

*Universidad de Matanzas
Sede "Camilo Cienfuegos"
Facultad de Ciencias Técnicas*



***OPTIMIZACIÓN DEL DISEÑO DE TORRES AUTOSOPORTADAS,
REVISIÓN DE LITERATURA***

Trabajo de Diploma en Ingeniería Civil

Autor: David Rodríguez Cardoso

Tutores: Ing: Juan Carlos Verdecia Somoano

Ing: Javier A García Miranda

Matanzas, 2018

DECLARACIÓN DE AUTORIDAD

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

NOTA DE ACEPTACIÓN

DEDICATORIA

A María y Miguelo a quienes les hubiera gustado verme.

AGRADECIMIENTOS

- **A mis padres.**
- **A Gladys por ayudarme cuando me hizo falta.**
- **A mis amigos cuando me tiraron un cabo.**
- **A mis tutores.**
- **A toda la gente que me tendió una mano o que me empujó para que llegara hasta aquí.**

RESUMEN

Las torres autoportadas son estructuras muy sensibles a los procesos de optimización debido a que tradicionalmente su diseño se hacía por métodos basados en la experiencia del proyectista y en ensayos de prueba y error, lo que devenía en que fuesen construidas con muchas reservas en cuanto al aprovechamiento de la capacidad estructural del acero, provocando costos extras. En las últimas décadas han sido desarrolladas un sinnúmero de técnicas de optimización de estructuras que han sido empleadas en torres autoportadas. Sin embargo, los procesos actuales de diseño utilizados en Cuba siguen siendo tradicionales. En este trabajo se realizará una revisión bibliográfica de la literatura existente sobre el tema de optimización de torres autoportadas a fin de definir cuáles son sus métodos y parámetros más empleados por los autores. También se caracterizarán algunas de las técnicas de optimización de estructuras más usada en el mundo. Como consecuencia en la siguiente investigación se obtuvo un documento guía para la posterior profundización de la optimización de torres autoportadas. Metodológicamente puede ser utilizado como una base para el estudio y aplicación en múltiples ramas de la Ingeniería Civil y de forma práctica proporciona una herramienta para facilitar y definir los métodos y parámetros para la optimización de torres autoportadas.

Palabras claves: Torres autoportadas, optimización de estructuras, revisión de literatura, Ingeniería Civil.

ABSTRACT

Self-supporting towers are structures that are very sensitive to optimization processes because their design was traditionally done by methods based on the experience of the designer and trial and error. What happened in that they were built with many reserves regarding the use of the structural capacity of steel, causing extra costs. In the last decades have been developed a number of techniques for optimizing structures that have been used in self-supporting towers. However, the current design processes used in Cuba are still traditional. In this work, a literature review of the existing literature on the topic of optimization of self-supported towers will be carried out in order to define which methods and parameters are most used by the authors. It will also characterize some of the most used structures optimization techniques in the world. As a consequence, in the following investigation a guide document was obtained for the further deepening of the optimization of self-supported towers. Methodologically it can be used as a basis for the study and application in multiple branches of Civil Engineering and in a practical way, it provides a tool to facilitate, define the methods and parameters for the optimization of self-supported towers.

Key words: Self-supporting towers, optimizing structures, literature review, Civil Engineering.

TABLA DE CONTENIDO

Introducción	1
Capítulo 1 Acerca de las técnicas de optimización.....	4
Introducción del capítulo.	4
1.1 Tipos de optimización.....	4
1.1.1 Optimización del tamaño.	4
1.1.2 Optimización de material	5
1.1.3 Optimización de la geometría:	5
1.1.4 Optimización topológica:	5
1.1.5 Optimización multiobjetivo.....	6
1.2 Técnicas de optimización.....	7
1.2.1 Programación Matemática:	7
1.2.2 Programación Lineal:	7
1.2.3 Programación no lineal:	8
1.2.4 Programación Cuadrática Secuencial:.....	8
1.2.5 Programación lineal mixta. (MILP)	9
1.2.6 Programación dinámica (PD).....	10
1.3 Procesos Heurísticos y Metaheurísticos.	10
1.4. Algoritmos Evolutivos.....	11
1.4.1 Estrategias Evolutivas	11
1.4.2 Programación Evolutiva:.....	12
1.4.3 Algoritmos Genéticos:	12
1.5 Otras técnicas metaheurísticas.	12
1.5.1 Optimización por enjambre de partículas.....	12
1.5.2 Optimización por algoritmo recocido simulado del acero.	14
1.5.3 Optimización por algoritmo Ondas del Agua (WWO)	16
1.5.4 Optimización por <i>Teaching-Learning-Based Optimization</i> (TLBO).....	16
1.5.5 Optimización por colonia de Hormigas (ACO)	17
1.5.5 Optimización por algoritmo <i>Colliding Bodies Optimization</i> (ECBO).....	17
1.5.6 Optimización por algoritmo <i>Vibrating Particles Sistem</i> (VPS)	17
1.6 Conclusiones parciales del capítulo	18
Capítulo 2 Revisión de trabajos acerca de la optimización de torres autosoportadas	19
Introducción del capítulo.....	19
2.1 Optimización de torres empleando programación dinámica.....	19

2.2 Optimización mediante programación matemática no lineal	20
2.3 Optimización mediante programación lineal secuencial.....	21
2.4 Optimización mediante métodos de optimización propios de los autores	21
2.5 Optimización de torres empleando algoritmos genéticos	34
2.6 Optimización de torres empleando el algoritmo recocido simulado del acero .	57
2.7 Optimización de torres empleando el algoritmo ondas del agua (WWE).....	69
2.8 Optimización de torres empleando los algoritmos <i>Colliding Bodies Optimization (CBO)</i> , <i>Enhanced Colliding Bodies Optimization (ECBO)</i> , <i>Vibrating Particles System (VPS)</i> y el algoritmo híbrido llamado MDVC-UVPS	71
2.9 Conclusiones parciales del capítulo.	74
Conclusiones.....	75
Recomendaciones	76
Referencias bibliográficas	78

INTRODUCCIÓN

Las torres autosoportadas de celosía de acero son estructuras relativamente jóvenes. Su desarrollo solo fue posible después de la Revolución Industrial, momento en que fue posible producir el acero en cantidades industriales. El mayor auge de estas estructuras ocurre a partir del siglo XX con la expansión de la electrificación, el surgimiento de las grandes centrales generadoras de energía y el desarrollo explosivo de las telecomunicaciones.

En Cuba sus principales usos son como portadoras de antenas y como soportes de líneas de alta tensión. En cualquiera de estos casos se trata de estructuras esbeltas, muy ligeras que tienden a colapsar ante los embates de vientos huracanados. Debido al alto precio de los materiales, representan del 30 al 40% del costo de una línea de transmisión de alta tensión (Alzate Acevedo, 2016; Sony y Airin, 2016) y al fuerte impacto que provoca en la sociedad su colapso, los efectos sociales para la sociedad del colapso de una torre de telecomunicaciones son similares a los que produciría un puente o estructura similar (Efthymiou, *et al.*, 2009); es necesario lograr un equilibrio entre resistencia y costo.

Tradicionalmente el proceso de diseño de una torre de celosía autosoportada se basa principalmente en la experiencia del diseñador. Parte de un predimensionamiento inicial y se comprueba la resistencia a la acción de las cargas. Luego se repite el proceso variando dimensiones hasta obtener un diseño a que a juicio del diseñador mejor cumple con los criterios de resistencia y economía.

Sin embargo, la tendencia en los últimos años es a incrementar el empleo de herramientas informáticas para obtener diseños con un comportamiento estructural más eficientes que los tradicionales, lo que incide directamente en su costo. Pero en Cuba la metodología para tal efecto no está muy desarrollada.

Lo que conduce a la siguiente situación problemática:

La necesidad de definir el estado actual de las técnicas de optimización en el diseño de torres autosoportadas según la bibliografía más actualizada del tema.

El Objetivo General es definir el estado actual del conocimiento sobre la optimización del diseño de torres autosoportadas.

Objetivos específicos:

1. Caracterizar el surgimiento y evolución de las técnicas de optimización.
2. Definir los métodos y parámetros para la optimización estructural de torres autoportadas según la bibliografía consultada.

Como resultado de la investigación se espera obtener un documento guía para la posterior profundización de esta temática. Este trabajo posee valor metodológico, porque puede ser utilizado como una base para el estudio y aplicación de métodos de optimización del diseño de torres autoportadas; el valor práctico radica en proporcionar una herramienta que facilita y orienta la comprensión del funcionamiento de los métodos y técnicas de optimización aplicados a torres autoportadas.

Métodos Teóricos

Histórico – Lógico: Permitirá recorrer los antecedentes hasta llegar a la situación actual de los algoritmos genéticos en la resolución de problemas de optimización. El método de análisis histórico estudia la trayectoria real de los fenómenos y acontecimientos en el curso de su historia y el método lógico investiga las leyes generales de funcionamiento y desarrollo.

Análisis – Síntesis: Para el procesamiento de las fuentes de información a fin de determinar los diferentes enfoques y criterios relacionados con la resolución de problemas de optimización aplicando algoritmos genéticos.

Inducción – Deducción: Aplicado por el autor para generalizar los rasgos más importantes obtenidos a partir de la documentación para definir los modelos y analizar las investigaciones ingenieras.

Métodos Empíricos: Para desarrollar la investigación se consultaron diferentes fuentes en formato digital.

El trabajo de diploma queda estructurado de la manera siguiente: resumen, introducción, desarrollo, conclusiones, recomendaciones, bibliografía consultada y anexos. En la Introducción se define el problema científico de la investigación, así como la importancia del tema para el país. Además se precisa el objetivo general y los objetivos específicos.

El desarrollo consta de dos partes fundamentales, en la primera se caracterizan algunas de las técnicas de optimización de estructuras más usadas que han sido empleadas para

optimizar torres autoportadas. En la segunda se definen los métodos y parámetros para optimizar torres según la bibliografía más actual del tema.

CAPÍTULO 1 ACERCA DE LAS TÉCNICAS DE OPTIMIZACIÓN

Introducción del capítulo.

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

Las 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 las la optimización combinatoria, la computación evolutiva y otras técnicas heurísticas y metaheurísticas.

En este capítulo se caracterizarán algunas de las técnicas de optimización utilizadas en el mundo, haciendo especial énfasis en aquellas que han sido empleadas en torres autosoportadas.

1.1 Tipos de optimización.

1.1.1 Optimización del tamaño.

La optimización del tamaño, 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)

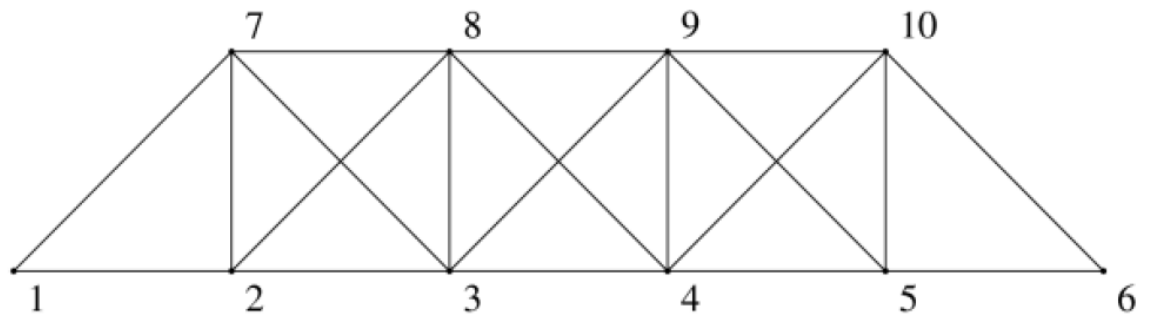


Figura 1.1 Optimización del tamaño. (Sanchez-Caballero, et al., 2012)

1.1.2 Optimización de material

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

1.1.3 Optimización de la geometría:

La reducción de peso se logra cambiando la conectividad entre nodos, manteniendo fijas las coordenadas nodales y las secciones rectas. (Sanchez-Caballero, et al., 2012) (Figura 2)

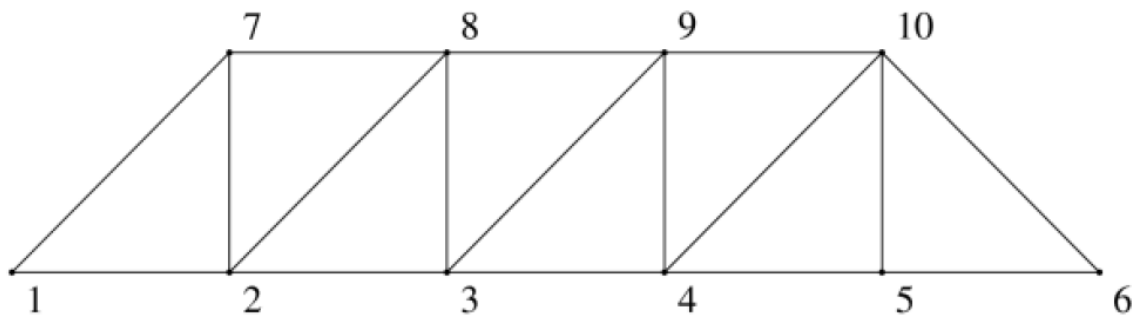


Figura 1.2. Optimización de la geometría. (Sanchez-Caballero, et al., 2012)

1.1.4 Optimización topológica:

La reducción de peso se logra cambiando las coordenadas y conectividad nodales, manteniendo fijas las secciones rectas. (Figura 3)

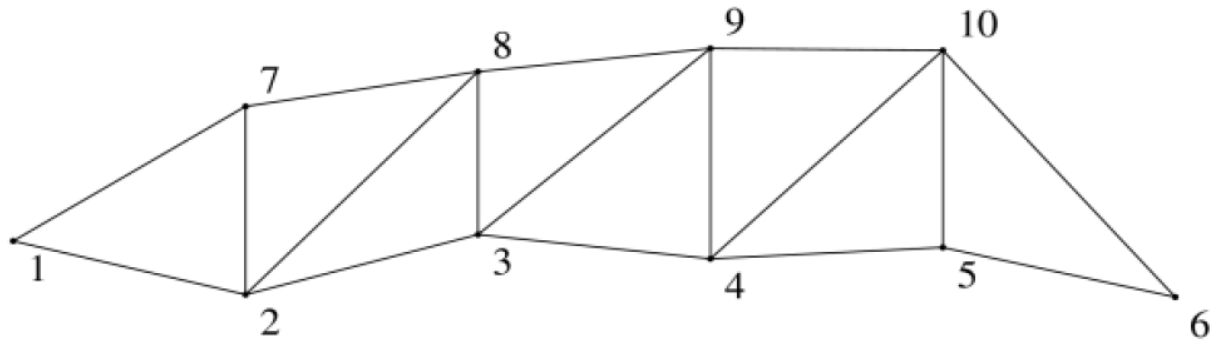


Figura 1.3. Optimización topológica (Sanchez-Caballero, et al., 2012)

1.1.5 Optimización multiobjetivo

Un problema de optimización multiobjetivo posee la siguiente forma:

$$\begin{aligned} \min \vec{f}(\vec{x}) &= (f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})) \\ \text{s.a. } \vec{x} &\in X, \vec{f} \in Y \end{aligned} \quad (1.1)$$

Donde $X \subset R^n$ es el espacio de decisión y $\vec{x} = (x_1, x_2, \dots, x_n) \in X$ es el vector de decisión, $Y \subset R_m$ es el espacio objetivo $\vec{f} \in Y$ es el vector objetivo compuesto por las funciones objetivo m ; $f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})$.

Cuando dos objetivos entran en conflicto la mejora de uno provoca el detrimento de otro, no existe una solución sencilla que optimice ambos objetivos al mismo tiempo. Como consecuencia existe un grupo de soluciones óptimas que "comercian" con los objetivos. En optimización multiobjetivo se dice que una solución x_1 domina a otra x_2 (denotado como $\vec{x}_1 \succ \vec{x}_2$) si:

$$\begin{cases} \forall_i \in 1, 2, \dots, m: f_i(\vec{x}_1) \leq f_i(\vec{x}_2) \\ \exists j \in 1, 2, \dots, m: f_j(\vec{x}_1) < f_j(\vec{x}_2) \end{cases} \quad (1.2)$$

Si una solución \vec{x} no puede ser dominada por otra solución \vec{x} se dice que se alcanzó el óptimo de Pareto y la unión de todas las \vec{x}^* es llamada Frente de Pareto. (Zhou, et al., 2011)

1.2 Técnicas de optimización.

1.2.1 Programación Matemática:

Los problemas de optimización se pueden, en primer lugar, clasificar en problemas que involucran solamente variables continuas y en problemas con variables continuas y discretas (o solamente discretas). Los principales problemas de optimización con variables continuas incluyen la Programación Lineal (LP) y la Programación no lineal (NLP)ⁱ. Una importante subclase de la programación lineal es el problema lineal complementario (LCP) y subproblemas importantes de programación no lineal incluyen la programación cuadrática (QP). En el caso de programación no lineal hay dos distinciones importantes a tener en cuenta. La primera es si el problema no lineal es convexo y la segunda es si el problema es diferenciable. Si el problema incluye variables discretas se puede nuevamente distinguir entre problemas lineales con variables enteras (MILP) y problemas no lineales con variables enteras (MINLP) ambas siglas del inglés *Mixed Integer (non) Linear Programming*.

Un problema de programación matemática presenta la forma:

$$\begin{aligned} \min & z = f(x, y) \\ \text{s.a.} & h(x, y) \\ & g(x, y) \leq 0 \\ & x \in X \subseteq \mathfrak{R}^n \\ & y \in \{0,1\}^m \end{aligned} \tag{1.3}$$

Donde f representa una función objetivo escalar, h y g son restricciones del problema en forma de igualdades y desigualdades, las variables x son variables continuas y las variables y son variables binarias. (Caballero y Grossman, 2007)

1.2.2 Programación Lineal:

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

$$\begin{aligned} \min & : z = c^T x \\ \text{s.a.} & Ax \leq b \\ & x \geq 0 \end{aligned} \tag{1.4}$$

El método estándar para resolver un problema lineal es el método simplex, desarrollado por Dantzing en los años cuarenta. Véase (Dantzing, 1963). (Caballero y Grossman, 2007)

1.2.3 Programación no lineal:

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

$$\begin{aligned} \min : & f(x) \\ \text{s.a.} & h(x) = 0 \\ & g(x) \leq 0 \end{aligned} \tag{1.5}$$

Dentro de los algoritmos para resolver NLPs aquellos que requieren un menor número de evaluaciones de la función objetivo son los basados en la programación cuadrática sucesiva (SQP –*successive quadratic programming*–) que ha dado lugar a una serie de algoritmos que se adaptan muy bien a un gran número de problemas con diferentes estructuras. (Caballero y Grossman, 2007)

1.2.4 Programación Cuadrática Secuencial:

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

La SQP se basa en aplicar el método de Newton a las condiciones de optimalidad de *Karush –Khun-Tucker* (KKT) del problema (3), que vienen dadas por las siguientes ecuaciones:

$$\begin{aligned}
& \nabla f(x^*) + \nabla h(x^*)\lambda + \nabla g(x^*)\mu = 0 \\
& h(x^*) = 0 \\
& \zeta g(x^*) + s = 0 \\
& SMe = 0 \quad C1 \\
& (s, \mu) \geq 0 \quad C2
\end{aligned} \tag{1.6}$$

Donde $e = [1, 1, 1, \dots, 1]^T$, λ es el vector de multiplicadores de Lagrange asociado a las restricciones de igualdad y μ el vector de multiplicadores de Lagrange de las desigualdades. $S = \text{diag} \{s\}$; $M = \text{diag} \{ \mu \}$

Las condiciones de complementariedad –ecuaciones (c1) y (c2) del problema (4)- presentan una importante dificultad cuando se resuelve el conjunto de ecuaciones asociado a las condiciones de KKT: Cerca de la solución las ecuaciones (c1) y los límites activos (c2) son dependientes y por lo tanto forman un sistema de ecuaciones mal condicionado. Los algoritmos basados en SQP solucionan este problema de dos formas diferentes: utilizando un método de conjunto activo o utilizando funciones de barrera. (Caballero y Grossman, 2007)

1.2.5 Programación lineal mixta. (MILP)

Un problema de programación lineal mixta se puede describir de la siguiente manera:

$$\begin{aligned}
& \min : c^T x + d^T y \\
& \text{s.a } Ax + By \leq b \\
& x \in X \subseteq \mathfrak{R}^n \\
& y \in \{0,1\}
\end{aligned} \tag{1.7}$$

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

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

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

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

1.2.6 Programación dinámica (PD)

La programación dinámica es un procedimiento matemático diseñado principalmente para mejorar la eficiencia de cálculo de problemas de programación matemática seleccionados, descomponiéndolos en subproblemas de menor tamaño y por consiguiente más fácil de calcular. La PD normalmente resuelve el problema en etapas. Los cálculos en las diferentes etapas se enlazan a través de cálculos recursivos de manera que se genere una solución óptima factible a todo el problema. (Rojas, 2005)

1.3 Procesos Heurísticos y Metaheurísticos.

“Se califica de heurístico a un procedimiento para el que se tiene un alto grado de confianza en que encuentra soluciones de alta calidad con un coste computacional razonable, aunque no se garantice su optimalidad o su factibilidad, e incluso, en algunos casos, no se llegue a establecer lo cerca que se está de dicha situación. Se usa el calificativo heurístico en contraposición a exacto“.

Los problemas altamente complejos son difíciles de resolver y se les denomina Problemas NP, se clasifican en conjuntos o clases de complejidad (L, NL, P, P Completo, NP, NP-Completo, NP Duro...). La mayor parte de los problemas en teoría de la complejidad tienen que ver con los problemas de la clase NP de decisión, y corresponden a problemas a los cuales no podemos garantizar encontrar una mejor solución en un tiempo polinómico razonable. Es aquí donde los métodos heurísticos se convierten en procedimientos eficientes para encontrar “buenas soluciones” aunque no se pueda comprobar que sean óptimas. En estos métodos, la rapidez del proceso (que siempre es menor que el tiempo de una solución por otros métodos) es tan importante como la calidad de la solución obtenida. (Suárez, 2011)

El término metaheurística fue introducido por Fred Glover en 1986 y desde entonces se han presentado muchas propuestas de pautas para diseñar mejores procedimientos de

solución a problemas combinatorios. Los profesores Osman y Kelly (1995) introdujeron la siguiente definición: “Los procedimientos Metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas de difícil optimización combinatoria, en los que los heurísticos clásicos no son efectivos”. (Suárez, 2011)

1.4. Algoritmos Evolutivos.

Este término es empleado para describir sistemas de resolución de problemas de optimización o búsqueda basados en el ordenador empleando modelos computacionales de algún mecanismo de evolución conocido como elemento clave en su diseño e implementación.

Los algoritmos evolutivos trabajan con una población de individuos, que representan candidatos a la solución de un problema. Esta población se somete a ciertas transformaciones y después a un proceso de selección que favorece a los mejores. Cada ciclo de transformación y selección constituye una generación, de forma que después de cierto número de generaciones se espera que el mejor individuo de la población esté cerca de la solución buscada. Los algoritmos evolutivos combinan la búsqueda aleatoria, dada por las transformaciones de la población, con una búsqueda dirigida dada por la selección.

Principales Componentes:

Población de individuos, que son una representación (no necesariamente directa) de posibles soluciones.

Procedimiento de selección basado en la aptitud de los individuos para resolver el problema.

Procedimiento de transformación para construir nuevos individuos a partir de los anteriores.

1.4.1 Estrategias Evolutivas

Estrategias Evolutivas: Técnica desarrollada por Rechenberg y Schwefel y extendida por Herdy, Kursawe, Ostermeier, Rudolph, y otros, fue diseñada inicialmente con la meta de resolver problemas de optimización discretos y continuos, principalmente experimentales y considerados difíciles. Trabaja con vectores de números reales y con desviaciones estándar que codifican las posibles soluciones de problemas numéricos. Utiliza recombinación o cruce (crossover aritmético), mutación y la operación de

selección, ya sea determinística o probabilística, elimina las peores soluciones de la población y no genera copia de aquellos individuos con una aptitud por debajo de la aptitud promedio. (Rechenberg, 1973; Schwefel, 1977)

1.4.2 Programación Evolutiva:

Técnica introducida por Fogel y extendida por Burgin, Atmar y otros, inicialmente fue diseñada como un intento de crear inteligencia artificial. La representación del problema se realiza mediante números reales (cualquier estructura de datos), y emplea los mecanismos de mutación y selección. El procedimiento es muy similar a las estrategias evolutivas con la diferencia de que no emplea la recombinación, de tal forma que son denominadas en conjunto algoritmos evolutivos como una manera de diferenciarlas de los algoritmos genéticos. (Fogel, 1967)

1.4.3 Algoritmos Genéticos:

Los Algoritmos Genéticos (AG): Fueron desarrollados por John H. Holland y sus colaboradores. Utilizaron inicialmente la codificación binaria, aunque en la actualidad también se ha extendido a la codificación con números reales. Emplea operadores genéticos de selección, recombinación y mutación teniendo mayor relevancia los dos primeros operadores. Para la formación de los descendientes la reproducción es sexual, es decir, hay intercambio del material genético de los padres mediante los operadores genéticos, cruce y mutación. En este caso, el operador mutación se considera un operador de segundo orden frente al operador cruce. (Holland, 1975)

1.5 Otras técnicas metaheurísticas.

1.5.1 Optimización por 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)

Estructura básica de un algoritmo PSO.

1. 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.
2. 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.
3. 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 .
4. Se actualiza la posición de cada partícula usando su posición previa y se actualiza la velocidad usando las ecuaciones:

$$\begin{aligned} x_{k+1}^i &= x_k^i + v_{k+1}^i \Delta t \\ v_{k+1}^i &= wv_k^i + c_1 r_1 \frac{(p_k^i - x_k^i)}{\Delta t} + c_2 r_2 \frac{(p_k^g - x_k^i)}{\Delta t} \end{aligned} \quad (1.8)$$

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 $\{0, 1\}$, los términos c_1 y c_2 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 y Behdinan, 2007)

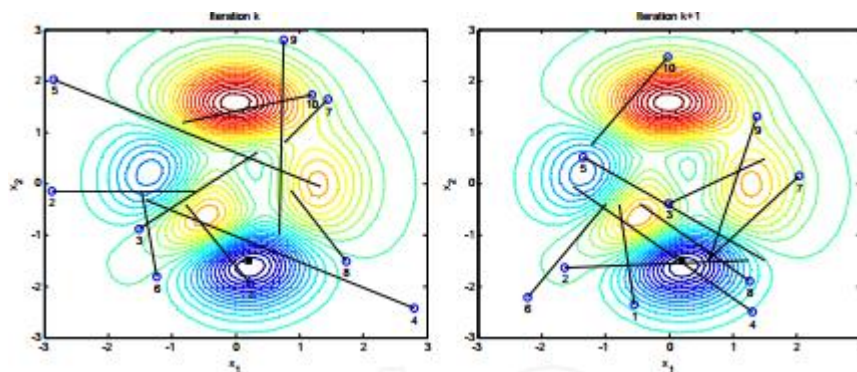


Figura 1.5. Actualización de los vectores velocidad y posición. (Perez y Behdinan, 2007)

1.5.2 Optimización por algoritmo recocido simulado del acero.

El algoritmo de «recocido simulado», o «*simulated annealing*» en literatura anglosajona, se basa en la analogía con el recocido de los metales, proceso que busca la configuración cristalina que proporciona una menor energía interna del sistema. Así, este procedimiento es análogo a un proceso de optimización en el que se busca la distribución de variables y que minimiza la función objetivo del problema, tal y como se puede observar en la tabla 1.1.

Tabla 1.1 Analogía entre la termodinámica y la optimización (Martínez, et al., 2012)

<i>Termodinámica</i>	<i>Optimización</i>
Configuración	Solución factible
Configuración fundamental	Solución óptima
Energía de la configuración	Función objetivo
Temperatura	Estado de la optimización

Así, el algoritmo comprueba de forma aleatoria, a partir de un estado o una configuración inicial, la modificación de la energía del sistema entre sus estados vecinos

(entorno factible), de forma que, si la energía es menor, el estado se acepta y se reinicia el proceso a partir de esa nueva configuración.

Función de energía:

La función de energía refleja el estado del sistema en un instante determinado del proceso de optimización, aportando la información necesaria sobre la función objetivo.

La temperatura

La temperatura es el parámetro fundamental que gobierna la evolución y el comportamiento del algoritmo. En consecuencia, es necesario llevar a cabo un estudio y una caracterización detallada del proceso de enfriamiento para garantizar su correcto funcionamiento. En este sentido, la temperatura gobierna no solo el número de pruebas o la longitud del proceso evolutivo, sino que también condiciona las probabilidades de transición de un estado de energía a otro.

La temperatura en el algoritmo se caracteriza por 4 aspectos fundamentales:

- La temperatura inicial.
- La constante de optimalidad.
- El esquema de enfriamiento que proporciona y controla la velocidad de enfriamiento.
- La temperatura de parada.

La constante de optimalidad:

Otro de los aspectos que influye notablemente sobre las probabilidades de transición es la constante de optimalidad. Dicha constante es fruto de la analogía con el recocido de los metales, siendo la equivalencia numérica de la constante de Boltzman.

El esquema de enfriamiento:

No existe un consenso general acerca del tipo de esquema que se debe emplear. Sin embargo, sí existe una aceptación general de que este tipo de algoritmos mejora su funcionamiento cuando la velocidad de enfriamiento disminuye, como ocurre con el enfriamiento de los metales. (Martínez, *et al.*, 2012)

1.5.3 Optimización por algoritmo Ondas del Agua (WWO)

El algoritmo WWO se inspira en los modelos de onda de agua superficial para resolver problemas de optimización. Supongamos que tenemos un problema de minimización con la función objetivo f . En WWO, el espacio de solución X es análogo al área del lecho marino, y la evaluación de la función objetivo de un punto $x \in X$ se mide directamente proporcional por su profundidad del lecho marino: cuanto más corta sea la distancia al nivel del agua inmóvil, menor será el valor de $f(x)$.

WWO mantiene una población de soluciones, cada una de las cuales es análoga a una onda que tiene una altura (o amplitud) $h \in \mathbb{Z}^+$ y una longitud de onda $\lambda \in \mathbb{R}^+$

Luego de la inicialización para cada onda, h se establece como una constante h_{\max} y λ se establece en 0,5.

Durante el proceso de solución del problema WWO realiza tres operaciones:

- 1) Propagación: El operador de propagación hace que las ondas con mejor valor de función objetivo busquen pequeñas áreas y las ondas de peor valor de función objetivo exploren grandes áreas.
- 2) Refracción: El operador de refracción ayuda a las ondas a escapar del estancamiento en un mínimo local y, así, mejora la diversidad de la población y reduce la convergencia prematura.
- 3) Ruptura: El operador de ruptura permite una búsqueda intensiva en un área (potencialmente) prometedora.

La combinación de los tres operadores proporciona al algoritmo un buen equilibrio entre exploración y explotación. (Millán-Páramo, 2017)

1.5.4 Optimización por *Teaching-Learning-Based Optimization* (TLBO)

Este método trabaja sobre el efecto de la influencia de un profesor en los alumnos. Al igual que otras metaheurísticas inspiradas en la naturaleza, TLBO también es un método basado en la población y utiliza una población de soluciones para proceder a la solución global. La población se considera como un grupo de aprendices o una clase de aprendices. El proceso de TLBO se divide en dos partes: la primera parte consiste en 'Fase del profesor' y la segunda parte consiste en la 'Fase del alumno'.

'Fase del profesor' significa aprender del maestro y "Fase de aprendizaje" significa aprender mediante la interacción entre los estudiantes. (Rao, *et al.*, 2011)

1.5.5 Optimización por colonia de Hormigas (ACO)

Los algoritmos ACO (*Ant Colony Optimization*) son programas constructivos.

En cada iteración del algoritmo cada hormiga construye una solución al problema a través de un grafo.

Cada arista representa las posibles opciones que el insecto puede tomar y tiene asociada el siguiente tipo de información:

- Información heurística: en ésta se mide la preferencia heurística que tienen las hormigas para moverse de un nodo a otro. El camino recorrido de un nodo a otro es una arista. Esta información no es modificada durante la ejecución del algoritmo.
- Información de los rastros de feromonas artificiales: en ésta se mide la deseabilidad aprendida en el movimiento de un nodo a otro, lo cual busca imitar la feromona real que depositan las hormigas naturales. Este tipo de información es modificada mientras que se ejecuta el algoritmo dependiendo de las soluciones encontradas por los insectos. (Algarin, 2010)

1.5.5 Optimización por algoritmo *Colliding Bodies Optimization* (ECBO)

Este algoritmo está conceptualizado utilizando las dimensiones unidireccionales entre cuerpos, donde cada agente solución comienza considerada como un objeto con masa. Después de la colisión entre dos cuerpos en movimiento, habiendo especificado masa y velocidad esos cuerpos son separados con nuevas velocidades. Esta colisión causa que los agentes soluciones se muevan hacia mejores posiciones. (Kaveh y Mahdavi, 2015)

1.5.6 Optimización por algoritmo *Vibrating Particles System* (VPS)

En este método las soluciones candidatas son consideradas como partículas que gradualmente se aproximan a sus posiciones de equilibrio. Las posiciones de equilibrio son archivadas dentro de la población y la mejor posición del historial es la que tiene un equilibrio apropiado entre exploración y explotación. (Kaveh y Ghazaan, 2016)

1.6 Conclusiones parciales del capítulo

Del análisis crítico de la revisión bibliográfica se concluyen las ideas que a continuación se mencionan.

1. Las técnicas heurísticas y metaheurísticas han probado obtener mejores resultados que las técnicas basadas en programación matemática debido a su adaptabilidad y a poder emplear variables continuas y discretas simultáneamente.
2. Las técnicas de optimización más modernas siguen la tendencia de utilizar procesos estocásticos inspirados en la naturaleza para resolver el problema de optimización y se alejan de la necesidad de encontrar el óptimo para buscar soluciones buenas aunque no sean las mejores.
3. Los algoritmos genéticos son de las técnicas heurísticas de optimización una de las más robustas debido a la simplicidad de su funcionamiento.

CAPÍTULO 2 REVISIÓN DE TRABAJOS ACERCA DE LA OPTIMIZACIÓN DE TORRES AUTOSOPORTADAS

Introducción del capítulo

Las torres de celosía de acero autosoportadas 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 autosoportadas tienen como objetivo la reducción del peso de la estructura. Para lograrlo existen una infinidad de parámetros y variables que pueden ser modificados por los diseñadores. El principal objetivo de este capítulo es realizar una extensa revisión bibliográfica por la literatura disponible sobre el tema, a fin de definir cuáles son los métodos y parámetros para la optimización estructural de torres autosoportadas

2.1 Optimización de torres empleando programación dinámica

Palmer y Sheppard (1972) En este trabajo se plantearon la minimización del peso de la parte inferior de la estructura (la situada entre los apoyos y la primera cruceta). (Figura 2.1)

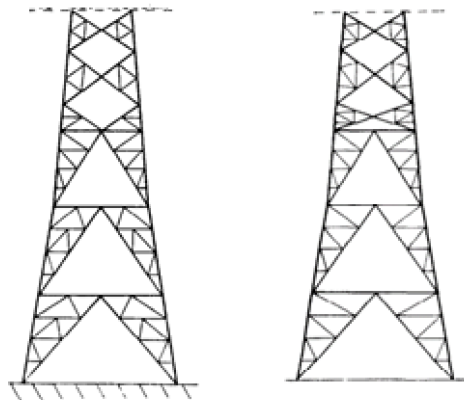


Figura 2.1. Parte inferior de la torre objeto de optimización (izquierda), Estructura optimizada (derecha). (Palmer y Sheppard, 1972)

Para ello fue dividida en varios segmentos (figura 2.2). El sistema de optimización tenía que determinar la altura y el diseño estructural de cada segmento (debía escoger entre seis tipos de módulos predefinidos) sin modificar la altura total y la forma externa de la torre, y dimensionar correctamente todas las barras. El problema de optimización fue resuelto mediante un algoritmo de programación dinámica, lo que exige desacoplar el

problema completo en varias etapas (el diseño de cada segmento) que se habrán de resolver secuencialmente. Además, no se consideró el peso propio de la estructura y sólo se tuvo en cuenta la acción del viento de una forma muy simplificada.

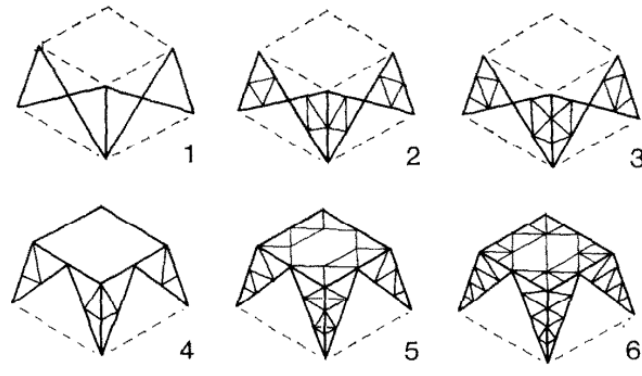


Figura 2.2. División en segmentos de la torre. (Palmer y Sheppard, 1972)

Raj y Durrant (1976) Plantearon la minimización del peso en toda la torre. Para ello dividieron la estructura en cuatro zonas. El sistema de optimización determinaba algunos parámetros geométricos y el diseño estructural de cada zona (para lo que cuenta también con varios tipos de módulos predefinidos), y dimensionaba correctamente todas las barras. El problema de optimización se resolvía mediante un algoritmo de programación dinámica. El cálculo estructural fue realizado separadamente para cada una de las cuatro caras de la estructura, que se analizan como estructuras articuladas planas independientes.

2.2 Optimización mediante programación matemática no lineal

Hanssen (1981) En este trabajo se planteó la minimización del coste (del acero estructural) de la torre. La torre se consideraba formada por varias subestructuras. El sistema de optimización debía determinar algunas variables que controlan la geometría de la torre y dimensionar correctamente todas las barras. Imponía restricciones (sencillas) sobre las variables de tipo geométrico. La función de coste se consideraba discontinua, ya que las secciones de las barras tienen que elegirse entre un cierto número de perfiles normalizados de acero laminado. El problema de optimización se 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.

2.3 Optimización mediante programación lineal secuencial

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

2.4 Optimización mediante métodos de optimización propios de los autores

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

Para resolver el problema de optimización propone un método desarrollado por el autor. Este método se basa en un criterio de 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.

J.Felix y G.N.Vanderplaats (1987) Desarrollaron un método general para la optimización de estructuras articuladas espaciales. La función objetivo es la minimización del peso de la estructura. Consideraron como variables de diseño las secciones de las barras y la posición de los nodos e impusieron restricciones para limitar la tensión a tracción y para evitar el pandeo local, además de restringir también los

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

La segunda formulación emplea las dimensiones de cada sección como variables de diseño, longitudes de las alas del perfil angular y el espesor para un total de 27 variables de diseño continuas.

La función objetivo empleada fue el peso de la estructura definida como:

$$W = \sum_{i=1}^{\text{número de miembros}} W_i, \quad W_i = A_i L_i \quad (2.1)$$

Donde W es el peso de la estructura, W_i es el peso del elemento i y A_i y L_i son el área transversal y la longitud de cada elemento respectivamente.

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

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

El método de dos fases:

En la primera fase el método resuelve el problema formulado utilizando un algoritmo de optimización de variables continuas SQP, asignando tres variables de diseño a cada perfil, pero la solución óptima no está disponible en perfiles comerciales. En la segunda fase el método usa algoritmos genéticos y variables discretas para encontrar la sección comercial óptima.

El método ADAM:

El método ADAM resuelve el problema de optimización en múltiples fases utilizando algoritmos de variables continuas. En la primera fase las secciones de la solución son

comparadas con las secciones comerciales disponibles, si cualquier sección se acerca a una sección comercial, entonces es fijada.

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

También el problema es resuelto empleando algoritmos genéticos.

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

El GA analizó $27,7 \times 10^6$ diseños por $6,18 \times 10^4$ el método de dos fases pero empleó más tiempo computacional $146,606 \text{ s}$ contra $120,710 \text{ s}$ por lo que los autores llegan a la conclusión de que el método de GA obtiene mejores resultados pero requiere muchos recursos informáticos.

Navarrina, *et al.* (2006) Realizaron la optimización del diseño de una torre de la tipología FECSA/GL-110KV.

Las torres de dicha tipología se dividen en bloques estructurales como sigue:

BLOQUE K (Fig. 1), con 12 nodos y 25 barras (figura 2.4a)

BLOQUE X o SUPER X (Fig. 2), con 12 nodos y 25 barras (figura 2.4 b)

CRUCETA (Fig. 3), con 10 nodos y 32 barras (figura 2.4c)

CRUZ (Fig. 4), with 8 nodos y 12 barras, (figura 2.4d)

CABEZAL (Fig. 5), con 9 nodos y 16 barras.(figura 2.4e)

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.

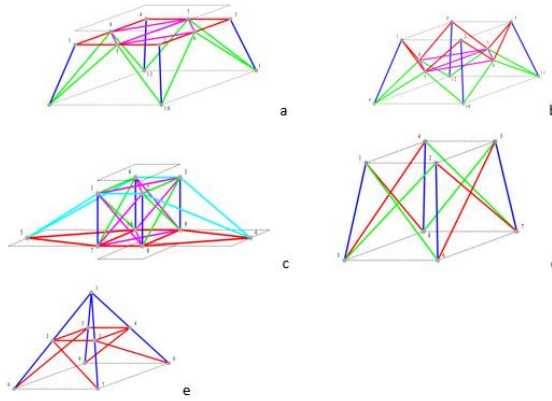


Figura 2.4 Bloques de la torre (Navarrina, et al., 2006)

Junto a las cargas por peso propio de la torre, los cables y los efectos por la dilatación térmica consideran cuatro estados de carga:

S1) carga de viento (sobre la estructura y sobre los conductores).

S2) carga de hielo (sobre los conductores).

F1) desequilibrio de tracciones con carga de hielo.

F2) rotura de un cable conductor con carga de hielo.

Siendo S1 y S2 aplicado al estado límite de servicio y F1 y F2 al estado límite de fallo.

En el análisis estructural consideraron la estructura como una armadura articulada tridimensional.

La función objetivo planteada por los autores es la siguiente:

$$\begin{aligned} &\text{minimizar } W = \sum_i \rho_i * A_i L_i \\ &\text{verificando } N_{i\ell\phi\ell} - \sigma_i^e A_i \leq 0 \quad \forall_i, \forall_\ell \quad (2.4) \\ &N_{i\ell\phi\ell} - \frac{\sigma_i^e A_i}{K_i} \leq 0 \quad \forall_i, \forall_\ell \end{aligned}$$

Donde: ρ_i y σ_i^e son el peso específico y el límite elástico del acero, respectivamente, A_i es el área de la sección transversal del perfil, L_i es la longitud de la barra, y K_i es el coeficiente de pandeo de la barra, y (para cada caso de carga ϕ) ϕ es el coeficiente de

seguridad que corresponde al caso de carga. El coeficiente de pandeo K_i de cada barra es calculado automáticamente por el sistema de diseño óptimo, teniendo en cuenta la disposición del sistema secundario de riostras del elemento estructural en cuestión.

Los autores toman como variables de diseño la longitud del lado de la base del bloque tipo k y del lado inferior del elemento tipo cabezal, manteniendo la altura de los elementos que componen la altura de la torre invariables y la sección de las barras. Además, incluyen algunas restricciones adicionales para evitar resultados sin sentido.

El problema de optimización de los variables que definen las formas es continuo y el de selección de la sección de las barras es discreto.

Para resolver el problema de optimización continua, utilizan un algoritmo de programación lineal secuencial con búsqueda unidireccional cuadrática (SLP-QLS o *Sequential Linear Programming with Quadratic Line Search*).

“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.

La torre objeto de estudio de los autores está conformada por un elemento tipo BLOQUE K, un elemento tipo BLOQUE SUPER X, tres elementos tipo BLOQUE X, siete elementos tipo CRUZ, un elemento tipo CRUCETA, tres elementos tipo CRUZ, un elemento tipo CRUCETA, tres elementos tipo CRUZ, un elemento tipo CRUCETA y el elemento tipo CABEZAL.

Las dimensiones iniciales de las variables de diseño son $x_1 = 6.55$ m y $x_2 = 1.25$ m con un peso $W = 4349.93$ Kp.

Para demostrar la robustez del algoritmo para la optimización de formas realizaron tres ejemplos de optimización con dimensiones iniciales muy diferentes entre sí y obtuvieron el mismo resultado.

Tabla 2.1 Validación del algoritmo (Navarrina, *et al.*, 2006)

<i>Diseño</i>	<i>X1</i>	<i>X2</i>
C1	15.00	2.5
C2	1.00	1.00
C3	2.1	2.0
Óptimo	7.43	1.0

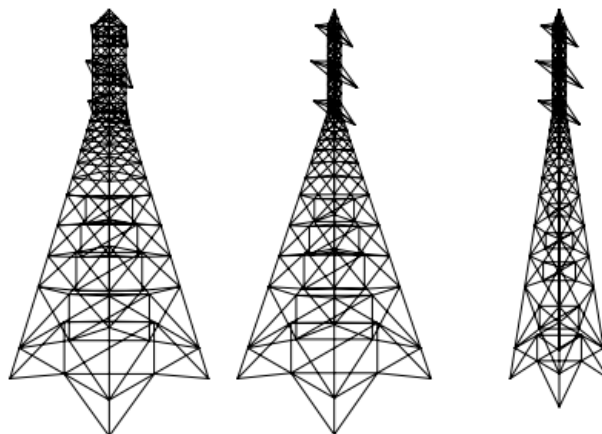


Figura 2.5 Evolución del diseño estructural del ejemplo C1. (Navarrina, *et al.*, 2006)

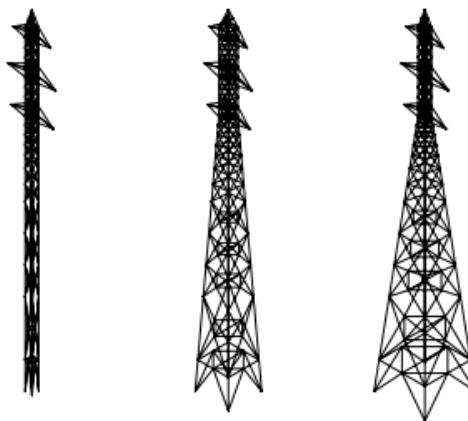


Figura 2.6 Evolución del diseño estructural del ejemplo C2. (Navarrina, et al., 2006)

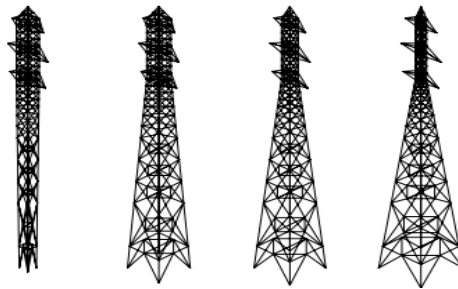


Figura 2.7 Evolución del diseño estructural del ejemplo C3. (Navarrina, et al., 2006)

Luego muestran el comportamiento del algoritmo SLP-QLS en combinación con la estrategia propuesta para el dimensionamiento de las secciones.

El ejemplo T1, las dimensiones iniciales son las mismas que las del diseño óptimo de los ejemplos C1, C2 y C3. En el ejemplo T2 las dimensiones son las mismas que en el ejemplo T1 pero los perfiles son diferentes, en lugar de utilizar un perfil L100.8 emplea un L100.7 y el ejemplo T3 se muestra una torre modificada a la que se le sustituyó los dos elementos BLOQUE X por un elemento BLOQUE SUPER X.

Tabla 2.2, comparación entre los diseños, (Navarrina, et al., 2006)

Diseño	X1	X2	Peso	% ahorro
T1	6.71	1.00	3892.79 kg	10.5
T2	8.25	1.00	3753.91	13.7
T3	9.30	1.0	3611.41	17.0

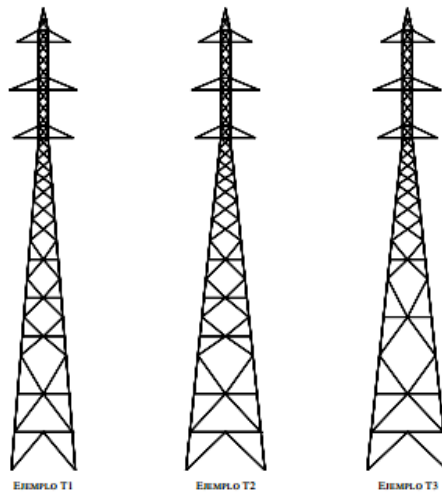


Figura 2.8 Comparación entre los diseños finales obtenidos. (Navarrina, et al., 2006)

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

- a) Costo inicial de construcción.
- b) Costo de operación.
- c) Costos de inspección y mantenimiento.
- d) Costo estimado si ocurre la falla.

El costo esperado si ocurre la falla está dado por la probabilidad de que esta ocurra:

$$\text{Costo esperado de falla } (\mathbf{z}, \mathbf{X}, Pf) = \text{costo de falla}(\mathbf{z}) \cdot Pf(\mathbf{z}, \mathbf{X}) \quad (2.5)$$

Donde:

Z: vector que contiene todos parámetros determinísticos del sistema estructural como factores de seguridad parciales, parámetros de inspección y programas de mantenimiento, etc.

X: Vector que incluye las características geométricas, resistencia, propiedades de los materiales de los miembros estructurales.

Pf: probabilidad de falla.

En los costos de falla incluyeron costos de reparación de elementos estructurales dañados, remoción de la estructura colapsada, reconstrucción, costo de inhabilitación,

costos por daños o muerte a los empleados o usuarios generales, penalizaciones por daños al medio ambiente, etc. Todos expresados en unidades monetarias.

El costo estimado total es:

$$\begin{aligned}
 & \text{Costo esperado } (z) \\
 & + \text{costo de operación}(z) \\
 & + \text{costo de inspección y mantenimiento}(z) \\
 & + \sum_{\text{tipos de fallas}} \text{costos de falla}(z) * P_f(z, X)
 \end{aligned} \tag{2.6}$$

El problema de optimización es planteado de la siguiente manera:

$$\begin{aligned}
 & \text{minimizar: costo total esperado } (z, X, P_f) \\
 & \text{sujeto a: } P_f(z, X) < P_f^{\text{admisible}}
 \end{aligned} \tag{2.7}$$

La probabilidad de falla admisible se aplica cuando hay riesgo de pérdida de vidas humanas.

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. (Figura 2.9)

La parte mecánica del problema es solucionada por el programa comercial de elementos finitos ANSYS, el módulo de fiabilidad estructural (StRAnD) fue desarrollado por el departamento de ingeniería de la universidad de Sao Paulo y fue escrito en FORTRAN. Para la optimización emplea interpolación cuadrática para la búsqueda lineal y método de regula falsi.

El módulo de optimización (RiskOPT) fue desarrollado in-house.

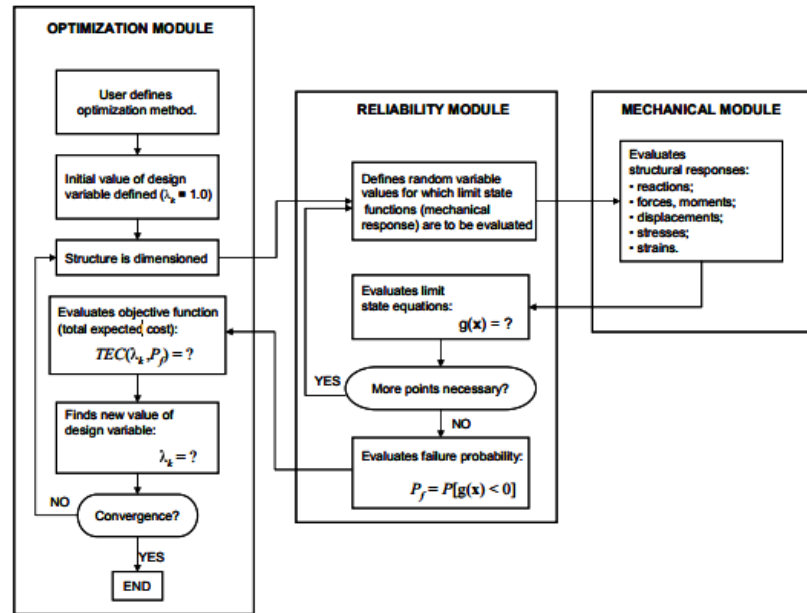


Figura 2.9 interacción entre los módulos. (Beck y Verzenhassi, 2008)

La función objetivo fue escrito en términos de la variable λ_k , los métodos de búsqueda lineal disponibles son interpolación cuadrática y regla false.

$$\lambda_k = RD / SD$$

RD : Resistencia nominal de los miembros. (2.8)

SD : Efectos nominales de las cargas.

En el ejemplo ejecutado por los autores consideraron dos situaciones para la función objetivo:

Para viento máximo de tormenta durante 50 años:

$$CTE_i(\kappa\lambda_k) = CI(\lambda_k) + CF_i(\lambda_k) * P[E / W_{50}] \quad (2.9)$$

Incluyendo tornado

$$CTE_i(\lambda_k) = CM(\lambda_k) + CF_i(\lambda_k) * P[E] \quad (2.10)$$

Donde:

CTE: costo del evento colapso de la torre.

E: evento colapso de la torre.

CI: costo inicial de la construcción definido como:

$$CI(\lambda_k) = CM(\lambda_k) + CW \quad (2.10)$$

CM: costo de materiales

CW: costo de mano de obra

Fi: costo de falla en función de las consecuencias de la falla.

Finalmente, los resultados finales son valores óptimos de λ_k para cada condición de falla.

Tabla 2.3. Resultados del proceso de optimización. (Beck y Verzenhassi, 2008)

Consecuencias de falla	λ_k óptimo		Índice de fiabilidad óptima β	
	W50	tornados	W50	tornados
menores	0.60	0.70	1.282	0.999
moderadas	0.68	0.80	2.054	1.292
grandes	0.75	2.10	2.423	1.798
extremas	0.80	2.27	2.647	1.837

Muda, *et al.* (2013) Analizaron el diseño estructural sostenible de torres de transmisión basado en un algoritmo topográfico

La función objetivo utilizada fue el peso de la torre y la variable de diseño es el ángulo de inclinación de la torre (figura 2.10). 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 (figura 2.11) extrae los archivos de entrada de datos .std de STAAD Pro y altera las coordenadas automáticamente de acuerdo con el ángulo de inclinación de las patas definidas por el usuario. (Figura 2.13)

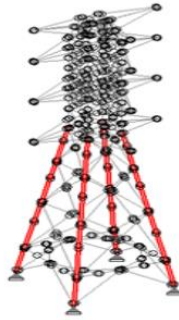


Figura 2.10 Variables de diseño. (Muda, et al., 2013)

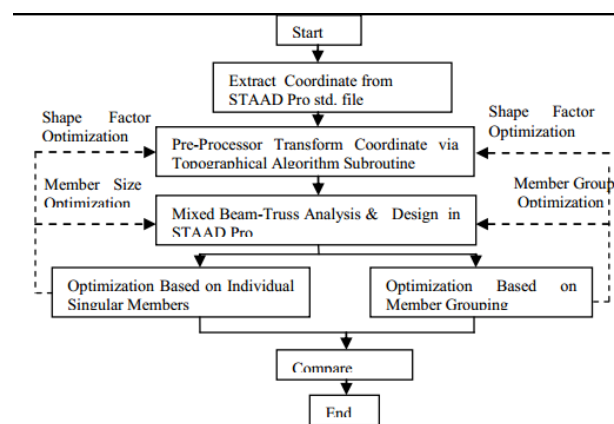


Figura 2.11. Esquema del algoritmo. (Muda, et al., 2013)

Los autores emplearon Visual Basic para desarrollar la interface del algoritmo.

Las combinaciones de cargas que consideraron fueron:

1. Carga de viento, peso propio y condición normal de los cables.
2. Carga de viento, peso propio y cable de tierra roto.
3. Carga de viento, peso propio y los conductores de la cima y fondo rotos.
4. Carga de viento, peso propio y los conductores del medio y de fondo rotos o el conductor de la cima y del fondo rotos.

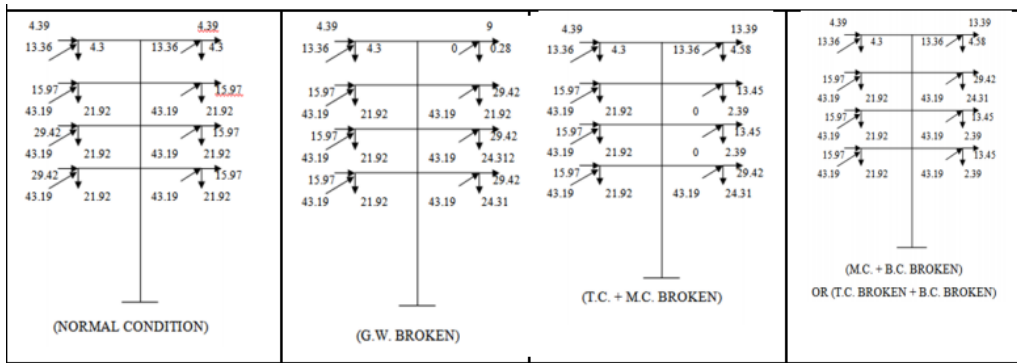


Figura 2.12. Combinaciones de carga. (Zwick, et al., 2012)

Los autores prueban el software para cinco ángulos de inclinación diferentes (5; 10; 15; 20; 25; 30). El peso de la torre al inicio varía entre 22000 kg y 65000 kg en dependencia del ángulo y el peso óptimo obtenido entre 10871kg y 25044kg para miembros singulares y 18366 y 52425 para grupos de miembros.

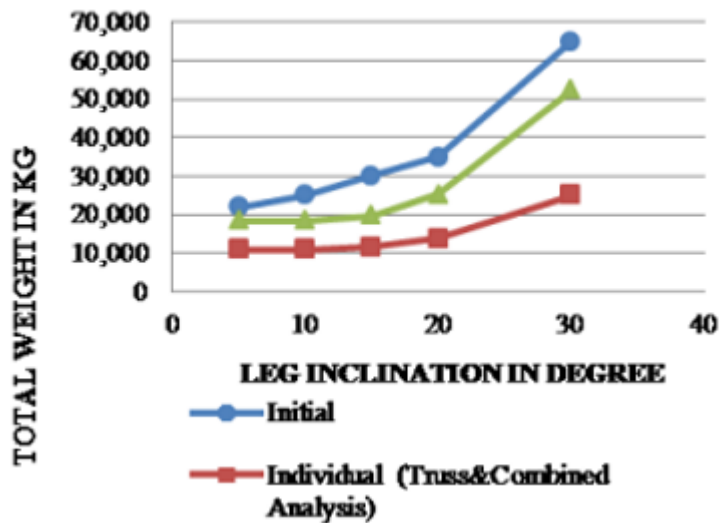


Figura 2.13. Variación del peso en función del ángulo. (Muda, et al., 2013)

Finalmente, concluyen que el peso óptimo es de 10924 kg para miembros singulares y de 18430 kg para grupos de miembros y ángulo de inclinación de 10 grados.

2.5 Optimización de torres empleando algoritmos genéticos

Aleli (1994) Presentó un método de resolución de optimizaciones de grandes estructuras realizando simultáneamente la evaluación de la función de aptitud de cada cadena de

cromosomas en la misma iteración. De la misma manera, la mutación de cada cadena y la operación de cruzamiento para cada par de cadenas (Aleli, 1994) y lo aplicaron para la optimización de una torre autosoportada de 942 elementos y 242 nodos.

La primera estrategia (figura 2.14) 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.

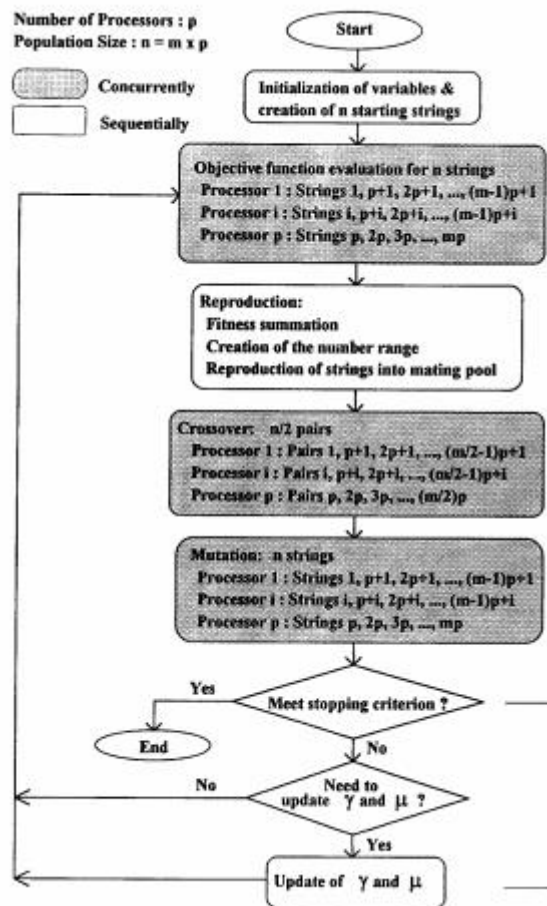


Figura 2.14: Algoritmo genético lagrangiano paralelizado. [Aleli, 1994]

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.

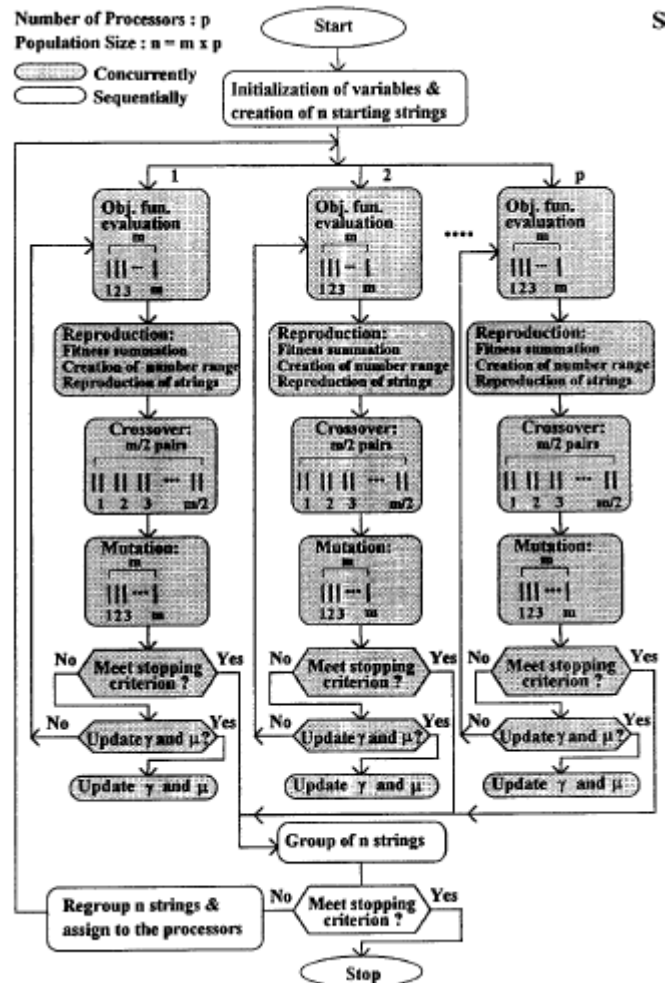


Figura 2.15 Esquema del algoritmo genético lagrangiano aumentado paralelizado.

[Aleli, 199]

Los autores aplicaron su metodología a una torre de 942 elementos y 244 nodos.

Las cargas verticales que consideraron fueron: 13.344 kN en cada nodo de la primera sección, 26.688 kN en cada nodo de la segunda sección y 40.032 kN en cada nodo de la tercera sección. Las cargas horizontales son 6.672 kN en el eje x en cada nodo del

lado izquierdo, 4.448 kN en el eje x de cada nodo del lado derecho, 4448kN en la dirección del eje y en el lado frontal.

Los límites de las áreas son 0.0065m^2 y 0.129m^2 .

La tensión límite permisible es de 170 MPa para tracción y compresión.

El desplazamiento es de 0.384m en cualquier dirección para los cuatro nodos del tope de la torre. Un valor de 150000 es usado para normalizar el factor Lr. El tamaño de la población fue de 100.

El tiempo computacional promedio para el algoritmo genético lagrangiano aumentado secuencial usando dos puntos y 20% de cruzamiento uniforme es mostrado en la tabla 2.4.

Los porcentajes de tiempo para las fases de iniciación, reproducción, cruzamiento y mutación también son mostrados en dicha tabla. Las figura 2.16 muestra las curvas de *speedup* para dos algoritmos paralelos sin y con vectorización respectivamente. 2.17 muestra el historial del mínimo valor de la función objetivo.

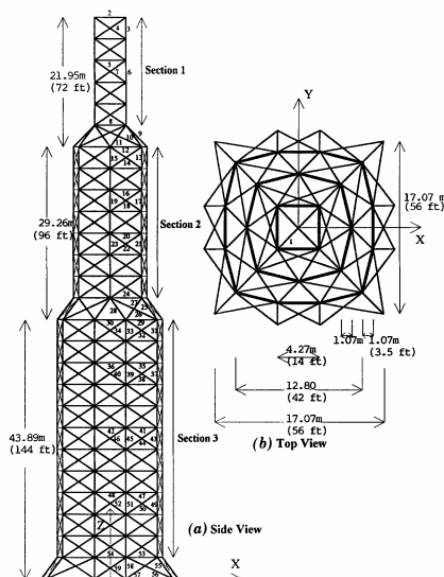


Figura 2.16: torre autoportada 26-story. [Aleli, 1994]

Tabla 2.4: Tiempo computacional promedio para el algoritmo genético lagrangiano aumentado secuencial usando dos puntos y 20% de cruzamiento uniforme. [Aleli, 1994]

Secuencia	Two-point crossover	20%-uniform crossover
Tiempo total CPU	2850 s	2850 s
Inicialización (%)	1,2	1,2
Reproducción (%)	98.1	98.1
Crossover (%)	0.3	0.3
Mutación (%)	0.4	0.4

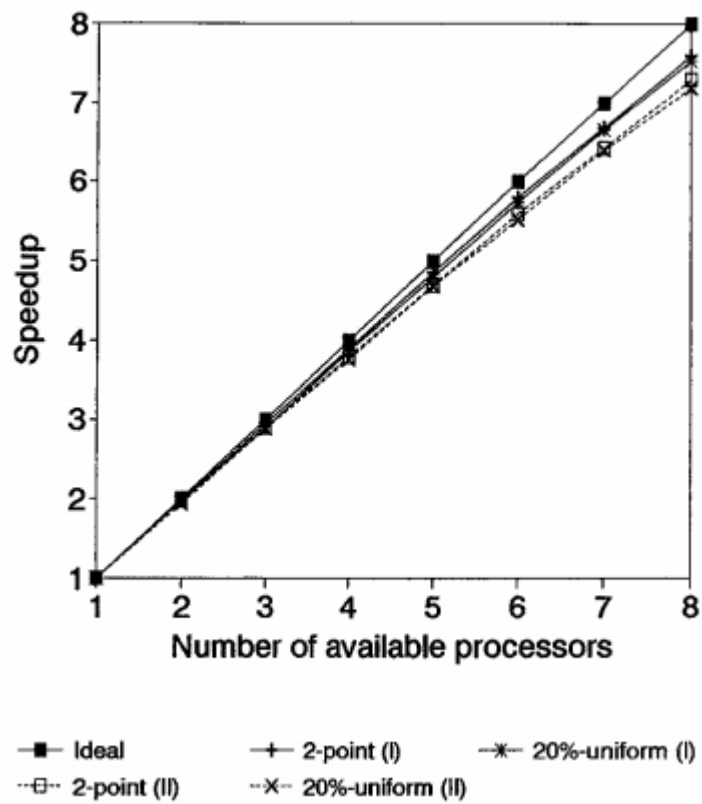


Figura 2.16: Speedup vs número de procesadores sin vectorización. [Aleli, 1994]

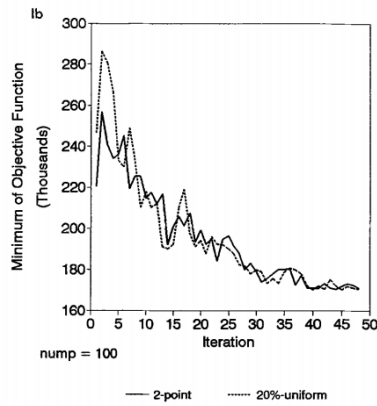


Figura 2.17 Historial del mínimo valor de la función objetivo. [Aleli, 1994]

Sivakumar, *et al.* (2004) analizaron la optimización orientada a objeto de una torre autosoportada aplicando algoritmos genéticos aplicando una metodología en la que la torre era dividida en objetos menores (figura 2.18 y 2.19) 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.

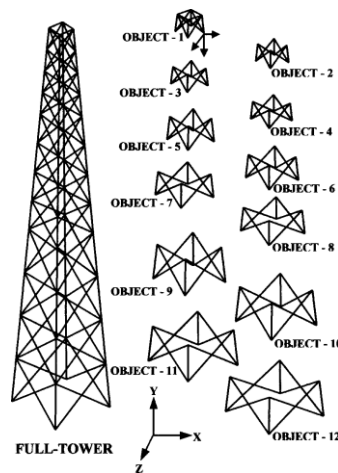


Figura 2.18 torre completa y paneles de objetos. [Sivakumar *et al.*, 2004]

La función objetivo fue el peso de la torre que quedó definida como:

$$\begin{aligned}
 \text{Minimizar } W &= f(X) \sum_{k=1}^{nm} W_k \\
 \text{sujeto a } g_i(X) &\leq 0; i = 1, 2, \dots, k \\
 X_j^K &\leq X_j \leq X_j^U; j = 1, 2, \dots, k
 \end{aligned}
 \tag{2.11}$$

Donde: w =función objetivo; Wk =peso del miembro k ; nm =número de miembros; X =vector de variables de diseño; g_i =restricciones debido a la respuesta estructural; n =número de variables; k =número de restricciones; X_j^L = límite inferior de la variable j y X_j^U límite superior de la variable j .

La función de aptitud fue:

$$\text{Valor de mérito} = \frac{K}{K + w} \quad (2.12)$$

Donde: w es el peso de la torre y K es una variable arbitraria que los autores por conveniencia fijan aproximadamente igual a w .

Las áreas de las secciones transversales de los miembros son tomados como variables discretas y son seleccionadas del catálogo de perfiles disponibles en India.

Las restricciones de tensión admisible para barras en compresión no pueden exceder $0.6 f_y$. La tensión axial en compresión no permisible fue calculada por la fórmula:

$$S_{ac} = \frac{f_{cc} * f_y}{[(f_{cc})^n + (f_y)^n]^{1/n}} \quad (2.13)$$

En tracción la tensión actuante no podía exceder:

$$S_{at} = 0.6 * f_y \quad (2.14)$$

Donde: S_{ac} tensión permisible a compresion (N/mm^2), S_{at} tensión permisible a tracción (N/mm^2), f_{cc} : tensión crítica en compresión ($= \frac{\pi^2 E}{\lambda^2}$): E =módulo de elasticidad del acero ($2 \times 10^5 N/mm^2$); λ =esbeltez del elemento (l/r) l : longitud efectiva, r : radio de giro y n : factor asumido como 1.4.

Los autores consideran l/r como una restricción para garantizar la estabilidad, para los miembros de la pierna de la torre está limitado a un valor máximo de 120 y para otros miembros es de 180.

La función de penalización empleada es:

$$\begin{aligned}
p_1 &= \frac{\sum V}{N_{v1}} * V_f \\
p_2 &= \frac{\sum V}{N_v} * V_f \\
\text{valor de aptitud (modificado)} &= \frac{\text{valor de aptitud}}{p_1 * p_2}
\end{aligned}
\tag{2.15}$$

Donde: V_f =factor de violación [$=1+(N_{v1}/N)$]; p_1 =factor de penalización por restricciones de tensión; p_2 =factor de penalización por restricciones de estabilidad; V =*ratio* crítico (*ratio* de tensión actual y de tensión admisible); N_{v1} =número de miembros que violan la restricción de tensión, N_{v2} =número de miembros que violan la restricción de estabilidad y N =número de miembros de la estructura.

Para la realización de la optimización de la torre completa como un solo elemento los autores consideraron la torre como una sola estructura en el análisis. Los soportes fueron asumidos como restringidos en todas las direcciones y sin restricciones de rotación.

En el análisis orientado a objeto cada panel de objetos fue considerado sometido a las cargas externas iguales que a la de la torre, además de las cargas externas fueron consideradas las reacciones del panel superior, la transferencia de carga del panel superior al inferior ocurre en los nodos.

Primero el panel superior es considerado de forma individual como un objeto independiente y se le aplican las fuerzas actuantes sobre la torre. El objeto es analizado y los resultados son aplicados en los nodos superiores del subsecuente panel. Los resultados obtenidos de las fuerzas son adicionados a las fuerzas actuantes sobre la torre. Un proceso similar se continúa con los siguientes paneles.

Parámetros del algoritmo genético:

Los autores emplearon selección por rueda de ruleta y *crossover* por *single point* y un máximo número de generaciones de 500. Asumen que la convergencia se logra cuando el resultado del valor del peso de la torre o del objeto no varía en 100 generaciones.

En la optimización de la torre completa el problema contiene 25 variables. La longitud total de un cordón de población es de 85. La optimización de la configuración de la torre completa fue realizada usando GOAT (*Genetic Algorithm Approach Towers*). El óptimo se alcanzó en la generación número 230 y el peso óptimo fue de 25.3 kN.

En *object-oriented* análisis, la torre fue dividida en 12 objetos y todos fueron optimizados utilizando GOAT independientemente. La transferencia de cargas es establecida siguiendo un análisis secuencial para cada objeto. El tamaño de población, cargas externas y la elección de variables son las mismas que para la torre completa. La convergencia se obtiene en la generación número 70. El peso total fue de 20.81 kN.

Tabla 2.5: Detalles de las variables de diseño. [Sivakumar et al, 2004]

Variable	Límite inferior	Límite superior	Número de decisiones	Longitud del cordón
1	15	15	1	1
2,3,4,5	27	42	16	4
6,7	37	52	16	4
8,9,10,11,12,13	49	64	16	4
14,15,16,17	11	11	1	1
18,19,20,21	18	33	16	4
22,23,24,25	29	44	16	4

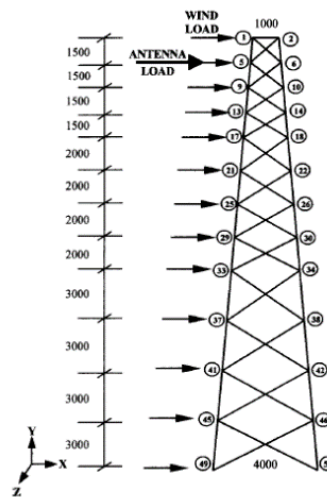


Figura 2.19 dimensiones de la torre. [Sivakumar et al, 2004]

Tabla 2.6 Miembros de la torre. [Sivakumar et al, 2004]

Panel	Pierna	Diagonal
1	65x65x5	35x35x5

2	65x65x5	35x35x5
3	65x65x5	35x35x5
4	65x65x5	35x35x5
5	90x90x6	50x50x5
6	90x90x6	50x50x5
7	100x100x6	50x50x5
8	100x100x8	50x50x5
9	100x100x8	70x70x5
10	100x100x10	70x70x5
11	100x100x10	70x70x5
12	100x100x10	70x70x5

Tabla 2.7 Comparación entre optimización orientada a objeto y optimización de la torre completa. [Sivakumar et al, 2004]

		<i>Toda la torre (a)</i>		<i>Objetos (b)</i>		
<i>Clasificación</i>	<i>variable</i>	<i>Número de sección</i>	<i>Área (mm²)</i>	<i>Objeto número</i>	<i>Sección número</i>	<i>Área (mm²)</i>
Cierre horizontal	1	15	378	1	15	378
pierna	2	39	1060	1	33	625
pierna	3	33	625	2	33	625
pierna	4	41	727	3	33	625
pierna	5	41	727	4	33	625
pierna	6	49	1050	5	49	1050
pierna	7	52	2020	6	49	1050
pierna	8	51	1700	7	53	1170
pierna	9	55	1900	8	53	1170
pierna	10	50	1380	9	53	1170
pierna	11	52	202	10	54	1540
pierna	152	61	2030	11	57	1710
pierna	13	58	2110	12	61	2030
diagonal	14	11	327	1	11	327
diagonal	15	11	327	2	11	327
diagonal	16	11	327	3	11	327
diagonal	17	21	327	4	11	327

diagonal	18	24	295	5	21	295
diagonal	19	20	560	6	21	295
diagonal	20	23	507	7	21	295
diagonal	21	37	479	8	21	295
diagonal	22	23	677	9	29	575
diagonal	23	41	727	10	33	625
diagonal	24	37	677	11	37	677
diagonal	25	37	677	12	37	677

Peso total a=25.2974kN

Peso total b=20.8172kN

Los autores llegan a la conclusión de que en la metodología propuesta por ellos *object-oriented optimization* se obtienen mejores resultados que optimizando la torre como un solo objeto.

Mohamed y otros autores (2012). Aplicaron algoritmos genéticos basados en FORTRAN a la optimización de torres de transmisión de electricidad utilizando distintas variables de diseño y luego compararon los resultados.

En los cálculos consideraron tres sistemas de cargas aplica dos a los nodos (figura 2.21) (tabla 2.8)

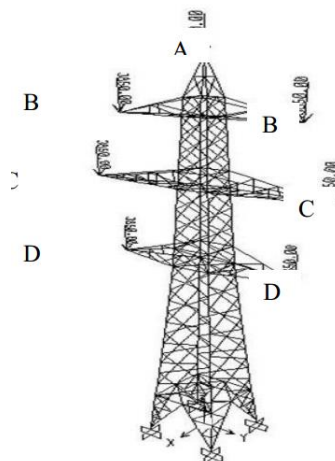


Figura 2.21 Nodos a los que se les aplicó las cargas. [Mohammed et al, 2012]

Tabla 2.8: Cargas en los nodos. [Mohammed et al, 2012]

Nodo	Cargas (N) en la dirección del eje Z
B, C, D	-3850
A	-1900

Carga transversal: Cargas de los cables sobre los puntos y las cargas de viento sobre la superficie transversal de la torre.

Carga longitudinal: Esta carga actúa en dirección paralela a la torre y es creada por la tensión diferencial de los conductores.

Carga vertical: este tipo de carga surge en los extremos de los brazos cruzados y en el punto pico del cable de tierra hacia abajo y consta de los siguientes componentes verticales:

- Peso del conductor en sí y el peso del hielo.
- Peso de aisladores y accesorios.
- Cargas extras para incluir acciones de mantenimiento.

Peso de la estructura, el peso de e la estructura es desconocido hasta que no se realice el diseño. Los autores suponen un diseño inicial con dimensiones razonables y luego comprueban.

Para el cálculo estructural usaron el método de elementos finitos.

El diseño óptimo se realizó sobre una torre de 672 elementos y 306 nodos. Figura 2.22

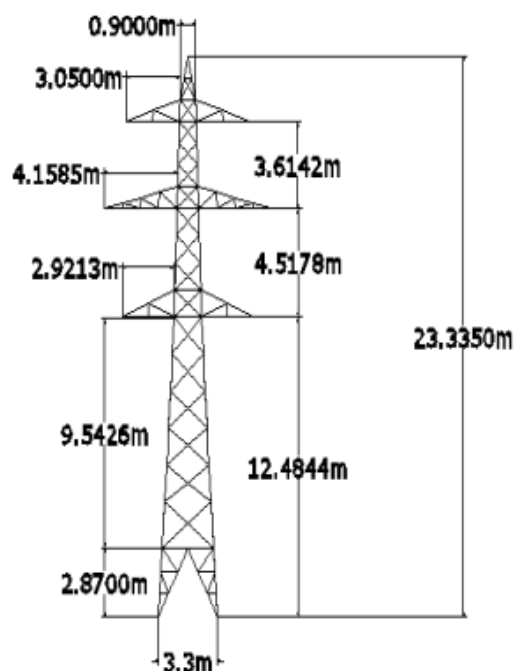


Figura 2.22 torre. [Mohammed et al, 2012]

Las restricciones aplicadas fueron:

Esfuerzo a tracción: $\sigma_t = 400.0 \times 10^6 \text{ N/m}^2$.

Esfuerzo a compresión $\sigma_c = -400.0 \times 10^6 \text{ N/m}^2$.

Máximo desplazamiento de los nodos: 0.1m en todas las direcciones.

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 las longitudes mínima y máxima son de 0.05 y 0.15m.

Los miembros fueron agrupados en doce grupos para la asignación de variables de diseño son mostrados en la figura 2.22 y clasificados como:

Sección transversal rectangular (con 12 grosores y 12 de ancho, en total 24 variables de diseño).

Sección transversal cuadrada (el ancho y el grosor son iguales, por lo que totalizan 12 variables de diseño).

Sección transversal rectangular (el ancho de las secciones es constante y el diseño de 12 espesores variables).

Sección transversal rectangular (el grosor de las secciones es constante y el diseño de 12 anchos variables).

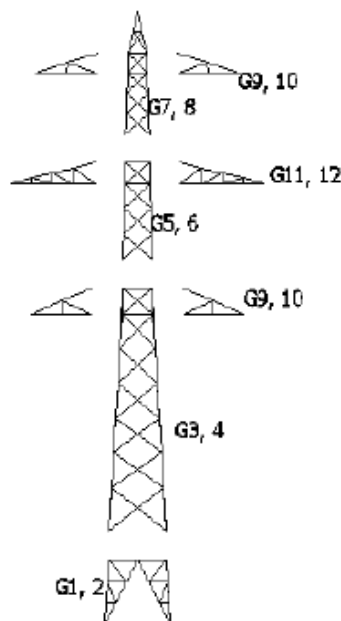


Figura 2.22 muestra los grupos de variables de diseño. [Mohammed et al, 2012]

Los parámetros del algoritmo genético empleado utilizado fueron:

Número de población 100, número de generación 100 y longitud de cadena binaria variable de diseño $m = 8$.

Para cada combinación de grupos de variables de diseño los autores obtuvieron un diseño óptimo (Tabla 2.9)

Tabla 2.9 Comparación entre los diseños óptimos. [Mohammed et al, 2012]

Variables de diseño (DV)	Valor optimizado por el algoritmo genético (m2)			
	Sección rectangular DV=24	Sección cuadrada DV=12	Ancho fijo DV= 12	Espesor de sección fijo DV=12
DV1	0,00054	0,00453	0,0081	0,0075
DV2	0,00873	0,0141	0,0119	0,00916
DV3	0,0119	0,00924	0,00022	0,00721
DV4	0.00131	0,00043	0,00697	0,0102
DV5	0.00765	0,00022	0,0066	0,00539
DV6	0,00683	0,011	0,00557	0,0116
DV7	0,00053	0,00687	0,00295	0,00344
DV8	0,00181	0,00873	0,0115	0,00152
DV9	0.00132	0,0108	0,00533	0,0022
DV10	0.00100	0,00645	0,00447	0,00405
DV11	0,000669	0,00127	0,00405	0,00447
DV12	0,00123	0,0135	0,0103	0,0115
Peso óptimo (kg)	17163,85	39729,3	33795,14	44085,71
% optimo	64,68	18,26	30,47	9,30

(Chunming, *et al.* (2012)) Optimizaron una torre de alta tensión de la tipología 5D-SZC3 (figura 2.23) empleando un algoritmo genético para optimización multiobjetivo denominado *non-dominated sorting genetic algorithm* (NSGA-II).

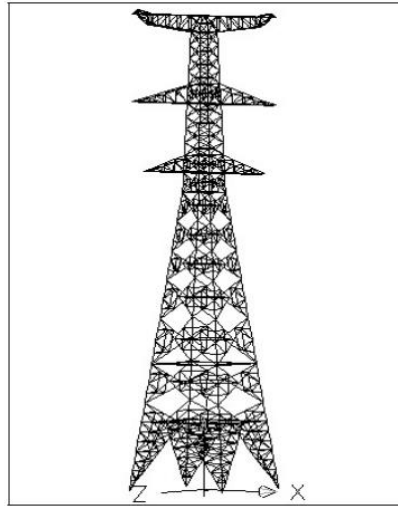


Figura 2.23. Torre tipología 5D-SZC3. (Chunming, et al., 2012)

Las funciones objetivo definidas por los autores fueron el costo de la torre y el desplazamiento en el eje X de las uniones con tierra de la torre. Los autores sometieron a la optimización solamente los miembros principales de la torre e ignoraron a los miembros diagonales y los miembros auxiliares.

Los miembros principales fueron divididos en 28 grupos definidos por los requerimientos de diseño. Dos variables de sección transversal fueron consideradas y dos variables de tipo de material. Para el algoritmo genético fueron definidas una población de 100 y una generación límite de 500.

Los autores seleccionaron tres diseños de las soluciones óptimas de Pareto llamados A, B y C y los compararon. En el diseño A minimizaron el desplazamiento nodal de la barra principal de la torre, en el diseño B minimizaron el costo de la estructura y en el diseño C persiguieron las dos metas, el costo de la estructura y los desplazamientos nodales. El diseño D es la torre original.

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.

Tabla 2.10 Comparación entre los tres diseños óptimos y el original. (Chunming, et al., 2012)

<i>Diseño</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
Desplazamiento (cm)	15,7	19,8	16,7	14,5
Costo (CNY)	54392	47317	51822	56395

Khodadadi y Bueleow (2014) Desarrollaron una técnica de búsqueda que usa conceptos de algebra Formex y optimización evolutiva basada en acercamiento de ParaGen. 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 *Ninzhy Novgorod*, Rusia; construida en 1896.



Figura 2.23 Torre de agua de Shukhov. [Khodadai & Buelow, 2014]

Los parámetros altura de la torre y diámetro superior fueron tomados de la torre. (Tabla 2.10)

Tabla 2.10 parámetros de la torre. [Khodadai & Von Buelow, 2014]

<i>Parámetros</i>	<i>Símbolo</i>	<i>Constante/Variable</i>	<i>Intervalo aceptable</i>
Altura de la torre	H	Constante	37m
Diámetro superior	D1	Constante	8m
Diametro inferior	D	variable	(8;24)m
Frecuencia de elementos a lo largo de la altura de la torre	N	Variable	(4;37)m
Número de lados de la base poligonal de la torre	m	Variable	(3;12)
Progenitores de la torre	Top	variable	(D1, D2, D3, D4, D5, D6, D7, D8)

Los autores procesan la configuración geométrica de la torre empleando un software de FORMEX. Luego la topología de la torre es insertada en STAAD para el análisis estructural y su evaluación. Los datos estructurales y las imágenes gráficas son guardados en una base de datos que propicia una interfaz para la exploración visual. En la siguiente etapa se seleccionan pares de torres para cruzarlos y obtener la siguiente generación. Las nuevas generaciones de torres producidas por iteración de estos pasos crean una serie de soluciones convenientes. Después, una función de aptitud elige las mejores. Finalmente la serie de soluciones convenientes es mostrada al diseñador para que pueda escoger las deseables atendiendo al desempeño estructural y la apariencia. (Figura 2.26)

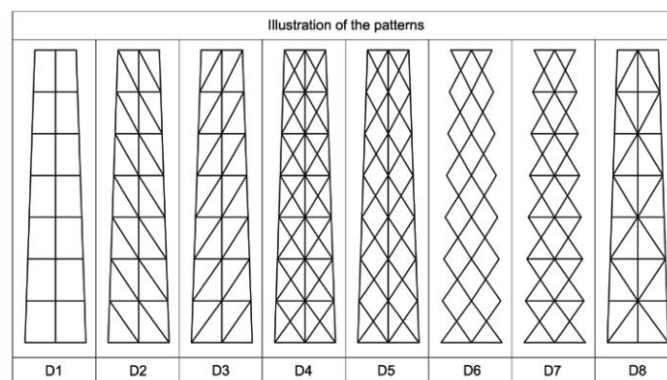


Figura 2.24: Progenitores. [Khodadai y Von Buelow, 2014]

Los autores emplearon un algoritmo genético no convencional denominado *Non-Destructive Dynamic Population GA* (NDDP GA). El cual tiene implementado cruzamiento tipo HUX (figura 2.25) y a diferencia de un GA tradicional este conserva en

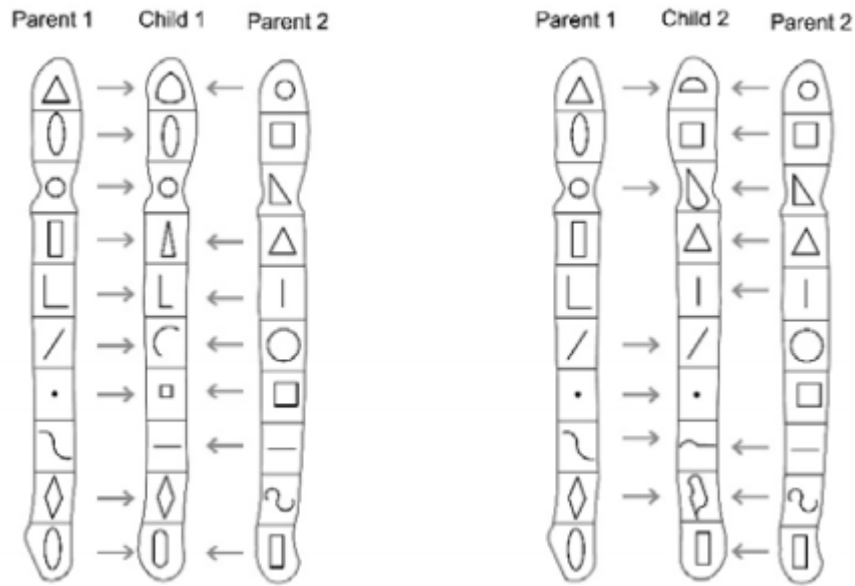


Figura 2.25. *Half Uniform Crossover (HUX)*. La descendencia puede tener las características exactas de uno de los progenitores o una combinación. [Khodadai y Von Buelow, 2014]

Descripción del proceso:

1. El problema es descrito en términos de variables paramétricas.
2. Un grupo de población inicial es generada y ordenada en una base de datos.
3. Un grupo de progenitores creado dinámicamente de entre la población.
4. Dos padres son seleccionados de forma aleatoria.
5. Un descendiente se genera de entre los progenitores.
6. Los cromosomas son traducidos a una solución geométrica.
7. La solución es evaluada por varios *softwares*.
8. Los valores de la evaluación, junto con los valores de los parámetros son subidos a una base de datos.

Las etapas de la 3 a la 8 se repiten hasta que se obtiene el óptimo. En la figura 2.26 se muestra el esquema del ciclo. Nótese que el diseñador debe intervenir en el proceso.

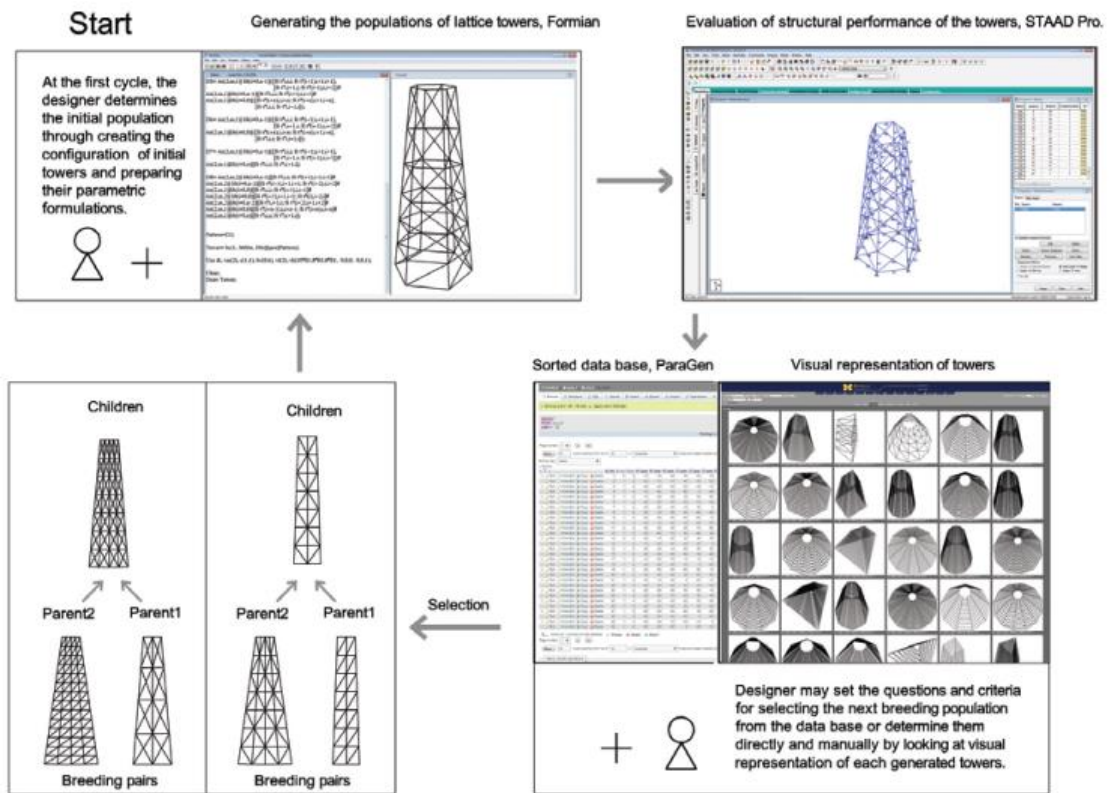


Figura 2.26: Esquema del ciclo. [Khodadai y Von Buelow, 2014]

En la figura 2.27 se, muestran las mejores soluciones filtradas con las propiedades mejores que la de la torre de agua de *Shukhov* mostradas en la tabla 2.11

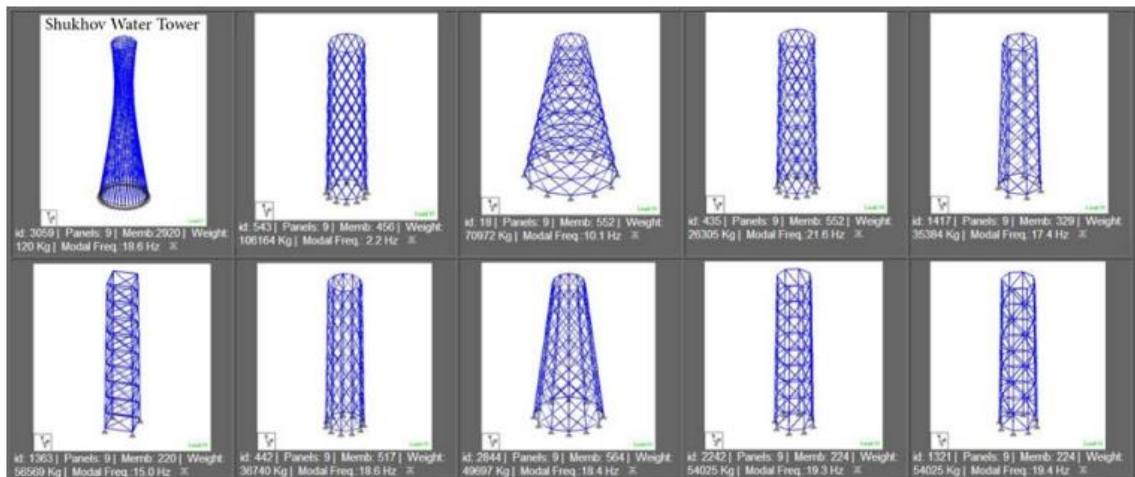


Figura 2.27. Presentación de las mejores soluciones [Khodadai & Von Buelow, 2014]

Tabla 2.11 Filtro de solución de propiedades. [Khodadai & Von Buelow, 2014]

<i>Propiedades</i>	<i>Valores aceptables</i>
Tamaño de la sección	=9
Diámetro de la base	>8
Número de juntas	≤1000
Peso total	≤120000kg

Magalhães, *et al.* (2014) Optimizaron el diseño de una torre autoportada 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 (C_m); costos de los materiales de los nodos (C_n), ensamblaje y mano de obra (C_b); los costos de los soportes a tierra (C_t); costos de manufactura de las diferentes secciones de la torre (C_{nm}); el número de barras que llegan a un nodo (costo de conectar una barra en un nodo); el ángulo de incidencia de una barra en el nodo (C_θ)

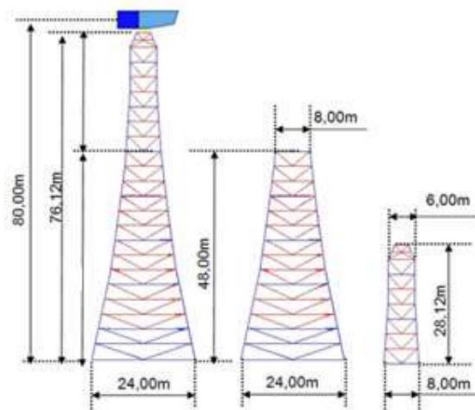


Figura 2.28 representación de la torre de un aerogenerador. (Magalhães, *et al.*, 2014)

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

$$C = C_m + C_n + C_b + C_t + C_{nm} + C_\theta \quad (2.16)$$

La estructura óptima fue generada en *Optistruct*® por el método de los elementos finitos. Los resultados de la optimización son mostrados en las figuras 2.29 y 2.30.

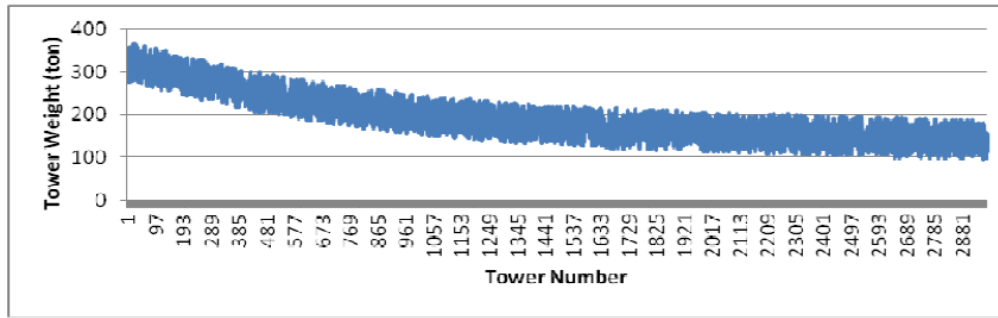


Figura 2.29 Evolución del peso de la torre. (Magalhães, et al., 2014)

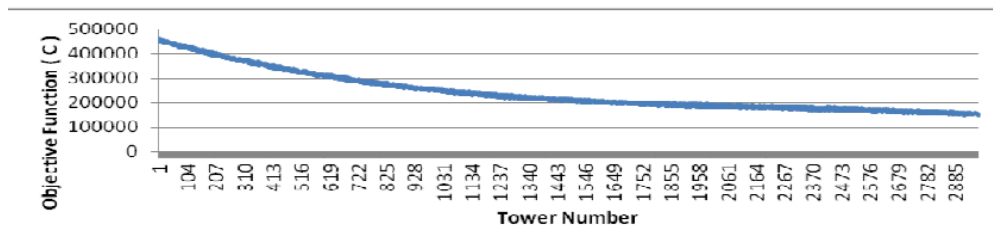


Figura 2.30 Relación entre el precio de la torre y su generación número. (Magalhães, et al., 2014)

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

En ANSYS la optimización hecha utilizando *Direct Optimization Toolbox*. Y en MATLAB es implementada por el comando „ga“ para algoritmos genéticos.

La interacción entre los programas quedó determinada por el siguiente algoritmo:

1. El algoritmo comienza con la creación de la población inicial aleatoria. La población inicial es exportada como documento de texto para convertirse en los parámetros escalares de entrada de ANSYS. Los archivos provenientes de MATLAB son corridos en ANSYS y producen archivos de salida.
2. Los resultados de ANSYS que se requieren ser utilizados como restricciones son tomados de los archivos de texto por MATLAB. Las restricciones son comprobadas.
3. Si se satisfacen las restricciones, entonces el volumen es tomado de los resultados de ANSYS, los otros resultados devuelven un valor invalido. El algoritmo crea la secuencia de una nueva población. En cada generación el algoritmo crea la siguiente con individuos de la actual.

4. El algoritmo se detiene cuando se cumple el criterio de parada.

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

Altura total: 30 m

Cuerpo de la torre: 18.2 m

Altura de la jaula: 8.95m

Armadura de las cruces: 2.527m

Altura de la cresta 2.85 m

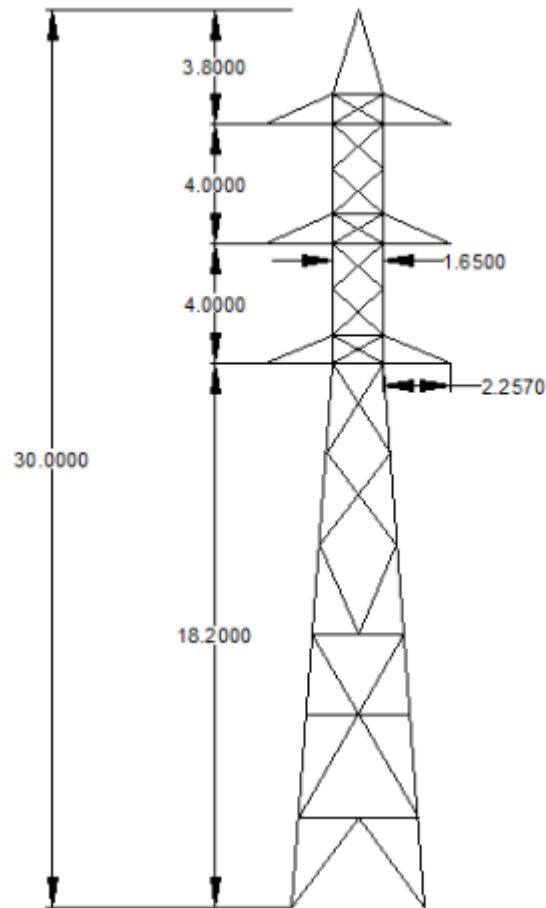


Figura 2.31 torre de transmisión (Sony y Airin, 2016)

Las propiedades de los materiales se muestran en la tabla 2.11

Tabla 2.11 propiedades de los materiales misma. (Sony y Airin, 2016)

Propiedad	Valor
Densidad del material	7850 kg/m ³
Módulo de Young	2.1 x 10 ⁵ N/mm ²
Coefficiente de Poisson	0.3

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 torres son consideradas fijas. Los parámetros de optimización son el ancho y el espesor de las secciones angulares usadas en cada miembro. Los miembros se agrupan en 12 grupos mostrados en la tabla 2.12

Tabla 2.12 parámetros de optimización traducción del original. (Sony y Airin, 2016)

<i>Parámetro</i>	<i>Descripción</i>	<i>Límites permitidos</i>
T1-T12	Espesores de las secciones angulares	5-8mm
W1-W12	Ancho de las secciones angulares	40-200mm
Tensión admisible	Tensión máxima admisible	Tensión admisible

ANSYS se ejecuta de MATLAB por la línea de comando mostrada en la figura 2.32

```
!C:\Program Files\ANSYS Inc\v162\ansys\bin\winx64\ANSYS162.exe' -b -p -i TrussModel.inp -o Output.txt
```

Figura 2.32 Línea de comando para que ANSYS corra en Batch mode. (Sony y Airin, 2016)

Los valores óptimos obtenidos en ANSYS y en MATLAB usando algoritmos genéticos pueden verse en la tabla 2.13. La reducción del peso en el caso de ANSYS *Workbench* fue de 69.46% y en el caso de los algoritmos genéticos en MATLAB fue de 78.66%. De acuerdo con esos resultados los autores concluyen que acoplando ANSYS y algoritmos genéticos se obtienen mejores resultados.

Empleado optimización directa en ANSYS, los autores analizaron 100 diseños y obtuvieron el valor óptimo en un tiempo de 75 minutos. Mientras que acoplando ANSYS con MATLAB y algoritmos genéticos con una población de 20 individuos y cien generaciones analizaron 2000 diseños en 140 minutos.

Tabla 2.13 resultados de optimización. (Sony y Airin, 2016)

<i>Parámetro</i>	<i>Tensión (MPa)</i>	<i>Peso (kg)</i>	<i>Porcentaje de variación</i>
Valor Inicial	64.295	10018	

Valor óptimo por ANSYS	238.25	3059,4	69.46%
Valor óptimo por MATLAB	230.86	2137,7	78.66%

2.6 Optimización de torres empleando el algoritmo recocido simulado del acero

Martínez, *et al.* (2012) 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 (figura 33). 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.

Las combinaciones de carga empleadas fueron las mismas de Navarrina 2006. La función objetivo utilizada fue el peso de la estructura definida como:

$$F(x, A) = \rho \sum_{i=1}^n l_i A_i \quad (2.17)$$

Donde ρ es la densidad del acero, l_i y A_i son la longitud y el área transversal de cada elemento de la estructura. La primera tiene un carácter continuo y la segunda tiene un carácter discreto.

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.

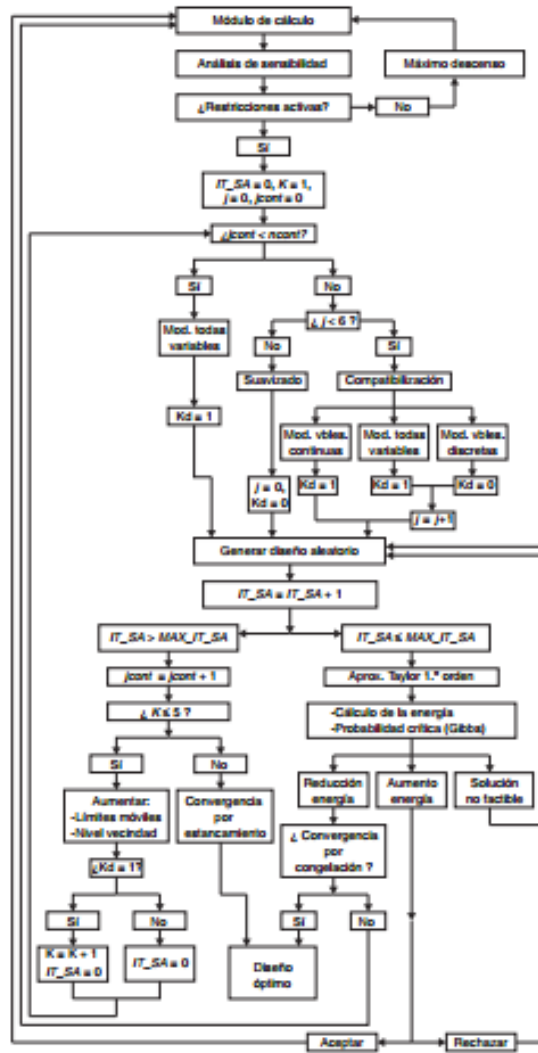


Figura 2.33. Esquema de la metodología utilizada por los autores. (Martínez, et al., 2012)

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.

El algoritmo de «recocido simulado» ejecuta una serie de pruebas para obtener una propuesta de diseño mejorado. En el caso de que el número de pruebas realizadas sea excesivo y todavía no se haya obtenido un diseño adecuado, el procedimiento

desarrollado reinicia el algoritmo del «recocido» incrementando la región de exploración de las variables de diseño. (Figura 2.34)

		Ancho de ala b (mm)								
		b_{k-4}	b_{k-3}	b_{k-2}	b_{k-1}	b_k	b_{k+1}	b_{k+2}	b_{k+3}	b_{k+4}
Espesor de ala t (mm)	t_{k-4}	*	*		*					
	t_{k-3}		*	*	*	*				
	t_{k-2}		*		*	*	*	*		
	t_{k-1}			*	*		*	*		
	t_k				*	*	*		*	
	t_{k+1}				*	*		*	*	
	t_{k+2}						*		*	*
	t_{k+3}						*	*	*	*
	t_{k+4}							*	*	*

Figura 2.34: Espacio de estudio, los valores con * representan combinaciones reales. (Martínez, et al., 2012)

Si el algoritmo de recocido simulado alcanza un número elevado de iteraciones sin obtener ningún resultado adecuado entonces el espacio de estudio se modifica.

La función de energía refleja el estado del sistema en un instante determinado del proceso de optimización, aportando la información necesaria sobre la función objetivo.

$$F(\xi) = f(\xi) + G(\xi) \quad (2.17)$$

Donde: $G(\xi)$ es la función que introduce la información necesaria sobre las restricciones del problema, y $f(\xi)$ es la función que incorpora la información relativa al peso total de la estructura. Debido a que durante el proceso de optimización se violan algunas restricciones y a que un diseño válido tendrá mayor energía que uno no válido la expresión de energía final es:

$$F(\xi) = \frac{f_k(\xi)}{f_0(\xi)} - \frac{\beta}{nrtot} \sum_{j=1}^{nrtot} \alpha_j^\gamma(\xi) \quad (2.18)$$

Donde $f_0(\xi)$ representa el valor de la función objetivo al inicio del proceso de optimización, $f_k(\xi)$ el valor de la función objetivo, $\alpha_j^\gamma(\xi)$ el aprovechamiento de la

función, en el instante k , $nrtot$ el número total de restricciones, y β es el coeficiente de ponderación de las restricciones. La temperatura inicial fue tomada como:

$$T_0 = \frac{(-0,20) \ nract}{\ln(0.50) \ nrtot} \quad (2.19)$$

Donde $nract$ es el número de restricciones actuantes y $nrtot$ es el número de restricciones total. La constante de optimalidad ha sido incluida por los autores dentro de la temperatura. El esquema de enfriamiento empleado fue:

$$T_k = \alpha^k T_0 \quad (2.20)$$

Donde α es el factor multiplicador tomado como 0.90 por los autores.

La temperatura de parada fue definida como:

$$T_{lim} = \frac{-0.001F_0}{\ln(0.001)} \sum_{j=1}^{nrtot} \frac{a_j(\xi)}{nrtot} \quad (2.21)$$

Donde:

F_0 el valor inicial de la función temperatura, a_j es el aprovechamiento estructural de la restricción j y $nrtot$ es el número de restricciones totales.

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.

Los autores emplean dos criterios de parada, el primero referente a la temperatura mínima permitida y el segundo si el algoritmo se estanca en un punto del dominio.

La torre analizada por los autores correspondió a una torre de alta tensión de 42 toneladas y una altura de 65 metros. La torre está conformada por los siguientes bloques: cúpula simple (figura 2.35 izquierda), súper lambda (figura 2.35 derecha), cruceta corta (figura 2.37), cruceta larga (figura 2.38), módulo X (fig. 9 izquierda), módulo súper X (fig. 9 derecha), módulo X arriostrado (fig. 10 izquierda), módulo K (fig. 10 derecha).

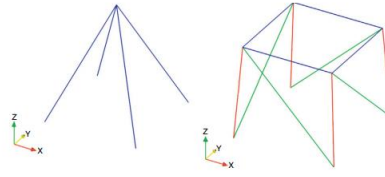


Figura 2.35 módulos cúpula simple (izquierda) y súper lambda (derecha) (Martínez, et al., 2012)



Figura 2.36 módulo cruceta corta. (Martínez, et al., 2012)



Figura 2.37 módulo cruceta larga. (Martínez, et al., 2012)

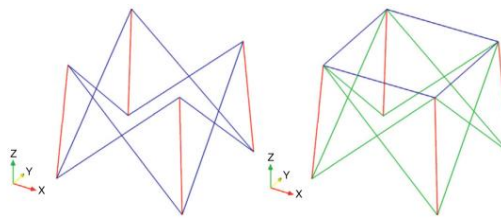


Figura 2.38 módulo tipo x y módulo tipo súper x. (Martínez, et al., 2012)

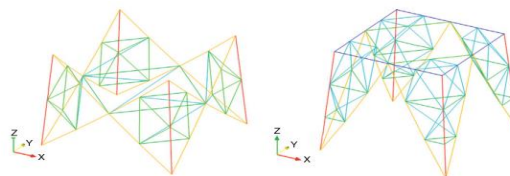


Figura 2.39 módulo súper X arriostrado módulo k. (Martínez, et al., 2012)

La torre está conformada por 23 bloques que contienen 267 nodos conectados por 829 elementos distribuidos en 77 secciones diferentes para un total de 154 variables de diseño discretas y 46 variables continuas.

Los resultados de la optimización logran un 19.85% de ahorro de peso de la estructura en 984 iteraciones consumiendo un tiempo de 8963 segundos, en un equipo con doble núcleo de 2,00 GHz y 4,00 GB de memoria RAM. (Figura 2.39)

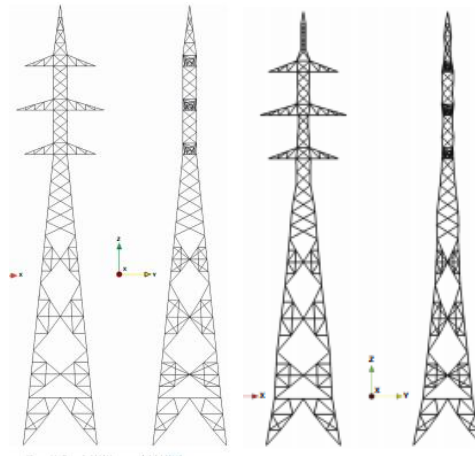


Figura 2.39 diseño inicial (izquierda) y diseño final (derecha) (Martínez, et al., 2012)

Tort, et al. (2016) Desarrollaron un método para optimizar torres de transmisión de energía integrando un algoritmo de recocido simulado en dos fases con el software para optimizar torres autoportadas en función del peso (acorde con los requerimientos y especificaciones de diseño de la ASCE 10-97) llamado PLS-TOWER.

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.

Para realizar la optimización los autores definieron dos vectores **A** y **S** que contienen las variables de diseño. El vector **A** contiene las variables de las secciones transversales de todos los miembros de la torre (N_m) definidos dentro de la base de datos de secciones comerciales disponible de un catálogo; por lo que el vector **A** contiene todas las variables de diseño discretas.

$$\mathbf{A} = [A_1; \dots; A_{N_m}]^T \quad (2.22)$$

$$\mathbf{S} = [S_1; \dots; S_{N_s}]^T \quad (2.23)$$

El vector **S** contiene todas las variables que pueden ser empleadas para alterar la geometría de la torre (N_s). En este estudio las variables consideradas son mostradas en la figura 2.40.

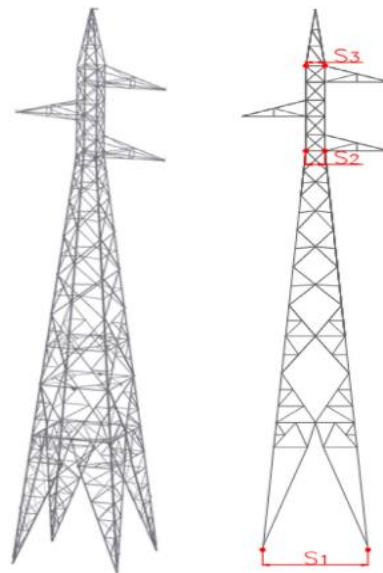


Figura 2.40. Variables de diseño definidas para alterar la geometría de la torre.

(Tort, et al., 2016)

La función objetivo fue el peso de la torre, definida como:

$$W = \sum_{i=1}^m \rho L_i A_i \quad (2.24)$$

Donde ρ es la densidad del acero, L_i es la longitud del miembro i y A_i es el área transversal del miembro i .

Los autores aplicaron las restricciones a la tensión y compresión axial impuestas por la ASCE 10-97 según el tipo de miembro. Adicionalmente añadieron restricciones geométricas ilustradas en la figura 2.41 y las ecuaciones 2.25 y 2.26.

$$w_{i+1} \leq w_i \quad (2.25)$$

$$t_{i+1} \leq t_i \quad (2.26)$$

Donde w y t representan el ancho y el espesor de los miembros i e $i+1$.

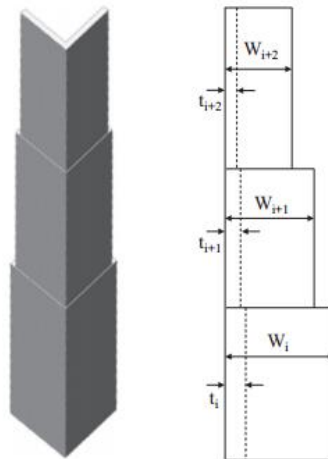


Figura 2.41. Restricciones geométricas.(Tort, et al., 2016)

Para realizar el diseño por estados límites los autores expresaron un set de funciones para manejar las restricciones:

$$C_i^c = \max \left[\frac{(P_{mc})}{(P_c)_i} - 1.0, 0 \right] \quad (2.27)$$

$$C_i^t = \max \left[\frac{(P_{mt})}{(P_t)_i} - 1.0, 0 \right] \quad (2.28)$$

$$C_i^\lambda = \max \left[\frac{(\lambda)_i}{(\lambda_{\max})_i} - 1.0, 0 \right] \quad (2.29)$$

$$C_1 = \sum_{i=1}^{Nm} \max(C_i^c, C_i^t, C_i^\lambda) \quad (2.30)$$

Donde: $(P_{mc})_i$, $(P_{mt})_i$, $(\lambda)_i$ son las cargas de tensión y compresión axial desarrolladas en el miembro i de la torre y sus radios de giro respectivamente; $(P_c)_i$, $(P_t)_i$ y $(\lambda_{\max})_i$ son los valores aceptables de las cantidades precedidas acorde con la ASCE 10-97. C_i^c , C_i^t y C_i^λ son las violaciones a la capacidad de compresión, capacidad de tensión y radio de giro del miembro i de la torre, y C_1 es la sumatoria de las máximas violaciones de las restricciones al estado límite.

De forma similar las restricciones geométricas quedaron planteadas como:

$$C_i^w = \max \left[\frac{w_{i+1}}{w_i} - 1.0, 0 \right] \quad (2.31)$$

$$C_i^t = \max \left[\frac{t_{i+1}}{t_i} - 1.0, 0 \right] \quad (2.32)$$

$$C_2 = \sum_{i=1}^{Nm} \max(C_i^w, C_i^t) \quad (2.33)$$

Donde: N_{tm} es el número total de miembros en la pierna de la torre, C_i^w y C_i^t son las violaciones a las restricciones geométricas del miembro i de las piernas de la torre y C_2 es la sumatoria de las máximas violaciones geométricas.

Finalmente, los autores introducen una función de penalización a la función objetivo:

$$\phi = W * (1 + r_1 * C_1 + r_2 * C_2) \quad (2.34)$$

Donde: r_1 y r_2 se refiere a los coeficientes de penalización para ajustar el grado de penalización por violaciones del estado límite de diseño y las restricciones geométricas respectivamente.

Para la realización de la optimización por un algoritmo SA los autores cumplieron las siguientes etapas:

1. Iniciación y creación del esquema de enfriamiento. En esta etapa se asigna un esquema de enfriamiento apropiado. Una base de datos es asignada para las formas de las variables y los límites de las variables de diseño son calculados.
2. Generación de un diseño inicial. El diseño inicial es generado aleatoriamente y le son asignadas secciones reales disponibles en la base de datos.
3. Generación de candidatos de diseño: Un número de candidatos de diseño son generados en la vecindad del diseño actual. (I) Una variable de diseño es seleccionada. (II) La variable de diseño es sometida a una pequeña perturbación en la vecindad predefinida. (III) Finalmente el candidato a diseño es generado asumiendo el valor perturbado de la variable, manteniendo el resto de los otros valores del diseño actual.
4. Evaluación del candidato a diseño y test de Metrópolis. Cuando un candidato de diseño es generado, la función objetivo es computada de acuerdo con la ecuación 2.34 y es comparado con la función objetivo del candidato actual. Si el candidato provee una mejor solución inmediatamente es aceptado y reemplaza al diseño actual. Además el llamado test de Metrópolis es empleado para determinar el ganador, en caso de que se aceptara un peor candidato de diseño.
5. Iteraciones del esquema de enfriamiento: Una iteración sencilla del esquema de enfriamiento es cuando todas las variables de diseño son seleccionadas a la misma vez y perturbadas para generar un nuevo candidato de diseño.
6. Reducir la temperatura: Cuando una iteración del esquema de enfriamiento es completada, la temperatura se reduce de acuerdo con la función de reducción de temperatura.
7. Criterio de parada. Las etapas de la 3 a la 6 se repiten hasta que el esquema de enfriamiento haya sido cumplido.

Las fases del algoritmo de recocido simulado en dos fases (figura 2.42).

Primera Fase.

1. Iniciación y elección del esquema de enfriamiento. Los parámetros son iniciados y un rápido esquema de enfriamiento es seleccionado.
2. Generación de un diseño inicial. El diseño inicial es generado de la manera usual por la asignación de variables aleatorias de acuerdo a los límites definidos.
3. Creación y reasignación de tamaño de un candidato a diseño. Un candidato a diseño es generado por perturbación de una variable de diseño y los tamaños de las secciones son reasignadas.
4. Evaluación del candidato y Test de metrópolis. El candidato obtenido en la etapa 3 es comparado con el diseño actual y si provee una mejor solución reemplaza al diseño actual. El ganador se determina usando el test de Metrópolis.
5. Iteraciones del esquema de enfriamiento: Una iteración sencilla del esquema de enfriamiento es cuando todas las variables de diseño son seleccionadas a la misma vez y perturbadas para generar un nuevo candidato de diseño.
6. Reducir la temperatura: Cuando una iteración del esquema de enfriamiento es completada, la temperatura se reduce de acuerdo con la función de reducción de temperatura.
7. Criterio de parada. Las etapas de la 3 a la 6 se repiten hasta que el esquema de enfriamiento haya sido cumplido.

Segunda fase:

En la segunda fase el proceso implementa las variables de diseño y las forma de manera junta, lo que en lugar de emplear un diseño aleatorio, emplea el mejor diseño de la etapa inicial.

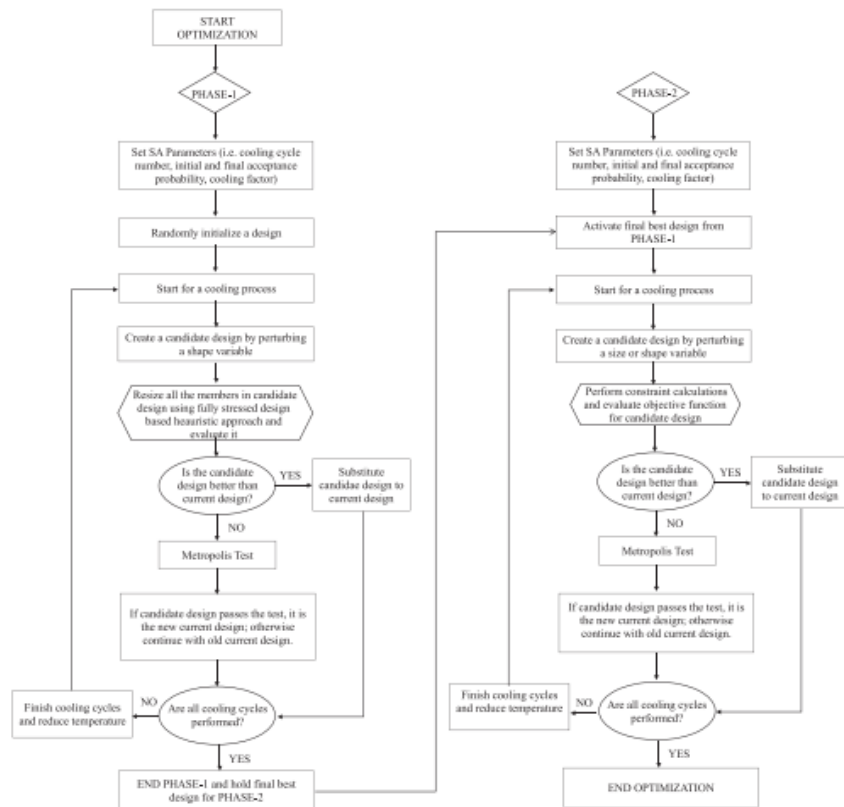


Figura 2.42. Algoritmo SA en dos fases. (Tort, et al., 2016)

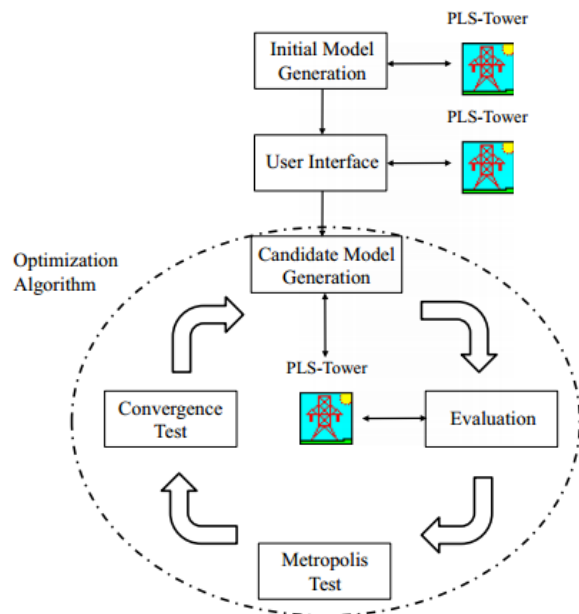


Figura 2.43 Interacción entre SA y PLS tower. (Tort, et al., 2016)

Para demostrar sus resultados los autores realizaron la optimización de tres tipos de torres de alta tensión, en este trabajo de diploma solo será mostrada la optimización de

la torre de 397 miembros (Tablas 2.14 y 2.15). Los autores llegan a la conclusión de que el algoritmo en dos fases obtiene mejores resultados

Tabla 2.14 Resultados de optimización por algoritmo SA. (Tort, et al., 2016)

			<i>Media</i>		<i>Desviación estándar</i>	
<i>Iteración</i>	<i>Peso óptimo(kg)</i>	<i>tiempo (min)</i>	<i>Peso (kg)</i>	<i>Tiempo (min)</i>	<i>Peso (kg)</i>	<i>Tiempo (min)</i>
1	4778,4	569	4754	549	89	30
2	4649,4	565				
3	4672,2	574				
4	4824,6	527				
5	4845	507				

Tabla 2.15 Resultados de optimización por algoritmo SA de en dos fases.(Tort, et al., 2016)

<i>Iteración</i>	<i>Peso (kg)</i>		<i>Tiempo (min)</i>			<i>Media</i>		<i>Desviación estándar</i>	
	<i>Primera fase</i>	<i>Segunda Fase</i>	<i>Primera fase</i>	<i>Segunda Fase</i>	<i>total</i>	<i>Peso (kg)</i>	<i>Tiempo (min)</i>	<i>Peso (kg)</i>	<i>Tiempo total</i>
1	5335,7	4695,5	21	153	174	4700,78	177,2	26	5
2	5322,2	4673,5	21	150	171				
3	5354,1	4742,3	21	160	181				
4	5333,8	4704,3	21	158	179				
5	5357,8	4688,3	21	160	181				

2.7 Optimización de torres empleando el algoritmo ondas del agua (WWE).

Millán-Páramo (2017) Solucionó el problema de optimización de una armadura espacial de 72 barras (figura 2.44) empleando el algoritmo ondas del agua. Para el análisis categorizó la armadura en 16 grupos de variables de diseño.

Las características del material utilizado fueron una densidad de $0,1 \text{ lb/m}^3$ y un módulo de elasticidad de 10000 ksi .

Las limitaciones impuestas fueron un desplazamiento de 0.25 in en todas las direcciones para los nodos superiores en cualquier dirección y un esfuerzo admisible de 25 ksi en cualquier elemento. Las áreas transversales admisibles varían entre 0.1 in^2 a 0.3 in^2 .

El autor consideró dos casos de carga:

Caso I, carga en el nodo 17 en las tres direcciones (x; y; z) con valores de $5,0; 5,0$ y $-5,0$ kips; Caso II, carga en los nodos 17;18;19 y 20 en la dirección z con valor de $-5,0$ kips.

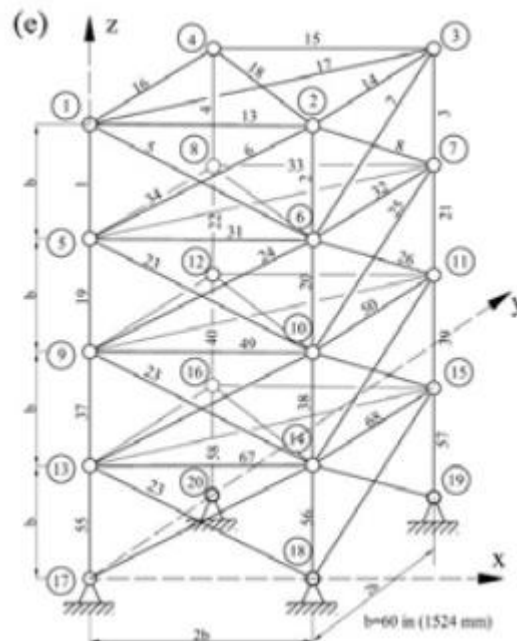


Figura 2.44 Armadura espacial de 72 barras (Millán-Páramo, 2017)

Otros parámetros fueron: $h_{\max} = 5$; coeficiente de reducción de longitud de onda

$\alpha = 1,001$; coeficiente de ruptura $\beta = 0,001$ y el número máximo de ondas solitarias

$k_{\max} = \min(12.D/2)$, donde D es la dimensión del problema.

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. Tabla 2.16 y llegan a la conclusión de que WWO proporciona mejores resultados.

Tabla 2.16 Comparación de resultados con otros métodos de optimización meta heurísticos. (Millán-Páramo, 2017)

	ACO	PSO	BB-BC	HBB-BC	TLBO	EHS	TLBO	WWO
Peso óptimo(lb)	380,2	381,9	379,9	379,7	379,6	381	379,6	379,6
Peso promedio	383,2	382,1	-	381,8	379,8	383,5	380,2	379,7
Diseños analizados	18500	-	19621	13200	21542	15044	1979	10200

2.8 Optimización de torres empleando los algoritmos *Colliding Bodies Optimization (CBO)*, *Enhanced Colliding Bodies Optimization (ECBO)*, *Vibrating Particles System (VPS)* y el algoritmo híbrido llamado MDVC-UVPS

Kaveh y Ghazaan (2018) Realizaron la optimización de tres torres de alta tensión autoportadas 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 servicialidad.

La función objetivo tomada fue el peso de la torre definida como:

$$\begin{aligned}
 &\text{Encontrar} \quad \{X\} = [x_1, x_2, \dots, x_{ng}] \\
 &\text{para optimizar} \quad W(\{X\}) = \sum_{i=1}^{ng} x_i \sum_{j=1}^{nm(i)} p_j L_j \quad (2.35) \\
 &\text{sujeto a} \quad \begin{cases} g_j(\{X\}) \leq 0, & j = 1, 2, \dots, n \\ x_{imin} \leq x_i \leq x_{imax} \end{cases}
 \end{aligned}$$

Donde: $\{X\}$ es el vector que contiene las variables de diseño, $W(\{X\})$ representa el peso de la estructura; $nm(i)$ es el número de miembros en el grupo i , p_j y L_j denotan la densidad del material y la longitud del miembro j , respectivamente. x_{imin} y x_{imax} representan los límites superior e inferior de la variable de diseño de la variable x_i ,

respectivamente. $g_j(\{X\})$ Denota la restricción al diseño j y nc es el número de restricciones.

Para manejar las restricciones los autores adoptaron una función de penalización que redefine la función objetivo como:

$$P(\{X\}) = (1 + \varepsilon_1 * \nu)^{\varepsilon_2} \times W(\{X\}) \quad (2.36)$$

Donde $P(\{X\})$ es la función de penalización y ν es la sumatoria de las violaciones de las restricciones. Aquí ε_1 es tomado como una unidad y ε_2 es calculado como:

$$\varepsilon_2 = 1.5 + 1.5 \times \frac{\text{iter}}{\text{iter}_{\max}} \quad (2.37)$$

Donde iter es la iteración actual e iter_{\max} es el número total de iteraciones.

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.

El primer ejemplo de diseño demostrado es una torre de 47 barras y 22 nodos (figura 2.45). Las variables de diseño fueron divididas en 27 grupos. Las áreas de las secciones transversales fueron tomadas de 64 valores posibles. Los nodos de la estructura fueron sujetos a tres casos de cargas:

Caso 1: 6.0 kips actuando en la dirección positiva del eje x y 14.0 kips actuando en la dirección negativa del eje y en los nodos 17 y 22.

Caso 2: 6.0 kips actuando en la dirección positiva del eje x y 14.0 en la dirección negativa del eje y en el nodo 17.

Caso 3: 6.0 kips actuando en la dirección positiva del eje x y 14.0 en la dirección negativa del eje y en el nodo 22.

El caso 1 corresponde a la condición normal de carga mientras que los casos 2 y 3 corresponden al desequilibrio de tensiones por ruptura de conductores.

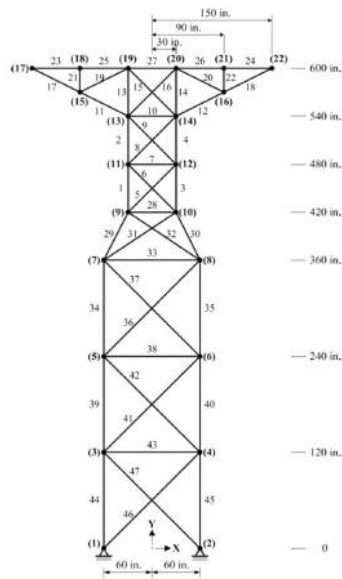


Figura 2.45. Esquema de la torre de 47 barras. (Kaveh y Ghazaan, 2018)

Los resultados del proceso de optimización son mostrados en la tabla 2.16

Tabla 2.16 Comparación de los resultados de la optimización. (Kaveh y Ghazaan, 2018)

	Áreas (in ²)			
	CBO	ECBO	VPS	MDVC-UVPS
Peso (libras)	2386.44	2375.35	2374.81	2374.09
Peso optimizado promedio (libras)	2462.76	2415.51	2415.07	2413.46
Desviación estándar del peso promedio	44.79	41.01	35.65	38.21

El óptimo fue obtenido por CBO en 9760 análisis, por ECBO en 16240 análisis y por VPS en 15,540 análisis. Además la figura 2.46 muestra que en las curvas de convergencia MDVC-UVPS alcanza la convergencia muy rápido. De estos resultados los autores concluyen que el algoritmo MDVC-UVPS obtiene mejores resultados. Los problemas de las torres de 160 y 244 barras obtienen resultados similares así que no serán referenciados en este trabajo de diploma.

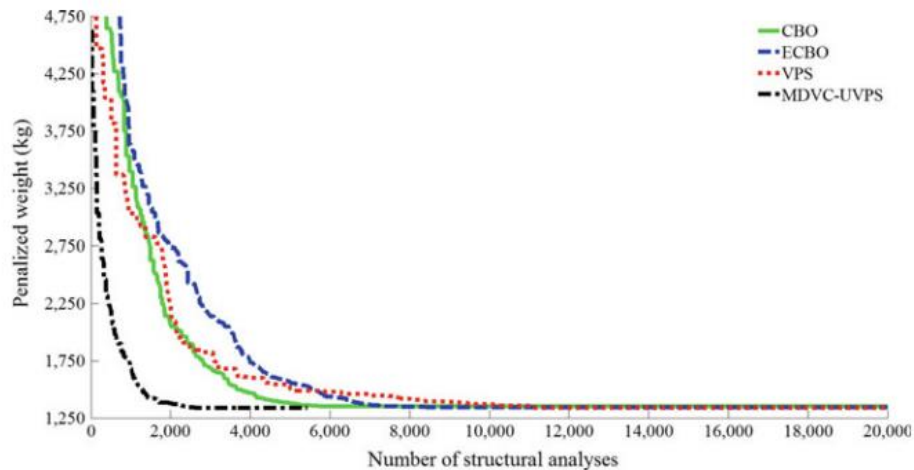


Figura 2.46. Curvas de convergencia de la torre de 160 barras. (Kaveh y Ghazaan, 2018)

2.9 Conclusiones parciales del capítulo.

1. De acuerdo con las bibliografías consultadas el objetivo de la mayoría de los trabajos de optimización de torres autoportadas se basan en la reducción del peso de la estructura de la torre, salvo (Beck y Verzenhassi (2008); Zwick, *et al.* (2012)) quienes consideran la optimización como un concepto más amplio que no se circunscribe solo al ahorro de material y definen como objetivo en su diseño óptimo el costo total de la torre.
2. La reducción de peso de la torre se logra en la mayoría de los casos variando únicamente la sección de la torre manteniendo la geometría original, sin embargo en los trabajos recientes esto solo se hace para comprobar o comparar las capacidades de un determinado grupo de algoritmos para la optimización de torres. En los trabajos más ambiciosos también se incluyen variables de diseño que modifican la forma de la torre.
3. Existen dos maneras de abordar el problema de optimización, la primera y más común es analizar íntegramente la torre como un único objeto. La segunda manera consiste en la división de la torre en segmentos o bloques constituyentes. Con este último método se obtienen mejores resultados según lo planteado por Sivakumar, *et al.* (2004).

4. No todos los elementos de la torre son objeto de optimización, algunos elementos como las barras de arriostre al pandeo no son considerados en la optimización por considerarse sus aportes estructurales despreciables.
5. A pesar del elevado número de variables que pueden influir, en los trabajos de optimización se suele fijar dos o tres variables independientes y el resto varía en función de estas. Las variables que más suelen ser tomadas como tal son las longitudes de los lados de la base. Las alturas de los elementos que conforman la torre suele ser dejada como restricción.
6. En cuanto al tratamiento de variables continuas y discretas: En los trabajos que requieren el uso de variables de naturaleza diferente de forma simultánea los autores introducen métodos para compatibilizarlas y evitar que el proceso de optimización se estanque. Estos métodos generalmente requieren la variación de un solo tipo de variable a la vez. En los trabajos con algoritmos genéticos esto no representa un problema.
7. Los miembros que conforman la torre suelen ser organizados en grupos de variables de diseño de acuerdo con la posición y función que tengan estos dentro de la torre. De esta manera según (Mohamed, *et al.* (2012); Muda, *et al.* (2013)) se obtienen mejores resultados.
8. Las variables de diseño que controlan la sección de la torre pueden ser tratadas de manera diferente, la primera es considerando la sección completa como una variable discreta disponible en un catálogo, la segunda es considerando los espesores y los anchos de las alas del perfil como variables discretas dentro de un espacio de búsqueda conformado por las combinaciones disponibles en el mercado de estos elementos.
9. Basado en el análisis de los resultados de (Kaveh y Ghazaan (2018); Tort, *et al.* (2016)) las técnicas de optimización compuestas desarrolladas específicamente para la optimización de torres autoportadas obtienen mejores resultados que aquellos algoritmos primitivos en los que se basaron.
10. Para evitar el inconveniente de trabajar con unidades de medida diferentes y las complicaciones que esto lleva consigo es recomendable emplear relaciones para volver adimensionales las restricciones. (Martínez, *et al.*, 2012; Navarrina, *et al.*, 2006)

CONCLUSIONES

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

RECOMENDACIONES

- A la empresa GDM torres o a cualquier otra entidad que se dedique al diseño de torres autosoportadas: Desarrollar un método para simplificar el diseño

optimizado de torres de manera similar a los métodos GAOT, ADAM, el algoritmo SA en dos fases y otros métodos compuestos mencionados en la presente investigación.

- A quien vaya a ser uso de esta investigación: Profundizar en los algoritmos CBO, ECBO, TLBO, BB-BC, HBB-BC, EHS, y el algoritmo híbrido MDVC-UVPS de los que no fue posible encontrar información confiable en webs libres de pago o membresía. Debido al limitado alcance de una cuenta de estudiante.

REFERENCIAS BIBLIOGRÁFICAS

- Aleli, H., 1994. "Concurrent Genetic Algorithms for Optimization of Large Structures". *Journal of Aerospace Engineering*, 7, pp. 276-297.
- Algarin, C.R., 2010. "Optimización por colonia de hormigas: aplicaciones y Tendencias". *Revista Ingeniería Solidaria*, 6, pp. 83-89.
- Alzate Acevedo, C., 2016. *Diseño Conceptual de torres de transmisión de energía utilizando el método de optimización topológica considerando análisis tridimensional y cargas producidas por cables y viento. digital*. Medellin, Colombia: Universidad Nacional de Colombia.
- 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, pp. 187-203.
- Caballero, J.A. ; Grossman, I.E., 2007. "Una revision del estado del arte en optimización". *Revista Iberoamericana de Automática e Informática Industrial*, 4, pp. 5-23.
- Caparrini, F.S., 2016. *PSO: Optimización por enjambres de partículas*.
- Chunming, W.; Tingting, S.; Bin, M. ; Jing, G., 2012 of Conference. "Research on the Optimal Layout of High-strength Steel in the Transmission Tower". En: (ed.)^(eds.), 2012 International Conference on Medical Physics and Biomedical Engineering. Jilin,China, vol., pp. 619-626.
- Efthymiou, E.; Kaziolas, D.N. ; Baniotopoulos, C.C., 2009 of Conference. "On the response of steel telecommunication masts under environmental actions and seismic loading.". En: (ed.)^(eds.), In Proc. of 11th Nordic Steel Conference (NSCC2009). Malmo, Sweden, vol., pp. pag 375-381.

- Fogel, D.B., 1967. *Artificial Intelligence Through Simulated Evolution*.
- Hanssen, L., 1981. "Computer program system for design of power transmission towers". 11th ONR Naval Structural Mechanics Symposium. pp. 12.37-12.45.
- Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems*.
- J.Felix ; G.N.Vanderplaats, 1987. "Configuration optimization of trusses subject to strength, displacement and frequency constraints ". *Journal of Mechanisms, Transmissions, and Automation in Design*, 109, pp. 223-241.
- Kaveh, A. ; Ghazaan, M.I., 2016. "Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints". *Acta Mech*, 228, pp. 307-322, DOI: DOI 10.1007/s00707-016-1725-z.
- 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, pp. pag 123-137, DOI: https://doi.org/10.1007/978-3-319-78780-0_8.
- Kaveh, A. ; Mahdavi, V.R., 2015. *Colliding Bodies Optimization Extensions and Applications*. ISBN 978-3-31-319-19658-9.
- 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 (2), pp. 197–204, DOI: DOI: 10.1061/~ASCE!0733-9445~2002!128:2~197!
- Magalhães, P.A.A.; Ferreira, T.S.; Junior, A.C.d.A.; de, O.A.; Filho, C. ; Magalhaes, C.A., 2014. "Structural analysis and structural optimization of self supporting truss towers to support a large wind turbine.". *International Journal of Optimization in Civil Engineering*, 4, pp. 341-359.
- Majid, K.I. ; Tang, X., 1984. "Optimum design of pin-jointed space structures with variable shape". *Structural Engineer*, 62, pp. 31-37.

- Martínez, S.; París, J.; Colominas, I.; Navarrina, F. ; Casteleiro, M., 2012. “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, pp. 121-135, DOI: <http://dx.doi.org/10.1016/j.rimni.2013.02.003>.
- Millán-Páramo, C., 2017. “Diseño óptimo de armaduras empleando optimización con ondas del agua”. *INGE CUC*, vol 13 (2), pp. 102-11, DOI: DOI: <http://dx.doi.org/10.17981/ingecuc.13.2.2017.11>.
- Mohamed, A.F.; Özaca, M. ; Taysil, N., 2012. “Optimal design of transmission towers using genetic algorithm”. *SDU International Technologic Science*, 4, pp. 115-123.
- Muda, Z.C.; Mustapha, K.N.; Omar, R.C.; Usman, F.; Alam, M.A. ; Thiruchelvam, S., 2013 of Conference. “Optimization of Structural Design for Sustainable Construction of Transmission Tower Based on Topographical Algorithm”. d.)^(eds.), 4th International Conference on Energy and Environment 2013 (ICEE 2013). vol., pp. 1-5.
- Navarrina, F.; Valera, A.; París, J.; Colominas, I. ; Casteleiro, M., 2006 of Conference. “Diseño óptimo de torres de alta tensión: Tratamiento conjunto de variables continuas y discretas.”. En: (ed.)^(eds.), 7º Simposio Internacional de Estructuras, Geotecnia y Materiales de Construcción. Cuba, vol., pp. 1-20.
- Palmer, A.C. ; Sheppard, D.J., 1972. “Optimal design of transmission towers by dynamic programing”. *Computers and Structures*, 2, pp. 455–468.
- Perez, R.E. ; Behdinan, K., 2007. Particle Swarm Optimization in Structural Design.
- Raj, P.P. ; Durrant, S.O., 1976. “Optimum structural design by dynamic programming”. *Journal of the Structural Division*, 102, pp. 1575-1589.
- Rao, R.V.; Savsani, V.J. ; Vakharia, D.P., 2011. “Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems”. *Computer-Aided Design* 43, pp. 303–315.

- Rechenberg, I., 1973. Evolutionsstrategie : Optimierung technischer Systemenach Prinzipien der biologischen Evolution.
- Rojas, W.J.R., 2005. Conceptos y ejemplos básicos de Programación Dinámica.
- Saka, M.P., 1984. Optimum design of space trusses with buckling constraints.
- Sanchez-Caballero, S.; Cantó, M.Á.S.; Pla-Ferrando, R. ; Raser, M.Á.P., 2012. “Nuevas Técnicas de Optimización de Estructuras.pdf”. 3ciencias, 4 (23), pp. 78-99.
- Schwefel, H.P., 1977. Numerische Optimierungun Computer-Modellenmittels der Ezdutionsstrategie,.
- Sigmund, O., 2000. “Topology optimization: A tool for the tailoring of structures”. Philosophical Transactions of the Royal Society A:Mathematical, Physical and Engineering Sciences, 358, pp. 221-227.
- Sivakumar, P.; Rajaraman, A.; Knight, G.M.S. ; Ramachandramurthy, D.S., 2004. “Object-Oriented Optimization Approach Using Genetic Algorithms for Lattice Towers”. JOURNAL OF COMPUTING IN CIVIL ENGINEERING, 18 (2004.18), pp. 162-171, DOI: 10.1061/~ASCE!0887-3801~2004!18:2~162!
- Sony, S.R. ; Airin, M.G., 2016. “Optimization of Transmission Tower using Genetic Algorithm ”. International Journal of Science and Research (IJSR) 5(9).
- Suárez, O.d.A., 2011. “Una aproximación a la heurística y metaheurísticas”. inge@UAN, 1 (1).
- Teles, M.L. ; Gomes, H.M., 2010. “Comparação de algoritmos genéticos e programação quadrática seqüencial para otimização de problemas em engenharia”. Teoria e Prática na Engenharia Civil, 15, pp. 29-39.
- Tort, C.; Sahin, S. ; Hasançebi, O.u., 2016. “Optimum design of steel lattice transmission line towers using simulated annealing and PLS-TOWER”. Computers and Structures vol 179, pp. 75-94, DOI: <http://dx.doi.org/10.1016/j.compstruc.2016.10.017>.

Zhou, A.; Qu, B.-Y.; Li, H.; Zhao, S.-Z.; Suganthan, P.N. ; Q. Zhang, 2011. “Multiobjective evolutionary algorithms: A survey of the state of the art”. *Swarm and Evolutionary Computation*, 1, pp. 32-49.

Zwick, D.; Muskulus, M. ; Moe, G., 2012. “Iterative optimization approach for the design of full-height lattice towers for offshore wind turbines”. *Energy Procedia* 24, pp. 297 – 304, DOI: doi: 10.1016/j.egypro.2012.06.112.