.
- FRASER, A. S. 1957. Simulation of genetic systems by automatic digital computers I. Introduction. *Australian Journal of Biological Sciences*, 10, 484-491.

- GANDOMI, A. H. & ALAVI, A. H. 2012. Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17, 4831-4845.
- GANDOMI, A. H. & GOLDMAN, B. W. 2018. Parameter-less population pyramid for large-scale tower optimization. *Expert Systems with Applications*, 96, 175-184.
- GOLDBERG, D. E. & HOLLAND, J. H. 1988. Genetic algorithms and machine learning. *Machine learning*, 3, 95-99.
- HANSSEN, L. 1981. A Computer Program System for Design of Power Transmission Towers. NORGES TEKNISKE HOEGSKOLE TRONDHEIM SELSKAPET FOR INDUSTRIELL OG TEKNISK ....
- HOLLAND, J. H. 1962. Outline for a logical theory of adaptive systems. *Journal of the ACM (JACM)*, 9, 297-314.
- KIRKPATRICK, S., GELATT, C. D. & VECCHI, M. P. 1987. Optimization by Simulated Annealing. In: FISCHLER, M. A. & FIRSCHEIN, O. (eds.) *Readings in Computer Vision*. San Francisco (CA): Morgan Kaufmann.
- KOCER, F. Y. & ARORA, J. S. 2002a. Optimal design of latticed towers subjected to earthquake loading. *Journal of Structural Engineering*, 128, 197-204.
- KOCER, F. Y. & ARORA, J. S. 2002b. Optimal Design of Latticed Towers Subjected to Earthquake Loading. *Journal of Structural Engineering*, 128, 197-204.
- MAGALHÃES, P.A.A.; FERREIRA, T.S.; JUNIOR, A.C.D.A.; DE, O.A.; FILHO, C. ; MAGALHAES, C.A., 2014. "Structural analysis and structural optimization of self supporting truss towers to support a large wind turbine.". *International Journal of Optimization in Civil Engineering*, 4, pp. 341-359.
- MAJID, K. & TANG, X. 1984. Optimum design of pin-jointed space structures with variable shape. *Structural Engineer*, 62, 31-37.
- MAKRIS, P. A. & PROVATIDIS, C. G. 2002. Weight minimisation of displacement-constrained truss structures using a strain energy criterion. *Computer Methods in Applied Mechanics and Engineering*, 191, 2187-2205.
- MARTÍN, P. & ELENA, V. 2012. Análisis sísmico de modelos cubanos de torres autoportadas de telecomunicaciones. *Revista Cubana de Ingeniería*, Vol. III, 25-34.

- MILLÁN-PÁRAMO, C., 2017. “Diseño óptimo de armaduras empleando optimización con ondas del agua”. INGE CUC, vol 13 (2), pp. 102-111, DOI: DOI: <http://dx.doi.org/10.17981/ingecuc.13.2.2017.11>.
- MOHAMMED, A. F., ÖZAKÇA, M. & TAYŞI, N. 2012. Optimal design of transmission towers using genetic algorithm. *SDU Int J Technol Sci*, 4.
- MUDA, Z. C., MUSTAPHA, K. N., OMAR, R. C., USMAN, F., ALAM, M. A. & THIRUCHELVAM, S. Optimization of structural design for sustainable construction of transmission tower based on topographical algorithm. IOP Conference Series: Earth and Environmental Science, 2013. IOP Publishing, 012023.
- NAVARRINA, F., VALERA, A., PARÍS, J., COLOMINAS, I. & CASTELEIRO, M. 2005. Optimización de torres de alta tensión mediante un método de programación lineal secuencial con búsqueda unidireccional cuadrática.
- RAJ, P. P. & DURRANT, S. O. 1976. Optimum structural design by dynamic programming. *Journal of the Structural Division*, 102, 1575-1589.
- RECHENBERG, I. 1965. Cybernetic solution path of an experimental problem. *Royal Aircraft Establishment Library Translation 1122*.
- SAKA, M. Optimum design of space trusses with buckling constraints. Proceedings of 3rd International Conference on Space Structures, 1984. University of Surrey.
- SANCHEZ-CABALLERO, S., SELLES CANTÓ, M. Á., PLA-FERRANDO, R. & PEYDRO, M. 2012. Nuevas técnicas de optimización de estructuras. *3C Tecnología*, 1, 1-19.
- SHEPPARD, D. J. & PALMER, A. C. 1972. Optimal design of transmission towers by dynamic programming. *Computers & Structures*, 2, 455-468.
- SIARRY, P. 2016. *Metaheuristics*, Springer.
- SIGMUND, O. 2000. Topology optimization: a tool for the tailoring of structures and materials. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 358, 211-227.
- SIVAKUMAR, P., RAJARAMAN, A., SAMUEL KNIGHT, G. & RAMACHANDRAMURTHY, D. 2004. Object-oriented optimization approach

- using genetic algorithms for lattice towers. *Journal of computing in civil engineering*, 18, 162-171.
- SONY, S.R. ; AIRIN, M.G., 2016. "Optimization of Transmission Tower using Genetic Algorithm ". *International Journal of Science and Research (IJSR)*.
- SOUZA, R. R. D., FADEL MIGUEL, L. F., LOPEZ, R. H., MIGUEL, L. F. F. & TORII, A. J. 2016. A procedure for the size, shape and topology optimization of transmission line tower structures. *Engineering Structures*, 111, 162-184.
- SUAREZ, O. 2013. Una aproximación a la heurística y metaheurísticas. *INGE@ UAN-Tendencias en la Ingeniería*, 1.
- TELES, M. L. & GOMES, H. M. 2010. Comparação de algoritmos genéticos e programação quadrática seqüencial para otimização de problemas em engenharia. *Teoria e Prática na Engenharia Civil*, 10, 29-39.
- TORT, C., ŞAHIN, S. & HASANÇEBİ, O. 2017. Optimum design of steel lattice transmission line towers using simulated annealing and PLS-TOWER. *Computers & Structures*, 179, 75-94.

## ANEXOS

### Anexo 1 Parametrización de los puntos de los triángulos.

$$4-\left(H1 \cdot \frac{D}{36} \cdot \frac{\sqrt{3}}{2}; H1 \cdot \frac{D}{36} \cdot \frac{1}{2}; H1\right)$$

$$5-\left(a - H1 \cdot \frac{D}{36} \cdot \frac{\sqrt{3}}{2}; H1 \cdot \frac{D}{36} \cdot \frac{1}{2}; H1\right)$$

$$6-\left(\frac{a}{2}; \frac{\sqrt{3}}{2} \cdot a - H1 \cdot \frac{D}{36}; H1\right)$$

$$7-\left(\frac{(H1+H2) \cdot D \cdot \sqrt{3}}{36}; \frac{(H1+H2) \cdot D \cdot 1}{36}; (H1+H2)\right)$$

$$8-\left(a - \frac{(H1+H2) \cdot D \cdot \sqrt{3}}{36}; \frac{(H1+H2) \cdot D \cdot 1}{36}; (H1+H2)\right)$$

$$9-\left(\frac{a}{2}; \frac{\sqrt{3}}{2} \cdot a - (H1+H2) \cdot \frac{D}{36}; (H1+H2)\right)$$

$$10-\left(D \cdot \frac{\sqrt{3}}{2}; D \cdot \frac{1}{2}; 15\right)$$

$$11-\left(a - D \cdot \frac{\sqrt{3}}{2}; D \cdot \frac{1}{2}; 15\right)$$

$$12-\left(\frac{a}{2}; a \cdot \frac{\sqrt{3}}{2} - D; 15\right)$$

**Anexo 2 Parametrización de los puntos intermedios de los triángulos.**

$$13- \left( \frac{a}{2}; H1 \cdot \frac{D}{15} \cdot \frac{1}{2}; H1 \right)$$

$$14- \left( \frac{a}{2}; (H1+H2) \cdot \frac{D}{15} \cdot \frac{1}{2}; H1+H2 \right)$$

$$15- \left( \frac{a}{2}; (H1+H2+H3) \cdot \frac{D}{15} \cdot \frac{1}{2}; H1+H2+H3 \right)$$

$$16- \left( H1 \cdot \frac{D}{15} \cdot \frac{\sqrt{3}}{2} + \left( \frac{a - H1 \cdot D \cdot \frac{\sqrt{3}}{15}}{4} \right); H1 \cdot \frac{D}{15} \cdot \frac{1}{2} + \left( \frac{a - H1 \cdot D \cdot \frac{\sqrt{3}}{15} \cdot \sqrt{3}}{4} \right); H1 \right)$$

$$17- \left( H1+H2 \cdot \frac{D}{15} \cdot \frac{\sqrt{3}}{2} + \left( \frac{a - (H1+H2) \cdot D \cdot \frac{\sqrt{3}}{15}}{4} \right); (H1+H2) \cdot \frac{D}{15} \cdot \frac{1}{2} + \left( \frac{a - (H1+H2) \cdot D \cdot \frac{\sqrt{3}}{15} \cdot \sqrt{3}}{4} \right); H1+H2 \right)$$

$$18- \left( H1+H2+H3 \cdot \frac{D}{15} \cdot \frac{\sqrt{3}}{2} + \left( \frac{a - (H1+H2+H3) \cdot D \cdot \frac{\sqrt{3}}{15}}{4} \right); (H1+H2+H3) \cdot \frac{D}{15} \cdot \frac{1}{2} + \left( \frac{a - (H1+H2+H3) \cdot D \cdot \frac{\sqrt{3}}{15} \cdot \sqrt{3}}{4} \right); H1+H2+H3 \right)$$

$$19- \left( a - (H1) \cdot \frac{D}{15} \cdot \frac{\sqrt{3}}{2} + \left( \frac{a - (H1) \cdot D \cdot \frac{\sqrt{3}}{15}}{4} \right); (H1) \cdot \frac{D}{15} \cdot \frac{1}{2} + \left( \frac{a - (H1) \cdot D \cdot \frac{\sqrt{3}}{15} \cdot \sqrt{3}}{4} \right); H1 \right)$$

$$20- \left( a - (H1+H2) \cdot \frac{D}{15} \cdot \frac{\sqrt{3}}{2} + \left( \frac{a - (H1+H2) \cdot D \cdot \frac{\sqrt{3}}{15}}{4} \right); (H1+H2) \cdot \frac{D}{15} \cdot \frac{1}{2} + \left( \frac{a - (H1+H2) \cdot D \cdot \frac{\sqrt{3}}{15} \cdot \sqrt{3}}{4} \right); H1+H2 \right)$$

$$21- \left( a - (H1+H2+H3) \cdot \frac{D}{15} \cdot \frac{\sqrt{3}}{2} + \left( \frac{a - (H1+H2+H3) \cdot D \cdot \frac{\sqrt{3}}{15}}{4} \right); (H1+H2+H3) \cdot \frac{D}{15} \cdot \frac{1}{2} + \left( \frac{a - (H1+H2+H3) \cdot D \cdot \frac{\sqrt{3}}{15} \cdot \sqrt{3}}{4} \right); H1+H2+H3 \right)$$



**Anexo 3 Parametrización de los nodos de las columnas**

$$22-\left(\frac{(H1+\frac{H2}{2})}{15}\cdot D\cdot\frac{\sqrt{3}}{2};\frac{(H1+\frac{H2}{2})}{15}\cdot D\cdot\frac{1}{2};(H1+\frac{H2}{2})\right)$$

$$23-\left(a-\frac{(H1+\frac{H2}{2})}{15}\cdot D\cdot\frac{\sqrt{3}}{2};\frac{(H1+\frac{H2}{2})}{15}\cdot D\cdot\frac{1}{2};(H1+\frac{H2}{2})\right)$$

$$24-\left(\frac{a}{2};a\cdot\frac{\sqrt{3}}{2}-\frac{(H1+\frac{H2}{2})}{15}\cdot D;(H1+\frac{H2}{2})\right)$$

$$25-\left(\frac{(H1+H2+\frac{H3}{2})}{15}\cdot D\cdot\frac{\sqrt{3}}{2};\frac{(H1+H2+\frac{H3}{2})}{15}\cdot D\cdot\frac{1}{2};(H1+H2+\frac{H3}{2})\right)$$

$$26-\left(a-\frac{(H1+H2+\frac{H3}{2})}{15}\cdot D\cdot\frac{\sqrt{3}}{2};\frac{(H1+H2+\frac{H3}{2})}{15}\cdot D\cdot\frac{1}{2};(H1+H2+\frac{H3}{2})\right)$$

$$27-\left(\frac{a}{2};a\cdot\frac{\sqrt{3}}{2}-\frac{(H1+H2+\frac{H3}{2})}{15}\cdot D;(H1+H2+\frac{H3}{2})\right)$$

#### **Anexo 4 Conectividad para lograr la configuración de la torre.**

Barra 1 ( 1, 4, A, E);

Barra 2;( 4, 22, A, E);

Barra 3 ( 22, 7, A, E);

Barra 4 ( 7, 10, A, E);

Barra 5 ( 2, 5, A, E);

Barra 6 ( 2, 23, A, E);

Barra 7 ( 23, 5, A, E);

Barra 8 ( 8, 11, A, E);

Barra 9 ( 3, 6, A, E);

Barra 10 ( 6, 24, A, E);

Barra 11 ( 24, 9, A, E);

Barra 12 ( 9, 12, A, E);

Barra 13 (4, 13, A, E);

Barra 14 (13, 5, A, E);

Barra 15 (5, 19, A, E);

Barra 16 (19, 6, A, E);

Barra 17 (6, 16, A, E);

Barra 18 (16, 4, A, E);

Barra 19 (7, 14, A, E);

Barra 20 (14, 8, A, E);

Barra 21 (8, 20, A, E);

Barra 22 (20, 9, A, E);

Barra 23 (9, 17, A, E);

Barra 24 (17, 7, A, E);

Barra 25 (10, 12, A, E);

Barra 26 (12, 11, A, E);

Barra 27 (11, 10, A, E);

Barra 28 (1, 13, A, E);

Barra 29 (13, 2, A, E);

Barra 30 (13, 22, A, E);

Barra 31 (22, 14, A, E);

Barra 32 (14, 23, A, E);

Barra 33 (23, 13, A, E);

Barra 34 (14, 10, A, E);

Barra 35 (14, 11, A, E);

Barra 36 (1, 16, A, E);

Barra 37 (16, 3, A, E);

Barra 38 (16, 22, A, E);

Barra 39 (22, 17, A, E);

Barra 40 (17, 24, A, E);

Barra 41 (24, 16, A, E);

Barra 42 (17, 10, A, E);

Barra 43 (17, 12, A, E);

Barra 44 (3, 19, A, E);

Barra 45 (19, 2, A, E);

Barra 46 (19, 23, A, E);

Barra 47 (23, 20, A, E);

Barra 48 (20, 24, A, E);

Barra 49 (24, 19, A, E);

Barra 50 (20, 11, A, E);

Barra 51 (20, 12, A, E);

## Anexo 5 Programación de la función y los parámetros.

```
function Y = f_towergaM(X)

%f_towerga Summary of this function goes here

% Detailed explanation goes here

global ELEMENTS NODES;

E = 19994.798; % Defining the Young modulus of the material [kN/cm2]

rho= 0.0078; % kN/cm^3

Fy= 25.310505; % kN/cm^2 tensión del acero en el límite de fluencia

Perfiles = [1.12 1.42 1.85 1.74 2.27 2.67 3.08 3.79
3.9 3.89 4.8 5.69 5.82 6.91 9.03 8.7 8.13
9.4 8.73 11.4 12.3 15.1 12.2 13.9 15.5 17.1
15.5 19.2 22.7 21.2 25.1 23.2 25.4 27.5 29.7
33.9 30 27.2 35 29.3 34.8 40.3 43 51
43.2 46.1 49 51.8 45.5 48.8 52.1 55.4 58.7
61.9 65.1 68.4 58.1 61.8 65.5 69.1 72.7 76.3
79.9 83.5 87.1 91 94.1 97.6 96.4 101 106 110
115 119 124 128 133];

RadiosGiro = [0.59 0.75 0.74 0.9 0.89 1.05 1.21
1.2 1.35 1.52 1.51 1.5 1.82 1.82 1.8 1.96
2.13 2.12 2.28 2.27 2.43 2.41 2.75 2.74 2.73
2.72 3.06 3.04 3.02 3.35 3.33 3.67 3.66 3.65
3.64 3.62 3.97 4.3 4.27 4.62 4.6 4.58 4.57
4.54 4.89 4.88 4.87 4.86 5.54 5.53 5.52 5.51
5.5 5.49 4.48 5.47 6.17 6.16 6.14 6.13 6.12
```

```
6.11    6.1    6.09    6.08    6.06    6.05    6.04    7.72    7.71
7.7 7.68    7.67    7.66    7.65    7.63    7.62];
```

```
a = X(:, 1);
```

```
D = X(:, 2);
```

```
H1 = X(:, 3);
```

```
H2 = X(:, 4);
```

```
H3 = 1500-H1-H2;
```

```
A1 = Perfiles(round(X(:, 5)));
```

```
R1 = RadiosGiro(round(X(:, 5)));
```

```
A2 = Perfiles(round(X(:, 6)));
```

```
R2 = RadiosGiro(round(X(:, 6)));
```

```
A3 = Perfiles(round(X(:, 7)));
```

```
R3 = RadiosGiro(round(X(:, 7)));
```

```
newProblem('truss3d'); % Creating a 2d truss FEM problem
```

```
addNode (0.0, 0.0, 0.0); % 1 1rst Triangle
```

```
addNode (a, 0.0, 0.0); % 2
```

```

addNode (a/2, sqrt(3)/2*a, 0.0);           % 3

addNode (H1*D/1500*sqrt(3)/2, H1*D/1500*1/2, H1);           % 4 2nd Triangle
addNode (a-H1*D/1500*sqrt(3)/2, H1*D/1500*1/2, H1);           % 5
addNode (a/2, sqrt(3)/2*a-H1*D/1500, H1);           % 6

addNode((H1+H2)/1500*D*sqrt(3)/2, (H1+H2)/1500*D*1/2, H1+H2) %7 3rd Triangle
addNode (a-(H1+H2)/1500*D*sqrt(3)/2, (H1+H2)/1500*D*1/2, H1+H2); % 8
addNode (a/2, a*sqrt(3)/2-(H1+H2)/1500*D, H1+H2); % 9

addNode (D*sqrt(3)/2, D*1/2, H1+H2+H3);           % 10 11th Triangle
addNode (a-D*sqrt(3)/2, D*1/2, H1+H2+H3);           % 11
addNode (a/2, a*sqrt(3)/2-D, H1+H2+H3);           % 12

addNode(a/2 (H1*D/1500)*1/2, H1);
% 13 Puntos Intermedios de triángulos cara paralela a la Horizontal

addNode(a/2, ((H1+H2)*D/1500)*1/2, H1+H2);
% 14

addNode(a/2, ((H1+H2+H3)*D/1500)*1/2, H1+H2+H3);
% 15

```



```
addNode( H1*D/1500*sqrt(3)/2+(a-H1*D*sqrt(3)/1500)/4, H1*D/1500*1/2+(a-
H1*D*sqrt(3)/1500)*sqrt(3)/4,H1) %16 Puntos intermedios de triángulos
parte izquierda
```

```
addNode( ((H1+H2)*D/1500)*sqrt(3)/2+(a-(H1+H2)*D*sqrt(3)/1500)/4,
(H1+H2)*D/1500*1/2+(a-(H1+H2)*D*sqrt(3)/1500)*sqrt(3)/4,H1+H2);
% 17
```

```
addNode( (H1+H2+H3)*D/1500*sqrt(3)/2+(a-(H1+H2+H3)*D*sqrt(3)/1500)/4,
(H1+H2+H3)*D/1500*1/2+(a-
(H1+H2+H3)*D*sqrt(3)/1500)*sqrt(3)/4,H1+H2+H3); % 18
```

```
addNode(a-(H1*D/1500*sqrt(3)/2+(a-H1*D*sqrt(3)/1500)/4),
H1*D/1500*1/2+(a-H1*D*sqrt(3)/1500)*sqrt(3)/4,H1);
% 19 Puntos intermedios de triángulos parte derecha
```

```
addNode(a-((H1+H2)*D/1500)*sqrt(3)/2+(a-(H1+H2)*D*sqrt(3)/1500)/4),
(H1+H2)*D/1500*1/2+(a-(H1+H2)*D*sqrt(3)/1500)*sqrt(3)/4,H1+H2);
% 20
```

```
addNode(a-((H1+H2+H3)*D/1500*sqrt(3)/2+(a-
(H1+H2+H3)*D*sqrt(3)/1500)/4), (H1+H2+H3)*D/1500*1/2+(a-
(H1+H2+H3)*D*sqrt(3)/1500)*sqrt(3)/4,H1+H2+H3); % 21
```

```
addNode( (H1+H2/2)/1500*D*sqrt(3)/2, (H1+H2/2)/1500*D*1/2, H1+ H2/2);
%22 Puntos de Rombo 1
```

```
addNode( a-(H1+H2/2)/1500*D*sqrt(3)/2, (H1+H2/2)/1500*D*1/2, H1+ H2/2);
%23
```

```
addNode( a/2, a*sqrt(3)/2-(H1+H2/2)/1500*D, H1+ H2/2);  
%24
```

```
addBar3D( 1, 4, A1, E);% Adding elements to the problem,columna izquierda
```

```
addBar3D( 4, 22, A2, E);
```

```
addBar3D( 22, 7, A2, E);
```

```
addBar3D( 7, 10, A3, E);
```

```
addBar3D( 2, 5, A1, E); % columna derecha
```

```
addBar3D( 5, 23, A2, E);
```

```
addBar3D( 23, 8, A2, E);
```

```
addBar3D( 8, 11, A3, E);
```

```
addBar3D( 3, 6, A1, E); % Columna de la punta
```

```
addBar3D( 6, 24, A2, E);
```

```
addBar3D( 24, 9, A2, E);
```

```
addBar3D( 9, 12, A3, E);
```

```
addBar3D(4, 13, A1, E);           % 1st Triangle

addBar3D(13, 5, A1, E);

addBar3D(5, 19, A1, E);

addBar3D(19, 6, A1, E);

addBar3D(6, 16, A1, E);

addBar3D(16, 4, A1, E);

addBar3D(7, 14, A2, E);         % 2nd triangle

addBar3D(14, 8, A2, E);

addBar3D(8, 20, A2, E);

addBar3D(20, 9, A2, E);

addBar3D(9, 17, A2, E);

addBar3D(17, 7, A2, E);

addBar3D(10, 12, A3, E);       % 3rd Triangle

addBar3D(12, 11, A3, E);

addBar3D(11, 10, A3, E);
```

```

addBar3D(1, 13, A1, E);           % Primer Semirombo
addBar3D(13, 2, A1, E);
addBar3D(1, 16, A1, E);
addBar3D(16, 3, A1, E);
addBar3D(3, 19, A1, E);
addBar3D(19, 2, A1, E);

addBar3D(16, 22, A2, E);         % Rombo 1 Cara Izquierda
addBar3D(22, 17, A2, E);
addBar3D(17, 24, A2, E);
addBar3D(24, 16, A2, E);

addBar3D(13, 22, A2, E); % Rombo 1 Cara Paralela a la horizontal
addBar3D(22, 14, A2, E);
addBar3D(14, 23, A2, E);
addBar3D(23, 13, A2, E);

addBar3D(19, 23, A2, E);         % Rombo 1 Cara derecha
addBar3D(23, 20, A2, E);

```

```
addBar3D(20, 24, A2, E);

addBar3D(24, 19, A2, E);

addBar3D(14, 10, A3, E);           % Semirombo Final

addBar3D(14, 11, A3, E);

addBar3D(20, 11, A3, E);

addBar3D(20, 12, A3, E);

addBar3D(17, 10, A3, E);

addBar3D(17, 12, A3, E);

initBC;           % Initializing boundary conditions

addConstraintAtNode(1, 'all');% Adding displacements constraints

addConstraintAtNode(2, 'all');

addConstraintAtNode(3, 'all');

addForceAtNode(10, 'x', 2);

addForceAtNode(11, 'x', 2);

addForceAtNode(12, 'x', 2);

addForceAtNode(7, 'x', 2);

addForceAtNode(8, 'x', 2);

addForceAtNode(9, 'x', 2);
```

```

addForceAtNode(4, 'x', 2);

addForceAtNode(5, 'x', 2);

addForceAtNode(6, 'x', 2);

solveFEM; % Solving the FEM problem

postFEM; % Postprocessing the FEM problem

showFEM; % Showing the outcomes

% graphFEM(1); % Plotting the outcomes graph

for i =1:51

    vectoraxial(i)=ELEMENTS{i}.Force; %Este ciclo genera vectores con los
    valores de axial, area, longitud y peso de cada elemento y con la funcion
    sum calcula el peso total de la estructura

    vectorarea(i) =ELEMENTS{i}.Area;

    D_X1(i) = NODES{ELEMENTS{i}.Node_I}.X;

    D_Y1(i) = NODES{ELEMENTS{i}.Node_I}.Y;

    D_Z1(i) = NODES{ELEMENTS{i}.Node_I}.Z;

    D_X2(i) = NODES{ELEMENTS{i}.Node_J}.X;

    D_Y2(i) = NODES{ELEMENTS{i}.Node_J}.Y;

    D_Z2(i) = NODES{ELEMENTS{i}.Node_J}.Z;

    vectorlongitud = sqrt((D_X2 - D_X1).^2 + (D_Y2 - D_Y1).^2+ (D_Z2 -
D_Z1).^2);

```

```

vectorpeso      =  vectorlongitud.*vectorarea.*rho;

Pesototal      =  sum(vectorpeso);

end

RadioGiro = [R1 R2 R2 R3 R1 R2 R2 R3 R1 R2 R2 R3 R1 R1 R1 R1 R1 R2 R2
R2 R2 R2 R2 R3 R3 R3 R1 R1 R1 R1 R1 R1 R2 R2 R2 R2 R2 R2 R2 R2 R2 R2 R2
R2 R3 R3 R3 R3 R3 R3]; %Este vector surge a partir de las equivalencias
de areas y radios de giro comerciales de las barras ,hacer for por radio
de giro

for i=1:1:length(ELEMENTS)

if  vectorlongitud(i)/RadioGiro(i)< 75

        klr(i) = 60+0.8*vectorlongitud(i)/RadioGiro(i);

        else

                klr(i) = 45+ vectorlongitud(i)/RadioGiro(i); % Hay que
poner una penalizacion para klr>200

end

if ( vectoraxial(i) >= 0)

        Pn(i) = 0.9.*vectorarea(i).*Fy;

else if ( vectoraxial(i)) < 0

        if (klr(i) < 4.71*sqrt(E/Fy))

                fe=((3.14^2)*E)/klr(i)^2;

                Fcr=(0.658^(Fy/fe))*Fy;

                Pn(i)= 0.9.*Fcr.*vectorarea(i);

        else

```





