

*Universidad de Matanzas Sede “Camilo Cienfuegos”
Facultad de Ciencias Técnicas
Departamento de construcciones*



Trabajo de Diploma en opción al título de Ingeniero Civil

**“EVALUACIÓN DE TÉCNICAS HEURÍSTICAS Y SU APLICACIÓN
EN TORRES AUTOSOPORTADAS”**

Autor: Lázaro Javier Gómez Biart

Tutor(es): Ing. Juan Carlos Verdecia Somoano

Ing. Javier Alejandro García Miranda

Matanzas, 2020

DECLARACIÓN DE AUTORIDAD

Por medio de la presente declare que Lazaro Javier Gómez Biart 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.

NOTA DE ACEPTACIÓN

Miembros del Tribunal:

Presidente

Secretario

Vocal

DEDICATORIA

Mi dedicatoria va dirigida a mis padres por estar siempre a mi lado en todo momento y en cualquier situación. Gracias por apoyarme siempre y ser la base de mi vida.

AGRADECIMIENTOS

- Quisiera agradecer a toda mi familia por haberme apoyado en todo y en especial a mis padres.
- A Juan Carlos y Javier mis tutores por haberme guiado y ayudado en la realización de este trabajo.
- A mi novia Laidelys porque a pesar de que nuestras carreras son totalmente diferentes siempre estuvo ahí para ayudarme sobre todo en el momento que más lo necesitaba.
- A todas las personas que colaboraron de una forma u otra.

RESUMEN

El campo de la optimización estructural ha sido una de las tendencias a nivel mundial en cuanto a la concepción y diseño de las torres autoportadas de telecomunicaciones, estructuras de vital importancia para nuestra sociedad y que unido al gran auge de las comunicaciones en nuestro país, su demanda ha ascendido considerablemente. Actualmente las técnicas heurísticas de optimización son una opción muy utilizada como herramienta para la optimización de estas estructuras ya que mediante su utilización se logran reducir costos y materiales, obteniendo una solución a un problema ingenieril que cumple con todas las restricciones y limitaciones impuestas, de ahí que la presente investigación tiene como objetivo principal: evaluar el desempeño de técnicas heurísticas en la optimización del diseño de las secciones transversales de los elementos de una torre autoportada. Para darle cumplimiento a dicho objetivo se muestra de forma detallada los algoritmos de dos de las técnicas evolutivas por excelencia utilizadas por varios autores: Algoritmo Genético y Enjambre de Partículas, haciendo análisis de sensibilidad en los parámetros que definen a estos algoritmos para encontrar óptimos globales, para lo cual se utiliza como herramienta computacional el software Matlab mediante una programación elaborada, basada en el método de elementos finitos. Como consecuencia se obtuvo en ambas técnicas un mismo valor de convergencia en cuanto al peso de la estructura, pero una de ellas mostró mayor eficiencia al encontrar este valor en un tiempo mucho menor, y que la hace un 82 % más efectiva. Esta investigación permite definir cuál de las técnicas heurísticas entre Algoritmo Genético y Enjambre de Partículas muestra un mejor desempeño en cuanto a la optimización de torres autoportadas.

Palabras claves: técnicas heurísticas; Algoritmo Genético; Enjambre de Partículas; optimización estructural; torres autoportadas

ABSTRACT

The structural optimization field has been one of the trends worldwide in terms of the conception and design of self-supporting telecommunications towers, structures of vital importance to our society and which together with the great boom in communications in our country, its demand has risen considerably. Currently heuristic optimization techniques are a widely used option as a tool for the optimization of these structures since by using them they reduce costs and materials, obtaining a solution to an engineering problem that complies with all the restrictions and limitations imposed, hence the main objective of this research is to evaluate the performance of heuristic techniques in optimizing the design of the cross sections of the elements of a self-supporting tower. In order to fulfill this objective, the algorithms of two of the evolutionary techniques par excellence used by various authors are shown in detail: Genetic Algorithm and Particle Swarm, doing sensitivity analysis on the parameters that define these algorithms to find global optimal ones, for this, the MATLAB software is used as a computational tool through elaborate programming, based on the finite element method. As a consequence, the same convergence value was obtained in both techniques in terms of the weight of the structure, but one of them showed greater efficiency finding this value in a much shorter time, and that makes it 82% more effective. This investigation allows to define which of the heuristic techniques between Genetic Algorithm and Particles Swarm shows a better performance in terms of optimization of self-supporting towers.

Keywords: heuristic techniques; Genetic Algorithm; Particles Swarm; structural optimization; self-supporting towers

TABLA DE CONTENIDO

Introducción	1
Capítulo 1 Análisis del estado del arte de las técnicas de optimización estructural y los métodos heurísticos de optimización en torres autosoportadas	5
1.1 Técnicas de Optimización.....	5
1.2 Tipos de optimización.....	6
1.3 Métodos de optimización.....	8
1.3.1 Clasificación de los métodos de optimización.....	9
1.4 Trabajos de optimización de torres autosoportadas.....	19
Conclusiones parciales.....	30
Capítulo 2 Parámetros y procedimientos para la optimización	31
2.1 Planteamiento del problema de optimización.....	31
2.2 Variables	31
2.3 Cargas	32
2.4 Función objetivo	33
2.5 Parametrización de la torre según las variables	33
2.6 Consideración del modelo articulado.....	36
2.7 Restricciones	36
2.8 Parámetros de Algoritmos Genéticos.....	38
2.9 Parámetros de Enjambre de partículas	38
Capítulo 3 Análisis de los Resultados.....	40
3.1 Análisis de los resultados de la optimización al aplicar algoritmos genéticos.	40
3.2 Análisis de los resultados de la optimización al aplicar Enjambre de partículas....	41
3.3 Tabla comparativa de los resultados obtenidos entre Algoritmo Genético y Enjambre Partículas.....	42
.....	43
Conclusiones parciales.....	44
Conclusiones	45
Recomendaciones	46
Referencias Bibliográficas	47
Anexos	51

INTRODUCCIÓN

Las torres de celosía autosoportadas 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 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.).

El diseño de estas torres se realiza mediante el método tradicional de prueba y error, donde el proyectista realiza modelos, según su experiencia y los evalúa hasta determinar el que a su juicio cumpla con las condiciones impuestas. Debido al gran número de variables que influyen en dicho diseño, como pueden ser secciones transversales, ancho de base, inclinación, ancho de los bloques, el determinar una combinación óptima de estas variables se hace una tarea muy compleja para los proyectistas promedio. Además, la optimización de la estructura de la torre es no lineal por lo que resulta un problema complejo que involucran varias restricciones, incluyendo pandeo, mínima y máxima tensión y restricciones geométricas. Pudiéndose concluir que, debido a su compleja configuración, estas torres son estructuras metálicas muy sensibles a los procesos de optimización.

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, aunque la tendencia mundial para enfrentar estos diseños sea el empleo de métodos heurísticos que logran arribar a óptimos globales para reducir costos y materiales, obteniéndose así una solución que cumpla con todas las restricciones y limitaciones impuestas.

Dentro del grupo de los algoritmos heurísticos se pueden distinguir los denominados métodos metaheurísticos, que imitan fenómenos simples observados en la naturaleza y que parecen estar asociados con la inteligencia artificial. Estos algoritmos tratan de adaptar el comportamiento de diferentes especies a soluciones de problemas altamente complejos mediante optimización. Entre las técnicas más utilizadas y que mayor desempeño han mostrado en la optimización estructural de torres autosoportadas se encuentran:

- Algoritmos Genéticos (*Genetic Algorithm – GA*)
- Enjambre de Partículas (*Particle Swarm Optimization - PSO*)

Problema Científico

La necesidad de definir cuál de las técnicas heurísticas entre Algoritmo genético y Enjambre de partículas muestra un mejor desempeño en cuanto a la optimización de torres autosoportadas.

Hipótesis

Aplicando técnicas heurísticas como Algoritmos Genéticos y Enjambre de Partículas en la optimización estructural de una torre autosoportada, permitirá definir qué técnica presenta mejor desempeño para este propósito.

Objetivo General

Analizar el desempeño de técnicas heurísticas como Algoritmos Genéticos y Enjambre de Partículas en la optimización estructural de una torre autosoportada.

Objetivos específicos

1. Definir el estado del conocimiento sobre las técnicas heurísticas de optimización, y su aplicación en el campo de la optimización de torres autosoportadas.
2. Aplicar técnicas heurísticas de optimización de Algoritmos Genéticos y Enjambre de Partículas al diseño de una torre autosoportada.
3. Analizar el desempeño de las técnicas heurísticas de Algoritmos Genéticos y Enjambre de Partículas en el diseño de una torre autosoportada.

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

Como **campo de acción** se tiene la optimización estructural.

El cumplimiento de los objetivos trazados conlleva a que se espere como resultado obtener un análisis de la técnica heurística entre Algoritmo Genético y Enjambre de Partículas que muestre un mejor desempeño para el diseño de la **configuración de una torre autosoportada**.

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 los mismos, además de la implicación con la conservación del medio ambiente que está implícita en el ahorro de los productos naturales. Esta reducción de la utilización de materiales sirve además como impulso a la elaboración de estos elementos utilizados en la construcción de viviendas, teniendo entonces un carácter social.

Posee además un **valor metodológico**, pues muestra de forma clara el algoritmo y características fundamentales para el diseño y optimización de los elementos estructurales componentes del sistema en cuestión.

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**:

1. Definición del estado del conocimiento sobre las técnicas heurísticas de optimización, y su aplicación en el campo de la optimización de torres autosoportadas.
2. Aplicación de técnicas heurísticas de optimización de Algoritmos Genéticos y Enjambre de Partículas al diseño de una torre autosoportada.

3. Análisis del desempeño de las técnicas heurísticas de Algoritmos Genéticos y Enjambre de Partículas en el diseño de una 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:

Métodos teóricos:

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.

Métodos empíricos:

Medición: Facilitó mediante la obtención de medidas, la realización y modelación de la torre.

Experimentación: Permitió mediante la realización de un modelo conocer el comportamiento de una torre autosoportada.

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

Las técnicas de optimización pueden ser clasificadas en dos grandes grupos. Aquellas que utilizan las derivadas de la función objetivo para resolver el problema de optimización y las que emplean procesos de búsqueda estocásticos y no calculan o simplemente no se conocen las derivadas de la función objetivo. Al primer grupo pertenecen la programación matemática, la programación lineal, la programación no lineal y todas sus componentes. Al segundo grupo la optimización combinatoria, la computación evolutiva y otras técnicas heurísticas y metaheurísticas. En este capítulo se exponen los conceptos fundamentales definidos a través de la revisión bibliográfica, asociados a los estudios de optimización de torres autosoportadas. Se caracterizarán algunas de las técnicas de optimización utilizadas en el mundo, haciendo especial énfasis en aquellas que han sido empleadas en torres autosoportadas. Además, se identifican las principales investigaciones de autores de referencia en el tema.

1.1 Técnicas de Optimización

Tradicionalmente, los problemas de optimización fueron resueltos usando métodos basados en cálculo, métodos basados en búsqueda aleatoria y en algunos casos técnicas de búsqueda enumerativa. En términos generales dichas técnicas de optimización se pueden dividir en dos grandes grupos, métodos derivativos y no derivativos, dependiendo de si se requiere la derivada de la función objetivo en el proceso de optimización. Los métodos derivativos son métodos basados en cálculo, los cuales son basados en búsqueda gradiente. El método de Newton, el método de Gauss–Newton, los métodos quasi-Newton, el método de la región de confianza y el método de Levenberg–Marquardt son algunos ejemplos de estas técnicas. Estos métodos clásicos se usaron eficientemente por mucho tiempo en el campo de la optimización, aunque no se desempeñan de forma aceptable cuando el número de variables de decisión es muy grande, o incluso cuando el espacio de decisión es discreto, lo cual ocurre en la gran mayoría de los problemas de optimización (Omid Bozorg-Haddad,

2018). Producto de esta desventaja el uso de métodos basados en inteligencia computacional se ha ido desarrollando en conjunto con el desarrollo de la tecnología.

La optimización de estructuras ha ido evolucionando junto con los métodos de optimización, y existen innumerables métodos utilizados para el fin estructural. El autor Millán-Páramo (2017) expone que, 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) en el caso de las estructuras de acero es minimizar el peso de las estructuras, al tiempo que se satisfacen todos los requerimientos impuestos por los códigos de diseño.

Las torres autoportadas de acero son estructuras que por su gran número de variables influyentes en sus diseños son muy sensibles a procesos de optimización estructural. La mayoría de los trabajos de optimización estructural de torres autoportadas tienen como objetivo la reducción del peso de la estructura, pues reduciendo el peso reducimos la cantidad de acero y consecuentemente el costo de la estructura. Para lograrlo existen una infinidad de parámetros y variables que pueden ser modificados por los diseñadores.

1.2 Tipos de optimización

La optimización estructural se puede hacer con diferentes alcances, pues se podría: solamente variar las secciones transversales de la estructura, modificar el propio material, e incluso se puede llegar a modificar la geometría de la estructura, así como una combinación de todos. A continuación, se explican los tipos de optimización mencionados anteriormente.

Optimización del tamaño

La optimización del tamaño mostrada en la 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)

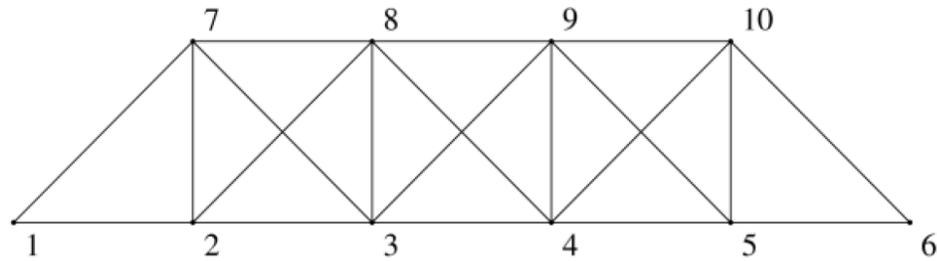


Figura 1.1 Optimización del tamaño. (Sanchez-Caballero et al., 2012)

Optimización del 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. (Sanchez-Caballero et al., 2012)

Optimización de la geometría

La reducción de peso mostrada en la Figura 1.2 se logra cambiando la conectividad entre nodos, manteniendo fijas las coordenadas nodales y las secciones rectas. (Sanchez-Caballero et al., 2012)

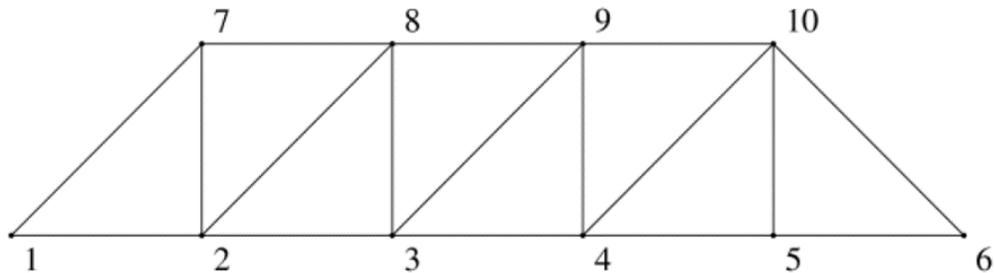


Figura 1.2. Optimización de la geometría. (Sanchez-Caballero et al., 2012)

Optimización topológica

La reducción de peso se logra cambiando las coordenadas y conectividad nodales, manteniendo fijas las secciones rectas. (Figura 1.3)

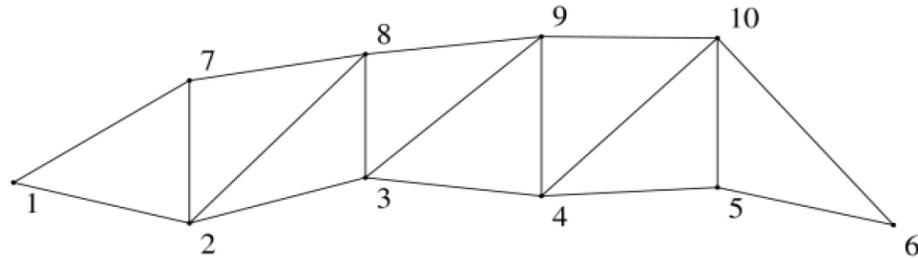


Figura 1.3 Optimización topológica (Sanchez-Caballero et al., 2012)

Cualquiera de los tipos de optimización expuestos puede ser utilizado para lograr una estructura óptima, aunque la decisión depende de las restricciones y características propias de la estructura, así como las condiciones computacionales con las que se cuente. Una vez definido el tipo de optimización es necesario la utilización de un método para llevarla a cabo. La utilización de métodos de optimización y su evolución será el objetivo del siguiente acápite.

1.3 Métodos de optimización

Existen muchos tipos de métodos de optimización con sus respectivas ventajas y desventajas. Recientemente las técnicas de optimización metaheurísticas están siendo el centro de atención y se están aplicando a muchos problemas de optimización en la ingeniería. En esta sección se describen algunos de los métodos de optimización más utilizados y aplicados a torres autosoportadas.

La optimización es un método para encontrar la mejor solución en determinadas circunstancias. Los ingenieros deben decidir sobre varios temas tecnológicos y gerenciales para el diseño, fabricación, construcción, así como mantenimiento de sistemas. La minimización de los esfuerzos con la maximización de la salida deseada es el aspecto más crucial de toda la decisión de búsqueda, así los esfuerzos necesarios para dar u obtener las salidas deseadas es una función de ciertas variables y pueden ser optimizadas para alcanzar el óptimo. (Rao, 2009, Beightler et al., 1979, Datta and Davim, 2019)

Durante el proceso de diseño, un sistema de ingeniería es definido por un conjunto de variables, de las cuales depende el desempeño del sistema. Generalmente algunas de las variables se fijan en el inicio como parámetros asignados de antemano. Los otros parámetros son analizados durante el proceso de optimización, y son llamados variables de diseño o decisión. (Rao, 2009, Datta and Davim, 2019) En la mayoría de los casos las variables de diseño deben satisfacer criterios prescritos, por tanto, no pueden ser escogidas arbitrariamente. Para realizar un diseño aceptable estas restricciones deben ser cumplidas. Cualquier diseño computarizado generalmente lleva a una situación, la cual proporciona un camino viable y aceptable hacia la solución del problema. Naturalmente otros diseños del mismo problema pueden solucionarlo con mejor o menor desempeño, para obtener la solución óptima el mejor diseño entre todos los posibles, debe ser escogido. (Datta and Davim, 2019)

El factor decisivo para la optimización de un diseño es llamado función objetivo, la misma debe ser formulada dependiendo del mecanismo inherente del problema. (Datta and Davim, 2019, Rao, 2009). Por ejemplo, en la optimización del diseño estructural de una torre autosoportada de acero, la función objetivo tiende a ser el peso de la torre, lo cual garantiza un diseño con el peso mínimo y que cumple restricciones geométricas y de tensión de los miembros, con el que se logra el costo mínimo. El algoritmo conduce al sistema, a la solución más óptima dentro de las soluciones existentes en diferentes estados, como un sistema de auto-organización. (Yang, 2014, Datta and Davim, 2019)

1.3.1 Clasificación de los métodos de optimización.

Los problemas de optimización se clasifican según varios aspectos. Pueden ser divididos dependiendo de la presencia de restricciones. También se pueden clasificar de acuerdo a los objetivos. En un problema mono-objetivo tanto la búsqueda del mínimo como del máximo conlleva a una única solución. Si el problema tiene más de un objetivo donde la mejora de uno significase el empeoramiento del otro, se denomina optimización multi-objetivo, aquí un conjunto de soluciones no dominadas surge y se representa en forma de diagrama o frontera de Pareto. Si el número de objetivos excede a tres, el problema es llamado optimización de muchos objetivos. Estos problemas pueden ser clasificados

basado en la naturaleza de la función objetivo, así como de las restricciones. Un problema es denominado lineal o no lineal en dependencia de la función objetivo y de las restricciones, siendo estas lineales o no lineales. Si la función objetivo es polinómica, es denominado problema geométrico, si las restricciones están integradas con la función objetivo y no pueden ser separadas, entonces el problema es denominado problema de programación no separable. Cuando se trata de restricciones temporales, la optimización puede ser distinguida dentro de dos tipos principales, optimización online y offline. Si el problema requiere ser resuelto en pocos segundos o milisegundos es denominado online. (Datta and Davim, 2019, Weise, 2009). En tales casos el objetivo del algoritmo es la velocidad. La actualización de horarios de trabajo, procesos de transporte y motores de búsqueda son un ejemplo de dicha optimización.

Estos ejemplos indican que la optimización online tiene que llevarse a cabo de manera repetitiva para atender diferentes órdenes que llegan continuamente al sistema sin una ponderación significativa. En el caso donde el tiempo no es tan importante y los usuarios están listos para esperar por el resultado óptimo, la optimización offline es aceptable. (Datta and Davim, 2019, Weise, 2009) En la optimización offline el proceso es llevado a cabo una sola vez. Aquí las estrategias de optimización son más importantes y deben ser decididas antes de comenzar el proceso. Los algoritmos determinísticos y heurísticos son dos clases de algoritmos de optimización (Datta and Davim, 2019). En el caso de los algoritmos determinísticos existe una relación obvia entre los atributos del sistema. Cuando la relación entre los parámetros del sistema y la relación función aptitud del sistema son complicadas se vuelve difícil resolver un problema determinísticamente.

Un algoritmo heurístico reúne información acerca del sistema y prueba la aptitud de soluciones aleatorias, así toma una decisión para generar la siguiente solución. De esta forma estos métodos son dependientes de la naturaleza del problema. Por otro lado, una técnica metaheurística combina las heurísticas y la función objetivo eficientemente sin depender de la estructura del problema. Así, un algoritmo metaheurístico hace que los métodos heurísticos sean aplicables a un amplio espectro de problemas (Datta and Davim, 2019, Glover and Kochenberger, 2003, Blum and Roli, 2003). La optimización con técnicas metaheurísticas puede encontrar soluciones de alta calidad para problemas de

optimización combinatoria en un tiempo razonable (Dorigo and Stützle, Datta and Davim, 2019).

Un algoritmo como los que utilizan métodos constructivos o mejora iterativa genera un número limitado de soluciones e incluso se detiene en un óptimo local (Schreiber and Martin, 1999, Johnson and McGeoch, 1997, Datta and Davim, 2019). Si una función es difícil desde el punto de vista matemático, cuando el contexto es discontinuo, no diferenciable, o tiene múltiples óptimos, un algoritmo de optimización determinístico o clásico debe converger a soluciones dentro del óptimo local o desde una pequeña parte del espacio de búsqueda. De ahí que el problema con algoritmos clásicos de optimización es la generalidad, pues resulta imposible determinar si la solución está situada en un espacio de óptimo local o global. De forma similar, se puede afirmar si la solución obtenida es el resultado de todo el espacio de búsqueda o solo de una parte de él. Esta cuestión es más importante para problemas multimodales, donde existen múltiples óptimos (Shekel, 1971, Žilinskas, 1978, Ursem, 2003, Schaffer et al., 1990).

Los conceptos de técnicas metaheurísticas de optimización están siendo cada vez más aceptados en aplicaciones de la vida real debido a su robustez y capacidad de lidiar con complejos problemas de diseño, (como se discutió anteriormente). Las técnicas metaheurísticas pueden dividirse en general en tres grupos. El primero es el grupo de computación evolutiva, que consiste en algoritmos genéticos, programación genética, estrategias evolutivas, evolución diferencial, etc. El segundo grupo puede ser denominado grupo de inteligencia de enjambre y consiste en optimización enjambre de partículas y optimización de colonia de hormigas, etc. Las técnicas dentro de estos dos grupos también pueden ser nombradas como técnicas de optimización bioinspiradas. El tercer grupo consiste en técnicas de optimización inspiradas en procesos físicos y el recocido simulado es la herramienta de optimización más común en este grupo. Los grupos se muestran la Figura 1.4.

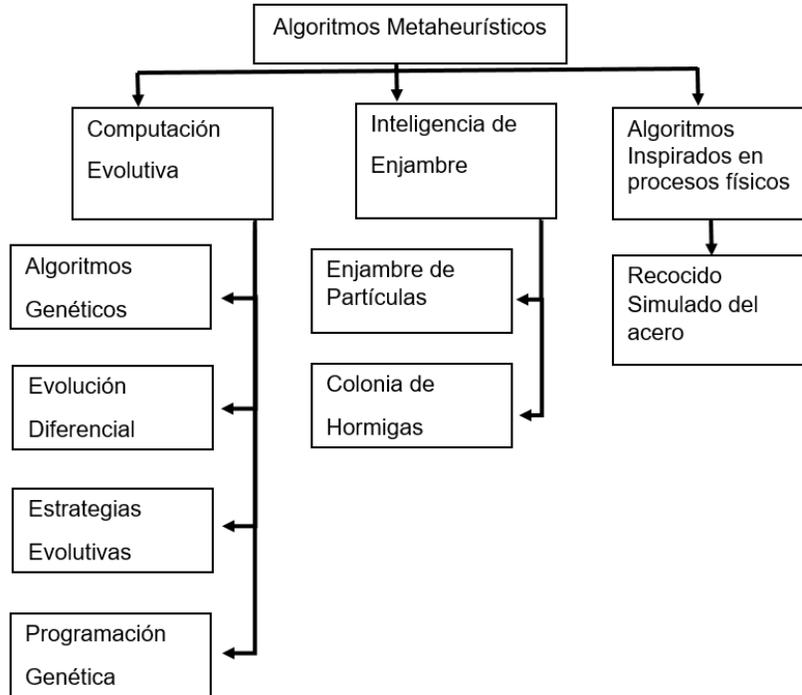


Figura 1.4 Optimización topológica. (Datta and Davim, 2019)

Computación evolutiva

En términos muy generales se podría definir la computación evolutiva como una familia de modelos computacionales inspirados en la evolución. Más formalmente, el término de computación evolutiva se refiere al estudio de los principios de la evolución natural (Alba and Tomassini, 2002). Todas las técnicas de la computación evolutiva se basan en el modelo de la evolución natural. Se utiliza una estrategia de aprendizaje colaborativo a partir de un conjunto de individuos. Normalmente, cada individuo representa o codifica un punto del espacio de búsqueda de soluciones en un problema dado. Cada individuo incorpora información adicional que permite llegar a la solución final del problema a partir de descendientes. La descendencia, obtenida a partir de los individuos de la población, se genera de forma pseudo-aleatoria mediante procesos de cruce y recombinación donde se intercambia información entre dos o más individuos actualmente existentes. La mutación es un proceso de auto-replicación errónea de los individuos que produce pequeños cambios en los mismos. Mediante la evaluación de los individuos, se consigue una medida de la adaptación de cada uno de ellos a su entorno. De acuerdo con esta medida de adaptación,

el proceso de selección favorece más a los individuos mejor adaptados, que serán por tanto los que se reproduzcan más frecuentemente (Goldberg, 1989).

Algoritmos genéticos

Los algoritmos genéticos son una rama de la inteligencia artificial basada en la teoría de la evolución de Darwin por lo que se pueden definir como métodos adaptativos que pueden utilizarse para resolver problemas reales de búsqueda y de optimización basándose en imitar los procesos genéticos de los organismos vivos. Este tipo de algoritmos, se utiliza para abordar una amplia variedad de problemas en un conjunto de campos sumamente diverso, demostrando claramente su capacidad y su potencial (Baker, 1987, Syswerda, 1991). Un algoritmo es un conjunto ordenado y finito de operaciones que permite ejecutar un procedimiento o resolver un problema, un algoritmo es genético cuando utiliza principios de la genética y la selección natural (Ai and Wang, 2011).

Estos trabajan sobre estructuras de cadena que son una representación de una posible solución al problema denominado cromosomas para comenzar, se inicializa la población completamente al azar. En la inicialización hay que tener en cuenta que la distribución de valores debe ser uniforme para cada rango representado por los cromosomas, estos evolucionan a través del tiempo de acuerdo a reglas de supervivencia que privilegian a los que presenten mejor desempeño dándoles una mayor probabilidad de que a partir de ellos se cree la siguiente generación de acuerdo con la función de adaptación; del mismo modo que en la naturaleza los individuos mejor dotados (es decir, los que ofrezcan las mejores soluciones) serán los que más probabilidades tendrán de ser seleccionados para reproducirse. Mientras que los individuos peor adaptados (los que ofrezcan peores soluciones) tendrán más difícil el propagar su material genético a las nuevas generaciones. Sucesivamente cada nueva generación contará con una mayor proporción de buenas características de forma que, si el algoritmo genético ha sido diseñado correctamente, la población convergerá hacia una solución óptima del problema. Mediante un proceso fuertemente estocástico se genera una nueva población de individuos tomando en cuenta su fitness. Básicamente durante la selección se decide cuales individuos serán padres de la

nueva generación. La nueva población puede reemplazar completamente a la población anterior o solamente a las peores soluciones (Fraser, 1958, Bremermann, 1962).

En algunos casos para que un elemento de la población pueda pasar a través de la función de aptitud debe ser codificado, este proceso de codificación es clave para permitir que el algoritmo converja de manera más eficiente hacia la solución del problema planteado, en principio la codificación puede realizarse con cualquier alfabeto finito, sin embargo, no todas las codificaciones resultan ser computacionalmente eficientes, es deber del programador dependiendo de la plataforma de software o hardware que elija escoger que codificación resulte más conveniente. Luego de haber decidido que codificación usar una serie inicial de cadenas (población) es creada aleatoriamente, entonces se aplican operadores a la población para crear nuevas generaciones que se espera tengan mejor desempeño.

Estas candidatas prometedoras se conservan y se les permite reproducirse. Se realizan múltiples copias de ellas, pero las copias no son perfectas; se introducen cambios aleatorios durante el proceso de copia. Luego, esta descendencia digital prosigue con la siguiente generación, formando un nuevo acervo de soluciones candidatas, y son sometidas a una ronda de evaluación de aptitud. Las candidatas que han empeorado o no han mejorado con los cambios en su código son eliminadas de nuevo por puro azar, las variaciones aleatorias introducidas en la población pueden haber mejorado a algunos individuos, convirtiéndolos en mejores soluciones del problema, más completas o más eficientes. De nuevo, se seleccionan y copian estos individuos vencedores hacia la siguiente generación con cambios aleatorios, y el proceso se repite. Las expectativas son que la aptitud media de la población se incrementará en cada ronda y, por tanto, repitiendo este proceso cientos o miles de rondas, pueden descubrirse soluciones muy buenas del problema. Con los cromosomas candidatos a ser padres de la nueva población se efectúan cruces y mutaciones los que garantizan la diversidad genética. Los cruces son intercambios de genes: el proceso consiste en intercambiar segmentos de los cromosomas de las parejas seleccionadas en forma aleatoria. Cuando un cromosoma sufre una mutación, el alelo de uno de sus genes cambia en forma aleatoria.

Al volver al principio del ciclo evolutivo se verifica nuevamente si nuestro mejor individuo supera los requisitos de la búsqueda, en caso contrario se repite todo el proceso para obtener una nueva generación. Este proceso se repetirá en tanto que no se cumpla la condición de parada impuesta previamente (Coello et al., 2007, Goldberg, 1989, Jong, 1975).

Koza (1992) define a Algoritmo genético como un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.

La Figura 1.5 muestra la implementación del Algoritmo Genético de la siguiente manera:

1. Se crea una población con diferentes individuos.
2. Se evalúa esa población mediante la función de aptitud o función objetiva.
3. En la condición de término el algoritmo debe detenerse cuando se encuentre la solución óptima que muchas veces se desconoce. Para ello se utiliza criterios de detención como por ejemplo un número máximo de generaciones o iteraciones, o cuando no exista cambios en la evolución.
4. Mientras no se cumpla la condición de término el algoritmo hará lo siguiente:
 - (a) Selección: Se selecciona un porcentaje de la población, los individuos con mejores aptitudes tienen mayores probabilidades de ser seleccionados.
 - (b) Cruzamiento: Utiliza dos individuos seleccionados para generar dos descendientes donde se combinan las características de ambos individuos.
 - (c) Mutación: Modifica parte del individuo permitiendo ampliar el espacio de búsqueda.

5. Se vuelve a evaluar a los individuos resultantes, seguidamente se reemplazan los mejores individuos para conformar la siguiente generación.
6. Si no se cumple la condición de termino se repite el ciclo hasta encontrar la mejor solución.

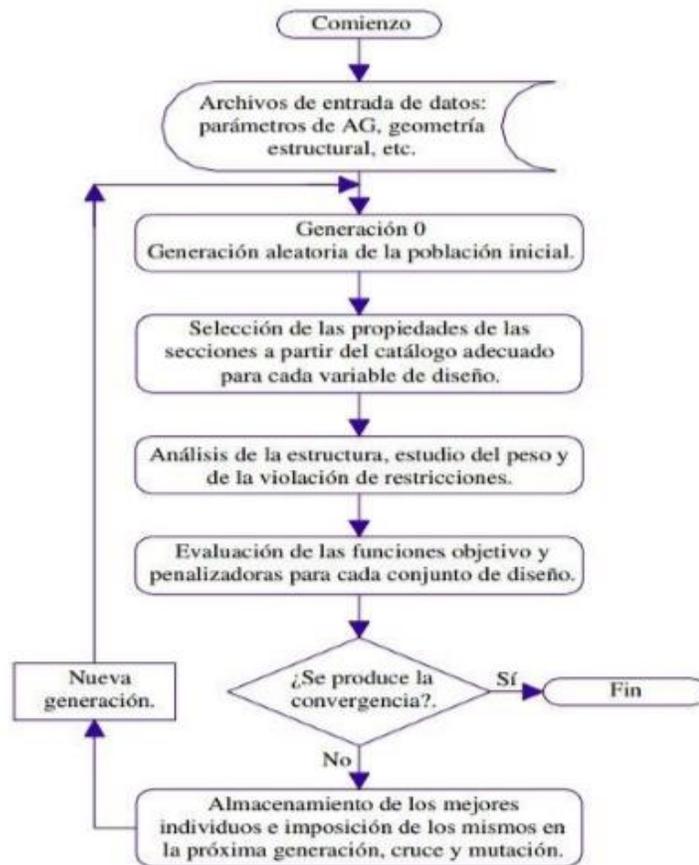


Figura 1.5 Esquema de un GA simple (PRENDES et al., 2002)

Inteligencia de Enjambres

Se basa en el estudio del comportamiento colectivo, descentralizado, de sistemas auto organizados naturales o artificiales. Están constituidos por una población de individuos que interactúan entre si y el medio. Los individuos siguen unas reglas muy simples y a pesar de no existir una estructura de control debido a las interacciones entre los individuos aparece una especie de inteligencia colectiva. En la naturaleza encontramos este tipo de

mecanismo en las colonias de hormigas, bandadas de pájaros, bancos de peces o el crecimiento de bacterias(Wang and Arora, 2004, Ai and Wang, 2011, Wang, 1991).

Enjambre de partículas

La Optimización por Enjambres de Partículas (conocida como PSO, por sus siglas en inglés, *Particle Swarm Optimization*) es una técnica de optimización/búsqueda descrito alrededor de 1995 por James Kennedy y Russell C y se inspira en el comportamiento de los enjambres de insectos en la naturaleza. Se supone que se tiene una función desconocida $f(x;y)$ que puede ser evaluada en los puntos deseados pero que no se puede conocer su expresión. El objetivo es, encontrar valores del espacio de búsqueda para los que la función sea máxima (o mínima, o bien verifica alguna relación extrema respecto a alguna otra función). A esa función se le suele llamar función de aptitud (*fitness*), ya que va a determinar cómo de buena es la posición actual para cada partícula (a la función de fitness a veces también se le llama "paisaje de *fitness*", ya que puede verse como un paisaje con valles y colinas formados por los valores que toma la función). PSO comienza situando partículas al azar en el espacio de búsqueda, pero dándoles la posibilidad de que se muevan a través de él de acuerdo a unas reglas que tienen en cuenta el conocimiento personal de cada partícula y el conocimiento global del enjambre. Proporcionándoles una capacidad simple de movimiento por este paisaje y de comunicación entre ellas pueden llegar a descubrir valores particularmente altos gastando pocos recursos computacionales (cálculos, memoria y tiempo). (Caparrini, 2016)

La Figura 1.6 muestra la estructura básica de un algoritmo PSO:

Se inicia con un set de partículas con posición x_0^i y velocidad y_0^i aleatoriamente distribuido dentro del espacio limitado por límites específicos. Se evalúa los valores de la función $f(x_0^i)$ usando las posiciones del espacio de diseño (x_0^i). Un total de n funciones objetivo pueden ser evaluadas en cada iteración donde n es el número total de partículas en el enjambre. Se actualiza la posición óptima de cada partícula p_k^i en la actual iteración k y la posición global óptima de la partícula p_k^g . Se actualiza la posición de cada partícula usando su posición previa y se actualiza la velocidad usando las ecuaciones:

$$\begin{aligned}
 x_{k+1}^i &= x_k^i + v_{k+1}^i \Delta t \\
 v_{k+1}^i &= w v_k^i + c_1 r_1 \frac{(p_k^i - x_k^i)}{\Delta t} + c_2 r_2 \frac{(p_k^g - x_k^i)}{\Delta t}
 \end{aligned}
 \tag{1.1}$$

Donde: v_{k+1}^i es la actualización del vector velocidad y Δt es el valor de tiempo típicamente considerado como una unidad, v_k^i es la velocidad de la partícula en la iteración k, p_k^i y p_k^g son respectivamente la mejor posición de la partícula i y la mejor posición global del enjambre en la iteración k, r representa un valor aleatorio en el intervalo {0}, los términos c1 y c2 representan el grado de confianza en la mejor solución encontrado por cada partícula individual y por el enjambre como una ola. El término w, es la inercia que es empleada para las habilidades de exploración del enjambre.

Se repite las etapas de la 2 a la 4 hasta que se cumpla con el criterio de parada. Para una implementación básica, el criterio de parada está definida por el número de iteraciones. (Perez and Behdinan, 2007b)

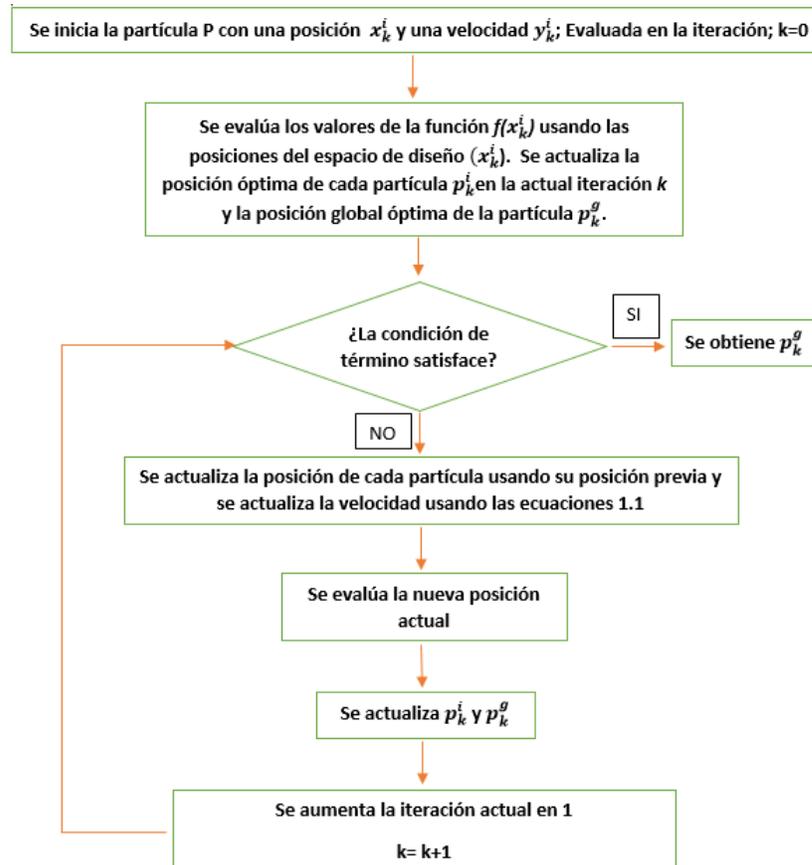


Figura 1.6 Esquema del PSO

En la Figura 1.7 cada partícula (individuo) tiene una posición, (que en 2 dimensiones vendrá determinado por un vector de la forma (x, y) , en el espacio de búsqueda y una velocidad, (que en 2 dimensiones vendrá determinado por un vector de la forma (V_x, V_y) , con la que se mueve a través del espacio. Además, como partículas de un mundo real físico, tienen una cantidad de inercia, que los mantiene en la misma dirección en la que se movían, así como una aceleración (cambio de velocidad), que depende principalmente de dos características: Cada partícula es atraída hacia la mejor localización que ella, personalmente, ha encontrado en su historia (mejor personal). Cada partícula es atraída hacia la mejor localización que ha sido encontrada por el conjunto de partículas en el espacio de búsqueda (mejor global).

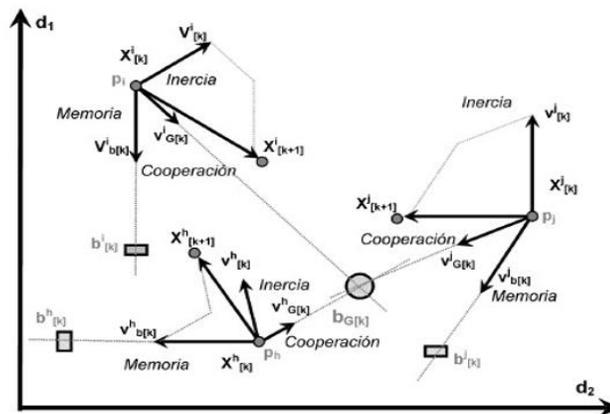


Figura 1.7 Movimiento de la partícula en el espacio de búsqueda.

1.4 Trabajos de optimización de torres autoportadas.

Los métodos de optimización se han aplicado a muchos problemas de ingeniería, específicamente en la ingeniería civil el diseño de torres autoportadas supone un reto para los ingenieros, la compleja configuración de las mismas y el alto número de variables que inciden en su comportamiento, hacen de las torres autoportadas un objetivo constante para la aplicación de estos métodos. Los resultados y simplificaciones utilizados por los autores que optimizan torres han variado con el desarrollo de nuevos métodos sobre todo los metaheurísticos y en conjunto con el desarrollo computacional, a continuación, se

resumen varios trabajos de optimización de torres autosoportadas de acero, analizando los métodos utilizados por los autores y los parámetros utilizados para la optimización.

Sheppard and Palmer (1972) plantean la minimización del peso de la parte inferior de una torre situada entre los apoyos y la primera cruceta. Para ello fue dividida en varios segmentos. 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.

Raj and Durrant (1976) logran 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.

Hanssen (1981) plantea 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.

Majid and Tang (1984a) analizan 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.

Saka (1984) analiza 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 optimalidad (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 optimalidad sólo es eficaz en la práctica cuando se aplica a este tipo de estructuras.

Felix and Vanderplaats (1987b) desarrollan 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 (2002) proponen 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 110kV de 316 miembros, 111 nodos y 666 grados de libertad. Para la realización del diseño dividieron las secciones en nueve tipos de acuerdo a sus longitudes. 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. También el problema es resuelto empleando algoritmos genéticos, los autores llegan a la conclusión de que el método de GA obtiene mejores resultados, pero requiere muchos recursos informáticos.

Navarrina et al. (2006) realizan la optimización del diseño de una torre de la tipología FECSA/GL-110KV. 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. 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 continuo, utilizan un algoritmo de programación lineal secuencial con búsqueda unidireccional cuadrática (SLP-QLS o Sequential Linear Programming with Quadratic Line Search). En cada iteración se plantea un problema lineal aproximado con restricciones laterales adicionales, que se resuelve mediante un algoritmo de programación lineal (Simplex). Para ello, es preciso realizar un análisis de sensibilidad completo de primer orden en cada iteración. La solución del problema lineal determina cuál será la dirección de búsqueda (o de avance) en la correspondiente iteración. A continuación, se calcula la aproximación cuadrática de la función objetivo y de las restricciones en la dirección determinada, lo que permite realizar una búsqueda unidireccional (line-search) aproximada con relativa sencillez. Por tanto, en cada iteración es necesario realizar un análisis de sensibilidad direccional de segundo orden en la dirección de avance. La solución del problema cuadrático determina cuál será el valor de las variables de diseño en la siguiente iteración. Para seleccionar los perfiles óptimos el sistema mantiene invariables los perfiles durante todo el proceso de optimización de formas y después asigna nuevos perfiles.

Beck and Verzenhassi (2008) tratan 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 y 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.

Muda et al. (2013) analizan 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. 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 de STAAD Pro y altera las coordenadas automáticamente de acuerdo con el ángulo de inclinación de las patas definidas por el usuario.

Adeli and Chengn.-T (1994) presentan 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 lo aplican 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 cómputos 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 (figura 2.15) 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.

Sivakumar et al. (2004) analizan 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 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.

Mohammed et al. (2012b) aplican 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 el diseño inicial consideraron áreas para las secciones rectangulares de 0.01 m², las áreas de las secciones fueron consideradas como pseudo-continuas y la función objetivo fue el peso de la torre. Las variables de diseño todas consisten en secciones transversales de los elementos la configuración de la torre no cambia.

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

Khodadadi and Bueleow (2014) desarrollan una técnica de búsqueda que usa conceptos de algebra Formex y optimización evolutiva. En su trabajo refieren que en el proceso de obtener la solución óptima se pierden muchas soluciones que pueden tener buenas prestaciones. Los autores demuestran la habilidad de un método de búsqueda de formas para el diseño multiobjetivo y para encontrar una serie de buenas soluciones que puedan ser comparadas en lugar de una única óptima. Para la puesta en práctica de la técnica los

autores exploran la topología de algunas torres autosoportadas de base poligonal y la comparan geoméricamente y estructuralmente con la torre de agua de Shukhov en Nínzhny Novgorod, Rusia; construida en 1896. Los autores emplearon un algoritmo genético no convencional denominado Non-Destructive Dynamic Population.

Magalhaes et al. (2014) optimizan el diseño de una torre autosoportada de un aerogenerador utilizando un algoritmo genético. 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; costos de los materiales de los nodos, ensamblaje y mano de obra; los costos de los soportes a tierra; costos de manufactura de las diferentes secciones de la torre; 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.

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

Martínez et al. (2014) expusieron una metodología para la optimización general del peso de las estructuras de transporte de energía basada en el algoritmo recocido simulado de los metales enunciado por *Kirkpatrick* a principios de los años ochenta. Este algoritmo permite adoptar soluciones que puedan empeorar la función objetivo y con la finalidad de mejorar la exploración del entorno factible y posibilitar la mejora de la solución final obtenida. El algoritmo propuesto por los autores compatibiliza la naturaleza discreta de las secciones de las barras con la naturaleza continua de las variables que definen la geometría y la forma global de la estructura con lo que evitan las explosiones combinatorias. Los autores aplican tres restricciones fundamentales: restricciones en esbeltez seccional, definen la capacidad resistente del perfil metálico empleado; restricciones en esbeltez elemental, pretenden cuantificar la sensibilidad de la estructura frente a fenómenos de pandeo elemental y

restricciones a los esfuerzos de tracción y compresión axial. La metodología empleada por los autores está basada en procesos estocásticos y considera dos alternativas diferentes, una primera basada en la dirección de máximo descenso, y una segunda basada, en el algoritmo del «recocido simulado». La elección de cada una de las metodologías depende de la existencia de restricciones activas, de forma que solo se empleará el primero de los métodos cuando el diseño no presente ninguna restricción activa. Para que el proceso de optimización diera buenos resultados los autores realizaron un proceso denominado por ellos como compatibilización de las variables que evita que el algoritmo modifique únicamente la forma o la sección haciendo que en las primeras pruebas de cada iteración modifique conjuntamente todas las variables, pero si el algoritmo es incapaz de encontrar una dirección viable durante esas pruebas entonces permite la elección aleatoria de las variables a modificar. La torre analizada por los autores correspondió a una torre de alta tensión de 42 toneladas y una altura de 65 metros.

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 autosoportadas en función del peso (acorde con los requerimientos y especificaciones de diseño de la ASCE 10-97) llamado PLS-TOWER.

Los autores compararon el algoritmo SA de dos fases desarrollado por ellos con una formulación de SA típica en tres problemas de optimización de torres de alto voltaje de 110 y 400kV. Los diseños optimizados y el tiempo de CPU empleado por las dos variantes de SA de cada problema fueron comparados con las configuraciones estructurales disponibles resultado de procesos de diseño convencionales y cuantificaron la economía de material lograda.

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. Los autores llegan a la conclusión de que el algoritmo en dos fases obtiene mejores resultados en cuanto al valor óptimo de la función objetivo y el tiempo computacional.

Millán-Páramo (2017) solucionó el problema de optimización de una armadura espacial de 72 barras empleando el algoritmo ondas del agua. Para el análisis categorizó la armadura en 16 grupos de variables de diseño. Los autores realizaron 100 corridas independientes del problema y utilizaron el mejor peso, el peso promedio y la desviación estándar encontrados por WWO para compararlo con otras técnicas de optimización y llegan a la conclusión de que WWO proporciona mejores resultados.

Kaveh and Ghazaan (2018) realizaron la optimización de tres torres de alta tensión autosoportadas empleando los algoritmos CBO, ECBO, VPS y el algoritmo híbrido MDVC-UVPS. Las variables de diseño consideradas son las áreas de las secciones transversales de todos los miembros de la torre las cuales fueron consideradas como variables discretas y son seleccionadas de una lista de secciones discreta basada en estándares de producción. El problema de optimización fue planteado como la minimización del peso de la estructura sujeto a restricciones de tensión y requerimientos de servicio. Los autores resolvieron cada ejemplo treinta veces de manera independiente y un máximo de 1000 iteraciones fue considerado como el criterio de parada. Para cada algoritmo consideraron una población de 20 partículas. Los algoritmos fueron codificados en MATLAB y las estructuras fueron analizadas usando un método propio de los autores. De los resultados obtenidos los autores concluyen que el algoritmo MDVC-UVPS obtiene mejores resultados.

Schutte and Groenwold (2003) aplicaron por primera vez con éxito un algoritmo PSO a la optimización de estructuras. El algoritmo imitaba el comportamiento social de una bandada de pájaros donde cada pájaro intercambia con los más próximos su posición, velocidad y aptitud mientras el comportamiento global de la bandada tiende a desplazarse a las regiones de mayor aptitud. Para evaluar el algoritmo emplearon las clásicas estructuras de diez y 25

barras, así como una estructura de 36 barras. Su algoritmo logró el mejor resultado para la estructura clásica de 10 barras de entre todos los PSO publicados hasta hoy.

Li et al. (2007) emplearon un algoritmo PSO para optimizar estructuras articuladas. En este trabajo intentaron evitar uno de los inconvenientes de los PSO: su rápida convergencia provoca que, en problemas complejos, la solución converja a un óptimo local. Para probar su algoritmo lo aplicaron sobre las estructuras clásicas de 10 y 25 barras, así como una estructura de 22 y 72 barras.

Perez and Behdinan (2007a) utilizaron un algoritmo PSO para la optimización de estructuras donde emplearon la función de penalización de (Lemonge and Barbosa, 2004, Barbosa and Lemonge, 2003). Validaron el algoritmo con las tradicionales estructuras de 10, 25 y 72 barras.

Kaveh and Talatahari (2009a) Kaveh and Talatahari (2009b) presentaron un algoritmo híbrido HPSOACO que combinaba una Búsqueda Armónica (HS), un algoritmo PSO y un algoritmo ACO para la optimización de estructuras. El algoritmo emplea el PSO para la búsqueda global y el ACO para la búsqueda local. La Búsqueda Armónica se empleó para manejar las restricciones. Para probar su algoritmo utilizaron la estructura de 10 barras, obteniendo unos resultados bastante pobres, así como una cúpula de 120 barras y una torre de 582 barras. La ventaja de este algoritmo frente a los publicados anteriormente es el bajo número de evaluaciones de la función objetivo requeridas.

Li et al. (2009) publicaron un algoritmo HPSO que empleaba una técnica de Búsqueda Armónica (HS) para acelerar la convergencia del algoritmo. Utilizaron las estructuras clásicas de 10 y 25 barras, así como otras de 52 y 72 para probar la convergencia del algoritmo. Este algoritmo proporcionó el mejor resultado después de Schutte entre los PSO para la estructura de 10 barras.

Luh and Lin (2011) presentaron un algoritmo donde la optimización de la estructura se realizaba en dos etapas. En una primera etapa se optimizaba la topología mediante un PSO binario para posteriormente modificar el tamaño mediante un PSO inspirado en la atracción y repulsión de partículas. Para evaluar el funcionamiento del algoritmo empleó una Ground

Structure de 39 barras y 12 nodos. A pesar de afirmar en las conclusiones que el algoritmo proporcionaba mejores resultados que los publicados anteriormente, no se indica referencia alguna a los mismos.

Martins and Gomes (2011) emplearon un algoritmo PSO aplicado a estructuras sometidas a restricciones de resonancia. Para probar su algoritmo emplearon las estructuras clásicas de 10 y 72 barras, así como un puente de 37 y una cúpula de 52.

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. 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.
2. Los parámetros a optimizar son las secciones transversales de los miembros que conforman la torre, la geometría, las coordenadas nodales, y la función objetivo más empleada es el peso de la estructura, bajo las restricciones de tensiones y deformaciones en los elementos estructurales.
3. De las técnicas heurísticas revisadas en la investigación las más empleadas por los autores son: Algoritmos genéticos y Enjambre de partículas.

CAPÍTULO 2 PARÁMETROS Y PROCEDIMIENTOS PARA LA OPTIMIZACIÓN

Las torres de celosía de acero autoportadas son estructuras que se prestan a ser sometidas a procesos de optimización de su diseño estructural. La mayoría de los trabajos de optimización estructural de torres autoportadas tienen como objetivo la reducción del peso de la estructura. En el capítulo se realiza una descripción de la función objetivo, así como el procedimiento para su optimización, se definen las variables, cargas y restricciones tanto geométricas como de capacidad de los miembros.

2.1 Planteamiento del problema de optimización.

El problema de optimización 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. La estructura a optimizar es una torre de sección triangular compuesta por perfiles angulares de alas iguales, distribuidos en tres tramos como muestra la Figura 2.1.

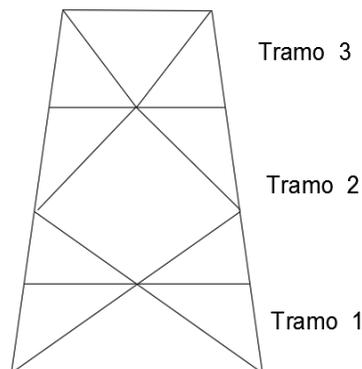


Figura 2.1 Configuración inicial de la armadura de una cara de la torre.

2.2 Variables

Las variables geométricas utilizadas para la optimización son, el área de las secciones transversales de los perfiles comerciales extraídos del catálogo de perfiles ((AISC), 1994) que serían variables de tipo discretas. Los perfiles son de sección angular de alas iguales y

están distribuidos en cada bloque, de modo que cada bloque tiene columnas, tranques y diagonales como variables, para un total de 9.

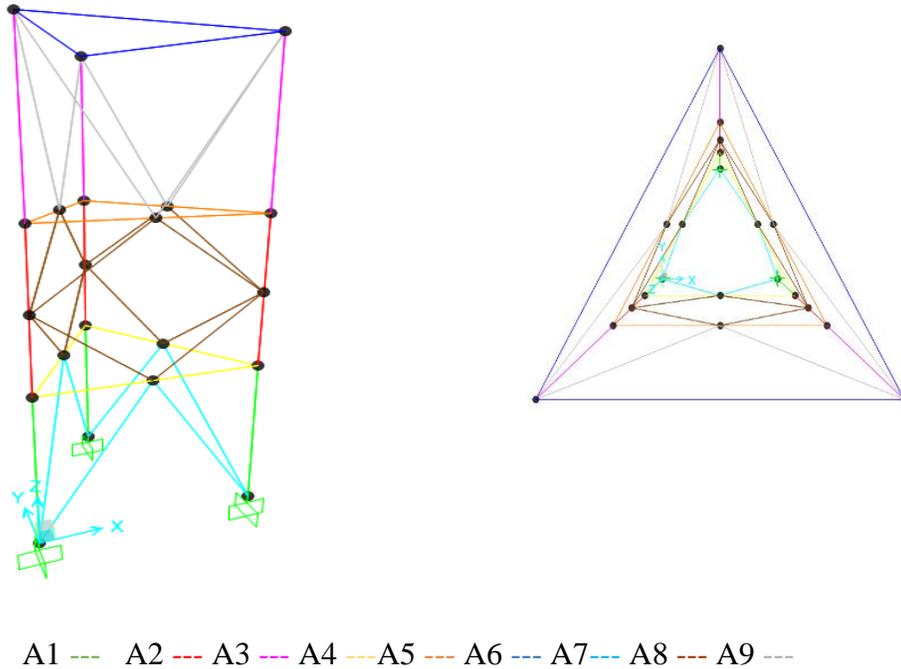


Figura 2.2 Variables de diseño en la función objetivo.

2.3 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 2.3.

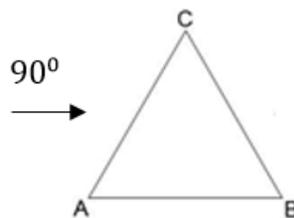


Figura 2.3 Direcciones de aplicación de la carga de viento.

2.4 Función objetivo

Como se demuestra según trabajos de diversos autores como (Tort et al. (2017)) (Mohammed et al. (2012a), Sivakumar et al. (2004), Felix and Vanderplaats (1987a)) (Sony and Airin, 2016) (Sheppard and Palmer (1972), Raj and Durrant (1976), Majid and Tang (1984b)), 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 ρ 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 se evita 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.

2.5 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 (Figura 2.4).

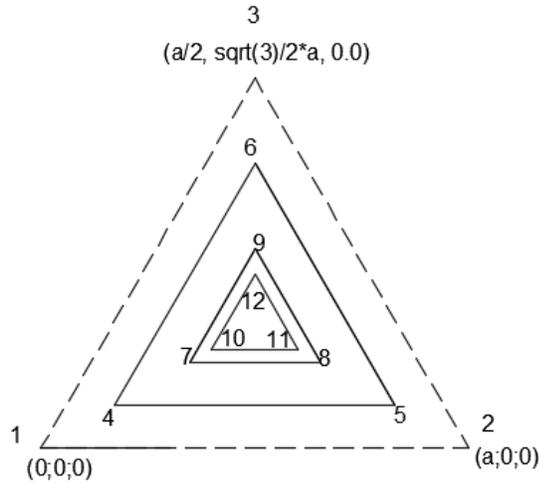


Figura 2.4 Nodos de los triángulos de la torre.

Los restantes puntos se describen en el ANEXO 1

Puntos intermedios de los triángulos

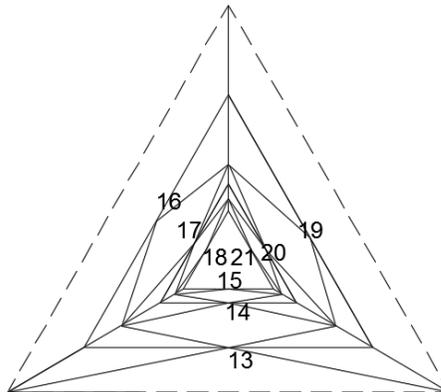


Figura 2.5 Nodos intermedios de los triángulos.

Los puntos intermedios de los triángulos mostrados en la (fig. 2.5) se parametrizaron de igual forma y se describen en el ANEXO 2.

Nodos de las columnas

Los nodos de las columnas mostrados en la Figura 2.6 se parametrizaron de igual forma siendo estos nodos los que conforman las columnas como se muestra en el ANEXO 3.

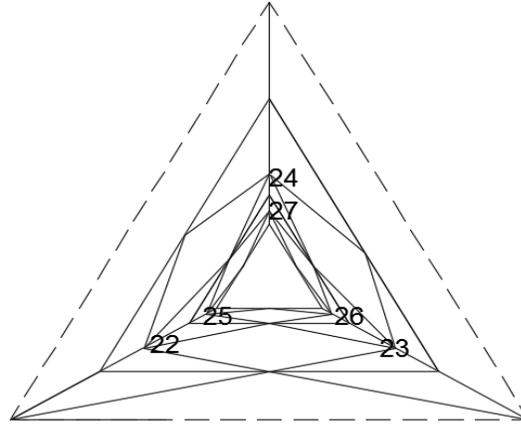
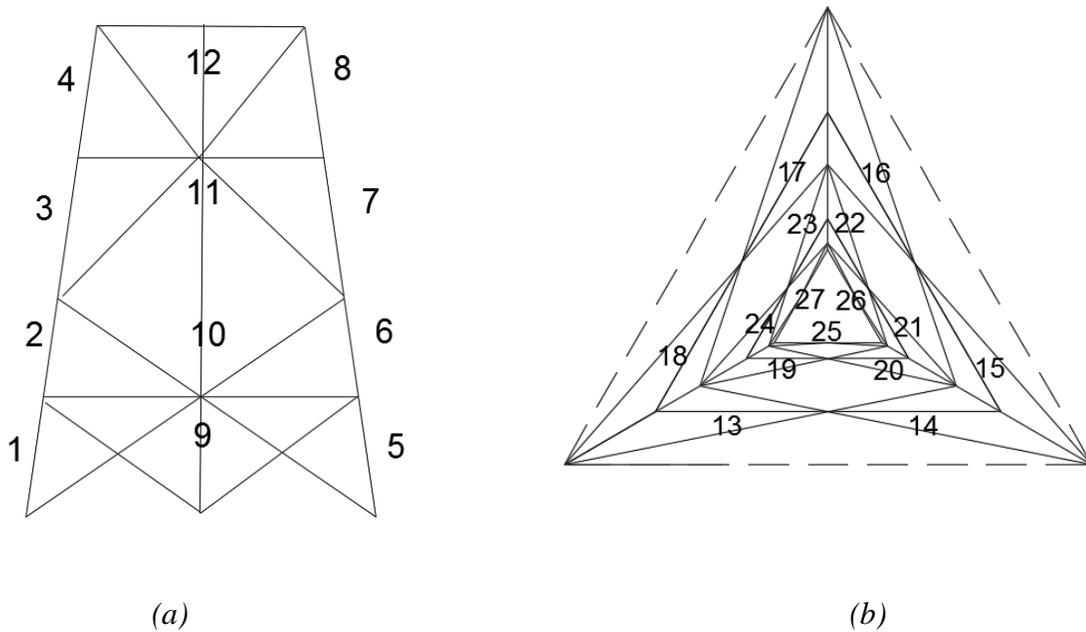
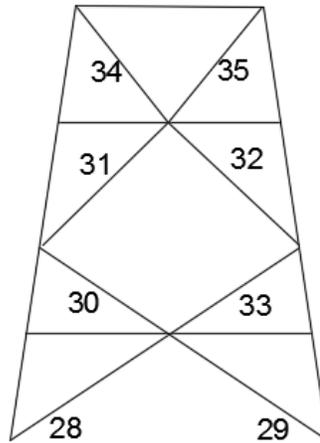


Figura 2.6 Nodos de las columnas.

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 Figura 2.7 y el análisis más detallado se muestra en el ANEXO 4.





(c)

Figura 2.7 (a) Barras que conforman las columnas (b) Barras que conforman los triángulos (c) Barras que conforman los rombos.

2.6 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.

2.7 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.

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, $\text{kgf}/\text{cm}^2 (\text{MPa})$

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:

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

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

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.

2.8 Parámetros de Algoritmos Genéticos

El Algoritmo Genético utiliza varios parámetros que influyen directamente en la obtención de los resultados finales. De estos parámetros se decidió variar la población con un valor inicial de 1500, el cual se fue incrementando en 100 hasta 2800 y luego en 200 hasta 5000, no siendo así con el parámetro generación cuyo valor fijo fue de 500. El resto de los parámetros se mantuvieron por defectos. (Ver Figura 1.5)

2.9 Parámetros de Enjambre de partículas

El enjambre de partículas utiliza varios parámetros fundamentales que permiten obtener valores que converjan con el óptimo global. Para ello se fijó un valor de iteración igual a 950 mientras que el número de partículas comenzó con 100 aumentando en un rango de 50 hasta llegar a 900. El resto de los parámetros se dejaron por defecto. (Ver Figura 1.6)

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 las técnicas heurísticas 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 variables del tipo discretas, correspondiente a las secciones transversales tipo angular de alas iguales.

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 Enjambre de Partículas, 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 los resultados obtenidos mediante las dos técnicas mencionadas anteriormente para de esta forma demostrar la eficiencia y superioridad de la aplicación de una de ellas en el proceso de diseño de una torre autosportada.

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 resultados se obtuvieron variando los valores de población comenzando con 1 500 hasta 5 000, hasta la población 2 800 se aumentaron los valores en rangos de 100 y a partir de 2 800 se aumentaron en rangos de 200 y la generación se mantuvo en 500 como se muestra en la Figura 3.1.



Figura 3.1 Resultados de los pesos obtenidos al aumentar las poblaciones.

En el Figura 3.1 se aprecia 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 3 400 donde se puede definir la convergencia a un único valor.

3.2 Análisis de los resultados de la optimización al aplicar Enjambre de partículas

Los resultados se obtuvieron variando los valores del número de partículas comenzando con 100 hasta 900, en rangos de 50 y la iteración se mantuvo en 950 como se muestra en la Figura 3.2.

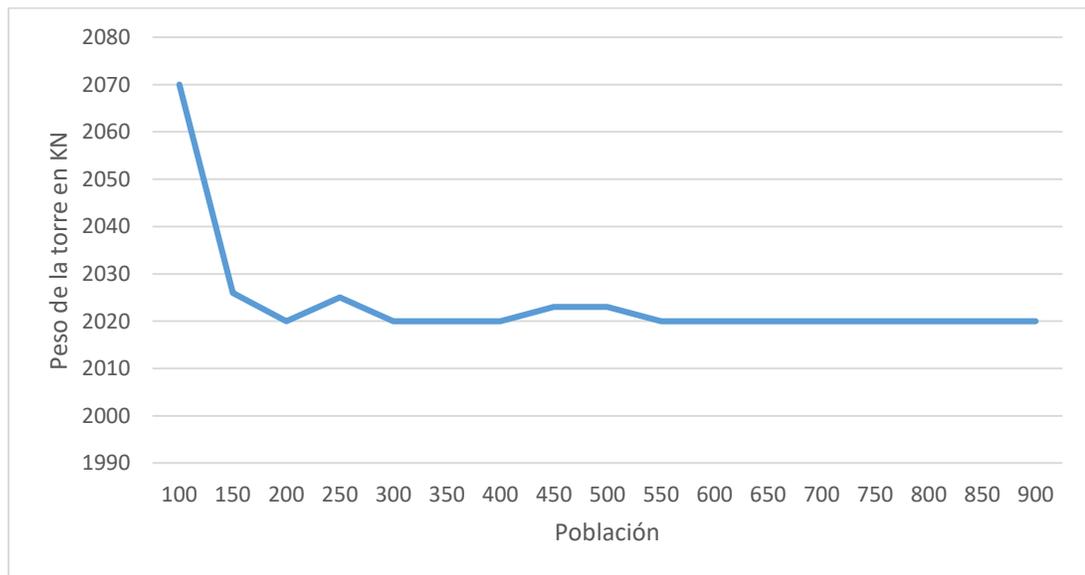


Figura 3.2 Resultados de los pesos obtenidos al aumentar el número de partículas.

En el Figura 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 550 donde se puede definir la convergencia a un único valor.

En la figura 3.3 se muestra el valor de óptimo global obtenido a través del algoritmo Enjambre de Partículas para un número de partículas igual a 900 y una iteración de 950, dicho valor fue encontrado en la iteración número 68.

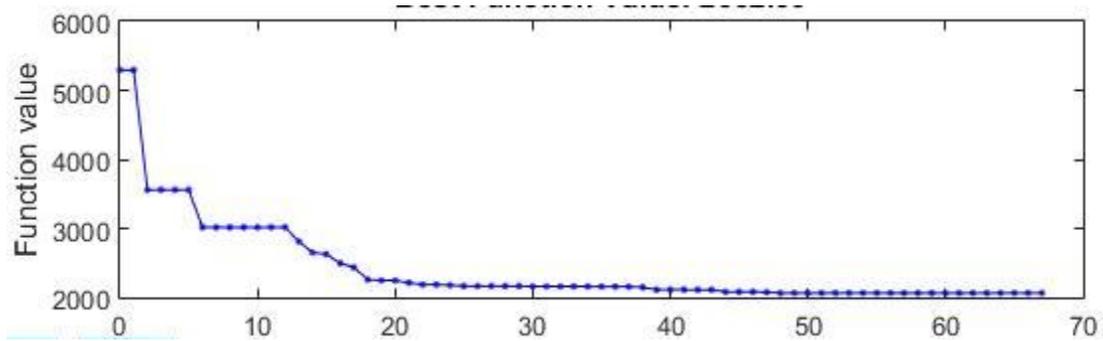


Figura 3.3 Valor de óptimo global en la iteración 68.

3.3 Tabla comparativa de los resultados obtenidos entre Algoritmo Genético y Enjambre Partículas

La tabla 3.1 precisa los valores obtenidos por cada una de las técnicas heurísticas utilizadas en la investigación en función de optimizar una torre autosoportada. Para ello se definieron las variables geométricas que en este caso están dadas por el área de las secciones transversales de los perfiles comerciales extraídos del catálogo de perfiles ACEROL. En el caso del Algoritmo Genético se fue variando los valores de población comenzando con 1 500 hasta 5 000 y la generación se mantuvo en 500, entre el rango de poblaciones de 3 400 hasta 5 000 existe convergencia entre los óptimos globales donde se obtuvo un valor de peso total de 2 020 kN en 10 028,125 segundos para un total de 25 corridas. Sin embargo, al utilizar el algoritmo de enjambre de partículas se fue variando los valores dirigidos al número de partículas comenzando con 100 hasta 900 y la iteración se mantuvo en 950 para un total de 17 corridas, entre el rango de poblaciones de 550 hasta 900 existe convergencia entre los óptimos globales, donde se obtuvo un valor de peso total igual al obtenido en el algoritmo anterior en un tiempo de 1 751,867 segundos, un 82 % más efectivo que Algoritmo Genético. (Ver Figura 3.4)

Tabla 3.1 Comparación entre ambas técnicas

Variable designada (cm2)	Algoritmo Genético		Enjambre de partículas	
	G 500	P 3 400	Np 550	Iter 950
A1	21.2		21.2	
A2	5.82		5.82	
A3	21.2		21.2	
A4	3.08		3.08	
A5	3.08		3.08	
A6	8.13		8.13	
A7	23.2		23.2	
A8	8.13		8.13	
A9	23.2		23.2	
Peso óptimo(KN)	2 020,30		2 020,30	
Total de corridas	25		17	
Tiempo mxm (s)	10 028,125 s		1 751,867	

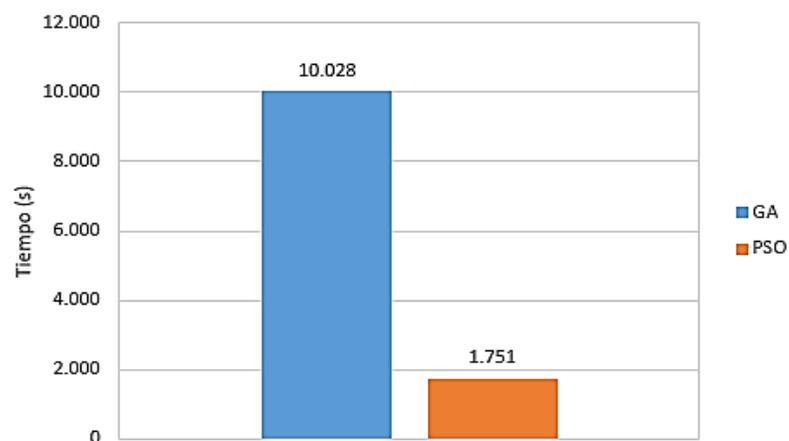


Figura 3.4 Valores del tiempo de ambas técnicas heurísticas.

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.

- 2- Al analizar los resultados obtenidos al aplicar los Algoritmos Genéticos en el proceso de optimización y los obtenidos al aplicar el algoritmo de Enjambre de Partículas se observa la eficiencia de la utilización de esta última ya que se obtiene valores de la función objetivo (peso) en igual magnitud, pero en un menor tiempo.

CONCLUSIONES

1. Debido a la complejidad y al carácter multifactorial del diseño de torres autoportadas, la tendencia mundial es la aplicación de heurísticas de optimización a dichas estructuras, siendo Algoritmos Genéticos y Enjambre de Partículas las más utilizadas.
2. La función optimizada es el peso de la estructura, las variables son del tipo discreta, correspondiendo al área de la sección transversal del catálogo de perfiles AISC.
3. Al analizar el resultado obtenido en el caso de estudio con la aplicación de Algoritmos Genéticos y el obtenido por Enjambre de Partículas, se demuestra la superioridad del segundo, que alcanza la convergencia en cuanto al valor óptimo de la función objetivo en menos corridas y en un 82 % menos de tiempo.

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. Ampliar el espectro de técnicas heurísticas a comparar utilizando funciones objetivas más complejas.

REFERENCIAS BIBLIOGRÁFICAS

- (AISC), A. I. O. S. C. 1994. Load and Resistance Factor Design Specification for Structural Steel Buildings. 2 ed.
- ADELI, H. & CHENGN.-T 1994. Concurrent genetic algorithms for optimization of large structures. *Journal of Aerospace Engineering*, 7, 276-296.
- AI, S. & WANG, Y. 2011. Application of improved genetic algorithms in structural optimization design. *Information and Management Engineering of Communications in Computer and Information Science*, 236, 480–487.
- ALBA, E. & TOMASSINI, M. 2002. Parallelism and evolutionary algorithms.
- BAKER, J. E. 1987. Reducing bias and inefficiency in the selection algorithm. *In Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application Hillsdale*, 14–21.
- BARBOSA, H. & LEMONGE, A. 2003. A new adaptive penalty scheme for genetic algorithms. *Information Sciences*, 156, 215–251.
- 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.
- BEIGHTLER, C. S., PHILLIPS, D. T. & WILDE, D. J. 1979. Foundations of optimization (2nd ed.). *Englewood Cliffs, NJ: Prentice Hall*. .
- BLUM, C. & ROLI, A. 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35, 268–308.
- BREMERMANN, H. J. 1962. Optimization through evolution and recombination. *In: Self-Organizing systems* 93–106.
- CAPARRINI, F. S. 2016. *PSO: Optimización por enjambres de partículas* [Online]. Available: <http://www.cs.us.es/~fsancho/?e=70> [Accessed 4/24/2018, 11:09 AM 2018].
- COELLO, C. A. C., LAMONT, G. B. & VELDUIZEN, D. A. V. 2007. Evolutionary Algorithms for Solving Multi-Objective Problems.
- 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. U., BIN, M. A. & JING, G. 2012. Research on the Optimal Layout of High-strength Steel in the Transmission Tower. *Physics Procedia*, 33, 619-625.
- DATTA, S. & DAVIM, J. P. (eds.) 2019. *Optimization in industry: Present practices and future scopes*, Cham (Switzerland): Springer Nature.
- DORIGO, M. & STÜTZLE, T. Ant colony optimization, a bradford book. *In: LONDON, E., MA: THE MIT PRESS CAMBRIDGE* (ed.).
- FELIX, J. & VANDERPLAATS, G. 1987a. Configuration optimization of trusses subject to strength, displacement and frequency constraints. *Journal of Mechanisms, Transmissions, and Automation in Design*, 109, 233-241.
- FELIX, J. & VANDERPLAATS, G. N. 1987b. Configuration optimization of trusses subject to strength, displacement and frequency constraints. *Journal of Mechanisms, Transmissions, and Automation in Design*, Vol.109, pp. 233–241.

- FRASER, A. 1958. Monte carlo analyses of genetic models. 181(4603), 208–209.
- GLOVER, F. & KOCHENBERGER, G. A. (eds.) 2003. *Handbook of Metaheuristics of International Series in Operations Research & Management Science*: Kluwer Academic Publishers/Springer, New York.
- GOLDBERG, D. E. 1989. Genetic Algorithms in Search, Optimization and Machine Learning.
- HANSEN, L. 1981. Computer program system for design of power transmission towers. *11th ONR Naval Structural Mechanics Symposium*, 12.37-12.45.
- JOHNSON, D. S. & MCGEOCH, L. A. 1997. The travelling salesman problem: A case study in local optimization. In E. H. L. Aarts & J. K. Lenstra (eds.), *Local search in combinatorial optimization*, 215–310.
- JONG, K. A. D. 1975. An analysis of the behavior of a class of genetic adaptive systems, University of Michigan.
- KAVEH, A. & GHAZAAN, M. I. 2018. Meta-heuristic Algorithms for Optimal Design of Real-Size Structures. *Springer International Publishing AG, part of Springer Nature 2018*, 123-137.
- KAVEH, A. & TALATAHARI, S. 2009a. Hybrid algorithm of harmony search, particle swarm and ant colony for structural design optimization. *Harmony Search Algorithms for Structural Design Optimization of Studies in Computational Intelligence*, 239, 159–198.
- KAVEH, A. & TALATAHARI, S. 2009b. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. *Computers & Structures*, 87, 267 – 283.
- KHODADADI, A. & BUELEOW, P. V. 2014. Form Exploration and GA-Based Optimization of Lattice Towers Comparing with Shukhov Water Tower.
- KOCER, F. Y. & ARORA, J. S. 2002. Optimal design of latticed towers subjected to earthquake loading. *Journal of Structural Engineering*, 128, 197-204.
- KOZA, J. R. 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. *MIT Press*, ISBN 0-262-11170-5.
- LEMONGE, A. & BARBOSA, H. 2004. An adaptive penalty scheme for genetic algorithms in structural optimization. *International Journal for Numerical Methods in Engineering*, 59, 703–736.
- LI, H., HUANG, H. X. & LIU, F. 2009. A heuristic particle swarm optimization method for truss structures with discrete variables. *Computers & Structures*, 87, 435 – 443.
- LI, L., HUANG, Z., LIU, F. & WU, Q. 2007. A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers & Structures*, 87, 435 – 443.
- LUH, G.-C. & LIN, D. 2011. Optimal design of truss-structures using particle swarm optimization. *Computers & Structures*, 89, 2221 – 2232.
- MAGALHAES, 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, 341-359.
- MAJID, K. & TANG, X. 1984a. The optimum design of pin-jointed space structures with variable shape. *Structural Engineer*, 62, 31-37.
- MAJID, K. & TANG, X. 1984b. Optimum design of pin-jointed space structures with variable shape. *Structural Engineer*, 62, 31-37.

- MARTÍN, P. & ELENA 2012. Análisis sísmico de modelos cubanos de torres autosoportadas de telecomunicaciones. *Revista Cubana de Ingeniería*, Vol.III, 25-34.
- MARTÍNEZ, S., PARÍS, J., COLOMINAS, I., NAVARRINA, F. & CASTELEIRO, M. 2014. Optimización mixta de estructuras de transporte de energía: aplicación del algoritmo de recocido simulado. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 30, 121-135.
- MARTINS, H. & GOMES 2011. Truss optimization with dynamic constraints using a particle swarm algorithm. *Expert Systems with Applications*, 38, 957 – 968.
- MILLÁN-PÁRAMO, C. 2017. Diseño óptimo de armaduras empleando optimización con ondas del agua. *INGE CUC*, 13, pp. 102-11.
- MOHAMMED, A. F., ÖZAKÇA, M. & TAYŞI, N. 2012a. Optimal design of transmission towers using genetic algorithm. *SDU Int J Technol Sci*, 4.
- MOHAMMED, A. F., ÖZAKÇA, M. & TAYŞI, N. 2012b. 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. 2006. Diseño óptimo de torres de alta tensión: Tratamiento conjunto de variables continuas y discretas. *7º Simposio Internacional de Estructuras, Geotecnia y Materiales de Construcción*.
- OMID BOZORG-HADDAD 2018. Advanced Optimization by Nature-Inspired Algorithms. *Studies in Computational Intelligence*, 720.
- PEREZ, R. & BEHDINAN, K. 2007a. Particle swarm approach for structural design optimization. *Computers and Structures*, 85, 1579–1588.
- PEREZ, R. E. & BEHDINAN, K. 2007b. Particle Swarm Optimization in Structural Design. In: FELIX T.S.CHAN & KUMARTIWARI, M. (eds.) *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*. InTech.
- PRENDES, B., BELLO, A. & DELCOZ, J. J. 2002. Entorno Gráfico Para la Optimización de Estructuras Metálicas en Base a Algoritmos Genéticos. .
- RAJ, P. P. & DURRANT, S. O. 1976. Optimum structural design by dynamic programming. *Journal of the Structural Division*, 102, 1575-1589.
- RAO, S. S. 2009. Engineering Optimization Theory and Practice(4thed.). *Copyright*.
- SAKA, M. P. 1984. Optimum design of space trusses with buckling constraints.
- SANCHEZ-CABALLERO, S., SELLES CANTÓ, M. Á., PLA-FERRANDO, R. & PEYDRO, M. 2012. Nuevas técnicas de optimización de estructuras. *3C Tecnología*, 1, 1-19.
- SCHAFFER, J. D., ESHELMAN, L. J. & OFFUTT, D. 1990. Spurious correlations and premature convergence in genetic algorithms. In *Proceedings of the First Workshop on Foundations of Genetic Algorithms (FOGA)*, 102–112.
- SCHREIBER, G. R. & MARTIN, O. C. 1999. Cut size statistics of graph bisection heuristics. *SIAM Journal on Optimization*, 10, 231–251.
- SCHUTTE, J. & GROENWOLD, A. 2003. Sizing design of truss structures using particle swarms. *Structural and Multidisciplinary Optimization*, 25, 261–269.

- SHEKEL, J. 1971. Test functions for multimodal search techniques. *In Proceedings of the Fifth Annual Princeton Conference on Information Science and Systems*. Princeton, NJ, USA: Princeton University Press.
- SHEPPARD, D. J. & PALMER, A. C. 1972. Optimal design of transmission towers by dynamic programming. *Computers & Structures*, 2, 455-468.
- 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. 2016. Optimization of Transmission Tower using Genetic Algorithm. *International Journal of Science and Research (IJSR)* 5.
- 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.
- SYSWERDA, G. 1991. A study of reproduction in generational and steady-state genetic algorithms. *Foundations of Genetic Algorithms 1*. Rawlins G.(Ed). Kaufmann, 94-101.
- TORT, C., ŞAHİN, S. & HASANÇEBİ, O. 2017. Optimum design of steel lattice transmission line towers using simulated annealing and PLS-TOWER. *Computers & Structures*, 179, 75-94.
- URSEM, R. K. 2003. *Models for evolutionary algorithms and their applications in system identification and control optimization*. University of Aarhus, Denmark, April 1, 2003.
- WANG, Q. & ARORA, J. S. 2004. Alternate formulations for structural optimization. *Palm Springs, CA*, . 2, 1413–1423.
- WANG, Q. J. 1991. The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. . *Water Resources Research*.
- WEISE, T. 2009. Global optimization algorithms—theory and application. *Version: June 26, 2009*. .
- YANG, X.-S. 2014. *School of science and technology*. London:Middlesex University London.
- ŽILINSKAS, A. 1978. Algorithm as 133: Optimization of one-dimensional multimodal functions. *Applied Statistics*, 27, 367–375.

ANEXOS

Anexo 1 Parametrización de los puntos de los triángulos.

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

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

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

$$7-\left(\frac{(H1+H2)}{36} \cdot D \cdot \frac{\sqrt{3}}{2}; \frac{(H1+H2)}{36} \cdot D \cdot \frac{1}{2}; (H1+H2)\right)$$

$$8-\left(a - \frac{(H1+H2)}{36} \cdot D \cdot \frac{\sqrt{3}}{2}; \frac{(H1+H2)}{36} \cdot D \cdot \frac{1}{2}; (H1+H2)\right)$$

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

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

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

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

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 17 (6, 16, A, E);

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Barra 16 (19, 6, 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);                                R1 = RadiosGiro(round(X(:, 5)));
D = X(:, 2);                                A2 = Perfiles(round(X(:, 6)));
H1 = X(:, 3);                               R2 = RadiosGiro(round(X(:, 6)));
H2 = X(:, 4);                               A3 = Perfiles(round(X(:, 7)));
H3 = 1500-H1-H2;                           R3 = RadiosGiro(round(X(:, 7)));

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

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 ntos 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(vectorpesc

```

```

end

RadioGiro = [R1 R2 R2 R3 R1 R2 R2 R3 R1 R2 R2 R3 R1 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

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

                Fcr = 0 77*(fe);

```

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

```
if klr(i)>200
```

```
    Pn(i)= 0.0001;
```

```
end
```

```
end
```

```
end
```

```
end
```

```
end
```

```
Vectorratio = [abs(vectoraxial./Pn)];
```

```
Penal = 1;
```

```
for i=1:1:length(ELEMENTS)
```

```
    if Vectorratio(i)>1
```

```
        Penal = Penal+100;
```

```
    end
```

```
end
```

```
Pesoptimo=Pesototal*Penal;
```

```
Y=Pesoptimo;
```

```
end
```