

**OPTIMIZACIÓN ESTRUCTURAL DE ARMADURAS EN 2D MEDIANTE UN
ALGORITMO RECOCIDO SIMULADO ALTERADO**

JESÚS ALBERTO SUÁREZ PERDOMO

**UNIVERSIDAD DE SUCRE
FACULTAD DE INGENIERÍAS
DEPARTAMENTO DE INGENIERÍA CIVIL
SINCELEJO
2017**

**OPTIMIZACIÓN ESTRUCTURAL DE ARMADURAS EN 2D MEDIANTE UN
ALGORITMO RECOCIDO SIMULADO ALTERADO**

JESÚS ALBERTO SUÁREZ PERDOMO

DIRECTOR

IC. MSc. CARLOS ANDRÉS MILLÁN PÁRAMO

**TRABAJO DE GRADO PRESENTADO COMO REQUISITO PARA OPTAR POR
EL TÍTULO DE INGENIERO CIVIL**

**UNIVERSIDAD DE SUCRE
FACULTAD DE INGENIERÍAS
DEPARTAMENTO DE INGENIERÍA CIVIL
SINCELEJO**

2017

AGRADECIMIENTOS

A la Universidad de Sucre y al departamento de Ingeniería Civil, porque debido a sus integrantes, profesores y demás, me formaron hasta obtener los mejores conocimientos y así cumplir con este objetivo.

Al director Carlos Andrés Millán Páramo por creer en mis facultades y brindar su apoyo para sacar este proyecto adelante.

DEDICATORIA

A Dios, quien siempre me ha dado la fortaleza y me ayuda a tomar las mejores decisiones para llevar a cabalidad mis metas.

A mis padres Blanca Perdomo y José Suárez quienes han sido una gran motivación para cumplir mis objetivos y así obtener este logro.

A mi hermano Eduar, por creer en mí y darme su apoyo incondicional, día a día, durante el transcurso de mi carrera universitaria.

A Lina quien con su amor y confianza ha contribuido de una u otra manera en el logro de mis metas.

Y a mis amigos, con quienes compartí estos años de estudio y se convirtieron en una familia en Sincelejo.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
1. OBJETIVO Y ORGANIZACIÓN DEL TRABAJO	3
2. ESTADO DEL ARTE	5
3. MARCO TEÓRICO	12
3.1. FUNDAMENTOS DEL ANÁLISIS MATRICIAL DE ESTRUCTURAS.....	13
3.1.1. Matriz De Rigidez De Un Elemento De La Armadura.....	13
3.1.2. Matrices De Transformación De Desplazamientos Y Fuerzas	15
3.1.2.1. Matriz De Transformación De Desplazamientos.....	16
3.1.2.2. Matriz De Transformación De Fuerzas	18
3.1.3. Matriz De Rigidez Global De Un Elemento	19
3.1.4. Matriz De Rigidez Global De La Estructura	20
3.2. OPTIMIZACIÓN.....	21
3.2.1. Metaheurística.....	22
3.2.2. Función Objetivo	22
3.2.3. Restricciones.....	23
3.3. ALGORITMO RECOCIDO SIMULADO (RS)	24
4. ALGORITMO RECOCIDO SIMULADO ALTERADO (ARSA).....	26
4.1. EVALUACIÓN DEL DESEMPEÑO DEL ARSA.....	27
4.1.1. Problemas Sin Restricciones	28
4.1.2. Problemas Con Restricciones	29
5. FORMULACIÓN Y RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE ARMADURAS	32



5.1. PROBLEMAS DE OPTIMIZACIÓN DE ARMADURAS.....	34
6. RESULTADOS Y DISCUSIÓN	35
6.1. OPTIMIZACIÓN DE ARMADURAS CON VARIABLES CONTINUAS ...	35
6.1.1. Armadura de 10 Barras	35
6.1.2. Armadura de 17 Barras	42
6.1.3. Armadura de 18 Barras	46
6.2. OPTIMIZACIÓN DE ARMADURAS CON VARIABLES DISCRETAS	49
6.2.1. Armadura de 10 Barras	49
6.2.2. Armadura de 15 Barras	52
6.2.3. Armadura de 52 Barras	55
7. CONCLUSIONES	63
8. TRABAJOS FUTUROS	64
REFERENCIAS	65

Lista de Figuras

Figura 1: Métodos de optimización [15].	6
Figura 2: Relación carga-desplazamiento.	14
Figura 3: Ángulos de transformación entre ejes locales y globales.	16
Figura 4: Esquema de transformación de desplazamientos.	17
Figura 5: Esquema de transformación de fuerzas.	18
Figura 6: Algoritmo básico SA [33].	25
Figura 7: Armadura plana compuesta por 10 barras.	36
Figura 8: Gráfica de convergencia para la armadura de 10 barras (caso 1).	40
Figura 9: Gráfica de convergencia para la armadura de 10 barras (caso 2).	41
Figura 10: Peso encontrado en cada corrida del algoritmo caso 1.	42
Figura 11: Peso encontrado en cada corrida del algoritmo caso 2.	42
Figura 12: Armadura plana compuesta por 17 barras.	43
Figura 13: Gráfica de convergencia para la armadura de 17 barras.	44
Figura 14: Peso encontrado en cada ejecución del algoritmo para la armadura de 17 barras.	46
Figura 15: Armadura plana compuesta por 18 barras.	47
Figura 16: Gráfica de convergencia para la armadura de 18 barras.	48
Figura 17: Peso encontrado en cada ejecución del algoritmo para la armadura de 18 barras.	49
Figura 18: Gráfica de convergencia para la armadura de 10 barras.	51
Figura 19: Peso encontrado en cada ejecución del algoritmo para la armadura de 10 barras.	52
Figura 20: Armadura plana compuesta por 15 barras.	52
Figura 21: Gráfica de convergencia para la armadura de 15 barras.	55
Figura 22: Peso encontrado en cada ejecución del algoritmo para la armadura de 15 barras.	55
Figura 23: Armadura plana compuesta por 52 barras.	56



Figura 24: Gráfica de convergencia para la cercha de 52 barras.	60
Figura 25: Peso encontrado en cada ejecución del algoritmo para la armadura de 52 barras.....	60

Lista de Tablas

Tabla 1: Funciones de prueba sin restricciones.	28
Tabla 2: Desempeño del SA clásico y el ARSA en funciones sin restricciones. ...	29
Tabla 3: Funciones de prueba con restricciones.	30
Tabla 4: Desempeño del ARSA en funciones con restricciones.....	31
Tabla 5: Comparación de resultados con otros autores para la armadura de 10 elementos (Caso 1).....	38
Tabla 6: Comparación de resultados con otros autores para la armadura de 10 elementos (Caso 2).....	39
Tabla 7: Comparación de resultados con otros autores para la armadura de 17 elementos.	45
Tabla 8: Comparación de resultados con otros autores para la armadura de 18 elementos.	47
Tabla 9: Comparación de resultados con otros autores para la armadura de 10 elementos.	50
Tabla 10: Comparación de resultados con otros autores para la armadura de 15 elementos.	54
Tabla 11: Valores discretos disponibles para áreas transversales de la norma AISC.	57
Tabla 12: Comparación de resultados con otros autores para la armadura de 52 elementos.	59
Tabla 13: Resumen de resultados armaduras con variables continuas.	61
Tabla 14: Resumen de resultados armaduras con variables discretas.	62

INTRODUCCIÓN

La optimización estructural (OE) se ha convertido en una de las ramas más activas de la ingeniería estructural en la última década. El número de variables de diseño, el tamaño del área de búsqueda y el número de restricciones de diseño son factores que influyen en el tiempo que necesitan los diseñadores para encontrar diseños optimizados. Tradicionalmente, realizar este proceso ha dependido en gran parte del criterio y la experiencia del diseñador. Esto ha llevado a que en muchos casos las configuraciones y dimensiones de los miembros estructurales no sean necesariamente las apropiadas.

El problema de OE es no convexo, debido a la naturaleza no convexa de la función objetivo y a la geometría compleja del dominio factible dado por las limitaciones de diseño, especialmente para el caso de que una o más de las dificultades anteriores están involucradas [1]. Por lo tanto, los enfoques deterministas son difíciles y requieren mucho tiempo para ser aplicados a estos problemas de optimización, también, un buen punto de partida para estos métodos es vital para obtener resultados exitosos. En tales circunstancias, los algoritmos metaheurísticos (estocásticos) pueden servir como alternativas apropiadas debido a la capacidad de buscar mínimos locales en espacios altamente modales y multidimensionales.

Un gran número de investigadores han introducido y aplicado diferentes algoritmos metaheurísticos para la optimización global en el diseño estructural. Entre las técnicas más destacadas se tienen los algoritmos genéticos (AG), la optimización por enjambre de partículas (PSO), la colonia de hormigas (AC), evolución diferencial (ED), recocido simulado (SA), entre otras. El SA es una técnica de optimización y búsqueda local basada en el proceso físico de enfriamiento de un metal. A pesar de que existen varios trabajos relacionados con la optimización de

estructuras mediante SA [2, 3], por lo general su formulación básica ha sido modificada para mejorar el comportamiento de convergencia [4, 5, 6, 7, 8].

En este sentido, el presente trabajo propone un Algoritmo Recocido Simulado Alterado (ARSA) para la resolución de problemas de OE (minimización de peso) de armaduras en 2D, desarrollado en el paquete computacional Matlab®. ARSA tiene dos características fundamentales que lo diferencian del SA básico: (i) paso de búsqueda y (ii) probabilidad de aceptación. Para evaluar el desempeño del algoritmo 6 armaduras clásicas reportadas en la literatura fueron analizadas. Los resultados numéricos indican que el ARSA puede minimizar el peso de armaduras mientras satisface restricciones de esfuerzos y desplazamiento.

1. OBJETIVO Y ORGANIZACIÓN DEL TRABAJO

Este trabajo tiene como objetivo desarrollar un algoritmo estable y eficiente para la resolución de problemas de optimización estructural (OE) de armaduras en 2D. La estructura es analizada por medio del Método Matricial de Rigidez y el problema de OE se resuelve utilizando el Algoritmo Recocido Simulado Alterado (ARSA). Los resultados se presentan para comprobar la validez del procedimiento utilizado.

El trabajo se encuentra organizado de la siguiente manera:

En el segundo capítulo se presenta el estado del arte de la investigación, se realiza una revisión de los estudios más relevantes pertenecientes a la OE en donde se encuentra involucrado el algoritmo recocido simulado.

En el tercer capítulo se muestran los conceptos básicos en los que está fundamentado la construcción del ARSA. Estos son los fundamentos del análisis matricial de estructuras, optimización y la técnica estocástica RS en la cual está basado el algoritmo propuesto.

En el cuarto capítulo se presenta el ARSA y los respectivos parámetros que lo controlan. El algoritmo propuesto es evaluado y validado a través de un conjunto funciones de prueba con y sin restricciones.

En el quinto capítulo se realiza la formulación del problema de OE de armaduras. Además, seis ejemplos de OE reportados en la literatura internacional son presentados.

En el sexto capítulo se expone los resultados obtenidos del proceso de optimización con el fin de comprobar la validez del procedimiento aquí planteado.



Finalmente en los capítulos 7 y 8 se presentan las conclusiones de esta investigación y algunas orientaciones y recomendaciones para la realización de futuros trabajos de investigación.

2. ESTADO DEL ARTE

En las ciencias aplicadas al igual que en el conocimiento teórico, existen problemas de optimización lineal o no lineal que involucran al mismo tiempo variables continuas y discretas. Cuando las dimensiones y la complejidad matemática del problema son grandes, aparece el fenómeno denominado explosión combinatoria, debido a la gran cantidad de soluciones factibles y no factibles que pueden aparecer.

En el caso de problemas con alta complejidad matemática y gran dimensión, las técnicas tradicionales (determinísticas) (ver Figura 1) requieren de tiempos de procesamiento muy altos y nunca alcanzan la solución óptima o soluciones subóptimas de buena calidad. Además, estos métodos no evalúan explícitamente todo el espacio de soluciones, y en su lugar evalúan unos pocos puntos y subespacios del problema. Desafortunadamente, estas metodologías sólo son exitosas en problemas muy acotados en tamaño y de media o baja complejidad. Es aquí donde las técnicas estocásticas de optimización surgen como una alternativa interesante, ya que exploran el espacio de soluciones de una manera controlada y encuentran soluciones cuasi óptimas de buena calidad, en tiempos de cómputo razonables. En este contexto, dichas técnicas son una buena opción cuando se emplean para la solución de OE.

En la actualidad existen revistas científicas (Journal of Global Optimization, Applied Mathematics and Computation, Applied Soft Computing, International Journal of Computer Mathematics, entre otras), dedicadas a éstas técnicas de optimización global, y puede decirse que su aplicación ha generado inmensos ahorros en recursos a los países y a los sectores productivos. Aunque estos procedimientos aparentan no poseer la rigurosidad matemática que si poseen las técnicas exactas, están soportadas sobre una base matemática sencilla, que se encuentra distribuida y de forma implícita en cada uno de los pasos que realizan,

además de contar con una componente de búsqueda aleatoria. Existe una gran variedad de técnicas estocásticas, y se encuentra que unas son más exitosas que otras en cada uno de los problemas, sin embargo, no existe en la actualidad un método que sea universal y que permita resolver, mejor que los otros métodos, la mayoría de los problemas existentes [9, 10]. Dentro de las técnicas más relevante se encuentran: Recocido Simulado – RS (Simulated Annealing – SA) [11], Algoritmos Genéticos (AG) [12], Optimización con Enjambre de Partículas (PSO) [13], Optimización con Colonia de Hormigas (ACO) [14], entre otras.

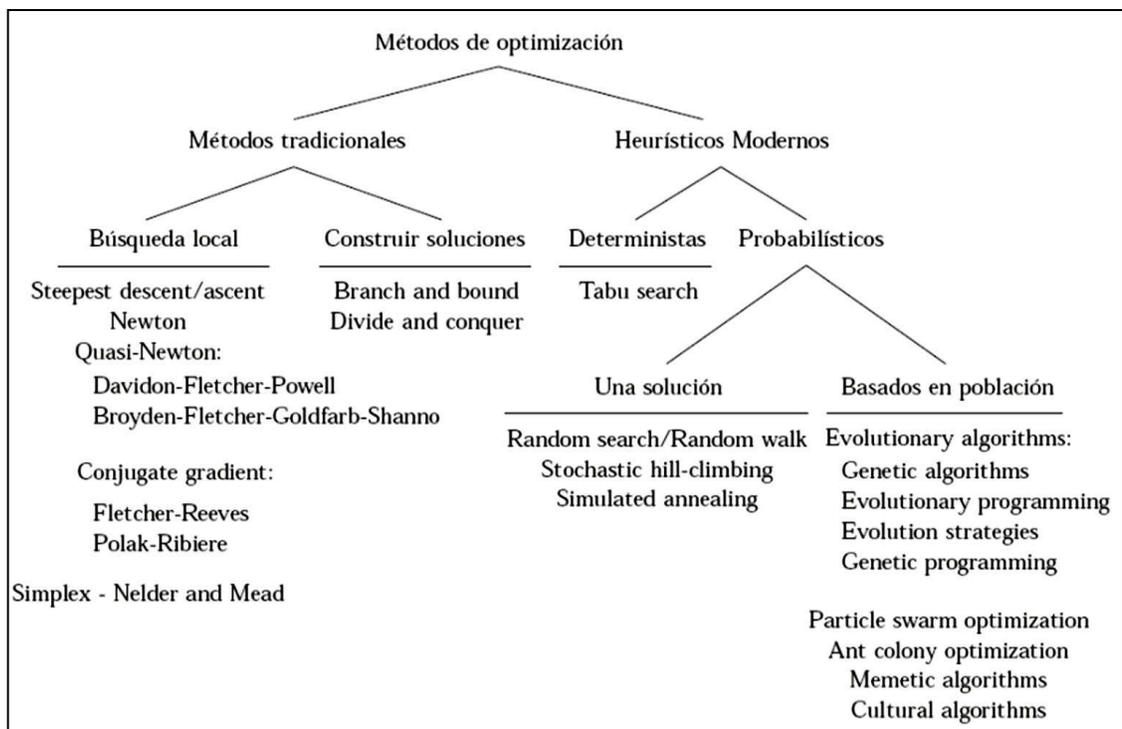


Figura 1: Métodos de optimización [15].

Estas técnicas por su simplicidad y buenos resultados en numerosos problemas, se han convertido en una herramienta muy popular, con cientos de aplicaciones en los más variados campos, sin embargo, estas contienen un conjunto de parámetros los cuales deben ser calibrados (sintonizados) para cada tipo de problema. En el caso del SA se deben calibrar la temperatura inicial y final del algoritmo, además, de la tasa de reducción de la temperatura y el número de

iteraciones. Lo mismo ocurre para el PSO, en el cual se deben ajustar el número de partículas, el parámetro cognitivo y social, entre otros. Así ocurre con las demás técnicas. Es por este motivo, que un estudio cuidadoso sobre el comportamiento de los algoritmos mencionados debe ser realizado debido a que su desempeño es dependiente del problema analizado.

Las tres principales ventajas de la optimización estocástica son: a) no son sensibles al punto de partida; b) tienen la facultad de buscar en espacios de solución grandes y, c) pueden escapar de puntos óptimos locales. Como es sabido, su mayor desventaja, la constituye su alto costo computacional. Entre las técnicas de optimización estocástica más populares está el Recocido Simulado [11]. La característica más importante del SA, es su alta confiabilidad al localizar óptimos globales en problemas con gran número de variables de diseño.

En cuanto a aplicaciones de técnicas estocásticas a problemas de OE, Sonmez [14], presentó el algoritmo Artificial Bee Colony – Adaptive Penalty (ABC - AP), que está basado en el comportamiento inteligente del enjambre de abejas integrado junto a una función de penalidad adaptativa propuesta por Hasançebi [16]. El algoritmo fue usado para resolver cinco problemas de prueba de la optimización estructural (armaduras planas y espaciales) cuyos resultados fueron comparados con algoritmos desarrollados por diversos autores y demuestra la aplicabilidad del ABC-AP.

Ho- Huu et al. [17] propusieron la metodología adaptive elitist Differential Evolution (aeDE), la cual fue implementada en seis problemas de optimización de cerchas mediante variables discretas, en donde señala las ventajas de la metodología propuesta respecto a su algoritmo base denotado como Differential Evolution [18], cuyo proceso se basa, al igual que otros algoritmos de optimización pertenecientes a la computación evolutiva como el AG y el PSO, en mantener una población de soluciones candidatas, las cuales se recombinan y mutan para producir nuevos aspirantes, siendo elegido para pasar a la siguiente generación el

que presente una mejor solución (según la función objetivo). Los resultados conseguidos fueron comparados con los obtenidos por diversos autores.

Lee y Geem [19], aplicaron el algoritmo Harmony Search (HS), el cual está basado en el proceso de componer música donde se busca la mejor armonía (mejor solución), generando un vector inicial de armonías (posibles soluciones), siendo la peor solución reemplazada por otra determinada estocásticamente y ajustada mediante parámetros propios del algoritmo. Los resultados obtenidos por esta metodología aplicada en problemas de referencia de optimización estructural fueron comparados con diversos autores demostrando así que el HS es capaz de resolver diferentes problemas estructurales con distintos grados de dificultad.

En cuanto al RS, su aplicación ha sido evidenciada en los trabajos de Balling [2] y Bennage [3], desde entonces, su forma original ha sido modificada para mejorar su desempeño. Shea y colaboradores [20, 21], desarrollaron un código para el diseño topológico de cerchas basado en la combinación de SA clásico y reglas de forma, denominado Shape Annealing Method (SAM). Mientras que en el SA básico el umbral de probabilidad para seleccionar cualquier movimiento es asignado por el usuario y no cambia en todo el proceso de optimización, en este trabajo utilizaron el método de Hustin [22] para la selección de reglas dinámicas. La probabilidad de selección de cualquier movimiento se basa en las tasas de éxito pasadas y factores de calidad se asignan a cada variable de diseño para seleccionar los movimientos más útiles en el proceso de optimización.

Shim y Manoochehri [23], resolvieron problemas de optimización de forma no lineal, conduciendo la búsqueda del SA mediante problemas aproximados en donde se linealizaron las restricciones de esfuerzos. Para asegurar la viabilidad del diseño, un factor de corrección ajusta los términos de tensión basados en la relación máxima del cambio en la tensión linealizada a la variación en los valores de tensión reales. Sin embargo, el proceso de ajuste puede implicar demasiada heurística.

Pantelides and Tzan [24] utilizaron el SA para optimizar sistemas estructurales bajo cargas dinámicas. En el proceso de optimización fue incluido un análisis de sensibilidad para identificar las variables de diseño que podían ser modificadas para reducir los desplazamientos globales. Cada vez que un nuevo diseño mejoró al anterior, se estipularon límites para restringir la generación aleatoria de diseños de prueba dentro de un área cercana al nuevo mejor valor. Sin embargo, esta estrategia posee dos grandes limitantes: i) Cambiar los límites no asegura caer dentro de la región factible; ii) el mismo porcentaje de movimientos es asignado a todas las variables a pesar las diferentes cantidades de no linealidad asociadas a cada variable [6].

Hasançebi y Erbatur [4], demostraron la posibilidad de optimizar simultáneamente el tamaño, la forma y topología de estructuras en celosía mediante la aplicación del SA básico. Del mismo modo, trabajaron en la optimización de problemas estructurales de armaduras en 2 y 3 dimensiones [5], reformulando el parámetro de Boltzmann intrínseco en el SA clásico para acelerar y realzar la productividad del algoritmo en términos de convergencia y fiabilidad.

Chen y Su [25], hicieron énfasis en que el SA clásico puede necesitar de muchos análisis estructurales para converger en el diseño óptimo. Por tal motivo dos mejoras al SA clásico fueron propuestas: i) las regiones factibles del espacio de diseño son estimadas por la linealización de restricciones y límites de movimiento que se asignan a las variables de diseño; ii) el proceso de optimización empieza comúnmente desde un punto de diseño factible.

Kripka [26], utilizó una técnica de penalización dinámica y la unió junto con el SA clásico para resolver problemas de optimización estructural, cuyas variables, áreas transversales de los elementos, fueron tomadas como variables discretas. Los resultados de este estudio se encuentran en capítulos posteriores.

Erdal y Sonmez [27], adoptaron un algoritmo basado en el SA clásico, denotado como Direct search Simulated Annealing (DSA), propuesto por Ali et al. [28], en el

cual un conjunto de configuraciones actuales son utilizadas en lugar de un solo diseño de prueba. Estas consisten en que cada vez que una prueba de diseño generada es mejor que la peor incluida en el conjunto actual de configuraciones, el peor diseño es reemplazado por el nuevo obtenido. La técnica presentó un buen comportamiento tanto en la optimización del diseño lay-up laminado como en la optimización de forma de estructuras en 2D [29].

Lamberti [6], desarrolló un algoritmo de optimización basado en el SA clásico, denotado como Corrected Multi-Level & Multi-Point Simulated Annealing (CMLPSA). Su mecanismo de búsqueda consiste en que cada candidato de diseño es seleccionado de una población de puntos de prueba generados aleatoriamente. Esta estrategia fue adoptada para diseños factibles y no factibles. En el primer caso, las perturbaciones dadas a las variables de diseño son forzadas a seguir la tasa actual de cambios exhibidos por la función de coste. En el último caso, una aproximación lineal de cuarto orden busca alrededor de cada punto de prueba factible generado en el actual ciclo de recocido del algoritmo. El CMLPSA posee la estrategia de multinivel en donde cada punto es generado para perturbar todas las variables de diseño simultáneamente (nivel global) o una por una (nivel local), la búsqueda global o local fue interpretada basado en la tendencia actual percibida en el proceso de optimización. Esta estrategia fue probada en 6 problemas de OE.

Hasançebi y Çarbaş [30], propusieron alterar el parámetro de probabilístico de Boltzmann para mejorar el comportamiento del SA clásico. El rendimiento de esta mejora fue investigado junto con 2 ejemplos de OE, los cuales fueron diseñados a nivel práctico, de acuerdo a lo previsto en las especificaciones de Allowable Stress Design (ASD). Martínez [31], de igual manera, demostró a nivel práctico (basado en la normativa española) la eficiencia del algoritmo SA clásico en la optimización de estructuras de transporte de energía. El SA clásico en este caso fue complementado con un análisis de sensibilidad de primer orden para reducir el

coste computacional, además de implementar funciones de penalización exterior para mejorar el tratamiento de las restricciones del diseño.

Asl [32], presentó un algoritmo híbrido que combina el poder de exploración del Algoritmo Genético (AG) con la capacidad de explotación de SA básico. Su capacidad fue comprobada al optimizar problemas de estructuras planas y espaciales.

Recientemente, Millán [33] propuso un Algoritmo Simulated Annealing Modificado (ASAM) con 3 modificaciones importantes (exploración preliminar, paso de búsqueda y probabilidad de aceptación) que lo diferencian del SA básico. El comportamiento eficaz de esta metodología en problemas de OE fue demostrado mediante la optimización (minimización de peso) de armaduras planas con variables discretas [7] y continuas [8].

Lo anterior demuestra que el SA es una técnica computacional de gran importancia, la cual se ha destacado en los últimos años por presentar una base estable para la creación de algoritmos robustos con la capacidad de resolver problemas de alta complejidad, con múltiples variables, presentes en la ingeniería estructural.

3. MARCO TEÓRICO

El diseño estructural, como rama importante en el campo de la ingeniería civil, se basa en encontrar las secciones que mejor se adecuen a la estructura, satisfaciendo las solicitaciones a las que está sometida. En este sentido, el diseñador debe asumir ciertas secciones de una base de datos disponible antes de empezar el análisis, de modo que se puedan hallar los esfuerzos y desplazamientos en los elementos. Esta metodología hace que el diseño sea iterativo, además, el encontrar un equilibrio entre economía y eficiencia (diseño óptimo) convierten este proceso en un problema de optimización.

El concepto de optimización está en la base de la ingeniería, puesto que la función clásica del ingeniero es diseñar sistemas novedosos, mejores, más eficientes, y menos costosos. En términos matemáticos, el problema del diseño óptimo se formula mediante la selección de un conjunto de variables de diseño, de forma que se minimice una función de coste, o función objetivo, y se verifiquen unas series de restricciones. El problema así descrito se conoce con el nombre de "problema de optimización", y al conjunto de métodos que permiten obtener su solución, se las denomina "técnicas de optimización".

El proceso de encontrar la mejor solución puede ser una tarea desgastante a nivel de diseño y por ende requiere de tiempos computacionales bastante altos cuando las estructuras son demasiado complejas. Por esta razón han surgido metodologías de diseños basadas en métodos heurísticos (estocásticos) para obtener soluciones aproximadas a problemas complejos, reduciendo los recursos computacionales. A continuación, se describen los fundamentos teóricos del diseño de armaduras (análisis matricial) y de un problema de optimización, y la teoría alrededor del algoritmo RS.

3.1. FUNDAMENTOS DEL ANÁLISIS MATRICIAL DE ESTRUCTURAS

El método matricial de rigideces se fundamenta en la relación de los desplazamientos y las fuerzas de los elementos de una estructura. A partir de las relaciones de rigidez entre desplazamientos y fuerzas, es posible plantear ciertas ecuaciones con el mismo fundamento, generadas aplicando las ecuaciones de movimiento y las ecuaciones constitutivas de los materiales. De esta manera, se obtiene un sistema con un número n de ecuaciones con n grados de libertad. Resolviendo este sistema para los desplazamientos de la estructura, es posible sustituir dichos desplazamientos conocidos en las ecuaciones matriciales de rigidez para hallar la fuerzas en los elementos [34].

Para la formulación de las ecuaciones del método, se parte de las siguientes hipótesis:

- El comportamiento es elástico, lineal, homogéneo e isotrópico.
- Desplazamientos infinitesimales.
- Cada punto del cuerpo tiene masa (continuidad).
- Todo el cuerpo al ser cargado tiene un régimen interno de tensiones.
- El comportamiento es adiabático.

3.1.1. Matriz De Rigidez De Un Elemento De La Armadura

La matriz de rigidez de un elemento representa la relación entre carga y desplazamiento presentes en un miembro estructural. Cada elemento solo puede desplazarse a lo largo de su eje local (eje x'), debido a que las cargas están aplicadas a lo largo de este, siendo posible entonces dos desplazamientos independientes. Como se observa en la Figura 2a, se impone un desplazamiento positivo d_N sobre el extremo inicial del miembro mientras el final se mantiene articulado, siendo así, las fuerzas desarrolladas en los extremos del elemento son [35]:

$$q'_N = \frac{A*E}{L} * d_N; \quad q'_F = -\frac{A*E}{L} * d_N;$$

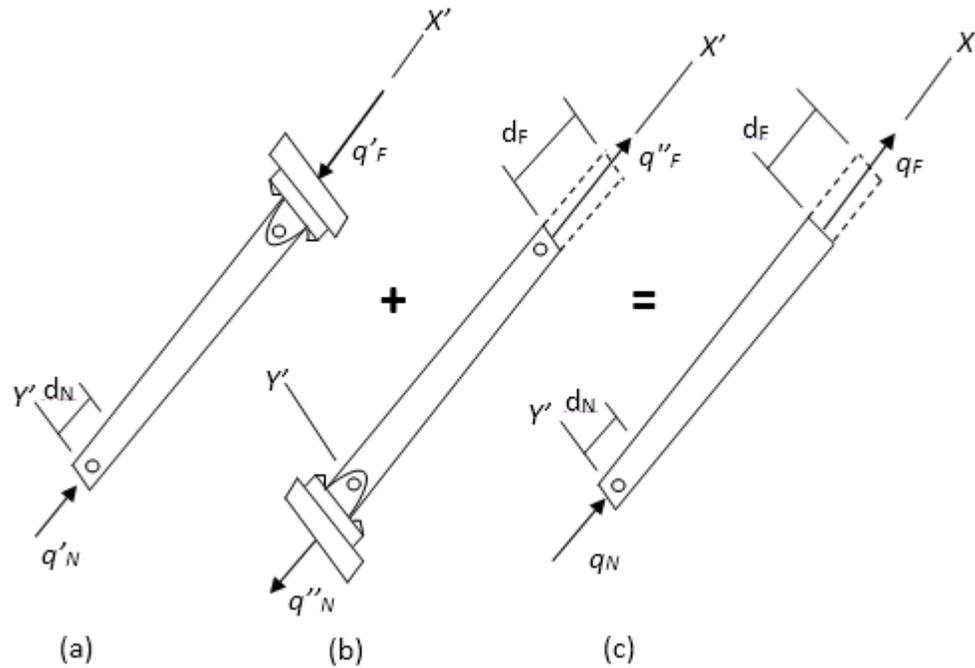


Figura 2: Relación carga-desplazamiento.

Del mismo modo, cuando se impone un desplazamiento positivo en el extremo final del elemento, Figura 2b, manteniendo el inicial articulado, se obtienen las siguientes fuerzas:

$$q''_N = -\frac{A*E}{L} * d_F; \quad q''_F = \frac{A*E}{L} * d_F;$$

Debido a la superposición de las Figura 2a y Figura 2b, se genera la Figura 2c, lo cual corresponde a:

$$q_N = \frac{A*E}{L} * d_N - \frac{A*E}{L} * d_F; \quad (1)$$

$$q_F = \frac{A*E}{L} * d_F - \frac{A*E}{L} * d_N; \quad (2)$$

Estas ecuaciones de relación carga-desplazamiento pueden ser representadas matricialmente de la siguiente manera:

$$\begin{bmatrix} q_N \\ q_F \end{bmatrix} = \frac{A*E}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} d_N \\ d_F \end{bmatrix};$$

o

$$q = k'd; \quad (3)$$

donde

$$k' = \frac{A*E}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}; \quad (4)$$

Esta matriz k' se denota Matriz de Rigidez de un elemento, siendo la misma para cada miembro que compone la estructura. Los cuatro elementos que conforman esta matriz se llaman Coeficientes de Influencia de Rigidez del Elemento y corresponden, por ejemplo, los de la primera columna, a la fuerza en el nodo inicial cuando se impone un desplazamiento unitario ($d_N=1$), estando el nodo final fijo. Por lo tanto, la fuerza se calcula con las posiciones de la matriz $i = j = 1$ de la ecuación 4, quedando la fuerza representada de la siguiente manera:

$$q_N = k'_{11} = \frac{A*E}{L};$$

Para el cálculo de la fuerza en el nodo final se tienen en cuenta las posiciones $i=2$, $j=1$, así obtenemos:

$$q_F = k'_{21} = -\frac{A*E}{L};$$

De la misma manera, para las otras dos posiciones de la matriz de rigidez (segunda columna), se calculan las fuerzas correspondientes en los extremos del elemento cuando se es inducido una deformación unitaria ($d_F=1$), en el nodo final.

3.1.2. Matrices De Transformación De Desplazamientos Y Fuerzas

Como las ecuaciones anteriormente mostradas están referenciadas en un sistema de ejes locales (x', y'), estas deben ser transformadas al sistema global (x, y) de la estructura, el cual se observa en la Figura 3, donde se encuentra un elemento estructural referenciado en ejes local y global con los ángulos θ_x y θ_y entre ellos. Los valores correspondientes a los cosenos de estos ángulos, son los encargados

de realizar las respectivas matrices de transformación de fuerzas y desplazamientos, los cuales son representados como $\lambda_x = \cos(\theta_x)$ y $\lambda_y = \cos(\theta_y)$, por lo tanto, teniendo en cuenta la Figura 3.

$$\lambda_x = \cos(\theta_x) = \frac{x_f - x_n}{L} = \frac{x_f - x_n}{\sqrt{(x_f - x_n)^2 + (y_f - y_n)^2}}; \quad (5)$$

$$\lambda_y = \cos(\theta_y) = \frac{y_f - y_n}{L} = \frac{y_f - y_n}{\sqrt{(x_f - x_n)^2 + (y_f - y_n)^2}}; \quad (6)$$

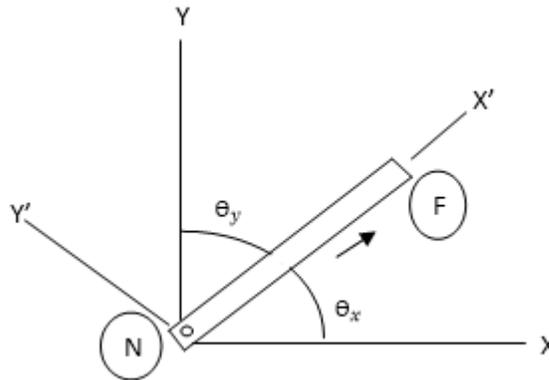


Figura 3: Ángulos de transformación entre ejes locales y globales.

3.1.2.1. Matriz De Transformación De Desplazamientos

En las coordenadas globales, cada extremo de un elemento posee dos grados de libertad, es decir, el nodo inicial (N) tiene desplazamientos en los ejes x y y (D_{NX} y D_{NY}), al igual que el nodo final (F) (D_{FX} y D_{FY}). Por tanto, en las Figura 4a y Figura 4b, cuando se es aplicado un desplazamiento D_{NX} en el nodo inicial, teniendo el otro articulado, se produce un desplazamiento $D_{NX} * \cos(\theta_x)$ en el eje local del elemento, del mismo modo, un desplazamiento global D_{NY} ocasionará un desplazamiento $D_{NY} * \cos(\theta_y)$ a lo largo del mismo eje local x' . El efecto de ambos desplazamientos conlleva a:

$$d_N = D_{NX} * \cos(\theta_x) + D_{NY} * \cos(\theta_y);$$

Del mismo modo, realizando un proceso equivalente en el extremo final manteniendo el inicial articulado, como lo muestran las Figura 4c y Figura 4d, el desplazamiento quedará expresado de la siguiente manera:

$$d_F = D_{FX} * \cos(\theta_x) + D_{FY} * \cos(\theta_y);$$

Ahora, teniendo en cuenta que $\lambda_x = \cos(\theta_x)$ y $\lambda_y = \cos(\theta_y)$ representan los cosenos directores del elemento, tenemos:

$$d_N = D_{NX} * \lambda_x + D_{NY} * \lambda_y;$$

$$d_F = D_{FX} * \lambda_x + D_{FY} * \lambda_y;$$

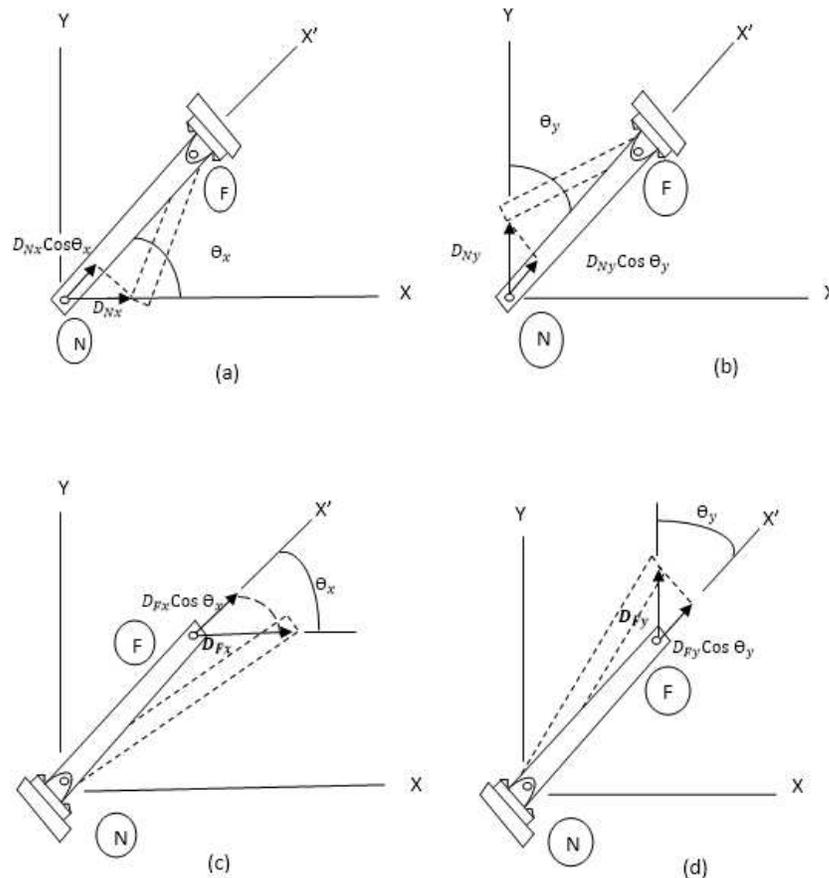


Figura 4: Esquema de transformación de desplazamientos.

Lo cual matricialmente puede expresarse de la siguiente manera:

$$\begin{bmatrix} dN \\ dF \end{bmatrix} = \begin{bmatrix} \lambda_x & \lambda_y & 0 & 0 \\ 0 & 0 & \lambda_x & \lambda_y \end{bmatrix} \begin{bmatrix} DN_x \\ DN_y \\ DF_x \\ DF_y \end{bmatrix}; \quad (7)$$

o

$$d = T * D; \quad (8)$$

donde

$$T = \begin{bmatrix} \lambda_x & \lambda_y & 0 & 0 \\ 0 & 0 & \lambda_x & \lambda_y \end{bmatrix}; \quad (9)$$

3.1.2.2. Matriz De Transformación De Fuerzas

Realizando un análisis similar al anterior, aplicando una fuerza q_N al extremo inicial manteniendo el final articulado, (Figura 5a), las componentes globales de esta fuerza quedarían de la siguiente manera:

$$Q_{NX} = q_N * \cos(\theta_x) \quad Q_{NY} = q_N * \cos(\theta_y)$$

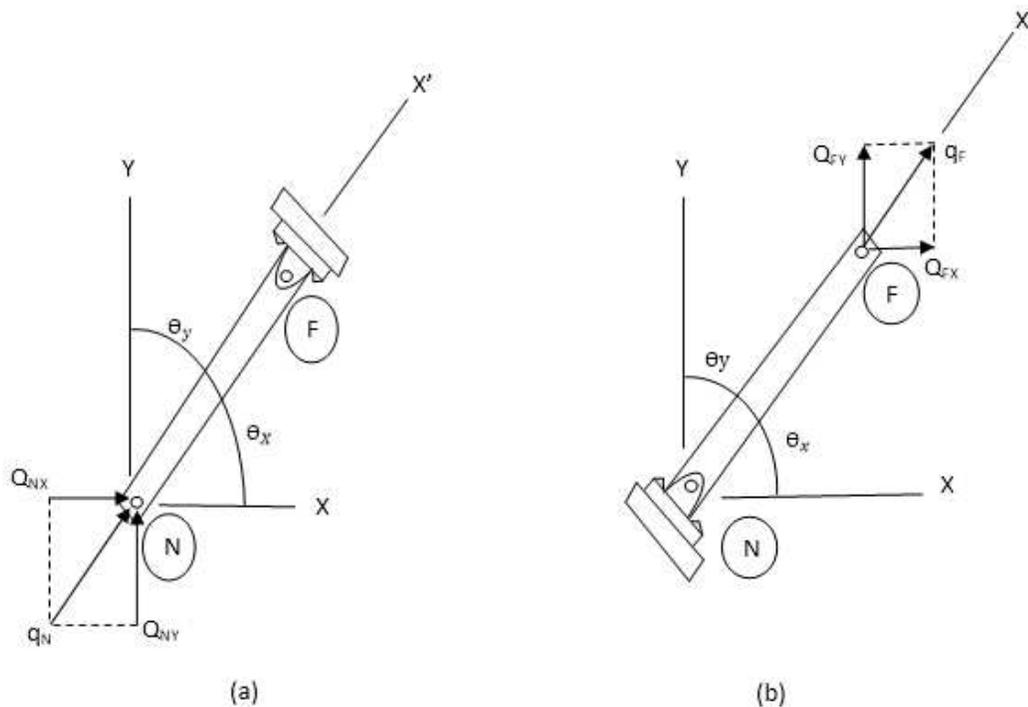


Figura 5: Esquema de transformación de fuerzas.

Del mismo modo, aplicando la fuerza q_F (Figura 5b), las componentes globales serían:

$$Q_{FX} = q_F * \cos(\theta_x) \quad Q_{FY} = q_F * \cos(\theta_y)$$

Ahora, mediante los cosenos directores $\lambda_x = \cos(\theta_x)$ y $\lambda_y = \cos(\theta_y)$, las ecuaciones se expresarían de la siguiente manera:

$$\begin{aligned} Q_{NX} &= q_N * \lambda_x & Q_{NY} &= q_N * \lambda_y \\ Q_{FX} &= q_F * \lambda_x & Q_{FY} &= q_F * \lambda_y \end{aligned}$$

Las cuales, de forma matricial, se expresan como sigue:

$$\begin{bmatrix} Q_{NX} \\ Q_{NY} \\ Q_{FX} \\ Q_{FY} \end{bmatrix} = \begin{bmatrix} \lambda_x & 0 \\ \lambda_y & 0 \\ 0 & \lambda_x \\ 0 & \lambda_y \end{bmatrix} \begin{bmatrix} q_N \\ q_F \end{bmatrix}; \quad (10)$$

o

$$Q = T^T q; \quad (11)$$

donde

$$T^T = \begin{bmatrix} \lambda_x & 0 \\ \lambda_y & 0 \\ 0 & \lambda_x \\ 0 & \lambda_y \end{bmatrix}; \quad (12)$$

Esta matriz T^T es la encargada de transformar las fuerzas locales q que actúan en los extremos de los elementos, en las cuatro componentes Q globales (x, y). Siendo la transpuesta de la matriz de transformación de desplazamientos, ecuación 9.

3.1.3. Matriz De Rigidez Global De Un Elemento

Realizando una combinación de las ecuaciones anteriores es posible obtener la matriz de rigidez global de un elemento. Por tanto, si se sustituye la ecuación 8 en

la ecuación 3, se obtienen las fuerzas q del elemento en términos de los desplazamientos globales D , esto es:

$$q = k' * T * D; \quad (13)$$

Sustituyéndola en la ecuación 11, obtenemos:

$$Q = T^T * k' * T * D;$$

o

$$Q = k * D; \quad (14)$$

donde

$$k = T^T * k' * T; \quad (15)$$

Siendo k la matriz de rigidez global del elemento y teniendo en cuenta que T^T , k' y T son conocidas, esta quedaría expresada de la siguiente manera:

$$k = \begin{bmatrix} \lambda x & 0 \\ \lambda y & 0 \\ 0 & \lambda x \\ 0 & \lambda y \end{bmatrix} \frac{A * E}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \lambda x & \lambda y & 0 & 0 \\ 0 & 0 & \lambda x & \lambda y \end{bmatrix};$$

$$k = \frac{A * E}{L} * \begin{matrix} & \begin{matrix} N_x & N_y & F_x & F_y \end{matrix} \\ \begin{matrix} \lambda x^2 & \lambda x \lambda y & -\lambda x^2 & -\lambda x \lambda y \end{matrix} & \begin{matrix} N_x \\ N_y \\ F_x \\ F_y \end{matrix} \\ \begin{matrix} \lambda x \lambda y & \lambda y^2 & -\lambda x \lambda y & -\lambda y^2 \end{matrix} \\ \begin{matrix} -\lambda x^2 & -\lambda x \lambda y & \lambda x^2 & \lambda x \lambda y \end{matrix} \\ \begin{matrix} -\lambda x \lambda y & -\lambda y^2 & \lambda x \lambda y & \lambda y^2 \end{matrix} \end{matrix} \quad (16)$$

Donde cada una de las filas y las columnas se encuentran relacionadas según los grados de libertad presentes en los nodos inicial (N) y final (F) del elemento, tal como se referencia en la notación correspondiente de la matriz (ecuación 16).

3.1.4. Matriz De Rigidez Global De La Estructura

La matriz de rigidez global (K) de la estructura se obtiene después de tener todas las matrices de rigidez de cada elemento (k). Para realizar el ensamble de esta

matriz global, se deben identificar los elementos y cada nodo de la estructura con sus respectivos grados de libertad, esto será necesario para identificar las filas y columnas de cada matriz de rigidez de cada miembro y referenciarla con números (códigos) como se muestra en la ecuación 16 (N_x , N_y , F_x , F_y). La matriz de rigidez global tendrá entonces un tamaño igual al número de grados de libertad total de la estructura, es decir, de número (código) “n” más alto asignado, ecuación 17.

Cuando se realiza el ensamble de las matrices k en la matriz global K , cada elemento de referencia de la matriz k , se coloca en su misma referencia en la matriz K . Cuando dos o más miembros de la estructura llegan a un mismo nodo, algunos de los elementos de cada una de las matrices k se asignan a la misma posición en la matriz K , cuando esto ocurre, los valores de estos elementos de posición en común son sumados algebraicamente.

$$\begin{matrix}
 N1x & N1y & \dots & Fnx & Fny \\
 \\
 K = \begin{bmatrix}
 & & \dots & & \\
 \vdots & \ddots & & & \\
 & & \ddots & & \\
 & & & \ddots & \\
 & & & & \dots
 \end{bmatrix} & \begin{bmatrix}
 N1x \\
 N1y \\
 \vdots \\
 Fnx \\
 Fny
 \end{bmatrix} & (17)
 \end{matrix}$$

3.2. OPTIMIZACIÓN

El tema de optimización, en la ingeniería, se considera un tema relativamente nuevo, en el cual se busca mejorar las condiciones de trabajos y costos de las piezas o estructuras a optimizar. Este proceso cualitativo y/o cuantitativo tiene como objetivo encontrar las mejores soluciones posibles a aquellos problemas de ingeniería (piezas, estructuras, procesos de manufactura, entre otros) en donde a su vez, se quiere disminuir el tiempo de búsqueda de estas mejores soluciones.

Debido a la gran cantidad de variables que se ven involucradas en un problema de ingeniería, muchas de sus combinaciones pueden generar una solución factible, aunque no necesariamente sea la mejor. Por lo tanto, el concepto de optimización se refiere a la selección de la mejor solución de un conjunto de soluciones

posibles, de acuerdo a un criterio de selección dado. Para esto, se deben emplear métodos de optimización que permitan conocer los mejores valores que puede asumir dicho conjunto de variables [34]. Tres conceptos fundamentales engloban la optimización de un problema; metaheurística, función objetivo y restricciones.

3.2.1. Metaheurística

La heurística es la capacidad de resolver un problema, sin que haya prueba matemática del desempeño de la solución [36]. Partiendo de esto, se puede describir el concepto de una metaheurística, palabra que viene del griego “Meta”, que quiere decir más allá y del griego “Heurística”, que quiere decir encontrar, como un método o técnica que permite el cálculo y búsqueda de soluciones óptimas en problemas de elevada complejidad, sin tener que evaluar todo el espacio de búsqueda, en tiempos de cómputo razonables.

Estas técnicas se caracterizan por emular fenómenos naturales que poseen una componente aleatoria que las clasifica como un proceso de optimización estocástico. Este hace referencia al proceso de búsqueda de una solución óptima (maximizar o minimizar una función objetivo) en el cual se encuentre presente la aleatoriedad de manera constructiva [9].

3.2.2. Función Objetivo

La función objetivo o de coste, es una formulación matemática (en función de las variables de diseño) que mide cualitativamente el funcionamiento del sistema en un proceso de optimización, de modo que lo que se busca es la maximización o minimización de esta, enmarcado en una gama de posibles soluciones. El problema así descrito puede ser expresado de la siguiente manera [34]:

- Dada: una función $f: A \rightarrow R$ donde A es un conjunto de números reales.
- Buscar: un elemento x_0 en A tal que $f(x_0) \leq f(x)$ para todo x en A ("minimización") o tal que $f(x_0) \geq f(x)$ para todo x en A ("maximización").

La complejidad del problema de optimización dependerá entonces de la cantidad de variables presentes, las cuales definen el espacio de búsqueda de soluciones. Existen diferentes tipos de función objetivo, de acuerdo al tipo de variable, formulación matemática y operatividad. Entre las más importantes, se tiene:

- Lineal y no lineal.
- Unimodal y multimodal.
- Estática y dinámica.
- Sin ruido y con ruido.
- Un objetivo y multiobjetivo

3.2.3. Restricciones

Las restricciones son relaciones que las variables deben satisfacer, estas limitan el espacio de búsqueda factible entre los cuales se puede mover cada variable de diseño, expresadas generalmente mediante ecuaciones e inecuaciones. Las restricciones o limitaciones pueden ser impuestas directamente por una o varias variables (restricción explícita), o puede ser impuesta por un fenómeno que no tiene relación directa con las variables de diseño (restricción implícita) [37]. Debido a que éstas son aplicadas en problemas de tipo físico-mecánico, se pueden clasificar en dos grandes grupos [34]:

- Funcionales: Se refieren a comportamiento físico-mecánico del sistema.
- Geométricas: Se refieren a las dimensiones y topología de los elementos.

En el ámbito estructural, las restricciones se refieren al comportamiento de la estructura y su estabilidad, y se imponen generalmente a los esfuerzos y deformaciones en los elementos (funcionales) y por ende a sus dimensiones mínimas y máximas (geométricas).

3.3. ALGORITMO RECOCIDO SIMULADO (RS)

Recocido Simulado (RS) pertenece a la clase de algoritmos de búsqueda local que permiten movimientos ascendentes para evitar quedar atrapado prematuramente en un óptimo local. Estos algoritmos juegan un rol especial dentro del campo de la optimización por 2 razones: en primer lugar, sus resultados han sido muy exitosos cuando se han aplicado a un amplio espectro de problemas prácticos; en segundo lugar, poseen una componente estadística que facilita su convergencia [33]. El origen de esta técnica estocástica y la elección del criterio de aceptación de la nueva solución generada se basan en el proceso físico de enfriamiento del metal propuesto por Metropolis [38] en los años cincuenta. Treinta años más tarde se creó el algoritmo actual [11]. El enfriamiento es un proceso termal a través del cual se alcanzan estados de energía mínima de un sólido en un baño de calor, y consiste en 2 pasos: primero se incrementa la temperatura en el baño de calor hasta un valor máximo al que se funde el sólido; luego se disminuye la temperatura cuidadosamente hasta que las partículas del sólido se reorganizan para conformar un reticulado altamente estructurado y la energía del sistema es mínima [39].

El funcionamiento de SA se puede describir de la siguiente manera. Comienza con un cierto estado S . A través de un proceso único crea un estado vecino S' al estado inicial. Si la energía o la evaluación del estado S' son menores que el estado S cambia el estado S por S' . Si la evaluación de S' es mayor que la de S puede estar empeorando, por lo que elige S' en vez de S con una cierta probabilidad que depende de las diferencias en las evaluaciones y la temperatura del sistema T . La probabilidad de aceptar un peor estado se calcula por la siguiente ecuación:

$$P(\Delta f, T) = e^{\left(\frac{\Delta f}{T}\right)}; \quad (18)$$

donde,

P : probabilidad de aceptar el nuevo estado.

Δf : diferencia de las evaluaciones de la función para cada estado.

T : temperatura del sistema.

e : número de Euler.

Inicialmente, con valores grandes para T , frecuentemente se aceptan soluciones con un mayor valor de función objetivo; a medida que el valor de T disminuye, tal tipo de soluciones raramente se aceptan, y cuando T se acerca a cero, solo se aceptan aquellas soluciones que mejoran la anterior. Varios estudios teóricos demuestran que sí T decrece con la suficiente lentitud, el proceso converge a la solución óptima [40]. La función para reducción de temperatura más utilizada es: $T_{(k+1)}=T_k \cdot \alpha$, donde $T_{(k+1)}$ es el nuevo valor ajustado de T , T_k corresponde al previo valor de T y α es una constante que está comprendida en el intervalo [0.8-0.99]. Por tanto, podemos definir un algoritmo básico de SA para problemas de optimización como se indica en la Figura 6.

```

Sea  $f(s)$  el coste de la solución  $s$  y sea  $G(s)$  su entorno.
Selecciona una solución inicial  $S$ ;
Seleccionar una temperatura inicial  $T_i > 0$ ;
Seleccionar una función de reducción de la temperatura  $\alpha$ ;
Seleccionar un número de iteraciones  $N$ ;
Seleccionar un criterio de parada;
REPETIR
    REPETIR
        Seleccionar aleatoriamente una solución  $S' \in G(s)$ ;
        Sea  $\Delta f = f(S') - f(S)$ ;
        SI  $\Delta f < 0$  ENTONCES  $S_{OPTIMO} = S'$ ;
        SINO
            Generar aleatoriamente  $t \in L(0, 1)$ ;
            SI  $t < e^{(\Delta f/T)}$  ENTONCES  $S_{OPTIMO} = S'$ ;
        FIN SINO
    HASTAQUE iteraciones= $N$ 
     $T_{i+1} = T_i \cdot \alpha$ ;
HASTAQUE criterio de parada=CUMPLA

La mejor solución encontrada será la solución dada por el algoritmo
    
```

Figura 6: Algoritmo básico SA [33].

4. ALGORITMO RECOCIDO SIMULADO ALTERADO (ARSA)

La técnica estocástica de optimización Recocido Simulado Alterado (ARSA), es propuesta con el fin de mejorar la capacidad de exploración del espacio de búsqueda y convergencia del algoritmo *Simulated Annealing* (SA) original. El SA comienza con una solución inicial escogida aleatoriamente en el espacio de búsqueda y la compara con otra que también se selecciona estocásticamente en el espacio de búsqueda, lo que afecta al algoritmo cuando se tienen funciones altamente dimensionales y modales generando mayores tiempos de búsqueda y soluciones subóptimas. Además, la probabilidad de aceptación de una solución peor se encuentra en un intervalo de entre 0 y 1, lo cual causa que a temperaturas iniciales el algoritmo acepte un gran número de soluciones de peor calidad (aumentando el riesgo de quedar atrapado en un óptimo local). Por tales razones, el algoritmo propuesto tiene dos modificaciones que lo diferencian del Recocido Simulado básico y son:

i) Reducir la probabilidad de aceptación, expresada de la siguiente manera:

$$P = \frac{1}{1+2*exp\left(\frac{\delta}{T}\right)} \quad (19)$$

donde:

P= Probabilidad de aceptación.

δ = Diferencia de soluciones.

T= Temperatura.

Esta probabilidad, como se muestra, se encuentra en un intervalo entre 0 y 1/3, lo que le permite al algoritmo, mientras realiza su proceso de optimización, tener un rango menor de aceptación de peores soluciones.

ii) Para crear un estado vecino (nueva solución) se genera un radio de búsqueda (R_k) que se reduce gradualmente a medida que desciende la temperatura del sistema. El radio de búsqueda viene dado por la siguiente ecuación:

$$R_{k+1} = \beta * R_k \quad (20)$$

donde:

R= Radio de búsqueda.

β = Factor de disminución del radio de búsqueda.

Esto permite que el algoritmo realice una exploración global a temperaturas altas (teniendo como zona inicial de búsqueda todo el espacio factible) y una exploración local (cuando el espacio de búsqueda se ha reducido a una pequeña región que contiene al mejor resultado encontrado hasta el momento) a temperaturas bajas, dando un equilibrio entre la exploración y la explotación del algoritmo.

4.1. EVALUACIÓN DEL DESEMPEÑO DEL ARSA

Para evaluar el desempeño del algoritmo propuesto se utilizaron dos conjunto de funciones (sin y con restricciones) conocidas en la literatura de la optimización combinatoria. En las funciones sin restricciones se realizó una comparación de resultados entre el SA clásico y ARSA, con el fin de evidenciar las bondades de las modificaciones propuestas. Se desarrolló un código en el paquete computacional Matlab® y el equipo utilizado fue un Intel (R) Core(TM) i5-3230M, 2.60 GHz y 8 Gb de RAM.

En cuanto al criterio de parada definido para el ARSA, fueron considerados: i) cuando la temperatura inicial (T_i) establecida fuese menor que una final (T_f) o ii) cuando el valor de la desviación estándar (DF) de los últimos valores óptimos generados fuese menor que una previamente establecida, esto con el fin de disminuir el costo computacional en la última fase del algoritmo en donde se encuentra haciendo su fase de búsqueda local.

4.1.1. Problemas Sin Restricciones

Los problemas sin restricciones seleccionados para evaluar el desempeño de ARSA, se encuentran en [41], donde hay una gran variedad de funciones para validar algoritmos estocásticos de optimización global. En la Tabla 1 se resumen las funciones utilizadas y las dimensiones de cada función.

Tabla 1: Funciones de prueba sin restricciones.

Problema N°	Nombre de las funciones	Valor óptimo	Número de dimensiones
1	Función Dixon Price (DX)	0.000	10
2	Función Ackley (AK)	0.000	50
3	Función Neumaier Trip (NM)	-210.000	10
4	Función Rosenbrock (R)	0.000	50
5	Función Sphere (SP)	0.000	50

Debido a que el algoritmo propuesto y probado es de carácter estocástico, se utiliza una base estadística para demostrar su desempeño. En este trabajo, cada ejecución del algoritmo se realizó 100 veces y se obtuvo el mejor valor de la función (**MF**), el peor valor de la función (**PF**), el promedio de los valores de la función (**MEF**), la desviación estándar de los valores de la función (**DF**) y el tiempo de ejecución promedio (**TP**). Se dice que un método heurístico de optimización es estable y preciso si su desviación estándar es baja, se cataloga que presenta exactitud cuando la diferencia entre la media de las pruebas y el valor óptimo analítico (conocido para la función de prueba) es pequeño, además, el algoritmo se puede describir como robusto si cuando se aplica en diferentes problemas presenta precisión y exactitud.

En todas las pruebas realizadas se utilizaron los siguientes parámetros de ARSA:
 i) Se consideró la temperatura inicial (T_i) como el promedio de los primeros valores obtenidos al evaluar la función objetivo con las primeras variables aleatorias generadas; ii).La temperatura final (T_f) se mantuvo en un rango entre 1×10^{-1} y

1×10^{-10} ; iii) El número de iteraciones (N_i) entre 5 y 4000 veces el número de variables de cada problema; iv) el factor de disminución de temperatura (α_t) se mantuvo en 0.95; v) para el radio inicial de búsqueda R_i se tuvo en cuenta todo el espacio de búsqueda y vi) para el radio final R_f solo se tuvo en cuenta variaciones de 1×10^{-5} unidades en cada variable.

La Tabla 2 presenta una comparación entre el algoritmo propuesto y el SA clásico. Se observa que el ARSA presenta mejoras evidentes respecto a la calidad de la solución. Además, el ARSA y el SA clásico poseen tiempos promedio de computo similares en el proceso de optimización, sin embargo, ARSA alcanza al valor óptimo global estipulado (siendo dirigido a este valor mediante el radio de búsqueda decreciente asignado), mientras que al SA clásico no consiguió llegar a los óptimos globales. Por último la DF arrojada por ARSA en cada uno de los problemas demuestra que es un algoritmo robusto y estable, capaz de lidiar con funciones de alta complejidad.

Tabla 2: Desempeño del SA clásico y el ARSA en funciones sin restricciones.

Algoritmo	F	Valor óptimo	MF	PF	MEF	DF	TP (s)	NI
SA clásico	DX	0.000	100.316	891.382	471.744	1.87E+02	14.86	6840000
	AK	0.000	17.931	19.033	18.706	1.96E-01	2.12	220000
	NM	-210.000	764.811	3558.383	2280.634	6.35E+02	0.06	26400
	R	0.000	569057.754	944584.303	765937.974	8.29E+04	40.69	9000000
	SP	0.000	186.400	249.418	225.719	1.28E+01	0.13	30500
ARSA	DX	0.000	0.000	0.000	0.000	8.94E-08	14.58	6840000
	AK	0.000	0.000	0.000	0.000	2.28E-05	2.06	220000
	NM	-210.000	-210.000	-210.000	-210.000	3.18E-09	0.05	26400
	R	0.000	0.000	3.987	0.439	1.25E+00	50.17	9000000
	SP	0.000	0.000	0.000	0.000	4.04E-08	0.18	30500

4.1.2. Problemas Con Restricciones

En la Tabla 3, se encuentra un resumen de las características de cada función con restricciones utilizada para realizar la evaluación del desempeño del algoritmo [41, 42, 43]. Los problemas están compuestos por funciones de diferentes dimensiones

las cuales pueden ser lineales, cuadráticas o no lineales. Además, estos problemas de optimización contienen restricciones con igualdades y desigualdades, tornando a su vez espacios de búsqueda factibles muy pequeños y de difícil convergencia.

Tabla 3: Funciones de prueba con restricciones.

Función	Valor óptimo	Número de dimensiones	Función	ρ	LI	NI	LE	NE
G1	-15.000	13	Cuadrática	0.00%	9	0	0	0
G2	-0.8036	20	No lineal	100.0%	1	1	0	0
G3	-1.000	10	No lineal	0.00%	0	0	0	1
G4	-30665.539	5	Cuadrática	27.01%	0	6	0	0
G5	5126.498	4	No lineal	0.00%	2	0	0	3
G6	-6961.814	2	No lineal	0.01%	0	2	0	0
G7	24.306	10	Cuadrática	0.00%	3	5	0	0
G8	-0.096	2	No lineal	0.86%	0	2	0	0
G9	680.630	7	No lineal	0.52%	0	4	0	0
G10	7049.331	8	Lineal	0.00%	3	3	0	0
G11	0.750	2	Cuadrática	0.10%	0	0	0	1
G12	-1.000	3	Cuadrática	4.77%	0	9	0	0
G13	0.054	5	No lineal	0.00%	0	0	1	2

LI indica el número de desigualdades lineales; **NI** el número de desigualdades no lineales; **LE** el número de igualdades lineales y **NE** el número de igualdades no lineales. ρ representa una estimación del espacio factible respecto a todo el espacio de búsqueda, y se obtiene generando un millón de soluciones aleatorias y se evalúa que porcentaje de ellas son factibles.

Estos problemas tienen diferentes niveles de complejidad, con características variables, como dimensiones y cantidad de restricciones. En el caso de los problemas G1, G3, G5, G7, G10 y G13, el tamaño del espacio factible es del 0.00%; problemas de elevada (G1, G2, G3, G7), media (G4, G9, G10, G13) y baja (G6, G8, G11, G12) dimensionalidad.

La Tabla 4 presenta los resultados alcanzados con ARSA en los problemas con restricciones. Se observa que el algoritmo llega al valor óptimo en la mayoría de los problemas, además, la desviación estándar (DF) constata la estabilidad de este. También es posible observar que el algoritmo propuesto realiza el proceso de optimización de una manera rápida, esto viéndose reflejado en los tiempos promedio de ejecución (TP). Los problemas con mayor dificultad fueron los G5 y G13, debido a que en estos, el espacio factible es muy pequeño ($\rho=0.00\%$).

Tabla 4: Desempeño del ARSA en funciones con restricciones.

Estadística	FUNCIONES DE PRUEBA					
	G1	G2	G3	G4	G5	G6
MF	-15.000	-0.8036	-1.000	-30665.539	Dificultad para cumplir restricciones	-6961.814
PF	-14.997	-0.7522	-1.000	-30665.539		-6961.806
MEF	-14.999	-0.7938	-1.000	-30665.539		-6961.812
DF	5.47E-04	1.31E-02	2.14E-06	4.86E-06		1.39E-03
TP (s)	17.95	42.57	1.29	1.06		6.54
NI	312000	1010000	86800	25650		1040
G7	G8	G9	G10	G11	G12	G13
24.350	-0.096	680.630	7049.301	0.750	-1.000	Dificultad para cumplir restricciones.
25.395	-0.096	680.634	7068.210	0.750	-0.994	
24.662	-0.096	680.631	7052.978	0.750	-1.000	
2.24E-01	6.43E-06	7.10E-04	3.34E+00	6.59E-05	5.67E-04	
108.60	1.38	68.94	749.60	0.08	26.78	
8600	18600	987000	52000	9600	4000	

Debido al desempeño mostrado por ARSA tanto en los problemas sin restricciones como en los problemas con restricciones, se utilizó para resolver el problema de optimización de armaduras, el cual se muestra en el siguiente capítulo.

5. FORMULACIÓN Y RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN DE ARMADURAS

El principal objetivo en este problema de optimización es minimizar las secciones transversales de los miembros de la estructura A_i con el fin de encontrar el peso mínimo de la estructura satisfaciendo simultáneamente las limitaciones que el problema de optimización impone. Por consiguiente la formulación matemática de este problema de optimización se describe de la siguiente manera:

$$\text{Minimizar: } W(\{x\}) = \sum_{i=1}^n \gamma_i A_i L_i(x) \quad (21)$$

Sujeto a:

$$\delta_{min} \leq \delta_i \leq \delta_{max}, \quad i = 1, 2, \dots, m \quad (22)$$

$$\sigma_{min} \leq \sigma_i \leq \sigma_{max} \quad i = 1, 2, \dots, n \quad (23)$$

$$A_{min} \leq A_i \leq A_{max} \quad i = 1, 2, \dots, n \quad (24)$$

Donde $W(\{x\})$ es el peso de la estructura; n es el número de miembros que componen la estructura; m es el número de nodos; γ_i es la densidad del material del miembro i ; L_i es la longitud del miembro i ; A_i es la sección transversal del miembro i elegida entre A_{min} y A_{max} ; σ_i y δ_i son los esfuerzos en los miembros y deflexiones en los nodos, respectivamente.

Las restricciones se deben formular y adecuar en el proceso de optimización. Estas serán planteadas de tal manera que cada vez que se genere una solución aleatoria (realice el análisis estructural de la armadura con valores de áreas aleatorias), se valide su cumplimiento, si no es así, se deben asignar nuevamente valores aleatorios de áreas para obtener una nueva solución y esta vuelva a ser evaluada.

Teniendo en cuenta lo anterior, el procedimiento completo para dar solución a los problemas de optimización estructural, quedaría descrito con el siguiente pseudocódigo:

Sea $f(s)$ la función a optimizar (minimizar en este caso)

$G(s)$ el espacio de búsqueda

Se adopta una temperatura inicial $T_i > 0$;

Se adopta un factor de reducción de temperatura α .

Se adopta un radio inicial y final de búsqueda R_k .

Se adopta un factor de reducción de radio de búsqueda β .

Se adopta un número de iteraciones del algoritmo (NI)

Se definen las restricciones de áreas, esfuerzos y deformaciones correspondientes;

() - Se definen áreas iniciales aleatorias;*

*- Se realiza el análisis por el **Método de la rigidez directa (Método Matricial)** con las áreas aleatorias generadas en donde se obtienen los valores respectivos de esfuerzos y desplazamientos de cada nodo.*

- Se valida el cumplimiento de las restricciones;

- Se calcula el peso de la estructura (S), tomando $S=S$ (“óptimo inicial”) como la solución óptima inicial $\in G(s)$;

Se adopta criterios de parada

Repetir

Repetir

Se selecciona aleatoriamente otra solución $S' \in G(s)$, (Repetiendo el proceso ());*

Sea $\delta = S' - S$;

Si $\delta < 0$; entonces $S' = S$ (óptimo)

Sino;

Se genera aleatoriamente un valor $P(\text{random})$ $0 \leq P \leq 1$;

Si $P < \frac{1}{1+2\exp(\frac{\delta}{T})}$; entonces $S' = S$ (óptimo)*

Fin Sino;

Hasta que NI= número de iteraciones;

$$T_i = \alpha * T_i$$

$$R_{k+1} = \beta * R_k$$

Hasta que el criterio de parada se cumpla.

Con el proceso anteriormente descrito se procede a realizar el proceso de optimización de las armaduras.

5.1. PROBLEMAS DE OPTIMIZACIÓN DE ARMADURAS

Para evaluar el desempeño de ARSA en problemas de OE de armaduras, se estudiaron 6 problemas reportados en la literatura, divididas en dos conjuntos:

a) Optimización de armaduras con variables continuas:

- Armadura plana de 10 barras.
- Armadura plana de 17 barras.
- Armadura plana de 18 barras.

b) Optimización de armaduras con variables discretas:

- Armadura plana de 10 barras.
- Armadura plana de 15 barras.
- Armadura plana de 52 barras.

6. RESULTADOS Y DISCUSIÓN

Se realizaron 100 ejecuciones independientes del algoritmo para cada problema. Los mejores diseños, peso de la estructura, peso promedio y desviación estándar (DF) logrados con ARSA son comparados con otras técnicas de optimización y son reportados en las tablas. La implementación del algoritmo fue realizada en Matlab®, bajo las siguiente características computacionales: Intel (R) Core(TM) i5-3230M, 2.60 GHz y 8 Gb de RAM.

En todas las pruebas realizadas en los problemas de OE se utilizaron los siguientes parámetros para el algoritmo ARSA: i) Se consideró la temperatura inicial (T_i) como el promedio de los primeros valores obtenidos al evaluar la función objetivo con las primeras variables aleatorias generadas; ii).La temperatura final (T_f) se mantuvo en un rango entre 1×10^{-2} y 1×10^{-5} ; iii) El número de iteraciones (N_i) entre 3 y 70 veces el número de variables de cada problema; iv) el factor de disminución de temperatura (α_t) se mantuvo en 0.9; v) para el radio inicial de búsqueda R_i se tuvo en cuenta todo el espacio de búsqueda y vi) para el radio final R_f se tuvo en cuenta variaciones entre 1×10^{-3} y 1×10^{-5} unidades en cada variable.

6.1. OPTIMIZACIÓN DE ARMADURAS CON VARIABLES CONTINUAS

6.1.1. Armadura de 10 Barras

Esta estructura, tipológicamente formada como se muestra en la Figura 7, cuenta con una densidad 0.1 lb/in^3 y con un módulo de elasticidad de 10000 ksi correspondiente a cada uno de los elementos de la armadura. El área de cada elemento debe estar comprendida entre 0.10 in^2 y 35 in^2 , así mismo los valores límites de esfuerzos y desplazamientos permitidos, son de $\pm 25 \text{ ksi}$ (para miembros sometidos a tensión y compresión) y 2 in (en cada dirección), respectivamente.

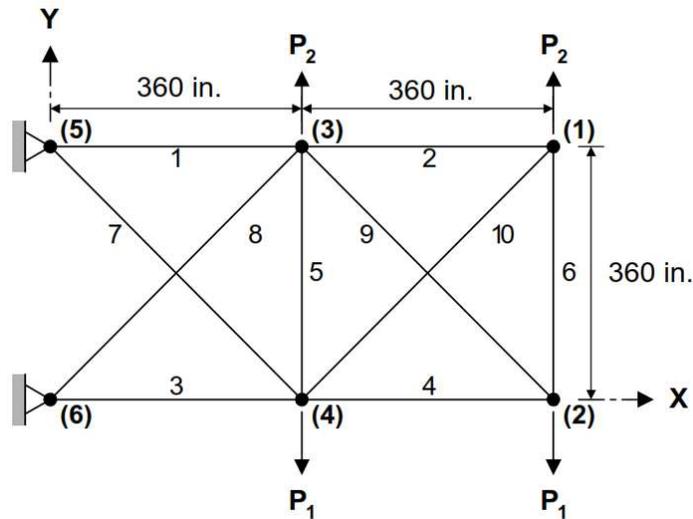


Figura 7: Armadura plana compuesta por 10 barras.

En el análisis realizado para esta armadura se tuvieron en cuenta dos casos de carga: Caso 1, $P_1=100$ kips y $P_2=0$ kips; y Caso 2, $P_1=150$ kips y $P_2=50$ kips.

Los valores de peso óptimo son mostrados junto a la desviación estándar y los valores de áreas, en la Tabla 5 y Tabla 6, donde se compara con los resultados obtenidos con los reportados por diversos autores. El mejor valor de peso obtenido por el ARSA fue de 5060.900 lb (para el caso 1) y 4677.036 lb (para el caso 2).

En la Tabla 5, el mejor valor de peso encontrado para la cercha de 10 barras en el caso de carga 1 es de 5060.900 lb, superando valores en hasta un 9.26 % a los encontrados por otros investigadores como Dizangian [44], Li [45], Kaveh [46], y Degertekin [47].

Otros autores como Sonmez [14], Baghlani [48, 49], Asl [32], y Charalampakis [50], encontraron resultados sutilmente inferiores al encontrado con el ARSA, mostrando solo variaciones en décimas de las variables (áreas), lo cual indica la competitividad del algoritmo. Los resultados reportados por Baghlani [48, 51], Lee [19], Kaveh [52] y Perez [53], fueron valores levemente mejores al encontrado con el algoritmo propuesto, pero esto en gran medida se debe a que en sus algoritmos

proponen violaciones a las restricciones planteadas, situación que el ARSA no permite en ningún momento en su proceso de optimización.

En la comparación realizada en la Tabla 6, el mejor valor de peso encontrado para la cercha de 10 barras en el caso de carga 2 es de 4677.036 lb, superando valores encontrados por investigadores como Sonmez [14], Baghlani [48, 51], Dizangian [44], Li [45], Kaveh [46] y Degertekin [47]. Autores como Asl [32], Baghlani [49], y Charalampakis [50], encontraron estructuras ligeramente livianas a las obtenidas con ARSA, mostrando solo variaciones en décimas de las variables (áreas).

Los resultados obtenidos por Baghlani [48, 51] con las metodologías Acelerated Firefly Algorithm (AFA) [51] y el Firefly Algorithm (FA) [48], Lee [19] y Kaveh [52] han encontrado valores levemente mejores al encontrado con el ARSA, esto, al igual que en el primer caso de carga, se debe a que en sus algoritmos proponen violaciones a las restricciones planteadas, situación que el algoritmo propuesto no considera.

Tabla 5: Comparación de resultados con otros autores para la armadura de 10 elementos (Caso 1).

Método	Peso (lb)	DF	Variables de diseño (in ²)									
			A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀
MS [44]	5064.40	--	31.17	0.10	23.10	14.72	0.10	0.41	7.57	21.16	21.42	0.10
FBSFA [48]	5060.86	--	30.55	0.10	23.21	15.23	0.10	0.55	7.46	21.03	21.50	0.10
FA [48]	5060.14	--	30.97	0.10	23.22	15.04	0.10	0.59	7.45	20.87	21.46	0.10
ABC [14]	5060.88	--	30.55	0.10	23.18	15.22	0.10	0.55	7.46	21.06	21.50	0.10
PSO [45]	5529.50	--	33.47	0.11	23.18	15.48	3.65	0.12	8.33	23.34	23.01	0.19
PSOPC [45]	5061.00	--	30.57	0.10	22.97	15.15	0.10	0.55	7.49	21.16	21.56	0.10
HPSO [45]	5060.92	--	30.70	0.10	23.17	15.18	0.10	0.55	7.46	20.98	21.51	0.10
HS [19]	5057.88	--	30.15	0.10	22.71	15.27	0.10	0.54	7.54	21.56	21.45	0.10
HPSACO [52]	5056.56	--	30.31	0.10	23.43	15.51	0.10	0.52	7.44	21.08	21.23	0.10
MCSS [46]	5086.90	--	29.58	0.11	23.81	15.89	0.11	0.10	8.60	21.68	20.30	0.11
IMCSS [46]	5064.60	--	30.03	0.10	23.63	15.97	0.10	0.52	7.46	21.44	20.74	0.10
FA [51]	5060.14	9.43	30.97	0.10	23.22	15.04	0.10	0.59	7.45	20.87	21.46	0.10
FA-R [51]	5060.07	7.67	30.37	0.10	23.77	15.05	0.10	0.62	7.39	20.74	21.65	0.10
AFA [51]	5059.22	2.96	30.30	0.10	23.20	15.21	0.10	0.54	7.44	20.98	21.74	0.10
PSOST [49]	5060.86	--	30.53	0.10	23.21	15.22	0.10	0.56	7.45	21.02	21.54	0.10
DE3 [50]	5060.86	0.01	30.53	0.10	23.20	15.23	0.10	0.55	7.46	21.05	21.51	0.10
H-SAGA [32]	5058.66	--	30.51	0.10	23.20	15.19	0.10	0.56	7.46	21.07	21.47	0.10
PSO [53]	5024.21	--	33.50	0.10	22.77	14.47	0.10	0.10	7.53	20.47	20.39	0.10
EHS [47]	5062.39	--	30.21	0.10	22.70	15.28	0.10	0.53	7.56	21.56	21.49	0.10
SAHS. [47]	5061.42	--	30.39	0.10	23.10	15.49	0.10	0.53	7.49	21.19	21.34	0.10
ARSA	5060.90	0.72	30.62	0.10	23.23	15.11	0.10	0.56	7.46	21.04	21.51	0.10

Tabla 6: Comparación de resultados con otros autores para la armadura de 10 elementos (Caso 2).

Método	Variables de diseño (in ²)											
	Peso (lb)	DF	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀
MS [44]	4682.48	--	25.00	0.10	25.00	14.43	0.10	1.99	12.43	12.36	19.98	0.10
FBSFA [48]	4677.08	--	23.48	0.10	25.30	14.29	0.10	1.97	12.41	12.91	20.31	0.10
FA [48]	4673.70	--	23.33	0.10	25.38	14.62	0.10	1.98	12.36	12.81	20.21	0.10
ABC [14]	4677.08	--	23.47	0.10	25.24	14.35	0.10	1.97	12.41	12.89	20.33	0.10
PSO [45]	4679.47	--	22.94	0.11	25.36	14.37	0.10	1.99	12.35	12.92	20.67	0.10
PSOPC [45]	4677.70	--	23.74	0.10	25.29	14.41	0.10	1.97	12.36	12.69	20.32	0.10
HPSO [45]	4677.29	--	23.35	0.10	25.50	14.25	0.10	1.97	12.36	12.89	20.36	0.10
HS [19]	4668.81	--	23.25	0.10	25.73	14.51	0.10	1.98	12.21	12.61	20.36	0.10
HPSACO [52]	4675.78	--	23.19	0.10	24.59	14.22	0.10	1.97	12.49	12.93	20.95	0.10
MCSS [46]	4686.47	--	22.86	0.12	25.72	15.31	0.10	1.97	12.31	12.93	19.91	0.10
IMCSS [46]	4679.15	--	23.30	0.10	25.68	14.51	0.10	1.97	12.15	12.36	20.87	0.10
FA [51]	4678.64	83.31	23.89	0.10	25.60	14.23	0.10	1.99	12.38	12.77	20.04	0.10
FA-R [51]	4678.09	37.60	23.70	0.10	25.37	14.29	0.10	2.00	12.45	12.84	20.14	0.10
AFA [51]	4677.01	4.57	23.71	0.10	25.35	14.27	0.10	1.97	12.35	12.69	20.41	0.10
PSOST [49]	4676.96	--	23.54	0.10	25.18	14.31	0.10	1.97	12.42	12.86	20.39	0.10
DE3 [50]	4676.93	0.01	23.52	0.10	25.29	14.39	0.10	1.97	12.39	12.83	20.33	0.10
H-SAGA [32]	4675.43	--	23.32	0.10	25.58	14.66	0.10	1.97	12.27	12.65	20.34	0.10
EHS [47]	4679.02	--	23.59	0.10	25.42	14.49	0.10	1.98	12.36	12.68	20.32	0.10
SAHS. [47]	4678.84	--	23.53	0.10	25.43	14.49	0.10	1.99	12.35	12.70	20.34	0.10
ARSA	4677.04	1.46	23.45	0.10	25.32	14.24	0.10	1.97	12.38	12.85	20.44	0.10

La Figura 8 y Figura 9 muestran las gráficas de convergencia para el caso 1 y caso 2, respectivamente. Se observa como ARSA converge al peso óptimo conforme disminuye la temperatura del sistema. Los pesos de estructuras finales obtenidos fueron alrededor de 1.26 veces (caso 1) y 1.45 veces (caso 2) más liviano respecto al primer diseño factible generado.

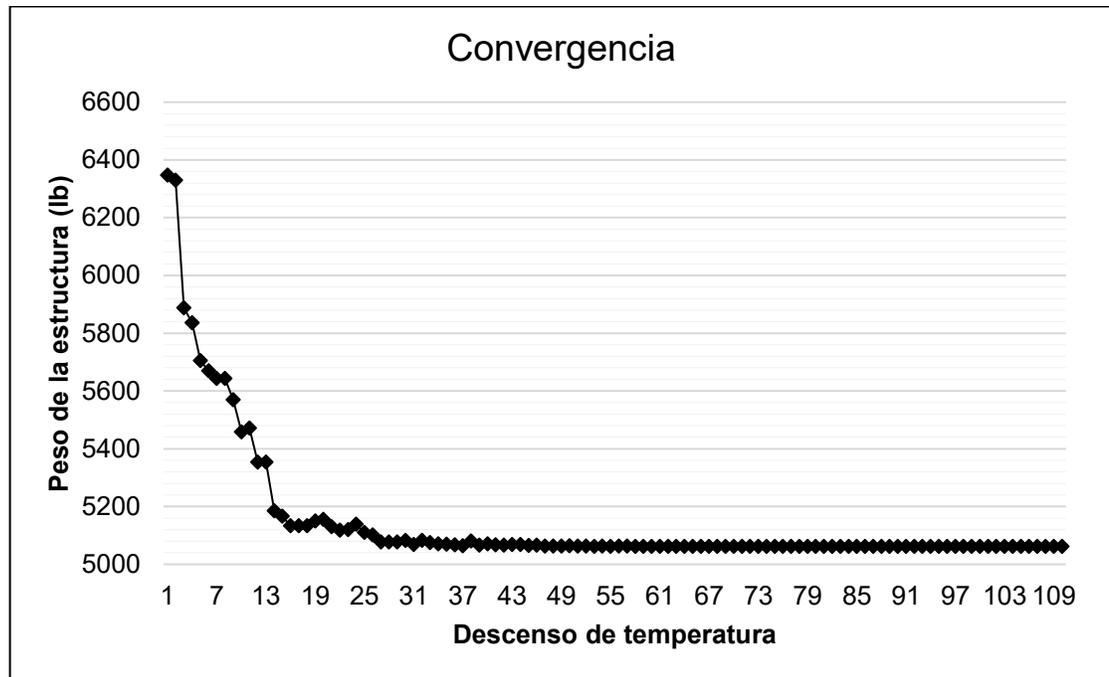


Figura 8: Gráfica de convergencia para la armadura de 10 barras (caso 1).

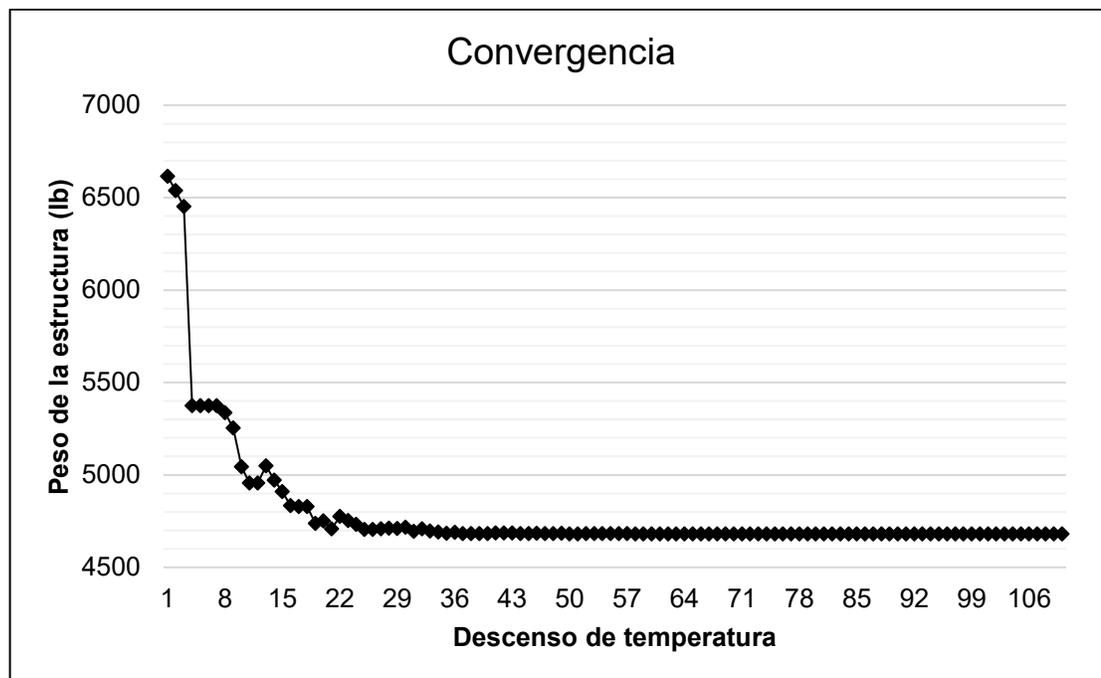


Figura 9: Gráfica de convergencia para la armadura de 10 barras (caso 2).

En la Figura 10 y Figura 11 es posible observar el peso encontrado por el algoritmo propuesto en cada una de las ejecuciones realizadas. En las 100 ejecuciones el algoritmo obtuvo una desviación estándar de 0.717 lb y un valor de peso promedio de 5061.838 lb para el caso 1; y con una desviación estándar de 1.46 lb y un valor de peso promedio de 4678.890 lb para el caso 2.

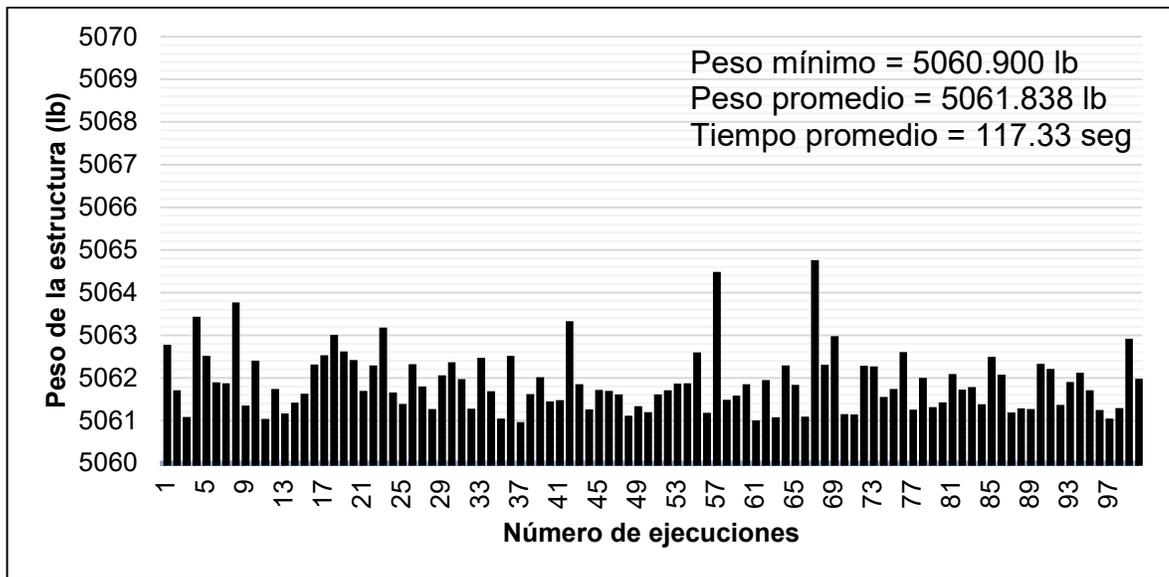


Figura 10: Peso encontrado en cada corrida del algoritmo caso 1.

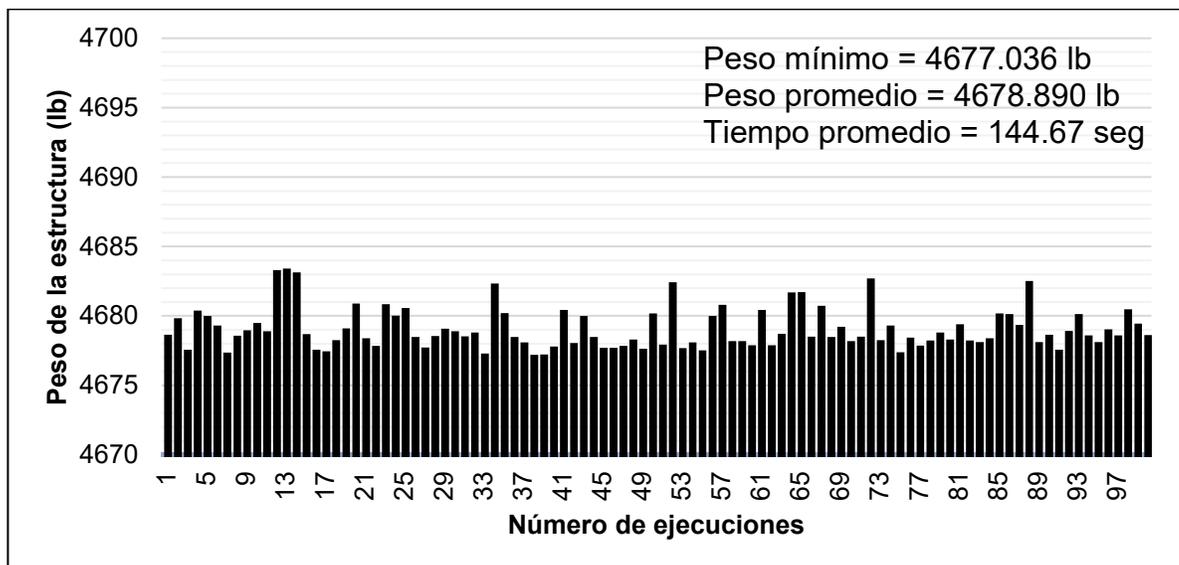


Figura 11: Peso encontrado en cada corrida del algoritmo caso 2.

6.1.2. Armadura de 17 Barras

La Figura 12 muestra la configuración de la armadura de 17 barras. Cuenta con una densidad de 0.268 lb/in^3 y con un módulo de elasticidad de 30000 ksi correspondiente a cada uno de los elementos de la cercha. El área transversal mínima requerida para cada elemento debe ser 0.10 in^2 , así mismo los valores

límites en esta cercha, de esfuerzos y desplazamientos, son de ± 50 ksi (para miembros sometidos a tensión y compresión) y 2 in (en cada dirección), respectivamente. La carga con la cual fue analizada la estructura corresponde a 100 kips colocada en el nodo 9.

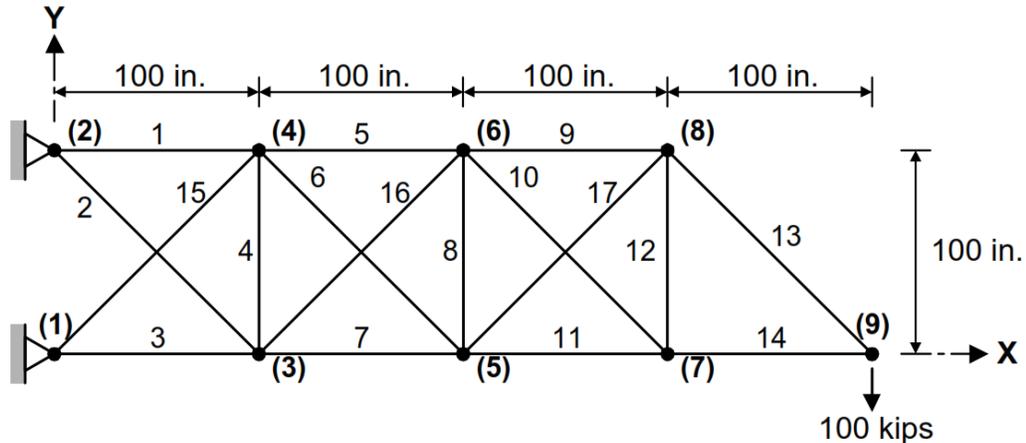


Figura 12: Armadura plana compuesta por 17 barras.

El mejor valor de peso de la armadura de 17 barras encontrado en el proceso de optimización fue de 2582.002 lb (ver Tabla 7) junto con la desviación estándar y los valores de las respectivas áreas. El mejor valor de peso obtenido con ARSA, es menor en hasta un 5.51 % que los encontrados por investigadores como Li [45] con los algoritmos Particle Swarm Optimizer (PSO) y Particle Swarm Optimizer with Passive Congregation (PSOPC), Adeli [54] y Miguel [55].

Los resultados de investigadores como Baghlani [48] con la metodología del Firefly Algorithm (FA), y Lee [19] han encontrado valores mejores al encontrado con el ARSA, esto en gran parte se debe, al igual que en problemas anteriores, a que en sus algoritmos proponen violaciones a las restricciones planteadas, escenario que el ARSA no permite en su proceso de optimización.

Baghlani [48] con la técnica híbrida Feasible Boundary Search with the Firefly Algorithm (FBSFA), Asl [32], Li [45] con el Heuristic Particle Swarm Optimization (HPSO); y Charalampakis [50], han encontrado resultados que son levemente inferiores al encontrado con el ARSA presentando mucha similitud, tanto en su

peso total, como en las variables (áreas) encontradas. Por último, la desviación estándar (0.677 lb) arrojada por ARSA es suficientemente baja para aprobar el comportamiento del algoritmo propuesto como estable y eficiente.

El proceso de optimización seguido por el algoritmo planteado en el presente problema es el mostrado en la Figura 13, donde se evidencia la convergencia del algoritmo con la obtención de mejores resultados a medida que disminuye la temperatura del ARSA, consiguiéndose una estructura alrededor de 1.51 veces más liviana respecto al primer diseño factible generado. La convergencia al valor óptimo encontrado se logra en el descenso de temperatura número 57.

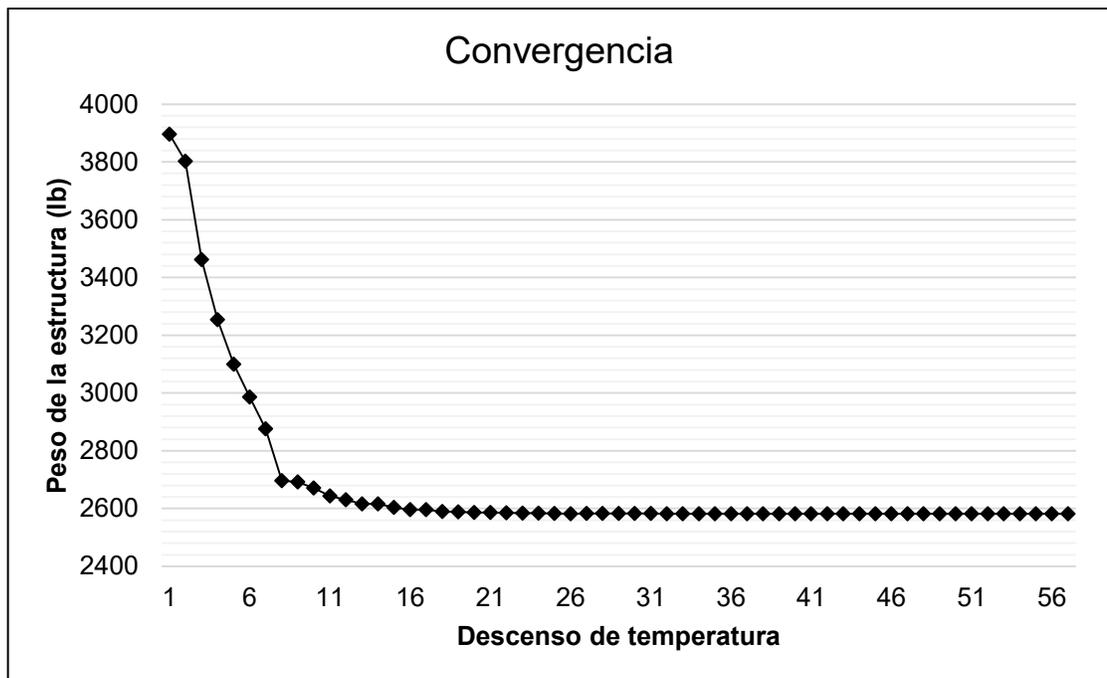


Figura 13: Gráfica de convergencia para la armadura de 17 barras.

Tabla 7: Comparación de resultados con otros autores para la armadura de 17 elementos.

Método	Peso (lb)	DF	Variables de diseño (in ²)																
			A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅	A ₁₆	A ₁₇
FBSFA [48]	2581.90	--	15.90	0.10	12.10	0.10	8.07	5.58	11.94	0.10	7.94	0.10	4.04	0.10	5.67	4.02	5.54	0.10	5.57
FA [48]	2577.57	--	15.94	0.10	12.02	0.10	8.04	5.52	11.78	0.10	7.92	0.10	4.08	0.10	5.63	4.03	5.60	0.10	5.62
H-SAGA [32]	2578.76	--	15.82	0.11	12.02	0.10	8.11	5.53	11.84	0.10	7.96	0.10	4.07	0.10	5.68	4.01	5.58	0.10	5.58
HS [19]	2580.81	--	15.82	0.11	12.00	0.10	8.15	5.51	11.83	0.10	7.93	0.10	4.09	0.10	5.66	4.06	5.66	0.10	5.58
PSO [45]	2724.37	--	15.77	2.26	13.85	0.11	11.36	3.92	8.07	0.10	5.85	2.29	6.31	3.38	5.43	3.92	3.53	2.31	3.54
PSOPC [45]	2582.85	--	15.98	0.10	12.14	0.10	8.10	5.57	11.73	0.10	7.98	0.11	4.07	0.13	5.67	3.99	5.56	0.10	5.56
HPSO [45]	2581.94	--	15.90	0.10	12.09	0.10	8.06	5.59	11.92	0.10	7.97	0.10	4.08	0.10	5.67	4.00	5.55	0.10	5.54
GA [54]	2594.42	--	16.03	0.11	12.18	0.11	8.42	5.72	11.33	0.11	7.30	0.12	4.05	0.10	5.61	4.05	5.15	0.11	5.29
DE3 [50]	2581.90	--	15.91	0.10	12.05	0.10	8.08	5.55	11.96	0.10	7.96	0.10	4.06	0.10	5.65	4.01	5.57	0.10	5.57
HS [55]	2586.51	--	16.19	0.12	12.23	0.11	8.00	5.27	11.76	0.11	7.73	0.15	4.30	0.10	5.73	3.87	5.57	0.22	5.66
FA [55]	2582.65	--	15.80	0.10	11.85	0.10	8.25	5.46	11.86	0.10	7.92	0.10	4.01	0.10	5.78	4.17	5.65	0.10	5.59
ARSA	2582.00	0.68	15.92	0.11	12.20	0.10	8.02	5.52	11.89	0.10	7.99	0.10	3.99	0.10	5.65	4.04	5.56	0.10	5.59

La Figura 14, muestra los pesos alcanzados con ARSA en cada una de las ejecuciones realizadas, con un valor de peso promedio de 2582.935 lb y desviación estándar de 0.677 lb.

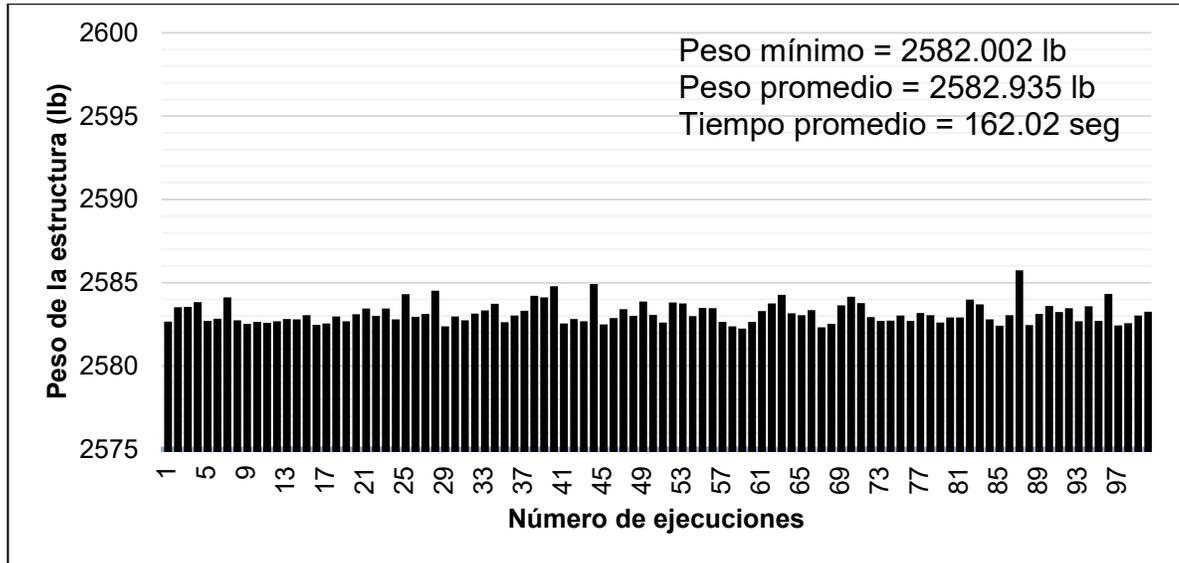


Figura 14: Peso encontrado en cada ejecución del algoritmo para la armadura de 17 barras.

6.1.3. Armadura de 18 Barras

La configuración de esta estructura se muestra en la Figura 15. Los elementos tienen una densidad 0.1 lb/in³ y con un módulo de elasticidad de 10000 ksi. El área de cada elemento debe estar comprendida entre 0.10 in² y 50 in², así mismo los valores límites en esta cercha, de esfuerzos son de ± 20 ksi (para miembros sometidos a tensión y compresión) y sin restricciones a desplazamientos. Adicionalmente, los elementos sometidos a compresión, son a su vez restringidos por el esfuerzo de pandeo, esto mediante la fórmula de Euler, mostrada a continuación:

$$\sigma_i = -\frac{K * E * A_i}{L_i^2};$$

Donde K, es una constante determinada por la forma y en este caso es tomada como 4. Los valores de A_i, E y L_i, corresponden al área, módulo de elasticidad y longitud de cada elemento.

Para el análisis de esta armadura se proporcionaron cargas en los nodos superiores de armadura (1, 2, 4, 6, 8), correspondientes a $P=20$ kips. El número de variables independientes (área) se reduce a cuatro grupos de elementos, agrupados de la siguiente manera:

Grupo 1 (G1): Elementos 1, 4, 8, 12 y 16.

Grupo 2 (G2): Elementos 2, 6, 10, 14 y 18.

Grupo 3 (G3): Elementos 3, 7, 11 y 15.

Grupo 4 (G4): Elementos 5, 9, 13 y 17.

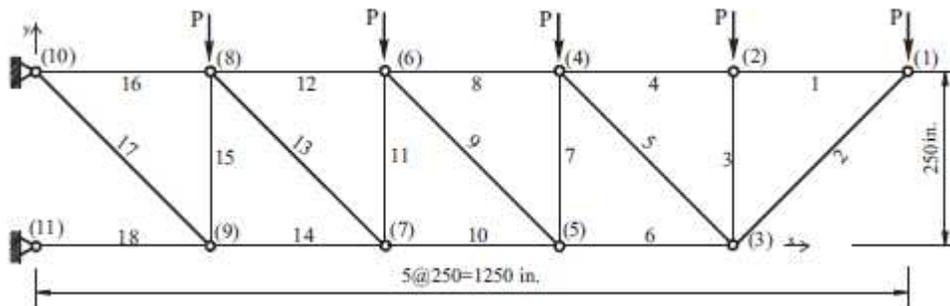


Figura 15: Armadura plana compuesta por 18 barras.

La Tabla 8 presenta una comparación del ARSA con los encontrados con otras técnicas metaheurísticas. El mejor valor de peso de armadura obtenido con ARSA fue de 6430.531 lb.

Tabla 8: Comparación de resultados con otros autores para la armadura de 18 elementos.

Método	Peso (lb)	DF	Variables de diseño (in ²)			
			G1	G2	G3	G4
H-SAGA [32]	6419.23	--	9.97	21.60	12.45	7.05
HS [19]	6421.88	--	9.98	21.63	12.49	7.06
ABC [14]	6430.53	--	10.00	21.65	12.50	7.07
MM [56]	6430.00	--	10.00	21.65	12.50	7.07
ARSA	6430.53	0.00	10.00	21.65	12.50	7.07

Los resultados reportados por Lee [19] y Asl *et al.* [32] permiten la violación de algunas restricciones. Sonmez [14], e Imai [56] han encontrado resultados que son iguales o levemente inferiores al encontrado con el ARSA presentando mucha

similitud, tanto en su peso total, como en las variables (áreas) encontradas, demostrando de esta manera que el algoritmo desarrollado, al igual que en los problemas anteriores, es contundente en llegar al valor óptimo global de peso de la estructura.

La Figura 16 muestra la gráfica de convergencia de ARSA para este problema, consiguiendo una estructura alrededor de 1.74 veces más liviana respecto al primer diseño factible generado. La convergencia al mejor valor óptimo encontrado se logra en el descenso de temperatura número 44.

La Figura 17, muestra los pesos alcanzados con ARSA en cada una de las ejecuciones realizadas, con un valor de peso promedio de 6430.538 lb y desviación estándar de 0.00434 lb. Estos resultados demuestran la estabilidad y precisión del algoritmo.

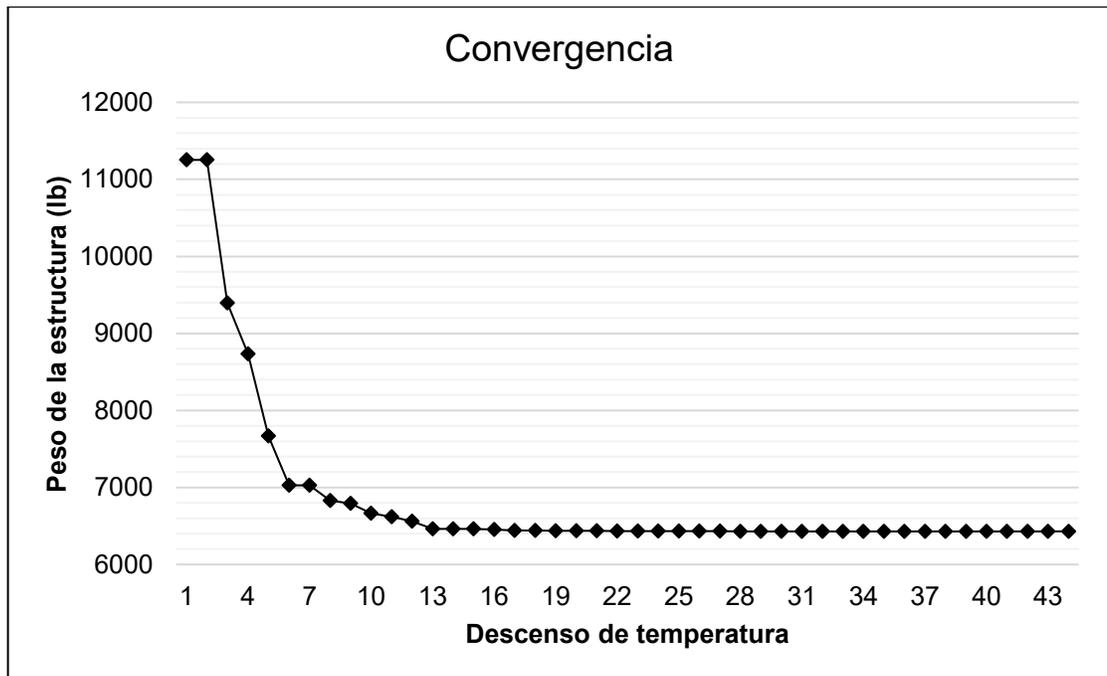


Figura 16: Gráfica de convergencia para la armadura de 18 barras.

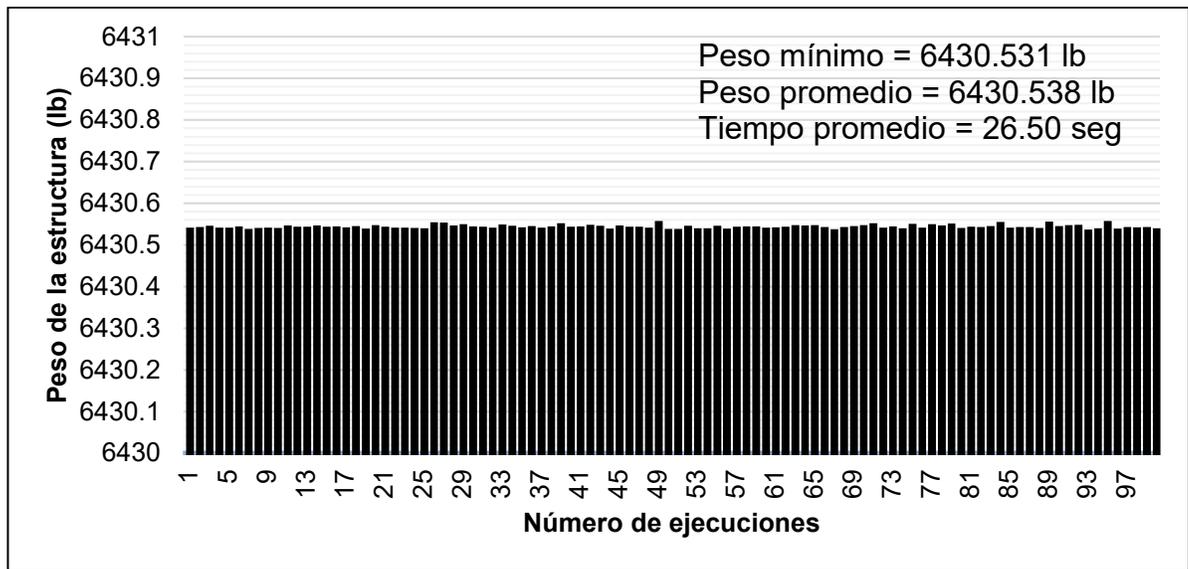


Figura 17: Peso encontrado en cada ejecución del algoritmo para la armadura de 18 barras.

6.2. OPTIMIZACIÓN DE ARMADURAS CON VARIABLES DISCRETAS

6.2.1. Armadura de 10 Barras

Esta armadura tiene la misma configuración mostrada en la Figura 7, pero con cargas de $P_1 = 100$ kips y $P_2 = 0$. Además, las variables (discretas) son escogidas del siguiente conjunto: $A_p\{1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, 33.50\}$ (in²).

La Tabla 9 muestra los resultados encontrados con ARSA junto con los reportados por otros autores utilizando diferentes técnicas. El mejor valor de peso de la cercha encontrado por ARSA fue de 5490.738 lb, siendo 1.87 % más liviano que los pesos reportados por Ho-Huu [17], Kripka [26], Eskandar [57], Baghlani [58], Camp [59, 60, 61], Li [62], Sadollah [63], Sonmez [64], Dede [65], y Barbosa [66], evidenciando la eficacia del algoritmo.

Tabla 9: Comparación de resultados con otros autores para la armadura de 10 elementos.

Método	Peso (lb)	DF	Variables de diseño (in ²)									
			A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀
DE [17]	5490.74	19.52	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62
aeDE [17]	5490.74	20.78	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62
ICACO [57]	5490.74	0.00	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62
ICA [57]	5491.72	0.00	33.50	1.62	22.90	15.50	1.62	1.62	7.97	22.00	22.00	1.62
FA [58]	5498.20	--	33.50	1.80	22.90	15.50	1.62	1.62	7.97	22.00	22.00	1.62
AFA [58]	5479.94	--	33.50	1.62	22.90	13.90	1.62	1.62	7.97	22.90	22.00	1.62
TLBO [59]	5490.74	20.33	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62
TLBO [65]	5490.74	--	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62
PSO [62]	5581.76	664.08	30.00	1.62	30.00	13.50	1.62	1.80	11.50	18.80	22.00	1.80
PSOPC [62]	5593.44	12.84	30.00	1.80	26.50	15.50	1.62	1.62	11.50	18.80	22.00	3.09
HPSO [62]	5531.98	3.84	30.00	1.62	22.90	13.50	1.62	1.62	7.97	26.50	22.00	1.80
MBA [63]	5507.75	11.38	30.00	1.62	22.90	16.90	1.62	1.62	7.97	22.90	22.90	1.62
SA [26]	5490.74	--	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62
GA [66]	5490.74	--	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62
ACO [60]	5490.74	23.19	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62
BB-BC [61]	5490.74	12.42	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62
ABC [64]	5490.74	21.51	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62
ARSA	5490.74	3.75	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62

El proceso de optimización seguido por el ARSA, en el presente problema es el mostrado en la Figura 18, donde se evidencia la convergencia del algoritmo con la obtención de mejores resultados a medida que disminuye la temperatura, consiguiendo una estructura alrededor de 1.18 veces más liviana respecto al primer diseño factible generado. La convergencia al mejor valor óptimo encontrado se logra en el descenso de temperatura número 53.

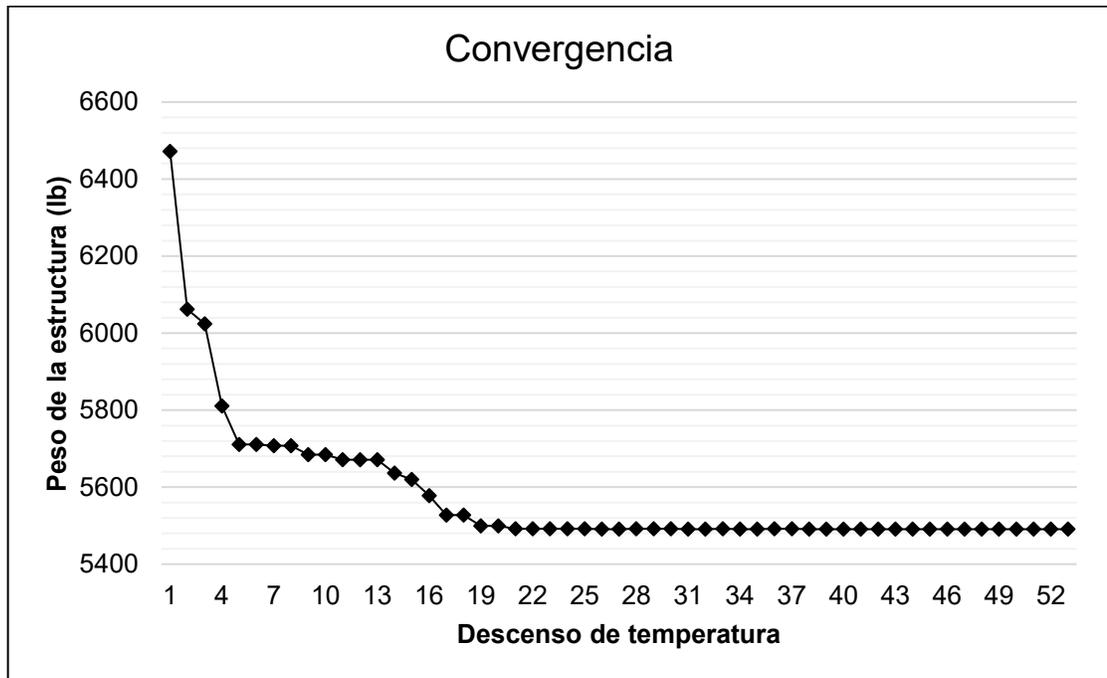


Figura 18: Gráfica de convergencia para la armadura de 10 barras.

En la Figura 19 se observa el valor óptimo de peso de la estructura obtenida con el ARSA en cada una de las ejecuciones. El valor del promedio de las ejecuciones fue 5491.478 lb con una desviación estándar de 3.752 lb.

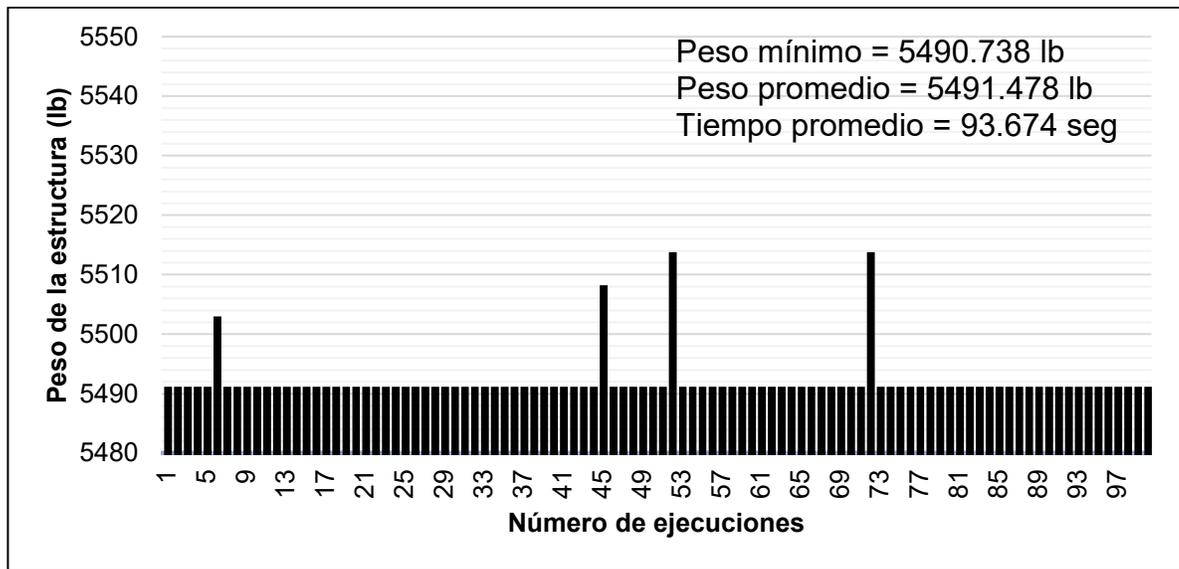


Figura 19: Peso encontrado en cada ejecución del algoritmo para la armadura de 10 barras.

6.2.2. Armadura de 15 Barras

Esta estructura, tipológicamente formada como se muestra en la Figura 20, cuenta con una densidad de $7.8 \times 10^{-6} \text{ kg/mm}^3$ y con un módulo de elasticidad de $2 \times 10^5 \text{ MPa}$ en cada uno de los elementos.

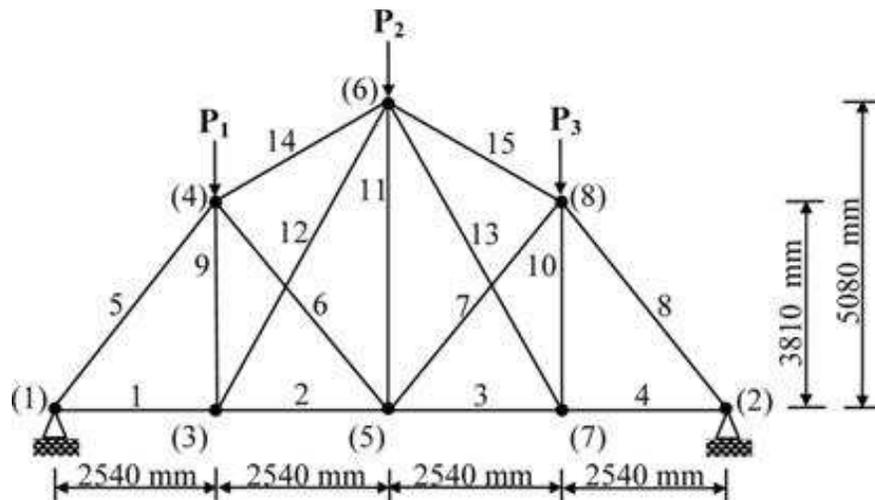


Figura 20: Armadura plana compuesta por 15 barras.

Los valores límites en esta cercha, de esfuerzos y desplazamientos, son de $\pm 120 \text{ MPa}$ (para miembros sometidos a tensión y compresión) y 10 mm (en cada

dirección), respectivamente. Las variables discretas (áreas), son seleccionadas del siguiente conjunto de datos: $A_p\{113.2, 143.2, 145.9, 174.9, 185.9, 235.9, 265.9, 297.1, 308.6, 334.3, 338.2, 497.8, 507.6, 736.7, 791.2, 1063.7\}$ (mm^2). Las cargas con las que fue analizada la estructura corresponden a 35 kN ($P_1=P_2=P_3=35$ kN) ubicadas en los nodos de la cuerda superior (4, 6, 8).

El mejor valor de peso encontrado en el proceso de optimización fue de 105.735 kg, mostrado en la Tabla 10. Este valor iguala y supera en hasta un 34.41 % a los encontrados por autores como Eskandar [57], Li [62], Sadollah [63], Dede [65], Zhang [67] y Cheng [68].

El proceso de obtención de mejores resultados se evidencia en la Figura 21, consiguiendo estructura alrededor de 1.42 veces más liviana respecto al primer diseño factible generado. Además, se puede apreciar que el algoritmo acepta ocasionalmente valores peores (descenso de temperatura número 7), esto con el fin de evitar quedar atrapado en óptimos locales. La convergencia al valor óptimo encontrado se logra en el descenso de temperatura número 49.

En la Figura 22 se observa el valor óptimo de peso de la estructura obtenida con el ARSA en cada una de las ejecuciones. El valor del promedio de las ejecuciones fue 105.737 kg con una desviación estándar de 0.012 kg.

Tabla 10: Comparación de resultados con otros autores para la armadura de 15 elementos.

Método	Peso (kg)	DF	Variables de diseño (mm ²)														
			A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅
ICACO [57]	105.74	0.00	113.2	113.2	113.2	113.2	736.7	113.2	113.2	736.7	113.2	113.2	113.2	113.2	113.2	334.3	334.3
ICA [57]	105.74	0.00	113.2	113.2	113.2	113.2	736.7	113.2	113.2	736.7	113.2	113.2	113.2	113.2	113.2	334.3	334.3
PSO [62]	108.84	--	185.9	113.2	143.2	113.2	736.7	143.2	113.2	736.7	113.2	113.2	113.2	113.2	113.2	334.3	334.3
PSOPC [62]	108.96	--	113.2	113.2	113.2	113.2	736.7	113.2	113.2	736.7	113.2	113.2	113.2	113.2	185.9	334.3	334.3
HPSO [62]	105.74	--	113.2	113.2	113.2	113.2	736.7	113.2	113.2	736.7	113.2	113.2	113.2	113.2	113.2	334.3	334.3
TLBO [65]	105.74	--	113.2	113.2	113.2	113.2	736.7	113.2	113.2	736.7	113.2	113.2	113.2	113.2	113.2	334.3	334.3
MBA [63]	105.74	0.00	113.2	113.2	113.2	113.2	736.7	113.2	113.2	736.7	113.2	113.2	113.2	113.2	113.2	334.3	334.3
HGA [67]	142.12	--	308.6	174.9	338.2	143.2	736.7	185.9	265.9	507.6	143.2	507.6	279.1	174.9	297.1	235.9	265.9
HHS [68]	105.74	1.10	113.2	113.2	113.2	113.2	736.7	113.2	113.2	736.7	113.2	113.2	113.2	113.2	113.2	334.3	334.3
ARSA	105.74	0.01	113.2	113.2	113.2	113.2	736.7	113.2	113.2	736.7	113.2	113.2	113.2	113.2	113.2	334.3	334.3

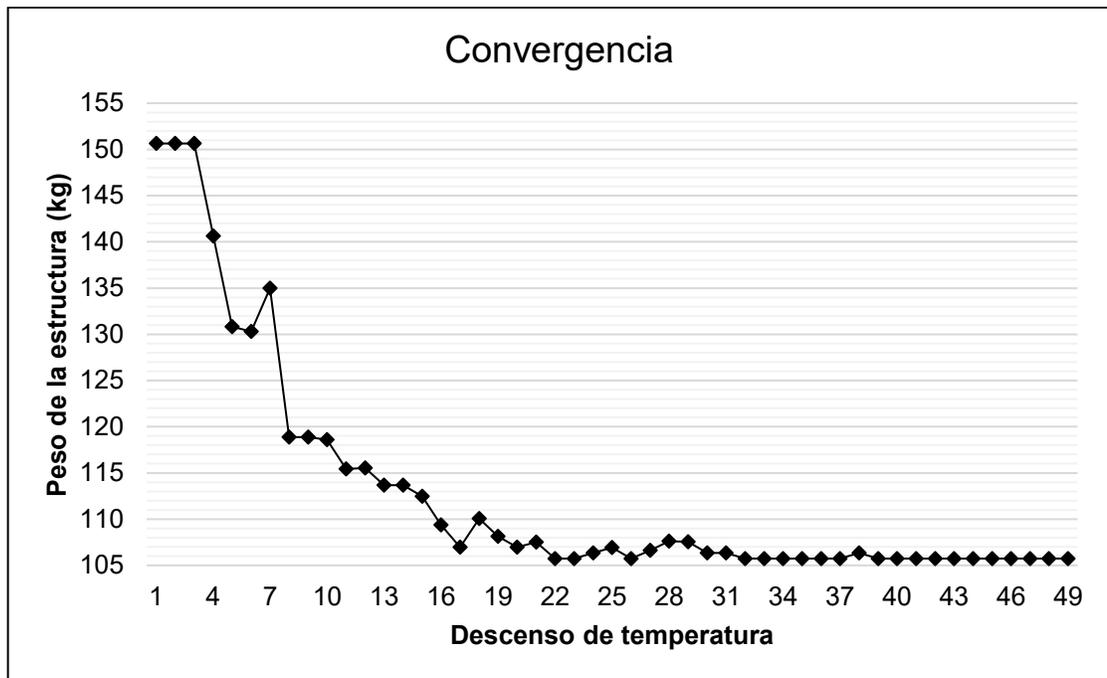


Figura 21: Gráfica de convergencia para la armadura de 15 barras.

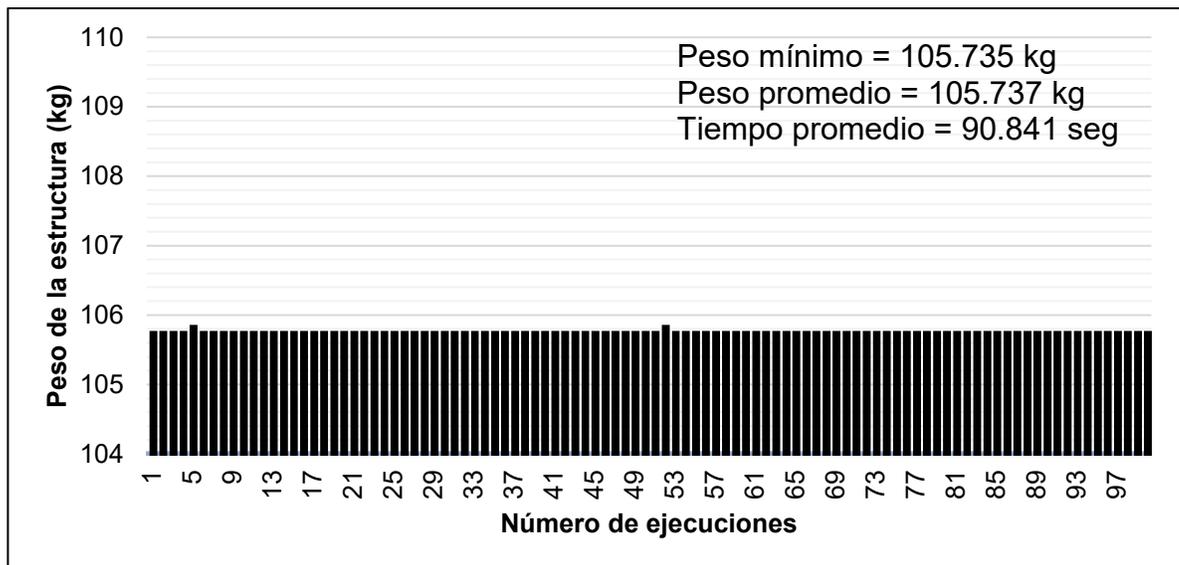


Figura 22: Peso encontrado en cada ejecución del algoritmo para la armadura de 15 barras.

6.2.3. Armadura de 52 Barras

Esta estructura, tipológicamente formada como se muestra en la Figura 23, cuenta con una densidad de 7860 kg/m^3 y con un módulo de elasticidad de 207000 MPa

correspondiente a cada uno de los elementos de la cercha. Los valores límites en esta cercha, de esfuerzos son de ± 180 MPa (para miembros sometidos a tensión y compresión) y sin restricción a desplazamientos. Para el análisis de esta armadura se proporcionaron cargas en los nodos superiores (17, 18, 19, 20), correspondientes a 100 kN en la dirección X y 200 kN en la dirección Y ($P_x = 100$ kN y $P_y = 200$ kN).

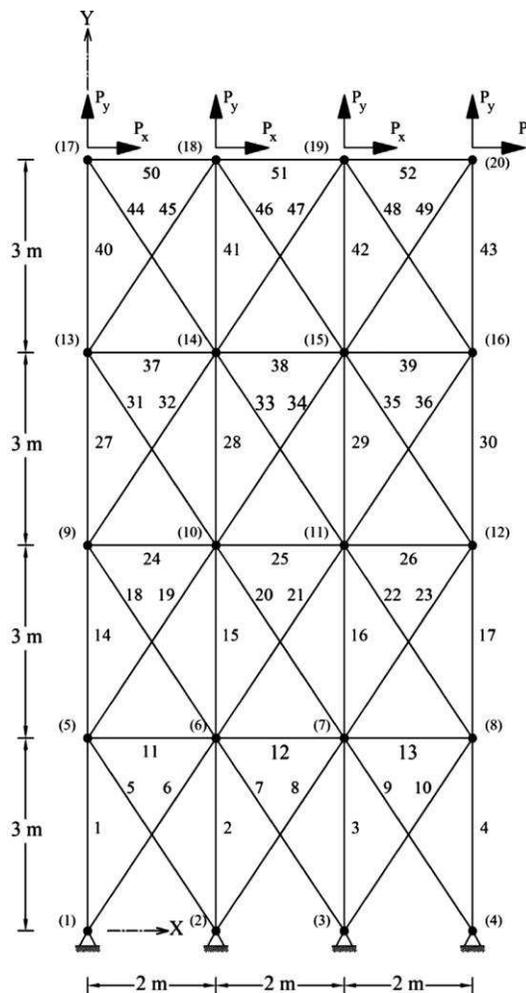


Figura 23: Armadura plana compuesta por 52 barras.

El número de variables independientes (área) se reduce a doce grupos de elementos asociados de la siguiente manera:

Grupo 1 (G1): Elementos 1, 2, 3 y 4.

Grupo 2 (G2): Elementos 5, 6, 7, 8, 9 y 10.

Grupo 3 (G3): Elementos 11, 12 y 13.

Grupo 4 (G4): Elementos 14, 15, 16 y 17.

Grupo 5 (G5): Elementos 18, 19, 20, 21, 22 y 23.

Grupo 6 (G6): Elementos 24, 25 y 26.

Grupo 7 (G7): Elementos 27, 28, 29 y 30.

Grupo 8 (G8): Elementos 31, 32, 33, 34, 35 y 36.

Grupo 9 (G9): Elementos 37, 38 y 39.

Grupo 10 (G10): Elementos 40, 41, 42 y 43.

Grupo 11 (G11): Elementos 44, 45, 46, 47, 48 y 49.

Grupo 12 (G12): Elementos 50, 51 y 52.

Las variables discretas (áreas), son seleccionadas del conjunto de datos del código AISC mostrados en la Tabla 11.

Tabla 11: Valores discretos disponibles para áreas transversales de la norma AISC.

No.	in ²	mm ²									
1	0.111	71.613	17	1.457	939.998	33	3.84	2477.414	49	11.5	7419.340
2	0.141	90.968	18	1.563	1008.385	34	3.87	2496.769	50	13.5	8709.660
3	0.196	126.451	19	1.62	1045.159	35	3.88	2503.221	51	13.9	8967.724
4	0.25	161.290	20	1.8	1161.288	36	4.18	2696.769	52	14.2	9161.272
5	0.307	198.064	21	1.99	1283.868	37	4.22	2722.575	53	15.5	9999.980
6	0.391	252.258	22	2.13	1374.191	38	4.49	2896.768	54	16	10322.560
7	0.442	285.161	23	2.38	1535.481	39	4.59	2961.284	55	16.9	10903.204
8	0.563	363.225	24	2.62	1690.319	40	4.8	3096.768	56	18.8	12129.008
9	0.602	388.386	25	2.63	1696.771	41	4.97	3206.445	57	19.9	12838.684
10	0.766	494.193	26	2.88	1858.061	42	5.12	3303.219	58	22	14193.520
11	0.785	506.451	27	2.93	1890.319	43	5.74	3703.218	59	22.9	14774.164
12	0.994	641.289	28	3.09	1993.544	44	7.22	4658.055	60	24.5	15806.420
13	1	645.160	29	3.38	2180.641	45	7.97	5141.925	61	26.5	17096.740
14	1.13	729.031	30	3.47	2238.705	46	8.53	5503.215	62	28	18064.480
15	1.228	792.256	31	3.55	2290.318	47	9.3	5999.988	63	30	19354.800
16	1.266	816.773	32	3.63	2341.931	48	10.85	6999.986	64	33.5	21612.860

Los resultados de secciones óptimas para los grupos de áreas planteados se muestran en la Tabla 12, donde se logró obtener un peso óptimo de 1902.605 kg. El mejor valor de peso encontrado es más liviano en hasta un 17.22 % que los valores encontrados por otros investigadores como Ho-Huu [17], Kaveh [46, 52], Eskandar [57], Baghlani [58], Li [62], Sadollah [63, 69], Dede [65], Cheng [68] y Lee [70] quienes desarrollaron el problema con diferentes metodologías propuestas, referenciadas en la tabla correspondiente, demostrando de esta manera la eficacia del ARSA.

En la Figura 24 se muestra la gráfica de convergencia de ARSA para este problema. El algoritmo consiguió una estructura alrededor de 6.09 veces más liviana respecto al primer diseño factible generado. La convergencia al valor óptimo encontrado se logra en el descenso de temperatura número 68.

En la Figura 25 se observa la obtención del valor óptimo de peso de la estructura con el ARSA en todas las veces que fue ejecutado el algoritmo propuesto. El valor del promedio de las ejecuciones fue 1910.679 kg con una desviación estándar de 21.380 kg.

Tabla 12: Comparación de resultados con otros autores para la armadura de 52 elementos.

Método	Peso (kg)	DF	Variables de diseño (mm ²)											
			G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12
DE [17]	1902.61	4.99	4658.06	1161.29	494.19	3303.22	940.00	494.19	2238.71	1008.39	494.19	1283.87	1161.29	494.19
aeDE [17]	1902.61	6.68	4658.06	1161.29	494.19	3303.22	940.00	494.19	2238.71	1008.39	494.19	1283.87	1161.29	494.19
MCSS [46]	1904.05	--	4658.06	1161.29	363.23	3303.22	940.00	506.45	2238.71	1008.39	388.39	1283.87	1161.29	729.03
IMCSS [46]	1902.61	--	4658.06	1161.29	494.19	3303.22	940.00	494.19	2238.71	1008.39	494.19	1283.87	1161.29	494.19
WCA [69]	1902.61	7.09	4658.06	1161.29	494.19	3303.22	940.00	494.19	2238.71	1008.39	494.19	1283.87	1161.29	494.19
IMBA [69]	1902.61	1.13	4658.06	1161.29	494.19	3303.22	940.00	494.19	2238.71	1008.39	494.19	1283.87	1161.29	494.19
MBA [63]	1902.61	4.09	4658.06	1161.29	494.19	3303.22	940.00	494.19	2238.71	1008.39	494.19	1283.87	1161.29	494.19
FA [58]	1955.18	--	4658.06	1161.29	494.19	3303.22	940.00	645.16	2180.64	1008.39	816.77	1161.29	1374.19	729.03
AFA [58]	1903.37	--	4658.06	1161.29	363.23	3303.22	940.00	494.19	2238.71	1008.39	641.29	1283.87	1161.29	494.19
TLBO [65]	1902.61	--	4658.06	1161.29	494.19	3303.22	940.00	494.19	2238.71	1008.39	494.19	1283.87	1161.29	494.19
ICACO [57]	1902.61	4.50	4658.06	1161.29	494.19	3303.22	940.00	494.19	2238.71	1008.39	494.19	1283.87	1161.29	494.19
ICA [57]	1903.37	5.58	4658.06	1161.29	363.23	3303.22	940.00	494.19	2238.71	1008.39	641.29	1283.87	1161.29	494.19
HHS [68]	1902.61	1.31	4658.06	1161.29	494.19	3303.22	940.00	494.19	2238.71	1008.39	494.19	1283.87	1161.29	494.19
DHPSACO [52]	1904.83	-	4658.06	1161.29	494.19	3303.22	1008.39	285.16	2290.32	1008.39	388.39	1283.87	1161.29	506.45
PSO [62]	2230.16	--	4658.06	1374.19	1858.06	3206.44	1283.87	252.26	3303.22	1045.16	126.45	2341.93	1008.38	1045.16
PSOPC [62]	2146.63	--	5999.99	1008.38	2696.77	3206.44	1161.29	729.03	2238.71	1008.38	494.19	1283.87	1161.29	494.19
HPSO [62]	1905.50	--	4658.06	1161.29	363.23	3303.22	940.00	494.19	2238.71	1008.39	388.39	1283.87	1161.29	792.26
HS [70]	1906.76	--	4658.06	1161.29	506.45	3303.22	940.00	494.19	2290.32	1008.39	2290.32	1535.48	1045.16	506.45
CBO [71]	1899.35	106.01	4658.06	1161.29	388.39	3303.22	940.00	506.45	2238.71	1008.39	506.45	1283.87	1161.29	506.45
ARSA	1902.61	21.38	4658.06	1161.29	494.19	3303.22	940.00	494.19	2238.71	1008.39	494.19	1283.87	1161.29	494.19

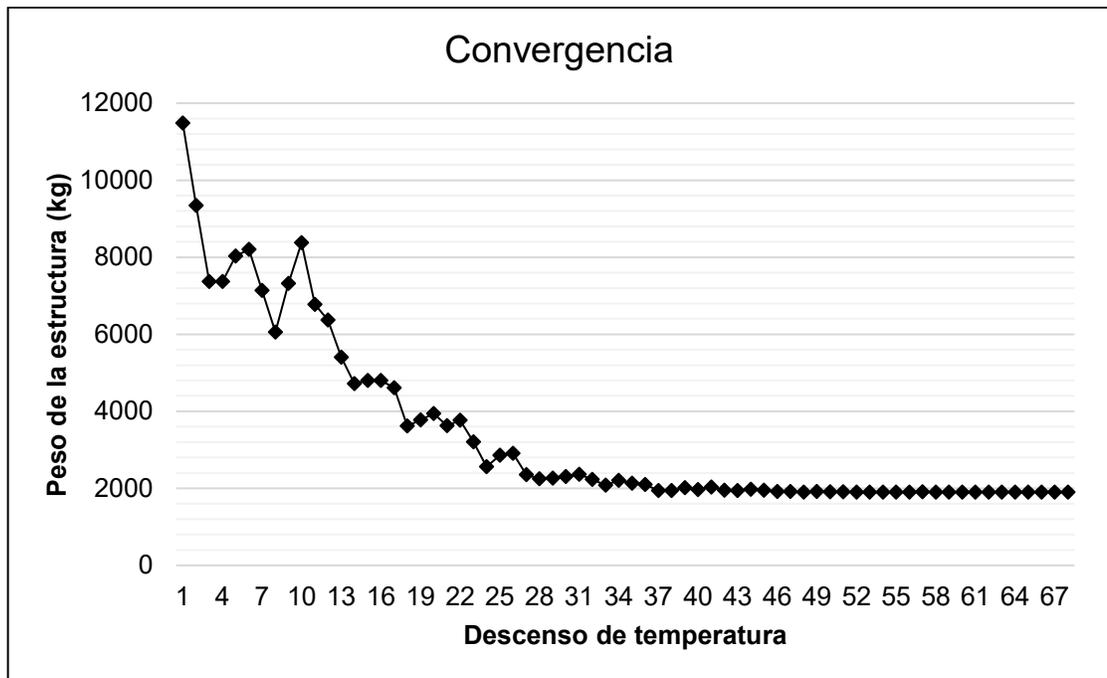


Figura 24: Gráfica de convergencia para la cercha de 52 barras.

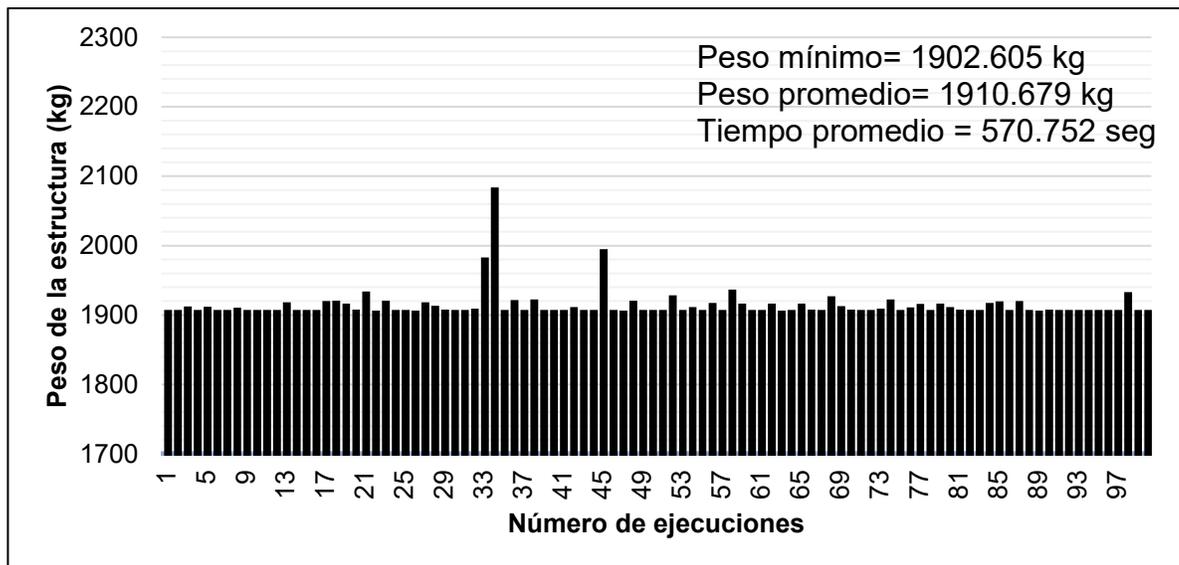


Figura 25: Peso encontrado en cada ejecución del algoritmo para la armadura de 52 barras.

Cómo se evidenció en el proceso de optimización de los problemas estructurales optimizados mediante variables discretas, el ARSA converge al valor óptimo de la estructura. El criterio de parada del algoritmo que se cumplió en este tipo de problemas es que la desviación estándar de los últimos valores no supere el valor

límite propuesto. Esto hace que no se desperdicie recursos computacionales y se evidencia en la fase final de las gráficas de convergencia de cada problema.

En las gráficas de convergencia, para todos los problemas optimizados (cerchas con variables continuas y discretas), se observa como a medida que estos aumentan su dificultad (mayor número de variables y restricciones), la gráfica muestra como el algoritmo escapa óptimos locales en su proceso de optimización respaldando de esta manera la capacidad del ARSA de desenvolverse eficazmente y conseguir dirigirse hacia el valor óptimo global.

El eficiente desempeño del ARSA queda demostrado en los valores de secciones transversales, peso, peso promedio y desviación estándar alcanzados. Además, el ARSA en todos los problemas cumple con las restricciones preestablecidas.

La Tabla 13 y Tabla 14 resumen los resultados obtenidos con ARSA en cuanto a mejor valor encontrado, promedio, desviación estándar y tiempos de ejecución, para las armaduras con variables continuas y discretas.

Tabla 13: Resumen de resultados armaduras con variables continuas.

Estadística	PROBLEMA			
	Armadura plana de 10 barras		Armadura plana de 17 barras	Armadura plana de 18 barras
	Caso 1	Caso 2		
MF (lb)	5060.900	4677.036	2582.002	6430.531
MEF (lb)	5061.838	4678.890	2582.935	6430.538
DF (lb)	7.17E-01	1.46E+00	6.77E-01	4.34E-03
TP (s)	117.33	144.67	162.02	26.50

Tabla 14: Resumen de resultados armaduras con variables discretas.

PROBLEMA			
Estadística	Armadura plana de 10 barras	Armadura plana de 15 barras	Armadura plana de 52 barras
MF (lb)	5490.738 lb	105.735 kg	1902.605 kg
MEF (lb)	5491.478 lb	105.737 kg	1910.679 kg
DF (lb)	3.752 lb	0.012 kg	21.380 kg
TP (s)	93.674	90.841	570.752

7. CONCLUSIONES

Se ha conseguido desarrollar un procedimiento para la resolución de problemas de optimización estructural de armaduras en 2D, donde se minimiza el peso de las estructuras (función objetivo). La optimización se realiza con la técnica estocástica propuesta llamada Algoritmo Recocido Simulado Alterado (ARSA). Los valores de pesos óptimos, áreas y desviaciones estándar obtenidos por ARSA, fueron comparados con los resultados presentados por otros autores, mostrando que son coherentes y satisfactorios, dando así validez al algoritmo.

En cuanto a la técnica empleada, se puede observar que ARSA presenta versatilidad, eficacia y exactitud para analizar diferentes tipos de problemas con diferentes números de elementos, esto se ve reflejado en los valores de pesos, áreas, tiempos de ejecución y desviación estándar.

Finalmente, ARSA presenta un buen equilibrio entre la búsqueda local y global (ver graficas de convergencias), esto causado por las modificaciones propuestas, siendo el algoritmo permisivo en las primeras fases, admitiendo valores peores al ya encontrado y en su fase final, solo aceptando valores de peso mejores al antes hallado.



8. TRABAJOS FUTUROS

Los resultados del presente trabajo han sido plenamente satisfactorios. No obstante, se abren nuevas vías de trabajo y futuras líneas de investigación de entre las que merece la pena destacar las siguientes:

- a) Desarrollar un procedimiento para la optimización de armaduras en 3D.
- b) Optimización de armaduras con restricciones de frecuencias.
- c) Implementar una interfaz gráfica.
- d) Proponer nuevos algoritmos de optimización.

REFERENCIAS

- [1] H. Li y Y. Ma, «Discrete optimum design for truss structures by subset simulation algorithm,» *Journal of Aerospace Engineering*, vol. 28, n° 4, p. 04014091, 2014.
- [2] R. Balling, «Optimal steel frame design by simulated annealing,» *J. Struct. Eng.*, vol. 117, n° 6, pp. 1780-1795, 1991.
- [3] W. Bennage, «Single and multiobjective structural optimization in discrete-continuous variables using simulated annealing,» *International Journal for Numerical Methods in Engineering*, vol. 38, n° 16, pp. 2753-2773, 1995.
- [4] O. Hasançebi y F. Erbartur, «Layout optimisation of trusses using simulated annealing,» *Advances in Engineering Software*, vol. 33, n° 7, pp. 681-696, 2002.
- [5] O. Hasançebi y F. Erbatur, «On efficient use of simulated annealing in complex structural optimization problems,» *Acta Mechanica*, vol. 157, n° 1, pp. 27-50, 2002.
- [6] L. Lamberti, «An efficient simulated annealing algorithm for design optimization of truss structures,» *Computers and Structures*, vol. 86, pp. 1936-1953, 2008.
- [7] C. Millán y E. Millán, «Algoritmo Simulated Annealing Modificado ASAM para Minimizar Peso en Cerchas Planas con Variables Discretas,» *INGE CUC*, vol. 12, n° 2, pp. 9-16, 2016.
- [8] C. Millán, A. Matoski y M. Wellington, «Algoritmo simulated annealing modificado para diseño óptimo de armaduras con variables continuas,» *Tecnología en Marcha*, vol. 30, n° 2, pp. 150-165, 2017.
- [9] D. Fouskakis y D. Draper, «Stochastic optimization: a review,» *International Statistitcal Review*, vol. 70, n° 3, pp. 315-349, 2002.
- [10] J. Gentle, W. Hardle y Y. Mori, *Handbook of Computational Statistics*, Springer, 2004.
- [11] S. Kirkpatrick, C. Gelatt y M. Vecchi, «Optimization by simulated annealing,» *Science*, vol. 220, pp. 671-680, 1983.
- [12] J. Holland, *Adaptation in natural and artificial systems*, The MIT Press, 1975.
- [13] J. Kennedy y R. Eberhart, «Particle swarm optimization,» *Neural Networks*, vol. 4, pp. 1942-1948, 1995.
- [14] M. Sonmez, «Artificial Bee Colony algorithm for optimization of truss structures,» *Applied Soft Computing*, vol. 11, pp. 2406-2418, 2011.
- [15] C. Millán y C. Celis, *Evaluación del desempeño de algoritmos de evolución paramétrica para optimización, Tesis de pregrado, Escuela de Ingeniería Civil, UIS, 2009.*

- [16] O. Hasancebi, «Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures,» *Computers and Structures*, vol. 86, n° 1, pp. 119-132, 2008.
- [17] V. Ho-Huu, T. Nguyen-Thoi, T. Vo-Duy y T. Nguyen-Trang, «An adaptive elitist differential evolution for optimization of truss structures with discrete design variables,» *Computers and Structures*, vol. 165, pp. 59-75, 2016.
- [18] R. Storn y K. Price, «Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces.,» *J Glob Optim* , n° 11, pp. 341-359, 1997.
- [19] K. Lee y Z. Geem, «A new structural optimization method based on the harmony search algorithm,» *Comput. Struct*, vol. 82, n° 9, pp. 781-798, 2004.
- [20] K. Shea, J. Cagan y S. Fennes, «A shape annealing approach to optimal truss design with dynamic grouping of members,» *Journal of Mechanical Design*, vol. 119, n° 3, pp. 388-394, 1997.
- [21] K. Shea y J. Cagan, «The design of novel roof trusses with shape annealing: assessing the ability of a computational method in aiding structural designers with varying design intent,» *Design Studies*, vol. 20, n° 1, pp. 3-23, 1999.
- [22] S. Hustin y A. Sangiovanni-Vincentelli, «TIM, a new standard cell placement program based on the simulated annealing algorithm,» de *In IEEE Physical Design Workshop on Placement and Floorplanning*, 1987.
- [23] P. Shim y S. Manoochehri, «Generating optimal configurations in structural design using simulated annealing,» *International journal for numerical methods in engineering*, vol. 40, n° 6, pp. 1053-1069, 1997.
- [24] C. Pantelides y S. Tzan, «Modified iterated simulated annealing algorithm for structural synthesis,» *Advances in Engineering Software*, vol. 31, n° 6, pp. 391-400, 2000.
- [25] T. Chen y J. Su, «Efficiency improvement of simulated annealing in optimal structural designs,» *Advances in Engineering Software*, vol. 33, n° 7, pp. 675-680, 2002.
- [26] M. Kripka, «Discrete optimization of trusses by simulated annealing,» *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 26, n° 2, pp. 170-173, 2004.
- [27] O. Erdal y F. Sonmez, «Optimum design of composite laminates for maximum buckling load capacity using simulated annealing,» *Composite Structures*, vol. 71, n° 1, pp. 45-52, 2005.
- [28] M. Ali, A. Törn y S. Viitanen, «A direct search variant of the simulated annealing algorithm for optimization involving continuous variables,» *Comput Operat Res* , vol. 29, pp. 87-102, 2002.
- [29] F. Sonmez, «Shape optimization of 2D structures using simulated annealing,» *Computer methods in applied mechanics and engineering*, vol. 196, n° 35, pp. 3279-3299, 2007.

- [30] O. Hasançebi, S. Çarbaş y M. P. Saka, « Improving the performance of simulated annealing in structural optimization,» *Structural and Multidisciplinary Optimization*, vol. 41, nº 2, pp. 189-203, 2010.
- [31] S. Martínez, J. París, I. Colominas, F. Navarrina y M. Casteleiro, «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*, vol. 30, nº 2, pp. 121-135, 2014.
- [32] R. Asl, M. Aslani y M. Panahi, «Sizing Optimization of Truss Structures using a Hybridized Genetic Algorithm,» *arXiv*, vol. 1, pp. 1306-1454, 2013.
- [33] C. Millán, O. Begambre y E. Millán, «Propuesta y validación de un algoritmo Simulated Annealing modificado para la solución de problemas de optimización,» *Rev. int. métodos numér. cálc. diseño ing.*, vol. 30, nº 4, pp. 264-270, 2014.
- [34] S. Ramírez E, *Metodología basada en Algoritmos Genéticos y Programación en Paralelo para el Diseño Óptimo de Armaduras de Acero*, Bogotá, 2014.
- [35] R. Hibbeler, *Análisis Estructural*, México: Prentice-Hall Hispanoamerica S.A., 1997.
- [36] G. Polya, *How to solve it: A new aspect of mathematics method*, Princeton: Princeton University Press, 1957.
- [37] M. Cerrolaza y W. Annicchiarico, *Algoritmos de optimización estructural basados en simulación genética*, CDCH UCV, 1996.
- [38] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller y E. Teller, «Equation of state calculations by fast computing machines,» *J. Chem. Phys.*, vol. 21, nº 6, pp. 1087-1092, 1953.
- [39] A. Corana, M. Marchesi, C. Martini y S. Ridella, «Minimizing multimodal functions of continuous variables with the "Simulated Annealing" Algorithm,» *ACM Transactions on Mathematical Software*, vol. 13, nº 3, pp. 262-280, 1983.
- [40] K. Dowland y B. Díaz, «Diseño de heurística y fundamentos del Simulated Annealing,» *Revista Iberoamericana de IA*, vol. 19, pp. 93-102, 2003.
- [41] A. R. Hedar y A. Ahmed, *Studies on metaheuristics for continuous global optimization problems*, Kyoto, 2004.
- [42] H. Liu, Z. Cai y Y. Wang, «Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization,» *Applied Soft Computing*, vol. 10, pp. 629-640, 2010.
- [43] T. P. Runarsson y X. Yao, «Stochastic ranking for constrained evolutionary optimization,» *IEEE Transactions on evolutionary computation*, vol. 4, nº 3, pp. 284-294, 2000.
- [44] B. Dizangian y M. Ghasemi, «A fast marginal feasibility search method in size optimization of truss structures,» *Asian Journal of Civil Engineering*, vol. 15, nº 5, pp. 567-585, 2015.
- [45] L. Li, Z. Huang, F. Liu y Q. Wu, «A heuristic particle swarm optimizer for

- optimization of pin connected structures,» *Computers and Structures*, vol. 85, pp. 340-349, 2007.
- [46] A. Kaveh, B. Mirzaei y A. Jafarvand, «An improved magnetic charged system search for optimization of truss structures with continuous and discrete variables,» *Applied Soft Computing*, vol. 28, pp. 400-410, 2015.
- [47] S. Degertekin, «Improved harmony search algorithms for sizing optimization of truss structures,» *Comput Struct.*, vol. 92, pp. 229-241, 2012.
- [48] A. Baghlani y M. Makiabadi, «Weight Optimization of Truss Structures by a New Feasible Boundary Search Technique Hybridized with Firefly Algorithm,» *KSCE Journal of Civil Engineering*, vol. 18, n° 4, pp. 1105-1118, 2014.
- [49] A. Baghlani y M. Makiabadi, «An enhanced particle swarm optimization for design of pin connected structures,» *Scientia Iranica*, vol. 20, n° 5, pp. 1415-1432, 2013.
- [50] A. Charalampakis, «Comparison of metaheuristic algorithms for size optimization of trusses,» de *11th HSTAM International Congress on Mechanics*, Greece, 2016.
- [51] A. Baghlani, M. Makiabadi y H. Rahnema, «A new accelerated firefly algorithm for size optimization of truss structures,» *Scientia Iranica*, vol. 20, n° 6, pp. 1612-1625, 2013.
- [52] A. Kaveh y S. Talatahari, «Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures,» *Comput. Struct.*, vol. 87, n° 5-6, pp. 267-283, 2009.
- [53] R. Perez y K. Behdinan, «Particle swarm intelligence for structural design optimization,» *Computers and Structures*, vol. 85, pp. 1579-1588, 2007.
- [54] H. Adeli y S. Kumar, «Distributed genetic algorithm for structural optimization,» *J Aerospace Eng, ASCE*, vol. 8, n° 3, pp. 156-163, 1995.
- [55] L. F. F. Miguel y L. F. Fadel Miguel, «Novel metaheuristic algorithms applied to optimization of structures,» *WSEAS Trans Appl Theor Mech*, vol. 7, pp. 210-220, 2012.
- [56] K. Imai y L. Schmidt, «Configuration optimization of trusses,» *J Structural Division, ASCE*, vol. 107, pp. 745-756, 1981.
- [57] H. Eskandar, M. Sabour y P. Salehi, «Imperialist Competitive Ant Colony Algorithm for Truss Structures,» *World Applied Sciences Journal*, vol. 12, n° 1, pp. 94-105, 2011.
- [58] A. Baghlani, M. Makiabadi y M. Sarcheshmehpour, «Discrete Optimum Design of Truss Structures by an Improved Firefly Algorithm,» *Advances in Structural Engineering*, vol. 17, n° 10, pp. 1516-1530, 2014.
- [59] C. Camp y M. Farshchin, «Design of space trusses using modified teaching-learning based optimization,» *Engineering Structures*, Vols. %1 de %262-63, pp. 87-97, 2014.
- [60] C. Camp y B. Bichon, «Design of space trusses using ant colony

- optimization,» *Journal of Structural Engineering*, vol. 130, nº 5, pp. 741-751, 2004.
- [61] C. Camp, «Design of space trusses using big bang–big crunch optimization,» *J Struct Eng*, vol. 133, nº 7, pp. 999-1008, 2007.
- [62] L. Li, Z. Huang y F. Liu, «A heuristic particle swarm optimization method for truss structures with discrete variables,» *Computers and Application of improved hybrid genetic algorithm to*, vol. 87, pp. 435-443, 2009.
- [63] A. Sadollah, A. Bahreininejad, H. Eskandar y M. Hamdi, «Mine blast algorithm for optimization of truss structures with discrete variables,» *Comput Struct*, Vols. %1 de %2102-103, pp. 49-63, 2012.
- [64] M. Sonmez, «Discrete optimum design of truss structures using artificial bee colony algorithm,» *Struct Multidisc Optim*, vol. 43, nº 1, pp. 85-97, 2011.
- [65] T. Dede, «Application of Teaching-Learning-based-Optimization Algorithm for the Discrete Optimization of Truss Structures,» *KSCE Journal of Civil Engineering*, vol. 18, nº 6, pp. 1759-1767, 2014.
- [66] H. Barbosa, A. Lemonge y C. Borges, «A genetic algorithm encoding for cardinality constraints and automatic variable linking in structural optimization,» *Eng Struct*, vol. 30, nº 12, pp. 3708-3723, 2008.
- [67] Y. Zhang, J. Liu, B. Liu, C. Zhu y Y. Li, «Application of improved hybrid genetic algorithm to optimized design of architecture structures,» *J South China Univ. Technol*, vol. 33, nº 3, pp. 69-72, 2003.
- [68] M. Cheng, D. Prayogo, Y. Wu y M. Lukito, «A Hybrid Harmony Search algorithm for discrete sizing optimization of truss structure,» *Automation in Construction*, vol. 69, pp. 21-33, 2016.
- [69] A. Sadollah, H. Eskandar, A. Bahreininejad y J. Hoon Kim, «Water cycle, mine blast and improved mine blast algorithms for discrete sizing optimization of truss structures,» *Computers and Structures*, vol. 149, pp. 1-16, 2015.
- [70] K. Lee, Z. Geem, S.-H. Lee y K.-W. Bae, «The harmony search heuristic algorithm for discrete structural optimization,» *Eng. Optim.*, vol. 37, nº 7, pp. 663-684, 2005.
- [71] A. Kaveh y V. Mahdavi, «Colliding Bodies Optimization method for optimum discrete design of truss structures,» *Computers and Structures*, vol. 139, pp. 43-53, 2014.