

Optimización topológica de estructuras de celosía

Luis Humberto Niño Álvarez

Trabajo de Grado para Optar el Título de Magíster en Ingeniería Civil

Director

Oscar Javier Begambre Carrillo

PhD. Ingeniería de Estructuras

Codirector

Leonardo Moreno De Luca

Magister en Ingeniería Civil

Universidad Industrial de Santander

Facultad de Ingenierías Fisico-Mecánicas

Escuela de Ingeniería Civil

Maestría en Ingeniería Civil

Bucaramanga

2020

### **Dedicatoria**

A Dios Padre todopoderoso y eterno, guía fundamental en todos los aspectos de mi vida, por permitirme culminar con éxito el programa de posgrado cursado, todos los objetivos, planes y proyectos durante toda la vida.

A mis padres Bertha y Luis y mi hermano Nixon por estar siempre ahí, en todo momento, aconsejándome y apoyándome incondicionalmente Son la inspiración que me impulsa a seguir adelante, para cumplir con los objetivos propuestos y crecer cada día.

A mis abuelos Saúl y Carmen por su amor y compañía. En memoria de mis abuelos Mercedes y Ulises por sus valiosos y sabios consejos.

### **Agradecimientos**

A la Universidad Industrial de Santander (UIS) por la beca de matrícula y sostenimiento otorgada para cursar el programa de posgrado en esta prestigiosa institución.

Al profesor Oscar Begambre por su orientación, ayuda, paciencia y la dedicación continua durante el desarrollo del trabajo de investigación. Al ingeniero y arquitecto Leonardo Moreno De Luca por su apoyo y asesoría.

A los profesores, por sus enseñanzas y todo lo aprendido durante mi formación. A la Escuela de Ingeniería Civil y al grupo de investigación INME, por el soporte, la ayuda, la colaboración y los espacios de trabajo brindados para el adecuado desarrollo de todas las actividades del programa de posgrado.

A todos los compañeros de la Maestría en Ingeniería Civil, muchas gracias.

**Tabla de contenido**

Introducción .....	21
1. Objetivos .....	30
1.1. Objetivo general .....	30
1.2. Objetivos específicos .....	30
2. Aplicaciones recientes de optimización topológica en estructuras de celosía: revisión de literatura .....	31
2.1. Combinación de algoritmos metaheurísticos con algoritmos clásicos y técnicas de mejora local .....	32
2.2. Combinación de algoritmos metaheurísticos con otros algoritmos metaheurísticos .....	35
2.3. Desarrollo y aplicación de nuevos algoritmos metaheurísticos en optimización topológica de estructura de celosías .....	36
2.4. Aplicaciones reales de optimización estructural de celosías, mediante algoritmos metaheurísticos .....	38
2.5. Aplicaciones de optimización topológica multiobjetivo de estructuras de celosía.....	39
2.6. Interacción del usuario en la optimización estructural .....	43
3. Optimización multiobjetivo y optimización estructural .....	43
3.1. Formulación matemática de un problema de optimización multiobjetivo.....	46
3.1.1. Definición de valor óptimo en optimización multiobjetivo.....	47
3.1.2. Conceptos básicos en optimización multiobjetivo.....	48
3.1.3. Metas de un problema de optimización multiobjetivo.....	50
3.2. Optimización estructural .....	52
3.2.1. Optimización de tamaño .....	53
3.2.2. Optimización de forma .....	54
3.2.3. Optimización topológica.....	55

3.2.4. Tipos de optimización estructural aplicables en estructuras de celosía.....	57
3.2.5. Enfoque clásico de optimización topológica de estructuras de celosía .....	59
3.2.6. Métodos para la generación de la estructura base.....	61
3.2.7. Formulación matemática de un problema de optimización topológica multiobjetivo de estructuras de celosía .....	65
3.3. Métodos de solución de problemas de optimización .....	68
3.3.1. Métodos de un solo objetivo.....	68
3.3.2. Métodos multiobjetivo.....	73
3.3.3. Técnicas para el manejo de restricciones en problemas de optimización.....	82
3.3.4. Medidas de desempeño de los algoritmos de optimización multiobjetivo .....	85
4. Algoritmo de optimización topológica multiobjetivo híbrido (AOTMOH) propuesto .....	89
4.1. Descripción de la formulación matemática aplicada al problema de optimización topológica multiobjetivo.....	90
4.2. Descripción de la estrategia para el manejo de las restricciones y funciones objetivo.....	91
4.3. Descripción de la estrategia de generación de la estructura base .....	93
4.3.1. Método de niveles de conectividad de nodos (NCN) .....	93
4.3.2. Método de trayectorias de esfuerzo y macroelementos (TEM) .....	94
4.4. Descripción de la estrategia de solución del problema de optimización topológica multiobjetivo.....	99
4.5. Descripción de la estrategia para remover o mantener elementos de la estructura base durante el proceso de optimización.....	101
4.6. Interfaz gráfica .....	102
5. Validación de los algoritmos programados.....	105
5.1. Validación de los algoritmos metaheurísticos de optimización multiobjetivo NSGA-II, MOPSO y AMOSA .....	105
5.1.1. Problemas de funciones matemáticas sin restricciones .....	107

5.1.2. Problemas de funciones matemáticas con restricciones .....	118
5.1.3. Conclusión general de desempeño de los algoritmos NSGA-II, MOPSO y AMOSA en problemas sin restricciones y con restricciones .....	127
5.2. Validación de la estrategia de generación de la estructura base usando el método TEM ...	128
5.3. Validación del algoritmo de análisis estructural .....	132
5.3.1. Armadura de 7 elementos .....	133
5.3.2. Armadura de 26 elementos .....	134
6. Aplicación del algoritmo de optimización topológica multiobjetivo híbrido (AOTMOH)....	137
6.1. Validación del algoritmo AOTMOH .....	137
6.1.1. Problema de optimización de tamaño multiobjetivo de la armadura de 10 elementos.....	138
6.1.2. Problema de optimización topológica multiobjetivo de la estructura base de 10 elementos .....	140
6.1.3. Problema de optimización topológica multiobjetivo de la estructura base de 14 elementos .....	143
6.1.4. Problema de optimización topológica multiobjetivo de la estructura base de 15 elementos .....	146
6.1.5. Análisis de resultados de desempeño en problemas de optimización multiobjetivo de estructuras de celosía 2D .....	148
6.2. Problemas de optimización topológica multiobjetivo con el método NCN .....	152
6.2.1. Estructura base de 10 elementos y 6 nodos. ....	153
6.2.2. Estructura base de 17 elementos y 9 nodos .....	153
6.2.3. Estructura base 39 elementos y 12 nodos .....	154
6.2.4. Estructura base 54 elementos y 15 nodos .....	155
6.2.5. Resultados de desempeño de los algoritmos en las estructuras base generadas usando el método NCN .....	156
6.2.6. Análisis de resultados de desempeño en las estructuras base generadas con el método NCN .....	159

6.3. Problemas de optimización topológica multiobjetivo con el método TEM .....	162
6.3.1. Viga cantiléver 1 .....	163
6.3.2. Viga cantiléver 2 .....	166
6.3.3. Viga simplemente apoyada .....	170
6.3.4. Viga cantiléver 3 .....	173
6.3.5. Análisis de resultados de desempeño en las estructuras base generadas con el método TEM .....	176
6.4. Problema de diseño óptimo de una torre de transmisión de energía eléctrica plana .....	180
6.4.1. Descripción del problema de diseño de la torre plana .....	180
6.4.2. Optimización de tamaño multiobjetivo de la torre plana .....	185
6.4.3. Optimización topológica multiobjetivo de la torre plana con el método TEM .....	189
6.4.4. Análisis de resultados del proceso de optimización topológica multiobjetivo de la torre plana con el método TEM .....	199
7. Conclusiones .....	203
Referencias bibliográficas .....	209
Apéndices .....	225

**Lista de figuras**

Figura 1. Torres de transmisión de energía eléctrica .....	24
Figura 2. Número de documento publicados por año en el tema de investigación .....	27
Figura 3. Representación del espacio de decisión $\Omega$ de las variables y el correspondiente espacio objetivo $\Lambda$ .....	47
Figura 4. Soluciones dominadas, no dominadas y frente de Pareto para un problema de minimización de dos funciones objetivo $f_1$ y $f_2$ .....	50
Figura 5. Las dos metas de la optimización multiobjetivo (izquierda). Un conjunto ideal de soluciones no dominadas que cumple con las dos metas (derecha).....	51
Figura 6. Ejemplos de aproximación de frente de Pareto para un problema de minimización de dos funciones objetivo.....	51
Figura 7. El problema de la optimización estructural encuentra la mejor estructura que transmite la carga $F$ al soporte .....	52
Figura 8. Pasos para formular un problema de optimización estructural .....	53
Figura 9. Ejemplos de estructuras donde la optimización de tamaño puede ser aplicada .....	54
Figura 10. Problemas de optimización de tamaño, forma y topológica en celosías .....	59
Figura 11. Optimización topológica de estructuras de celosía .....	60
Figura 12. Niveles de conectividad para una estructura base en el método estándar. ....	62
Figura 13. Tipos de malla para la estructura base usando el método de macroelementos .....	63
Figura 14. Macroelementos de malla cuadrada y voronoi.....	64
Figura 15. Estructura base usando el método de trayectorias de esfuerzo.....	64
Figura 16. Componentes de un algoritmo metaheurístico .....	72
Figura 17. Distancia de las soluciones extremas .....	86
Figura 18. Hipervolumen encerrado por las soluciones no dominadas de $Q = \{A, B, C, D, E\}$ .....	87
Figura 19. Representación de la estructura base usando el método NCN .....	94
Figura 20. Cálculo del estado de esfuerzo para un problema plano y continuo .....	95
Figura 21. Trayectorias de esfuerzo.....	95
Figura 22. Trayectorias de esfuerzo clave .....	98
Figura 23. Estructura de macroelementos.....	98

Figura 24. Estructura base usando el método TEM .....	99
Figura 25. Representación de frentes de Pareto para el problema de optimización topológica multiobjetivo calculados con los algoritmos NSGA.II (izquierda), MOPSO (centro) y AMOSA (derecha) .....	100
Figura 26. Representación del proceso de cálculo del frente de Pareto envolvente .....	100
Figura 27. Representación del frente de Pareto envolvente con las soluciones óptimas de la estructura base generada con el método NCN (izquierda) y TEM (derecha) .....	101
Figura 28. Interfaz gráfica en el software Rhinoceros-Grasshopper .....	103
Figura 29. Diagrama de flujo del algoritmo AOTMOH .....	104
Figura 30. Frentes de Pareto obtenidos en cada una de las 20 ejecuciones, para cada algoritmo y para cada una de los 8 problemas de funciones matemáticas sin restricciones .....	110
Figura 31. Comparación de todos los frentes de Pareto de los tres algoritmos (izquierda) con las soluciones de referencia (derecha), para los 8 problemas de funciones matemáticas sin restricciones .....	112
Figura 32. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de funciones matemáticas sin restricciones .....	115
Figura 33. Diagramas de cajas para distancia generacional (GD) de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de funciones matemáticas sin restricciones.....	115
Figura 34. Diagramas de cajas para propagación ( $\Delta$ ) de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de funciones matemáticas sin restricciones .....	116
Figura 35. Frentes de Pareto obtenidos en cada una de las 20 ejecuciones, para cada algoritmo y para cada una de los 6 problemas de funciones matemáticas sin restricciones .....	121
Figura 36. Comparación de todos los frentes de Pareto de los tres algoritmos (izquierda) con las soluciones de referencia (derecha), para los 6 problemas de funciones matemáticas con restricciones .....	122
Figura 37. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de funciones matemáticas con restricciones .....	124
Figura 38. Diagramas de cajas para hipervolumen (HV) de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de funciones matemáticas con restricciones .....	125
Figura 39. Armadura de 7 elementos .....	133
Figura 40. Armadura de 26 elementos .....	134

Figura 41. Convención utilizada para referenciar las soluciones extremas del frente de Pareto	137
Figura 42. Armadura de 10 elementos para optimización de tamaño multiobjetivo .....	139
Figura 43. Frentes de Pareto en cada una de las 20 ejecuciones, para cada algoritmo en problema de optimización de tamaño multiobjetivo armadura de 10 elementos.....	139
Figura 44. Comparación de frente de Pareto envolvente (este trabajo) con la solución de referencia (G. G. Tejani, Pholdee, et al., 2018) en el problema de optimización de tamaño multiobjetivo de la armadura de 10 elementos.....	140
Figura 45. Estructura base de 10 elementos para optimización topológica multiobjetivo .....	141
Figura 46. Frentes de Pareto en cada una de las 20 ejecuciones, para cada algoritmo en problema de optimización topológica multiobjetivo estructura base de 10 elementos.....	141
Figura 47. Comparación de frente de Pareto envolvente (este trabajo) con la solución de referencia (Su et al., 2011) en el problema de optimización topológica multiobjetivo de la estructura base de 10 elementos .....	142
Figura 48. Topologías del frente de Pareto envolvente para problema de optimización topológica multiobjetivo estructura base de 10 elementos .....	142
Figura 49. Estructura base de 14 elementos para optimización topológica multiobjetivo .....	144
Figura 50. Frentes de Pareto en cada una de las 20 ejecuciones, para cada algoritmo en problema de optimización topológica multiobjetivo estructura base de 14 elementos.....	144
Figura 51. Comparación de frente de Pareto envolvente (este trabajo) con la solución de referencia (Richardson et al., 2012) en el problema de optimización topológica multiobjetivo estructura base de 14 elementos.....	145
Figura 52. Topologías del frente de Pareto envolvente (este trabajo) y del frente de Pareto de referencia (Richardson et al., 2012) para problema de optimización topológica multiobjetivo estructura base de 14 elementos.....	145
Figura 53. Estructura base de 15 elementos para optimización topológica multiobjetivo .....	146
Figura 54. Frentes de Pareto en cada una de las 20 ejecuciones, para cada algoritmo en problema de optimización topológica multiobjetivo estructura base de 14 elementos.....	147
Figura 55. Comparación de frente de Pareto envolvente (este trabajo) con la solución de referencia (Su et al., 2011) en el problema de optimización topológica multiobjetivo de la estructura base de 15 elementos .....	147

Figura 56. Topologías del frente de Pareto envolvente para problema de optimización topológica multiobjetivo estructura base de 15 elementos .....	148
Figura 57. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de optimización multiobjetivo de celosías .....	150
Figura 58. Diagramas de cajas para hipervolumen (HV) de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de optimización multiobjetivo de celosías .....	150
Figura 59. Estructura base de 10 elementos y 6 nodos .....	153
Figura 60. Estructura base de 17 elementos y 9 nodos .....	154
Figura 61. Estructura base de 39 elementos y 12 nodos .....	155
Figura 62. Estructura base de 54 elementos y 15 nodos .....	156
Figura 63. Soluciones decodificadas del Frente de Pareto envolvente para las 4 estructuras base con el método NCN .....	159
Figura 64. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en las 4 estructuras base generadas con el método NCN .....	159
Figura 65. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en las 4 estructuras base generadas con el método NCN .....	159
Figura 66. Frentes de Pareto de los algoritmos metaheurísticos (NSGA-II, MOPSO y AMOSA) y frente de Pareto envolvente para el problema de la estructura base de 17 elementos .....	162
Figura 67. Generación de las estructuras base para la viga cantiléver 1 .....	164
Figura 68. Soluciones decodificadas del Frente de Pareto envolvente de las dos estructuras base para la viga cantiléver 1 .....	166
Figura 69. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 1 .....	166
Figura 70. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 1 .....	166
Figura 71. Generación de las estructuras base para la viga cantiléver 2 .....	167
Figura 72. Soluciones decodificadas del Frente de Pareto envolvente de las dos estructuras base para la viga cantiléver 2 .....	169
Figura 73. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 2 .....	169

Figura 74. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 2.....	169
Figura 75. Generación de las estructuras base para la viga simplemente apoyada.....	170
Figura 76. Soluciones decodificadas del Frente de Pareto envolvente de las dos estructuras base para la viga simplemente apoyada .....	172
Figura 77. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la viga simplemente apoyada .....	172
Figura 78. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver simplemente apoyada .....	172
Figura 79. Generación de las estructuras base para la viga cantiléver 3.....	173
Figura 80. Soluciones decodificadas del Frente de Pareto envolvente de las dos estructuras base para la viga cantiléver 3 .....	175
Figura 81. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 3 .....	175
Figura 82. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 3.....	175
Figura 83. Torre de transmisión plana de 47 barras.....	183
Figura 84. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización de forma de la torre plana de 47 elementos.....	187
Figura 85. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización de forma de la torre plana de 47 elementos .....	187
Figura 86. Soluciones decodificadas del frente de Pareto envolvente de la estructura base de 101 elementos .....	194
Figura 87. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 101 elementos	194
Figura 88. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 101 elementos.....	195
Figura 89. Soluciones decodificadas del frente de Pareto envolvente de la estructura base de 85 elementos .....	196
Figura 90. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 85 elementos ..	197

Figura 91. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 85 elementos.....	197
Figura 92. Soluciones decodificadas del frente de Pareto envolvente de la estructura base de 88 elementos .....	199
Figura 93. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 88 elementos ..	199
Figura 94. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 88 elementos.....	199

**Lista de tablas**

Tabla 1. Ejemplos donde optimización de forma puede ser aplicada.....	55
Tabla 2. Ejemplos donde la optimización topológica puede ser aplicada .....	57
Tabla 3. Parámetros empleados en el algoritmo NSGA-II .....	106
Tabla 4. Parámetros empleados en el algoritmo MOPSO .....	106
Tabla 5. Parámetros empleados en el algoritmo AMOSA.....	106
Tabla 6. Problemas de funciones matemáticas sin restricciones .....	107
Tabla 7. Criterio estadístico de Wilcoxon para GD y $\Delta$ para problemas de funciones matemáticas sin restricciones.....	116
Tabla 8. Problemas de funciones matemáticas con restricciones .....	118
Tabla 9. Criterio estadístico de Wilcoxon para HV para problemas de funciones matemáticas con restricciones .....	125
Tabla 10. Problemas de generación de la estructura base.....	129
Tabla 11. Coordenadas.....	133
Tabla 12. Conectividad .....	133
Tabla 13. Reacciones de la armadura de 7 elementos .....	133
Tabla 14. Desplazamientos de la armadura de 7 elementos .....	134
Tabla 15. Fuerzas internas de la armadura de 7 elementos.....	134
Tabla 16. Coordenadas de los nodos.....	135
Tabla 17. Conectividad de los elementos .....	135
Tabla 18. Reacciones de la armadura de 26 elementos .....	135
Tabla 19. Desplazamientos de la armadura de 26 elementos .....	135
Tabla 20. Fuerzas internas de la armadura de 26 elementos.....	136
Tabla 21. Desempeño del algoritmo AOTMOH en las 4 estructuras base generadas usando el método NCN .....	156
Tabla 22. Desempeño del algoritmo AOTMOH en la viga cantiléver 1 .....	164
Tabla 23. Desempeño del algoritmo AOTMOH en la viga cantiléver 2 .....	167
Tabla 24. Desempeño del algoritmo AOTMOH en la viga simplemente apoyada .....	171
Tabla 25. Desempeño del algoritmo AOTMOH en la viga cantiléver 3 .....	174

Tabla 26. Comparación de las soluciones extremas de las Secciones 6.2 y 6.3 .....	179
Tabla 27. Grupos de variables .....	<b>¡Error! Marcador no definido.</b>
Tabla 28. Desempeño del algoritmo AOTMOH en la optimización de tamaño de la torre plana de 47 elementos .....	186
Tabla 29. Comparación de desempeño para el problema de optimización de tamaño de la torre plana de 47 elementos.....	188
Tabla 30. Construcción de la estructura base para los 3 casos de carga de la torre plana con el método TEM.....	190
Tabla 31. Desempeño del algoritmo AOTMOH en la optimización de la estructura base 101 elementos .....	193
Tabla 32. Desempeño del algoritmo AOTMOH en la optimización de la estructura base 85 elementos .....	195
Tabla 33. Desempeño del algoritmo AOTMOH en la optimización de la estructura base 88 elementos .....	197

**Lista de Apéndices**

Apéndice A. Algoritmos de optimización metaheurísticos .....	225
Apéndice B. Trayectorias de esfuerzo .....	246
Apéndice C. Fundamentos del análisis y diseño estructural de celosías .....	272
Apéndice D. Descripción del contenido de la carpeta de anexos .....	283

**(Ver Apéndices adjunto en el CD y pueden visualizarlos en la Base de Datos de la Biblioteca)**

- Apéndice 1. Estrategia de Generación Estructural
- Apéndice 2. Algoritmos metaheurísticos
- Apéndice 3. Interfaz Grasshopper
- Apéndice 4. Interfaz Grasshoppercode
- Apéndice 5. Algoritmo Análisis Estructural

## Resumen

**Título:** Optimización topológica de estructuras de celosía\*

**Autor:** Luis Humberto Niño Álvarez\*\*

**Palabras clave:** Optimización topológica, celosía, multiobjetivo, algoritmos metaheurísticos, trayectorias de esfuerzo.

### Descripción:

En este trabajo se desarrolla un proceso de optimización topológica multiobjetivo (denominado algoritmo AOTMOH) que minimiza el peso y la energía de deformación (de forma simultánea) de estructuras de celosía (armaduras o cerchas) planas. Para generar la estructura base se usan los métodos NCN (niveles de conectividad de nodos) y TEM (trayectorias de esfuerzo y macroelementos). Como herramientas de optimización se utilizan los algoritmos metaheurísticos NSGA-II (Non-dominated Sorting Genetic Algorithm II), MOPSO (Multiobjective Particle Swarm Optimization) y AMOSA (Archived Multiobjective Simulated Annealing), para generar un frente de Pareto óptimo único (denominado frente de Pareto envolvente) con las mejores soluciones no dominadas proporcionadas por los tres algoritmos.

El proceso de optimización está implementado en el software MATLAB®. Los resultados obtenidos indican que el proceso de optimización desarrollado puede ser de gran utilidad y tener un impacto significativo, contribuyendo al progreso y desarrollo de metodologías de optimización topológica, cuya aplicación en el de diseño de estructuras de celosía (ej. torres de transmisión) puede disminuir los costos de diseño, haciendo un uso más eficiente de materiales, mejorar el desempeño estructural, automatizar el proceso de diseño con tiempos de cómputo razonables, y generar geometrías innovadoras para este tipo de estructuras, de gran importancia para la ingeniería civil.

---

\* Trabajo de grado

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Civil. Director: Oscar Javier Begambre Carrillo, Ph.D. Codirector: Leonardo Moreno De Luca, M.Sc.

**Abstract**

**Title:** Topology optimization of truss structures\*

**Author:** Luis Humberto Niño Álvarez\*\*

**Keywords:** Topology optimization, Truss, Multiobjective, Metaheuristic algorithms, Stress trajectories.

**Description:**

In this work, a multiobjective topology optimization procedure (called AOTMOH algorithm) that minimizes the weight and strain energy (simultaneously) of planar truss structures is developed. To generate the ground structure, the NCN (node connectivity levels) and TEM (stress trajectories and macroelements) methods are used. The metaheuristic algorithms NSGA-II (Non-dominated Sorting Genetic Algorithm II), MOPSO (Multiobjective Particle Swarm Optimization) and AMOSA (Archived Multiobjective Simulated Annealing) are used as optimization tools, to generate a single optimal Pareto front (called envelope Pareto front) with the best non-dominated solutions given by the three algorithms.

The optimization procedure is implemented in MATLAB® software. The results obtained suggest that the developed optimization procedure can be very useful and have a significant impact, thus contributing to the progress and development of topology optimization methodologies, whose application in the design of trusses structures (e.g. transmission towers) can reduce design costs, making a more efficient use of materials, improving structural performance, automating the design process with reasonable computing times, and generating innovative geometries for this type of structures, of great importance for civil engineering.

---

\* Degree work

\*\* Faculty of Physical Mechanical Engineering. School of Civil Engineering. Advisor: Oscar Javier Begambre Carrillo, Ph.D. Coadvisor: Leonardo Moreno De Luca, M.Sc.

## Introducción

El diseño es un proceso iterativo que implica analizar varias pruebas (o posibles soluciones) hasta obtener un diseño aceptable (Arora, 2017a). En la ingeniería civil este proceso tradicionalmente se basa en la experiencia, destreza y la intuición del diseñador, donde un modelo es creado, probado y actualizado a través de una serie de análisis estructurales con el objetivo de obtener un diseño que cumpla con las restricciones técnicas y económicas dadas (Descamps & Coelho, 2013). Este proceso consume mucho tiempo, puede volverse complejo e incluso descartarse para muchas estructuras, especialmente de gran escala (cientos de elementos (Cao, Qian, & Zhou, 2018; Ali Kaveh & Ilchi Ghazaan, 2018)), al ser muy difícil o imposible, para el diseñador, probar cada una de las combinaciones posibles que le permitan obtener la mejor opción (o alguna cercana), limitándose a realizar unas pocas iteraciones, para obtener un diseño satisfactorio que cumpla con los criterios normativos y demás requisitos establecidos. Por lo tanto, la aplicación de conceptos de optimización se convierte en una herramienta de apoyo muy importante en el diseño estructural, haciendo posible obtener soluciones óptimas (mejores soluciones) con tiempos y recursos razonables usando métodos de optimización (Saka, Hasaengebi, & Geem, 2016).

La optimización estructural es un proceso que permite realizar una distribución eficiente de los materiales en una estructura, con el fin de soportar y transmitir las cargas aplicadas. Los problemas de optimización estructural son modelados mediante funciones matemáticas (costo de construcción, peso estructural o rigidez, entre otros), que se deben minimizar (o maximizar) y que representan criterios técnicos (estructurales) usados para evaluar la calidad de la solución de un diseño, bajo ciertas limitaciones dadas (como exigencias de seguridad y requerimientos de servicio establecidos en las normas de diseño) denominadas restricciones (Saka et al., 2016). El tema ha

despertado gran interés dentro de la ingeniería civil y se ha convertido en objeto de investigación de gran importancia en los últimos años (Assimi, Jamali, & Nariman-zadeh, 2017).

En muchos de los problemas de optimización estructural, la formulación matemática del proceso de diseño incluye funciones objetivo y/o restricciones no diferenciables, no lineales, con un espacio de búsqueda no convexo, múltiples óptimos locales (multimodalidad) y una alta cantidad de variables (continuas y discretas), condiciones en las cuales es muy difícil o imposible encontrar soluciones por medio de los algoritmos clásicos basados en gradiente (Kanarachos, Griffin, & Fitzpatrick, 2017; Zadeh & Shirazi, 2013). En este caso, los algoritmos metaheurísticos se convierten en una herramienta de gran utilidad, ya que permiten abordar el problema, encontrando una solución aproximada solamente con la evaluación de las funciones objetivo, sin requerir de una relación explícita con las restricciones, aunque generalmente tienen una tasa de convergencia más lenta y un mayor costo computacional (en comparación con los algoritmos clásicos) (Du & Swamy, 2016; Hare, Nutini, & Tesfamariam, 2013; Saka et al., 2016). La mayor parte de los estudios recientes de optimización estructural se basan en algoritmos metaheurísticos (Talatahari & Kaveh, 2013). El rápido desarrollo y popularidad de estos algoritmos se basa principalmente en su fácil implementación y la habilidad para localizar soluciones muy cercanas al óptimo global de una manera eficiente, lo que los hace muy convenientes para el manejo de una amplia gama de problemas de optimización (Zadeh & Shirazi, 2013).

Por otro lado, una de las estructuras más populares y ampliamente usadas en la ingeniería civil son las estructuras de celosía (también conocidas como armaduras o cerchas), ya que son versátiles, ligeras, de fácil instalación y se adaptan a diversas configuraciones geométricas (Couceiro et al., 2016). Pueden encontrarse en techos, estadios deportivos, puentes, torres de transmisión de energía eléctrica, entre otras. La aplicación de técnicas de optimización en este tipo de estructuras puede

tener un impacto económico y ambiental significativo, ya que al ser estructuras muchas veces repetitivas (amplia aplicación de un mismo diseño), como suele suceder en una línea de transmisión, donde la cantidad de torres es elevada, los ahorros materiales que puedan obtenerse para una sola estructura, serán de gran importancia (un menor uso de materiales significa menos costo y una menor afectación del medio ambiente) (D'Amico & Pomponi, 2018; de Souza, Fadel Miguel, Lopez, Miguel, & Torii, 2016). En estudios de optimización estructural con algoritmos metaheurísticos aplicados en torres de transmisión de energía eléctrica, se ha logrado obtener un ahorro de materiales que varía entre un 6% y 12% (Couceiro et al., 2016; de Souza et al., 2016; Tort, Şahin, & Hasançebi, 2017), mostrando su gran utilidad a la hora de apoyar procesos de diseño estructural.

El uso de estructuras de celosía en Colombia es significativo, particularmente en el sector energético (Unidad de Planeación Minero Energética [UPME], 2018), aunque también se emplean en plantas industriales y grandes superficies. El “Plan Estratégico de Ciencia, Tecnología e Innovación en Energía y Minería 2013-2022”, elaborado por el Departamento Administrativo de Ciencia, Tecnología e Innovación (Colciencias, 2013), establece dentro de sus líneas de acción, el apoyo a investigaciones que contribuyan a la modernización de la red de transmisión y distribución de energía eléctrica (incluyendo la infraestructura), con el propósito de satisfacer las necesidades internas y que permitan posicionar a Colombia como un clúster energético en la región. En este caso, la aplicación de un proceso de optimización en las actividades de diseño de las estructuras que soportan las redes de transmisión de energía eléctrica (las torres de celosía, ver Figura 1), puede ayudar a contribuir con esta línea de acción, ya que el uso de este proceso podría representar, a nivel nacional, ahorros significativos en el costo total de las líneas de transmisión (menor uso de

materiales, disminuyendo costos asociados a la fabricación, transporte e instalación), lo que impactaría positivamente la economía del país.



Figura 1. Torres de transmisión de energía eléctrica  
Adaptado: <https://unsplash.com/s/photos/electricity>

El precio de las torres de celosía (como las mostradas en la Figura 1) oscila entre \$US1100-1500/Ton (de acuerdo al sitio web chino *alibaba.com*), para torres entre 110 y 500 kV. En el estudio de optimización de torres de energía realizado por Tort et al., (2017), obtuvieron un ahorro aproximado en peso de materiales del 12% (en comparación con el diseño convencional), para las dos torres optimizadas (de 110 kV y 400 kV) usando el algoritmo metaheurístico SA (*Simulated Annealing*). Suponiendo un costo medio de \$US1300/Ton, este porcentaje representó un ahorro aproximado de US\$2000 para la torre de 110 kV y de \$US5000 para la torre de 400 kV. Teniendo en cuenta que la separación media entre torres para una línea de transmisión es de 300-500 m (Alzate Acevedo, 2016), el ahorro en costos, aplicando un proceso de optimización, es de \$US6000 por kilómetro (considerando una red con torres de 110 kV cada 330 m con las características presentadas en el estudio de Tort et al., (2017)), lo cual es muy significativo cuando se plantea la construcción de una línea de transmisión de cientos de kilómetros.

Gran parte de los problemas de optimización estructural de celosías se han solucionado usando solamente una función objetivo (UFO), usualmente el peso estructural (Assimi & Jamali, 2018;

Assimi et al., 2017; Degertekin, Lamberti, & Ugur, 2017; Frans & Arfiadi, 2014; G. G. Tejani, Savsani, Patel, & Savsani, 2018). Sin embargo, en aplicaciones prácticas el enfoque UFO rara vez proporciona una medida representativa del desempeño estructural, ya que en realidad cuando un diseñador busca el diseño óptimo de una estructura se enfrenta a la pregunta de seleccionar las funciones objetivo más convenientes para medir la economía, la resistencia, la capacidad de servicio y otros factores que afectan el desempeño de una estructura. Una cantidad que tiene una influencia directa en el desempeño estructural puede considerarse como una función objetivo, cumpliendo algunas restricciones de diseño, definiendo límites inferiores y superiores. Cuando es difícil seleccionar los límites de las restricciones, entonces pueden ser tratadas como funciones objetivo independientes (Mitropoulou, Fourkiotis, Lagaros, & Karlaftis, 2013). Para la solución de este tipo de problemas, los algoritmos metaheurísticos multiobjetivo son una opción atractiva, ya que presentan un buen desempeño (cálculo directo del frente de Pareto en una ejecución) en este tipo de problemas y proporcionan soluciones satisfactorias (Noilublao & Bureerat, 2011). Su uso puede ayudar a ejecutar los cálculos de diseño de una manera automática, eficiente y rápida (en comparación con el proceso de diseño tradicional, en el cual el proceso de optimización de un diseño es un proceso iterativo-intuitivo que depende en gran parte de la experiencia del diseñador), obteniendo una amplia cantidad de diseños óptimos confiables (Arora, 2017a; Christensen & Klarbring, 2009).

En las estructuras de celosía pueden aplicarse tres tipos de optimización (Röber & Todtermuschke, 2014): La *optimización de tamaño* que busca el valor óptimo de las áreas de los elementos para una estructura con una configuración previamente definida; la *optimización de forma* que busca la posición óptima de los nodos de una estructura con una configuración definida (al igual que en la optimización de tamaño); y la *optimización topológica* discreta que busca el

patrón de conectividad óptimo de los nodos o la secuencia espacial de los elementos de una estructura que contiene todas las posibles configuraciones, permitiendo la presencia o remoción de elementos específicos.

La *optimización topológica* es el problema más general, ya que trata con todas las posibles configuraciones geométricas generadas, en vez de una sola en particular (en el caso de la *optimización de tamaño y de forma*), y aplicarla puede resultar en un mayor ahorro de materiales, al estar buscando potencialmente la mejor topología (G. G. Tejani, Savsani, et al., 2018).

Los estudios de *optimización topológica* de estructuras de celosía se ha abordado principalmente con el enfoque de la *estructura base* (Miguel, Lopez, & Miguel, 2013; Mortazavi & Toğan, 2016; G. G. Tejani, Savsani, Bureerat, Patel, & Savsani, 2019), en el cual la estructura inicial contiene un número de elementos posibles, de acuerdo a las potenciales conexiones entre los nodos que conforman el espacio de búsqueda (Bendsøe & Sigmund, 2004). Este enfoque está basado en un proceso de reducción, donde elementos y nodos son eliminados de la estructura inicial hasta encontrar la mejor solución posible (Rozvany, 1997). Como la *estructura base* tiene una gran influencia en la topología final de la estructura, el método usado para generarla es muy importante. Es probable que un buen resultado de optimización topológica, pueda lograrse mediante la construcción de una *estructura base* con una cantidad moderada de nodos en posiciones razonables, aunque aún no se han propuesto criterios o reglas que permitan juzgar su desempeño. Si un método puede generar una *estructura base* bien definida con solo pocos nodos y barras, esto puede ayudar a obtener una buena topología como solución y disminuir el costo computacional, lo cual es muy significativo para su aplicación en problemas prácticos, donde el número de variables es elevado (Gao, Liu, Li, & Qiao, 2017).

En este contexto, una vez realizada una búsqueda del tema optimización topológica de celosías<sup>1</sup> (truss topology optimization) en bases de datos como SCOPUS (ver Figura 2), se encontró que el número de estudios publicados en el periodo comprendido entre 2000-2018, a pesar de haber aumentado de manera significativa en los últimos años, son aún muy pocos, en comparación con otras disciplinas, lo que abre la posibilidad de desarrollar nuevos estudios que contribuyan al fortalecimiento de esta área de investigación de gran interés.

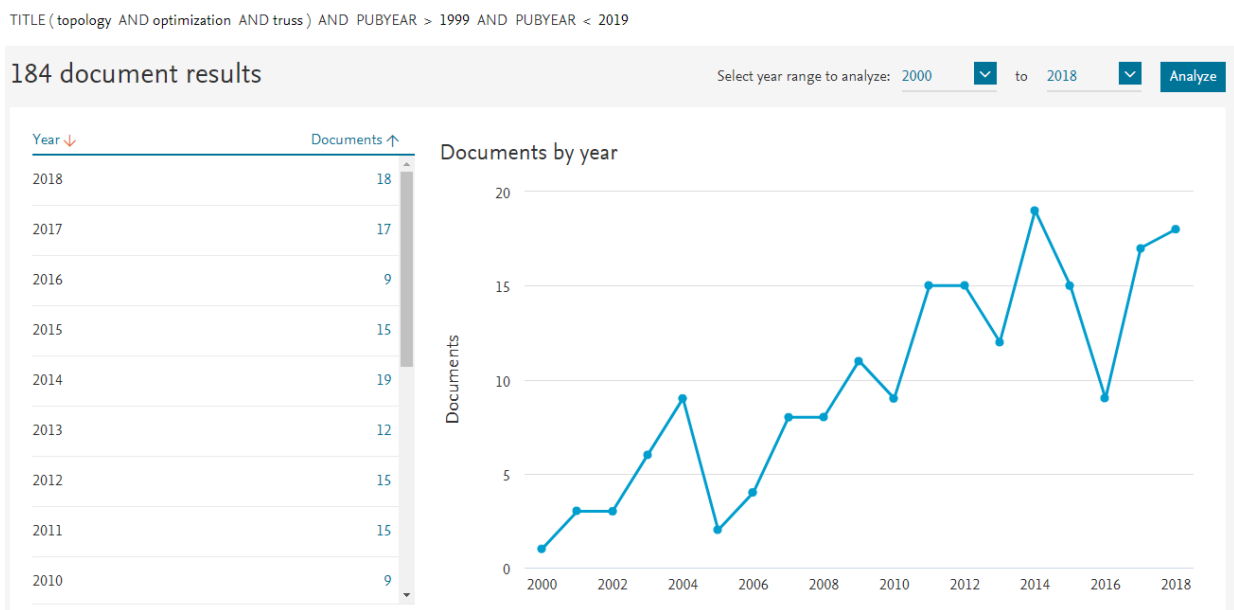


Figura 2. Número de documento publicados por año en el tema de investigación  
Adaptado: SCOPUS®

El desarrollo de estrategias optimización topológica es un área de investigación aún muy activa, ya que las metodologías disponibles, tienen un uso limitado para su aplicación directa en problemas de diseño práctico, principalmente por el gran número de variables, y se requiere el uso de algoritmos de optimización que permitan un alto nivel de exploración del espacio de búsqueda. No obstante, su uso como herramienta de apoyo en las actividades preliminares de diseño puede resultar en un ahorro significativo de recursos y mejoras en el desempeño estructural (G. Tejani, Savsani, & Bureerat, 2018).

<sup>1</sup> El término celosía (armadura o cercha) se puede traducir al idioma inglés como *truss* o *lattice*

Adicionalmente, realizando una consulta de información relacionada con la existencia de paquetes de software que incluyen optimización topológica, se encontraron herramientas de uso libre como TopOpt3D (Technical University of Denmark [DTU], 2014) y OpenLSTO (Multiscale Multiphysics Design Optimization Laboratory [M2DO], 2018). En la parte comercial se cuenta con softwares de uso propietario como Abaqus (Dassault Systèmes [3DS], 2018a), Tosca (Dassault Systèmes [3DS], 2018b), ANSYS (ANSYS Inc., 2018), MSC Nastran (MSC Software Corporation, 2018), Altair OptiStruct (Altair Engineering Inc., 2019) y midas NFX (MIDAS Information Technology Co. Ltd. [MIDAS IT], 2018).

La mayor parte de los programas mencionados anteriormente se basan en el enfoque de optimización topológica continua, donde a partir de un espacio de diseño continuo (lleno), se va removiendo material, hasta obtener una estructura óptima. Este enfoque es usualmente aplicado en áreas como la ingeniería mecánica y aeronáutica para optimizar el diseño de piezas mecánicas y estructuras en máquinas, vehículos y aeronaves (W. Zhang, Zhu, & Gao, 2016). Para estructuras comúnmente utilizadas en la ingeniería civil como las estructuras de celosía, no se encontró un software que permita aplicar la optimización topológica discreta (caso aplicable a este tipo de estructuras). Los programas de uso comercial comúnmente utilizados en el diseño de este tipo de estructuras como SAP2000 (Computers and Structures Inc. [CSI], 2018), aún no cuentan con un módulo de optimización topológica (discreta) para realizar este proceso.

Ante la situación presentada, la pregunta que se busca responder en esta investigación es: ¿Qué mejoras en el desempeño estructural (medido en términos del peso y la energía de deformación) y en el diseño (automatización del procedimiento en la fase conceptual, preliminar y definitiva) de estructuras de celosía, pueden lograrse con la aplicación de un proceso de optimización topológica multiobjetivo? La hipótesis planteada es que la aplicación de un proceso de optimización

topológica multiobjetivo, influye de manera significativa en el desempeño estructural (medido con relación a un diseño inicial satisfactorio) y en la automatización del procedimiento de diseño de estructuras de celosía.

Teniendo en cuenta lo anterior, en este trabajo se desarrolló un proceso de optimización topológica multiobjetivo aplicable a estructuras de celosía planas. Los resultados obtenidos mediante la aplicación de este procedimiento pueden tener un impacto significativo, contribuyendo al progreso y desarrollo de metodologías de optimización topológica, cuya aplicación en el proceso de diseño de estructuras de celosía (torres, armaduras, cerchas, entre otras) puede disminuir los costos de diseño, haciendo un uso más eficiente de materiales, mejorar el desempeño estructural, automatizar el proceso y generar geometrías innovadoras para este tipo de estructuras, de gran importancia para la ingeniería civil.

Finalmente, este trabajo está organizado de la siguiente manera. En primer lugar, se presentan los objetivos establecidos en la propuesta de investigación (Capítulo 1). Seguidamente, se presenta la revisión del estado del arte de optimización topológica en estructuras de celosía en el Capítulo 2. En el Capítulo 3 se abordan los fundamentos teóricos de optimización requeridos en este trabajo. En Capítulo 4 se propone el algoritmo AOTMOH (Algoritmo de Optimización Topológica Multiobjetivo Híbrido) y se explica su funcionamiento. En el Capítulo 5, los algoritmos de optimización metaheurísticos, de análisis estructural y de generación de la estructura base son validados con problemas de referencia. En el Capítulo 6 se muestran aplicaciones del AOTMOH en diversos problemas de estructuras de celosía planas y por último se realizan las conclusiones de este trabajo (Capítulo 7).

## **1. Objetivos**

### **1.1. Objetivo general**

Desarrollar un proceso de optimización topológica, por medio del uso de un algoritmo metaheurístico multiobjetivo, para minimizar el peso y la energía de deformación de estructuras de celosía planas.

### **1.2. Objetivos específicos**

Seleccionar el algoritmo metaheurístico multiobjetivo más conveniente, como método de optimización estructural.

Plantear una estrategia automática e interactiva de generación geométrica de la estructura base.  
Integrar el método de análisis estructural de las celosías, el algoritmo de optimización y la estrategia de generación geométrica de la estructura base.

Incorporar una interfaz gráfica, que permita representar de forma interactiva los resultados del proceso de optimización.

## 2. Aplicaciones recientes de optimización topológica en estructuras de celosía: revisión de literatura

La optimización estructural se ha convertido en un tema de investigación de gran importancia en los últimos años, ya que a través de este proceso es posible obtener un desempeño eficiente de las estructuras y un máximo provecho de los recursos disponibles (materiales y económicos) (Assimi et al., 2017). En estructuras de celosía (armaduras o cerchas) los problemas de optimización se han abordado usando técnicas de optimización que incluyen algoritmos clásicos y algoritmos metaheurísticos <sup>2</sup>.

En el estudio realizado por Miguel, Lopez, & Miguel (2013) se aplicó el algoritmo FA (Firefly Algorithm) en la optimización simultánea de tamaño, forma y topológica de estructuras de celosía, minimizando el peso estructural. Los autores obtuvieron múltiples soluciones óptimas al final de la optimización gracias a una característica especial del algoritmo denominada optimización multimodal, permitiendo su evaluación con base en criterios estéticos y constructivos por parte del diseñador.

Por su parte, Assimi et al. (2017) realizaron la optimización simultánea de tamaño y topológica en estructuras de celosía usando el algoritmo GP (Genetic Programming). Los resultados mostraron mejoras entre el 0.01% y el 7.38% en comparación con algoritmos populares como GA (Genetic Algorithm), PSO (Particle Swarm Optimization), HS (Harmony Search) y HPSO (Heuristic Particle Swarm Optimization).

---

<sup>2</sup> **Metaheurística:** La palabra “*meta*” significa *más allá, más alto o nivel superior*, y la palabra “*heurística*” denota el arte de descubrir nuevas estrategias (reglas) para resolver problemas (Talbi, 2009), entonces una “*metaheurística*” se puede definir como el arte de descubrir una metodología general independiente del problema, que proporcione una serie de reglas para resolver problemas de optimización (G. G. Tejani, Savsani, et al., 2018).

Frans & Arfiadi (2014) emplearon el algoritmo GA (Genetic Algorithm), usando codificación híbrida en la optimización simultánea de tamaño, forma y topología de cerchas de techo, minimizando el peso estructural. Usaron codificación binaria para el manejo de las variables de tamaño y codificación real para las variables de forma. Cazacu & Grama (2014) usaron el algoritmo GA (Genetic Algorithm) en la optimización simultánea de tamaño, forma y topología en estructuras de celosía, usando una técnica de codificación por bloques.

### **2.1. Combinación de algoritmos metaheurísticos con algoritmos clásicos y técnicas de mejora local**

Algunos problemas de optimización de estructuras de celosía, se han tratado aplicando metodologías basadas en algoritmos clásicos, con el fin de disminuir el costo computacional que pueden generar los algoritmos metaheurísticos y mejorando su convergencia. Faramarzi & Afshar (2012) desarrollaron una metodología que incluye el uso de CA (*Cellular Automata*), y un método basado en programación lineal LP (*Linear Programming*), para la optimización de tamaño y topología de estructuras de celosía. Los resultados de desempeño del algoritmo propuesto en ejemplos de referencia muestran su efectividad, logrando similares niveles en la calidad de la optimización, comparado con GA, con un menor costo computacional.

Descamps & Filomeno Coelho (2014) desarrollaron un método computacionalmente eficiente, para la formulación de optimización simultánea de forma y de topología en estructuras de celosía, usando para ello el método de la fuerza nominal, adaptado a la geometría y el método de punto interior con programación cuadrática secuencial como herramienta de optimización. Con esta formulación obtuvieron soluciones que pueden ser consideradas como candidatas en los diseños preliminares, necesitando un pos procesamiento de acuerdo a los requerimientos de los códigos de diseño.

Rahami, Kaveh, & Gholipour (2008) mejoraron el desempeño del algoritmo GA (Genetic Algorithm) en el proceso de optimización de tamaño, forma y topología de estructuras de celosía, para minimizar el peso estructural, usando una formulación en términos de conceptos energéticos, mediante la aplicación del método de la fuerza con un enfoque de energía para el análisis estructural, lo cual reduce el número de variables de entrada en el proceso de optimización por medio del algoritmo GA.

Assimi & Jamali (2018) aplicaron el algoritmo GP (Genetic Programming) en la optimización simultánea de tamaño y topología de estructuras de celosía, con variables discretas y continuas, integrando el algoritmo GP con el algoritmo de Nelder-Mead (método clásico), para mejorar la búsqueda local, la competencia y la convergencia. Los resultados de aplicación indicaron un buen desempeño del algoritmo híbrido generado HGP (Hybrid Genetic Programming), siendo similar e incluso mejor, en comparación con los algoritmos PSO (Particle Swarm Optimization), TLBO (Teaching Learning Based Optimization) y CSS (Charged System Search).

Una forma adicional de mejora de un algoritmo metaheurístico es modificando algunas de sus características, logrando un buen balance entre la exploración y la explotación. En las investigaciones de Savsani et al. (2017); G. G. Tejani et al. (2018) realizaron la optimización de tamaño y topología de estructuras de celosía 2D y 3D. En ambas investigaciones se presentaron modificaciones a los algoritmos metaheurísticos TLBO (Teaching Learning Based Optimization), HTS (Heat Transfer Search), WWO (Water Wave Optimization), y PVS (Passing Vehicle Search), integrando a cada uno de ellos una técnica de búsqueda local aleatoria denominada “random mutation based search”, para mejorar la búsqueda local y global, aumentando la diversidad y mejorando las capacidades de exploración y explotación de los algoritmos originales. Los resultados de optimización y comparación con la prueba estadística de Friedman en problemas de

referencia mostraron que los algoritmos modificados presentan un mejor desempeño que los algoritmos originales.

Gan, Hara, Han, Alisjahbana, & As'Ad (2017) mostraron la aplicación del algoritmo EACO (Evolutionary Ant Colony Optimization) para tratar la optimización de tamaño y topología en cerchas. Para ello combinaron algoritmo ACO (Ant Colony Optimization), con un operador de sensibilidad tomado del método BESO (Bidirectional Evolutionary Structural Optimization), para orientar el proceso de búsqueda en el proceso de optimización, obteniendo rapidez y estabilidad en la convergencia.

Millán Páramo (2013) desarrolló un procedimiento de optimización topológica, empleando técnicas estocásticas (Algoritmo Recocido Simulado Modificado ASAM y optimización con enjambre de partículas PSO) como métodos de optimización de las densidades del elemento. Para evitar problemas de inestabilidades numéricas (tablero de ajedrez, contornos discontinuos, entre otros), usó un filtro de imagen propuesto por Andreassen, Clausen, Schevenels, Lazarov, & Sigmund (2011), para obtener soluciones menos pixeladas, más suavizadas y menos dentadas. Para la validación de este procedimiento, abordó tres problemas reportados en la literatura especializada, (viga simplemente apoyada, viga en voladizo y viga en voladizo con hueco, cada problema fue analizado con tres tipos de malla (números de elementos finitos, sin y con el uso del filtro propuesto, todo esto con el fin de comparar los resultados obtenidos, en cuanto a topologías, valor de energía de deformación y tiempos. A partir de lo anterior, logró establecer que el procedimiento arroja menores tiempos computacionales a medida que se analizan los problemas con mallas refinadas, además, las distribuciones de del material en el dominio de diseño y valores de energía obtenidos, fueron similares a los reportados por la literatura. El autor recomienda incorporar más estados de carga y más tipos de carga aparte de las uniformemente distribuidas.

Un estudio adicional de mejoramiento de un algoritmo metaheurístico es el realizado por Mortazavi & Toğan (2016) en el que presentaron el algoritmo iPSO (Integrated PSO), para tratar la optimización simultánea de tamaño, forma y topología de estructuras de celosía. Este algoritmo integra las ventajas del PSO con un eficiente concepto de “Weight Particle”, el cual ayuda al algoritmo a evitar quedar atrapado en óptimos locales, mejorando su desempeño. Para el manejo de las restricciones usaron una técnica denominada “Improved Fly Back”. Los resultados presentados muestran un desempeño similar a los algoritmos GA (Genetic Algorithm) y FA (Firefly Algorithm), como comprobaron en los ejemplos de referencia aplicado.

## **2.2. Combinación de algoritmos metaheurísticos con otros algoritmos metaheurísticos**

Algunos algoritmos metaheurísticos se han combinado entre sí, para aprovechar sus características y lograr el desarrollo de nuevos algoritmos más eficientes. Un ejemplo de ello se observa en el estudio hecho por Kaveh & Talatahari (2009) para la optimización de estructuras de celosía con variables discretas, en el cual a partir del PSO (Particle Swarm Optimization) y PC (Passive Congregation), forman el algoritmo PSOPC como técnica de optimización global, y como herramientas auxiliares aprovechan las características del ACO (Ant Colony Optimization) para la búsqueda local y HS (Harmony Search) para el manejo de las restricciones, formando así el algoritmo híbrido DHPSACO (Discrete Heuristic Particle Swarm Ant Colony Optimization). Los resultados de desempeño mostraron su efectividad, al compararlo con GA (Genetic Algorithm) y PSO (Particle Swarm Optimization).

En un estudio reciente realizado por G. G. Tejani, Savsani, Bureerat, Patel, & Savsani (2019), se desarrolla la optimización simultánea de tamaño y topología en estructuras de celosía 2D y 3D, usando el algoritmo metaheurístico PVS (Passing Vehicle Search), y versiones modificadas del

PVS integradas con el algoritmo SA (Simulated Annealing). La convergencia de los algoritmos modificados mostró mejores resultados en comparación con el algoritmo original (PVS) y algoritmos populares como PSO (Particle Swarm Optimization), TLBO (Teaching Learning Based Optimization) y CSS (Charged System Search). Estudios similares sobre algoritmos híbridos y su aplicación a estructuras de celosía fueron realizados recientemente por Assimi & Jamali (2018); Cheng, Prayogo, Wu, & Lukito (2016); Kaveh & Mahdavi (2015); Lieu, Do, & Lee (2018); Luh & Lin (2011).

### **2.3. Desarrollo y aplicación de nuevos algoritmos metaheurísticos en optimización topológica de estructura de celosías**

En los años recientes se ha visto un rápido desarrollo de nuevos algoritmos metaheurísticos, ya que sus características permiten una potencial aplicación en un amplio rango de áreas dentro de la comunidad científica (Kar, 2016). Kanarachos et al. (2017) presentaron el algoritmo cm-FOA (Contrast-Based Multi-Parameter Fruit Fly Optimization Algorithm), un algoritmo que presenta pocos parámetros de ajuste. Validaron su funcionamiento en funciones matemáticas de referencia y por primera vez hicieron su aplicación en la optimización de tamaño y forma de estructuras de celosía, usando variables discretas y continuas, logrando un desempeño similar a los algoritmos GA (Genetic Algorithm), SA (Simulated Annealing), PSO (Particle Swarm Optimization) y TLBO (Teaching Learning Based Optimization). Gonçalves et al. (2015) mostraron el nuevo algoritmo SGA (Search Group Algorithm), y lo aplicaron en la optimización simultánea de tamaño, forma y topología de estructuras de celosía comunes en la literatura, mostrando mejores resultados en 5 de las 6 pruebas presentadas, siendo comparado su desempeño con algoritmos como PSO y CSS (Charged System Search).

De igual manera Degertekin, Lamberti, & Ugur (2017) hicieron la primera aplicación del algoritmo JA (Jaya Algorithm). En un detallado estudio, evaluaron la conveniencia de su aplicación en la optimización de tamaño forma y topología de estructuras de celosía. Compararon su rendimiento con algoritmos conocidos por su buen desempeño como GA, SA, PSO y HS. Los autores concluyeron que de acuerdo a las características particulares del algoritmo JA como la sencilla formulación, la no configuración de parámetros, la insensibilidad al tamaño de la población y la buena convergencia (similar a la de algoritmos basados en gradiente) y el requerimiento de un menor número de análisis estructurales, lo convierten en una buena opción para la solución de diversos problemas de optimización estructural.

Dentro de la amplia gama de algoritmos metaheurísticos que han surgido recientemente se encuentran: RO (Ray Optimization) (A. Kaveh & Khayatazad, 2013), ADS (Adaptive Dimensión Search) (Hasançebi & Azad, 2015), IGCMEA (Intelligent Garbage Can Decision-Making Model Evolution Algorithm) (Kuo, Chiu, & Lin, 2012) y FPA (Flower Pollination Algorithm) (Bekdaş, Nigdeli, & Yang, 2015).

El rápido desarrollo de los algoritmos metaheurísticos ha hecho muy difícil su seguimiento, ya que diferentes algoritmos se están introduciendo frecuentemente, y pocos estudios han identificado estos algoritmos de forma exhaustiva, explorando y comparando su capacidad en diferentes problemas, además con la limitada visibilidad entre algoritmos, los nuevos investigadores del área tienden a enfocarse en una gama limitada, que comprende principalmente los algoritmos más populares (Kar, 2016). además muchos de ellos dependen de parámetros de ajuste que no son conocidos previamente (en la actualidad existe la necesidad de tener algoritmos metaheurísticos en los cuales se utilice pocos parámetros) (Kanarachos et al., 2017).

#### **2.4. Aplicaciones reales de optimización estructural de celosías, mediante algoritmos metaheurísticos**

La optimización de estructuras de celosía (ej. torres de transmisión de energía) es una tarea difícil debido al gran número de variables que se incluyen en el diseño. A pesar del desarrollo teórico y las nuevas técnicas surgidas en el área de la optimización estructural su aplicación en la práctica es muy limitada debido principalmente a: a) Los algoritmos metaheurísticos pueden cumplir los requisitos de diseño, gracias a su fácil implementación, pero pueden el requerir un gran costo computacional en estructuras a gran escala sujetas a numerosos casos de carga. b) No se ha exhibido suficientemente la capacidad, el potencial y la utilidad de estos algoritmos en problemas reales de diseño, ya que en muchos de estudios presentados solo se prueba su eficiencia en algunos ejemplos de referencia tomados de la literatura, utilizando pocos casos de carga y omitiendo criterios de diseño, incluso si se aplica un código de diseño práctico. Esto hace que un diseñador no vea su utilidad a pesar de los resultados presentados. Se requiere de un mayor trabajo y esfuerzo para mostrar su potencial y extender las herramientas de la optimización estructural a nivel industrial (Tort et al., 2017).

En el estudio realizado por de Souza, Fadel Miguel, Lopez, Miguel, & Torii (2016), desarrollaron un procedimiento de optimización en estructuras de torres de transmisión, sujeto a la norma de diseño ASCE 10-97. Usaron dos algoritmos: FA (Firefly Algorithm) y BSA (Backtracing Search Algorithm). Aplicaron el proceso desarrollado en la optimización de tamaño, forma y topología (considerando módulos o bloques preestablecido de acuerdo a criterios constructivos), en dos torres de transmisión con diferentes casos de carga. Con la aplicación de los algoritmos, a nivel general el peso estructural para los dos problemas disminuyó un 6.4%, comparado con la metodología tradicional de diseño. Los autores señalan que para la ejecución de

la optimización se requirió alrededor de 12 h utilizando un equipo portátil con características normales, no obstante, el mismo diseño realizado por un ingeniero senior en el área tomaría alrededor de 1 semana, lo cual muestra la importancia de la aplicación de las técnicas de optimización estructural en problemas reales que se presentan en la industria.

Alzate Acevedo (2016) realizó el diseño conceptual de torres de transmisión de energía, usando el método de optimización topológica (MOT). Combinó el método de solución numérica (método de los elementos finitos (MEF)), el algoritmo de optimización PLS (Programación Lineal Secuencial) y el modelo SIMP (Solid Isotropic with Material Penalization) para la discretización del material. La función objetivo fue minimizar el peso de la estructura y se especificaron frecuencias de resonancias deseadas para la estructura. Adicionalmente el autor presentó un análisis de las condiciones de carga a la cual se encuentra sometida la estructura, teniendo como referencia para su cálculo la norma ASCE 74-09.

## **2.5. Aplicaciones de optimización topológica multiobjetivo de estructuras de celosía**

En el diseño de una estructura, generalmente se busca obtener: “mínimo costo y una mayor seguridad” o “mínimo peso y una mayor rigidez”. Estos objetivos son opuestos y generan conflicto entre sí, ya que mejorar uno de ellos implica perjudicar el otro. Las técnicas de optimización multiobjetivo, basadas en algoritmos metaheurísticos se convierten en una gran herramienta para la solución de este tipo de problemas, al ser capaces de manipular funciones objetivo no lineales y no diferenciables de forma simultánea, comúnmente encontrados en la ingeniería estructural (G. R. Zavala, Nebro, Luna, & Coello Coello, 2014).

Dentro de los estudios consultados que tratan el tema, se encuentra el realizado por Pandey, Datta, & Bhattacharya (2017), quienes establecieron una técnica de optimización de topología

constructiva sólida, basada en la geometría (CSG-TOM: Constructive Solid Geometry Based Topology Optimization Method), para el diseño de estructuras y mecanismos. Una de las novedades del método presentado radica en la capacidad para manipular vacíos, restricciones no asociadas al diseño y formas irregulares del contorno del espacio de búsqueda. Aplicaron el algoritmo NSGA-II (Non-dominated Sorting Genetic Algorithm II) como método de optimización. La técnica CSG-TOM fue aplicada satisfactoriamente en la optimización topológica de geometrías 2D y 3D.

En el estudio de Noilublao & Bureerat (2011) se presenta una técnica de diseño integrado, para la optimización simultánea de tamaño, forma y topología en estructuras de celosía tridimensionales. Las funciones objetivo planteadas fueron: *peso estructural*, *rigidez*, *frecuencias naturales*, *función de respuesta de frecuencia (FRF)* y *transmisibilidad de la fuerza (FT)*, agrupadas en cuatro problemas bi-objetivo. Los algoritmos metaheurísticos empleados para la solución de estos cuatro problemas de optimización fueron SPEA2 (Strength Pareto Evolutionary Algorithm 2), PBIL (Population Based Incremental Learning) y AMOSA (Archived Multiobjective Simulated Annealing). De acuerdo a los autores, los resultados obtenidos pueden tener aplicación práctica, si se realiza una fase adicional de diseño detallado, además el concepto de diseño desarrollado en este estudio puede extenderse a otro tipo de estructuras como marcos.

Richardson, Adriaenssens, Bouillard, & Coelho (2012), aplicaron el algoritmo metaheurístico MOGA (Multiobjective Genetic Algorithm), en la optimización topológica de estructuras de celosías, evaluando su convergencia usando un rastreador métrico durante la ejecución del proceso de optimización. Röber & Todtermuschke (2014) desarrollaron un proceso de optimización topológica multiobjetivo, para minimizar el peso y aumentar la rigidez en estructuras de celosía usadas en equipo de ensamblaje automotriz. Usaron una estrategia de dos fases, en la que

inicialmente usando el algoritmo ACO se remueve material innecesario para crear de forma rápida la topología básica de la estructura, la cual fue refinada posteriormente usando PSO, y un Método de Orden No Dominado, similar al NSGA II, para el tratamiento de los dos objetivos.

Hosseini, Hamidi, Mansuri, & Ghoddosian (2015) aplicaron el algoritmo MOPSO (Multiobjective Particle Swarm Optimization) en el problema de optimización de tamaño y forma estructuras de celosía planas, usando como funciones objetivo la minimización del peso y de la deflexión de los nodos. Los autores obtuvieron resultados satisfactorios en comparación con los algoritmos FSD-ES (Fully Stressed Design Evolution Strategy), Tchebycheff Method,  $\epsilon$ -Constraint Method y Weighting Method.

Moreno De Luca (2013) propuso y desarrolló un Procedimiento Auto-Configurado para el Diseño Multi-Objetivo de Estructuras Reticulares (PACDMOER), el cual está enfocado en generar y optimizar la geometría de dichas estructuras, teniendo como objetivos minimizar la energía de deformación, el peso, y optimizar el rendimiento acústico de la estructura. Como método de optimización usó un algoritmo heurístico auto-configurado y multi-objetivo (basado en la combinación del algoritmo PSO, con el algoritmo Unified PSO - UPSO). Como resultado, el procedimiento propuesto fue capaz de alcanzar los objetivos, optimizando significativamente las tres funciones objetivo planteadas, conduciendo un proceso de morfogénesis que resulta en la forma que se aproxima a ser óptima para una estructura reticular en particular.

Recientemente Mokarram & Banan (2018) presentaron un nuevo algoritmo metaheurístico de optimización multiobjetivo, llamado FC-MOPSO (Fast Converging Multiobjective Particle Swarm Optimization), aplicable a problemas de optimización con variables continuas y discretas, incluyendo diferentes tipos de restricciones. El algoritmo propuesto aplica la idea principal del algoritmo PSO original, en el tratamiento de problemas multiobjetivo con o sin restricciones. El

desempeño del algoritmo es validado en cinco problemas de optimización de estructuras de celosía comúnmente encontrados en la literatura, comparando resultados con los algoritmos MOPSO, NSGA-II y SPEA2, encontrando de manera efectiva una aproximación aceptable de los frentes de Pareto con un pequeño número de evaluaciones de las funciones.

La lista de algoritmos metaheurísticos multiobjetivo que han aparecido recientemente y que tienen una potencial aplicación en la optimización topológica de estructuras de celosía incluye: MOFA (Multiobjective Firefly Algorithm) (Yang, 2013), MOSHO (Multiobjective Spotted Hyena Optimizer) (Dhiman & Kumar, 2018), MOCBO (Multiobjective Colliding Bodies Optimization Algorithm) (A. Kaveh & Mahdavi, 2018), MOHTS (Multiobjective Heat Transfer Search) (G. G. Tejani, Kumar, & Gandomi, 2019).

A partir de la revisión de la literatura, se percibe que hay un desarrollo intensivo de nuevos algoritmos metaheurísticos multiobjetivo que pueden aplicarse en optimización topológica. De igual manera, el número de estos algoritmos es considerable. No obstante, la tendencia de desarrollo de algoritmos es proponer uno que supere a los anteriormente propuestos. En este sentido la idea que se propone en este trabajo es combinar las fortalezas de estos algoritmos existentes (con aplicaciones publicadas) para obtener las soluciones óptimas de un problema de optimización multiobjetivo. En este contexto, y de acuerdo búsqueda actualizada realizada en SCOPUS® durante el año 2019, dentro de los algoritmos metaheurísticos más populares para optimización multiobjetivo se encontraron el NSGA-II (Non-dominated Sorting Genetic Algorithm II) con 21170 citaciones, el MOPSO (Multiobjective Particle Swarm Optimization) con 3351 citaciones y el AMOSA (Archived Multiobjective Simulated Annealing) con 474 citaciones.

## **2.6. Interacción del usuario en la optimización estructural**

Finalmente, un aspecto a tener en cuenta para dar mayor visibilidad a la aplicación de los algoritmos metaheurísticos (con un objetivo y multiobjetivo) en el proceso de optimización estructural y que incentive una mayor aplicación en problemas reales, es tener en cuenta criterios estéticos, arquitectónicos y preferencias de usuario, para hacer una evaluación más detallada de los diseños obtenidos (G. R. Zavala et al., 2014).

Un estudio reciente de aplicación de la interacción del usuario en la optimización estructural fue realizado por Felkner, Chatzi, & Kotnik (2015), en el cual incorporaron las curvas NURBS (Non-Uniform Rational Basis Spline) en una herramienta interactiva de diseño semi-automatizado para el diseño de estructuras de celosía, sujeto a criterios objetivos (restricciones estructurales formales) y subjetivos (no cuantificables y dependientes de la perspectiva del usuario, también llamados estéticos), utilizando el algoritmo PSO como herramienta optimizadora. La ventaja de la metodología propuesta por los autores radica en la incorporación de “feedback” en el algoritmo de optimización, y una interfaz amigable que permite visualizar las soluciones candidatas generadas durante el proceso de optimización, lo que puede servir como fuente de inspiración para el diseñador, haciendo posible diferentes soluciones.

## **3. Optimización multiobjetivo y optimización estructural**

La optimización es un proceso esencial que tiene múltiples aplicaciones en la vida diaria de la sociedad (Ngatchou, Zarei, & El-Sharkawi, 2005). Se usa comúnmente en la industria automotriz y aeronáutica, con el fin de lograr un mayor desempeño y eficiencia en productos, herramientas y estructuras. Sin embargo, existe un creciente interés en áreas como la ingeniería civil y arquitectura (Descamps & Coelho, 2013), donde la optimización estructural se ha convertido en un tema de

gran importancia en los últimos años, siendo una de las áreas de investigación más activas (Assimi et al., 2017; A. Kaveh & Talatahari, 2009). En las décadas recientes se ha presentado un rápido desarrollo de estos métodos de optimización, gracias al avance en el rendimiento y la capacidad de los computadores y los algoritmos (W. Zhang et al., 2016). Su desarrollo se ha convertido en una herramienta de gran utilidad para los diseñadores estructurales, ya que ha permitido plantear el proceso de diseño como un problema de optimización matemático, que puede solucionarse mediante la aplicación de estos métodos de optimización, logrando así un eficiente desempeño estructural, y un ahorro en el uso de los recursos (comúnmente el peso estructural) (Saka et al., 2016).

En muchas de las áreas de aplicación (ingeniería, informática, logística, transporte, administración, negocios, finanzas, telecomunicaciones, medio ambiente, aeronáutica, entre otras), a menudo se presentan problemas de optimización complejos que involucran múltiples objetivos que son conflictivos entre sí (Ngatchou et al., 2005). Por ejemplo, en el diseño de un producto se quiere minimizar su costo, maximizar su calidad (en términos físicos, mecánicos o de servicio) y minimizar su impacto ambiental (Talbi, 2009).

La optimización multiobjetivo tiene sus orígenes en el siglo XIX con el trabajo de *Edgeworth* y *Pareto* en Economía (Talbi, 2009). Se ha usado en Economía y la Ciencias de la Administración por varias décadas, y luego en otras Ciencias e Ingeniería, donde actualmente es un área de gran importancia. La complejidad de los problemas de optimización multiobjetivo se hace más y más significativa, en términos del tamaño del problema a resolver (ej. número de objetivos y tamaño del espacio de búsqueda), además, el tiempo para la búsqueda de una solución debe ser razonable para la mayoría de los problemas encontrados en la práctica (complejos y a gran escala). Por lo tanto, para superar estas dificultades, desde el final de los años 1980s, el desarrollo de algoritmos

metaheurísticos multiobjetivo avanzados ha sido investigado ampliamente por la comunidad científica (Talbi, 2009).

En comparación con los problemas de optimización que incluyen un solo objetivo, los problemas multiobjetivo son más difícil de resolver porque no existe una única solución, sino un conjunto aceptable de soluciones consideradas igualmente importantes definidas como soluciones Pareto-óptimas. La optimización multiobjetivo se considera como una fase analítica del proceso de toma de decisiones y consiste en determinar todas las soluciones del problema multiobjetivo que son óptimas (en el sentido de Pareto). La solución preferida, es decir la más deseable por el diseñador o el tomador de decisiones se selecciona del conjunto de soluciones Pareto-óptimas disponible. La generación de un Pareto (el cual solo es posible porque los objetivos son conflictivos) tiene varias ventajas, ya que permite tomar una decisión informada, pues se tiene conocimiento de un amplio rango de opciones que son óptimas desde un punto de vista global. Esta característica proporciona un mejor entendimiento del problema, donde puede explorarse la consecuencia de la decisión tomada, con respecto a todos los objetivos involucrados (Ngatchou et al., 2005).

Existe una amplia gama de algoritmos de optimización multiobjetivo. Entre estos se encuentran las técnicas clásicas basadas en la optimización matemática y los algoritmos basados en enfoques metaheurísticos (como los algoritmos evolutivos o los algoritmos basados en enjambre). Los algoritmos metaheurísticos son adecuados para problemas multiobjetivo (permiten calcular el frente de Pareto en una sola ejecución), especialmente cuando no se conoce a priori información acerca de las preferencias o la prioridad de los objetivos, y cuando se quiere un amplio rango de alternativas de solución. Aunque la información adicional relacionada con el problema que pueda ser proporcionada por el tomador de decisiones (ya sea antes o después del proceso de búsqueda

y que puede traducirse en objetivos y restricciones adicionales) puede ayudar a guiar y refinar el proceso de optimización, lo que se busca es resolver un caso general donde no se cuenta con esta información (Ngatchou et al., 2005).

### 3.1. Formulación matemática de un problema de optimización multiobjetivo

En su forma general, un problema de optimización multiobjetivo se formula de acuerdo a la Ecuación (3.1) (Kalyanmoy Deb, 2001):

$$\begin{aligned} \text{Minimizar } F(\mathbf{x}) = \mathbf{z} = f_m(\mathbf{x}), \quad m = 1, 2, \dots, M \\ \text{Sujeto a,} \\ g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, J \\ h_k(\mathbf{x}) = 0, \quad k = 1, 2, \dots, K \\ x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \dots, n \end{aligned} \quad (3.1)$$

Donde,

- $F(\mathbf{x}) = \mathbf{z} = f_m(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})]$ , representa el vector con las  $M$  funciones objetivo a ser optimizadas.
- $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , es el vector que representa una solución de  $n$  variables de decisión,  $\mathbf{x} \in S$ .
- $g_j(\mathbf{x})$ , son las funciones de restricción de desigualdad
- $h_k(\mathbf{x})$ , son las funciones de restricción de igualdad.
- $x_i^{(L)}$  y  $x_i^{(U)}$ , son las condiciones de borde (límites) de la variable de decisión  $x_i$ .

El espacio de búsqueda  $\Omega$  representa el *espacio de decisión* o espacio de las variables para un problema de optimización multiobjetivo. Por otra parte, el espacio al cual pertenece el vector de las funciones objetivo  $F(\mathbf{x})$ , se denomina *espacio objetivo*  $\Lambda$ . El subespacio objetivo que satisface todas las restricciones se denomina espacio factible (feasible space). El vector  $F$  puede definirse

como una función de costo del *espacio de decisión*, en el *espacio objetivo*, que evalúa la calidad de cada solución  $(x_1, x_2, \dots, x_n)$ , asignando un vector objetivo  $[f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})]$ , que representa la calidad de la solución (o fitness) (Talbi, 2009). El mapeo entre un vector solución con tres variables ( $n = 3$ ) y un vector objetivo con dos funciones ( $m = 2$ ) se muestra en la Figura 3 (Kalyanmoy Deb, 2001):

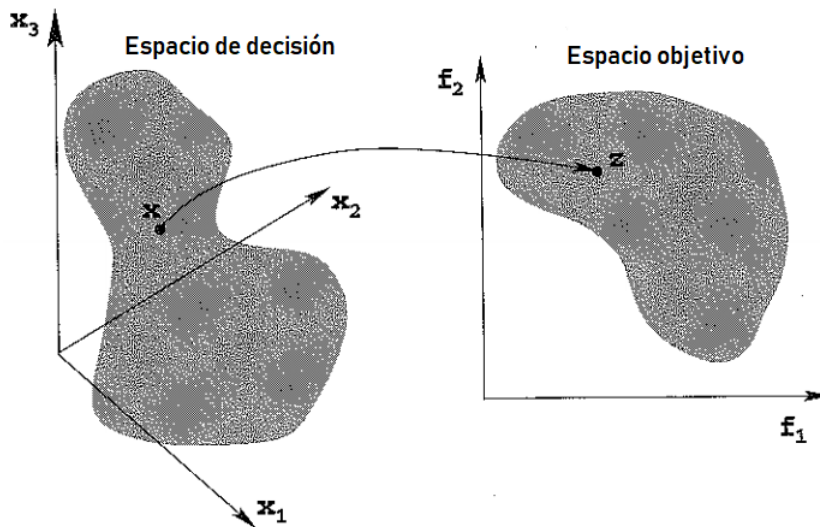


Figura 3. Representación del espacio de decisión  $\Omega$  de las variables y el correspondiente espacio objetivo  $\Lambda$   
Adaptado: Deb (2001)

El objetivo de un problema de optimización multiobjetivo es encontrar el vector  $\mathbf{x}$ , que optimice el vector de funciones  $f_m(\mathbf{x})$ , dando cumplimiento a las funciones de restricción  $g_j(\mathbf{x})$ ,  $h_k(\mathbf{x})$ , y las condiciones de borde (Kalyanmoy Deb, 2001).

**3.1.1. Definición de valor óptimo en optimización multiobjetivo.** En los problemas de optimización multiobjetivo, generalmente no es posible encontrar una única solución que optimice de manera simultánea todas las funciones objetivo, ya que, estas son contradictorias y generan conflicto entre sí, además, cuando se comparan las posibles soluciones entre sí es imposible valorar cuál de ellas es la mejor en término de todos los objetivos. Lo anterior constituye una propiedad básica en la formulación multiobjetivo (Ngatchou et al., 2005). La solución óptima de un problema

de optimización multiobjetivo es un conjunto de soluciones denominadas *soluciones Pareto-óptimas* o *soluciones no dominadas* que representan los diferentes *compromisos (trade-offs)* entre las funciones objetivo (Ngatchou et al., 2005; G. Zavala, Nebro, Luna, & Coello Coello, 2016). A continuación, se abordan en detalle los conceptos fundamentales en optimización multiobjetivo.

**3.1.2. Conceptos básicos en optimización multiobjetivo.** La resolución de un problema de optimización multiobjetivo, generalmente proporciona una alta cantidad de soluciones, las cuales no son necesariamente todas óptimas. La guía para la selección de las soluciones óptimas se basa en los conceptos introducidos por Francis Ysidro Edgeworth y generalizados por Vilfredo Pareto (Ngatchou et al., 2005):

**3.1.2.1. Relación de dominación de Pareto y optimalidad de Pareto.** Formalmente, un vector solución  $\mathbf{x}_1$ , domina otro vector solución  $\mathbf{x}_2$  (denotado como  $\mathbf{x}_1 \prec \mathbf{x}_2$ ) si y sólo si (Ngatchou et al., 2005):

- La solución  $\mathbf{x}_1$  es mejor o igual a  $\mathbf{x}_2$  para todas las funciones objetivo.
- La solución  $\mathbf{x}_1$  es estrictamente mejor que  $\mathbf{x}_2$  en al menos una función objetivo.

Lo anterior es lo que se llama *relación de dominación de Pareto* y se usa para comparar y ordenar los vectores de decisión (posibles soluciones) obtenidos del proceso de optimización. Se expresa matemáticamente (en el contexto de un problema de minimización) de acuerdo a la Ecuación (3.2) (Barr, 2013):

$$f_m(\mathbf{x}_1) \leq f_m(\mathbf{x}_2) \text{ para todos los } m = 1, 2, \dots, M$$

$$f_m(\mathbf{x}_1) < f_m(\mathbf{x}_2) \text{ para al menos un } m$$
(3.2)

El concepto generalmente usado en optimización multiobjetivo es el de *optimalidad de Pareto*. Su definición se deriva directamente de la *relación de dominación*, y fue propuesto por Francis Ysidro Edgeworth en 1881 y fue extendido por W. Pareto en 1896. Una solución  $\mathbf{x}$ , se dice que es Pareto óptima (solución aceptable, no inferior o no dominada), si y solo si no existe otra solución  $\mathbf{x}^*$  que la domine (G. Zavala et al., 2016). El vector de funciones objetivo  $f_m(\mathbf{x})$ , se llama vector dominante de Pareto o vector no dominado (Ngatchou et al., 2005). Una solución Pareto óptima indica que es imposible encontrar una solución que mejore el desempeño de una función objetivo sin disminuir la calidad de al menos otra de las funciones objetivo (Talbi, 2009). Los puntos que dominan los demás puntos y no son dominados entre sí, representan las soluciones óptimas (Pareto-óptimas) y se denominan *soluciones no dominadas* y pertenecen al rango 1 de dominación (Barr, 2013).

**3.1.2.2. Conjunto óptimo de Pareto.** Para un problema de optimización multiobjetivo dado, el conjunto óptimo de Pareto son los valores de las variables de decisión  $\mathbf{x}$  de las soluciones no dominadas pertenecientes al rango 1 de dominación, en el espacio de decisión  $\mathcal{S}$ .

**3.1.2.3. El frente de Pareto.** El frente de Pareto (también llamado superficie de compromiso) es la imagen del conjunto óptimo de Pareto en el espacio objetivo  $\mathcal{Q}$  y se forma con el conjunto de las soluciones que pertenecen al rango 1 de dominación. El frente de Pareto se define como “el borde entre el espacio de soluciones”, factible y no factible. El compromiso es el conjunto de soluciones no dominadas, representadas en el frente de Pareto (ver Figura 4) (Barr, 2013; Ngatchou et al., 2005). Generalmente, es imposible establecer una expresión analítica (al menos para los problemas reales) que permita calcular (determinar) el frente de Pareto (Ngatchou et al., 2005).

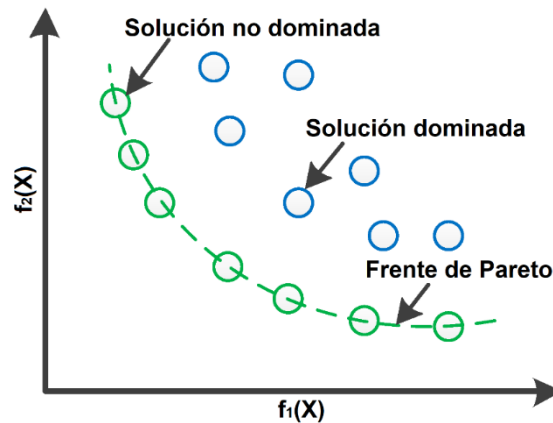


Figura 4. Soluciones dominadas, no dominadas y frente de Pareto para un problema de minimización de dos funciones objetivo  $f_1$  y  $f_2$ .

Adaptado: Adaptado de Mahesh, Nallagownden, & Elamvazuthi (2016)

**3.1.3. Metas de un problema de optimización multiobjetivo.** Obtener el frente de Pareto es la principal meta de un problema de optimización multiobjetivo. Una buena aproximación del frente de Pareto debe contener un número limitado de soluciones, que cumplan con las siguientes condiciones: (i) convergencia: estar lo más cerca posible al frente de Pareto verdadero (Pareto-óptimo), y (ii) diversidad: encontrar soluciones tan diversas como sea posible, es decir distribuidas uniformemente a lo largo del frente de Pareto no dominado encontrado y cubriendo toda la región Pareto-óptima. De otra manera la aproximación obtenida no es de mucha utilidad para el tomador de decisiones quien debería tener una información completa del frente de Pareto (Talbi, 2009). Las dos metas son ortogonales entre sí. La primera meta requiere una búsqueda hacia la región Pareto-óptima, mientras que la segunda requiere una búsqueda a lo largo del frente Pareto-óptimo (ver Figura 5 izquierda) (Kalyanmoy Deb, 2001).

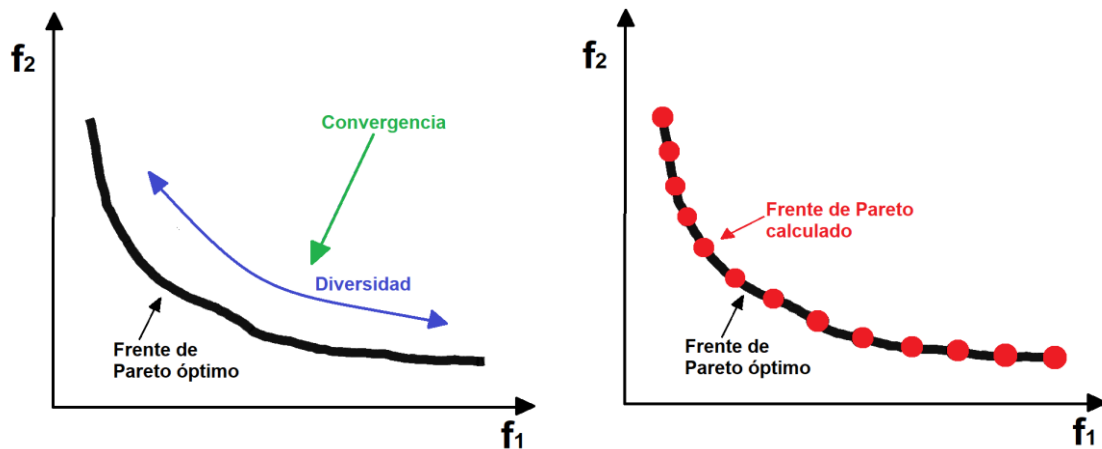


Figura 5. Las dos metas de la optimización multiobjetivo (izquierda). Un conjunto ideal de soluciones no dominadas que cumple con las dos metas (derecha)

Adaptado: Deb (2001)

En la Figura 5 (derecha) se muestra una solución ideal de un problema de optimización multiobjetivo donde se cumple con ambos objetivos: todas las soluciones no dominadas obtenidas se encuentran sobre el frente Pareto-óptimo verdadero y también se mantiene una distribución uniforme a lo largo de la región Pareto-óptima. Sin embargo, debido a diferentes dificultades asociadas con los problemas y/o ineficiencias asociadas con el algoritmo seleccionado, una buena convergencia y distribución de las soluciones no siempre puede lograrse para un problema arbitrario (Kalyanmoy Deb, 2001). A continuación (ver Figura 6), se examinan tres frentes de Pareto que representan varias aproximaciones (puntos discretos) del frente de Pareto-óptimo verdadero (línea continua) para un problema de optimización con dos funciones objetivo:

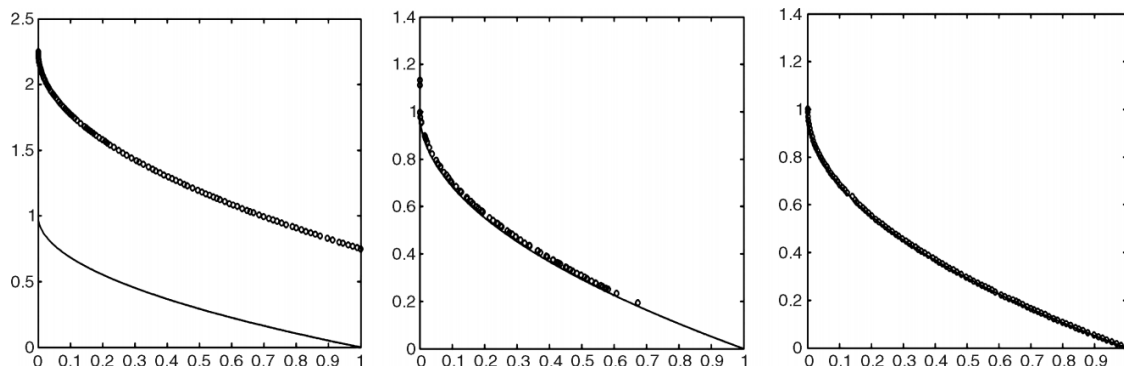


Figura 6. Ejemplos de aproximación de frente de Pareto para un problema de minimización de dos funciones objetivo

Adaptado: Talbi (2009)

La primera gráfica (izquierda) muestra un frente de Pareto que tiene una muy buena diversidad de soluciones, pero los puntos están lejos del frente de Pareto verdadero. Este frente no es atractivo porque no proporciona soluciones Pareto-óptimas. El segundo ejemplo (centro) contiene un conjunto de soluciones muy cercano al frente de Pareto verdadero, aunque algunas regiones no son cubiertas, con lo cual el tomador de decisiones podría perder información importante del frente de Pareto-óptimo. Finalmente, el último frente (derecha) tiene las dos propiedades deseables de un frente de Pareto: una buena convergencia y diversidad (Talbi, 2009).

### 3.2. Optimización estructural

La optimización estructural es el proceso para determinar la mejor distribución posible de material dentro de un espacio de diseño establecido, para soportar y transmitir de manera segura las cargas aplicadas, teniendo en cuenta las restricciones impuestas por condiciones de uso y de manufactura (ver Figura 7) (Querin, Victoria, Alonso, Ansola, & Martí, 2017). El término “mejor” u óptimo, puede tener un significado diferente dependiendo las especificaciones del problema, aunque en general implica que un sistema es económico, eficiente, confiable y durable (Arora, 2017a).

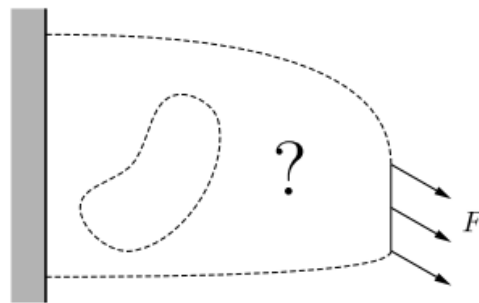


Figura 7. El problema de la optimización estructural encuentra la mejor estructura que transmite la carga  $F$  al soporte

Adaptado: Christensen & Klarbring (2009)

La optimización estructural formula el diseño estructural como un problema de optimización matemático (Sahab, Toropov, & Gandomi, 2013; Saka et al., 2016) que sigue (para la mayoría de problemas de optimización estructural) los cinco pasos (Arora, 2017c) mostrados en la Figura 8:

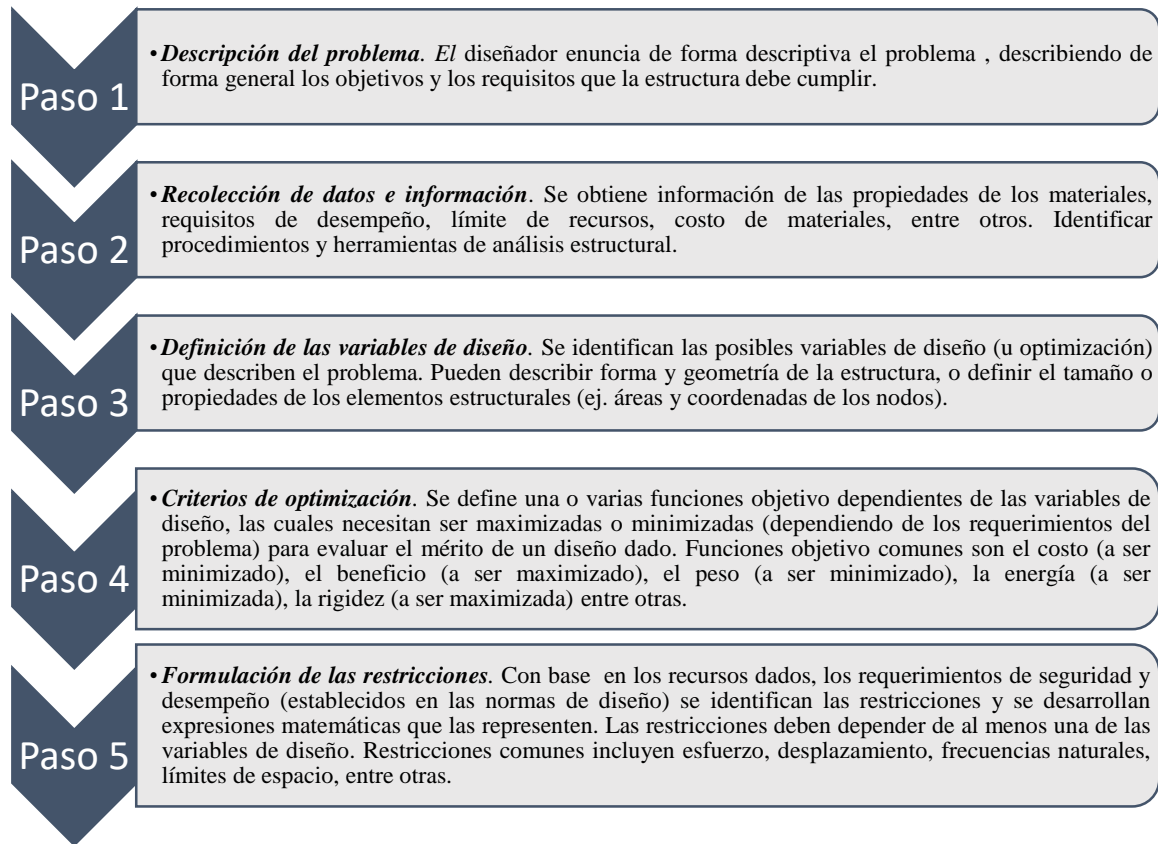


Figura 8. Pasos para formular un problema de optimización estructural  
Adaptado: Arora (2017)

Hay principalmente tres clases de problemas de optimización estructural, los cuales se pueden clasificar dependiendo de las características geométricas que el vector de variables de diseño  $\mathbf{x}$ , representa dentro del tipo de estructura que se quiere optimizar (Christensen & Klarbring, 2009), y son los siguientes:

**3.2.1. Optimización de tamaño.** El diseñador conoce como se verá la estructura, pero no conoce el tamaño (dimensión) de sus componentes. En este caso  $\mathbf{x}$  representa variables de espesor

(en placas, láminas cascarones y membranas) o variables de área (de las secciones transversales de los elementos en celosías, marcos o vigas). Las variables pueden ser continuas, discretas o mixtas (Christensen & Klarbring, 2009; Descamps & Coelho, 2013). A continuación, en la Figura 9 se presentan algunos ejemplos de aplicación de este tipo de optimización:

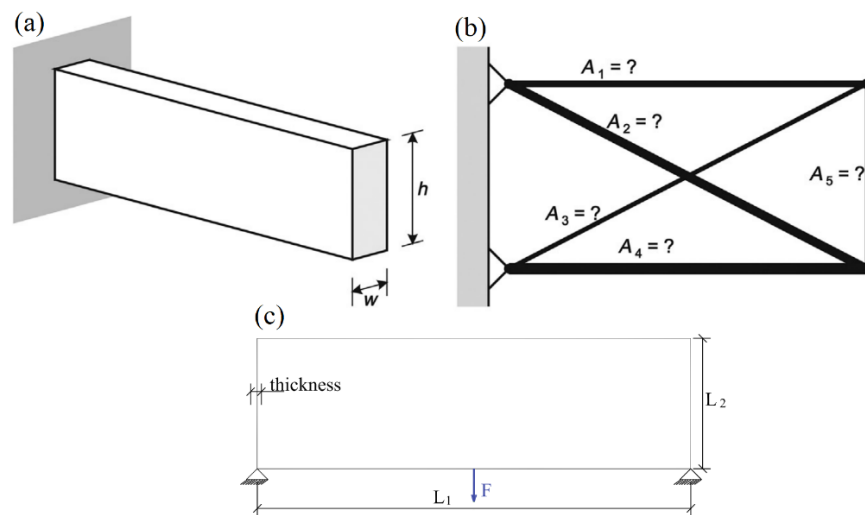


Figura 9. Ejemplos de estructuras donde la optimización de tamaño puede ser aplicada  
 Adaptado: Querin et al. (2017); G. R. Zavala et al. (2014)

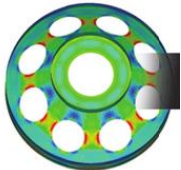
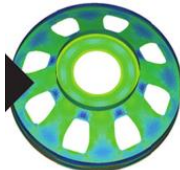



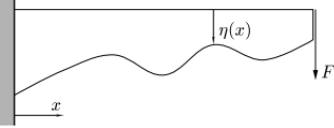
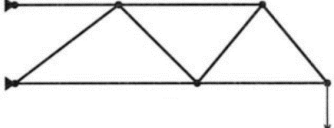
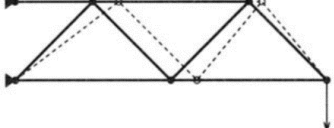
En la Figura 9, (a) representa una viga cantiléver (Querin et al., 2017), donde las variables  $w$  y  $h$  que representan el área de la sección transversal son desconocidas; (b) es una celosía (Querin et al., 2017), donde el área de cada elementos es desconocida; y (c) es una lámina (G. R. Zavala et al., 2014) donde el espesor es desconocido, siendo fijos y conocidos los datos  $L_1$ ,  $L_2$  y  $F$ .

**3.2.2. Optimización de forma.** Busca encontrar los límites estructurales, es decir la forma óptima. En este caso  $\mathbf{x}$  representa las variables geométricas de forma o de contorno del dominio de diseño. Este tipo de optimización actúa ya sea directamente en los nodos (caso de celosías), o mediante funciones de diseño geométrico asistidas por computador (ej. curvas NURBS). La conectividad de los nodos no es cambiada con la aplicación de este tipo de optimización

(Christensen & Klarbring, 2009; Descamps & Coelho, 2013). A continuación, se presentan algunos ejemplos de aplicación de este tipo de optimización (ver Tabla 1):

Tabla 1

*Ejemplos donde optimización de forma puede ser aplicada*

Estructura	Diseño inicial	Diseño optimizado
(a)		
(b)		
(c)		
(d)		

Nota: Adaptado de Bendsøe & Sigmund (2004); Christensen & Klarbring (2009); Rozvany (1997); Zhang et al. (2016)

En la Tabla 1, para el caso (a) pieza mecánica (W. Zhang et al., 2016) y (b) viga simplemente apoyada (Bendsøe & Sigmund, 2004), la optimización de forma permite principalmente mejorar la distribución de esfuerzos. Para el caso (c) viga cantiléver (Christensen & Klarbring, 2009), la optimización de forma encuentra la función  $n(x)$  que describe la forma, y en (d) celosía (Rozvany, 1997) se busca las coordenadas óptimas de los nodos.

**3.2.3. Optimización topológica.** En optimización estructural el término topología significa patrón de conectividad o secuencia espacial de los elementos en una estructura (Rozvany, 1997). Es la forma más general de optimización y busca asistir al diseñador en la definición del tipo de

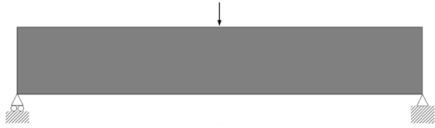
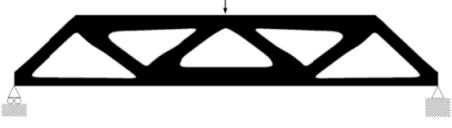
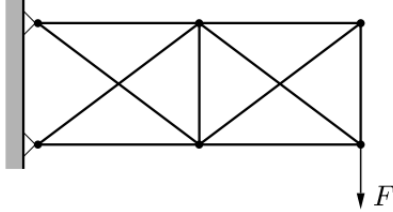
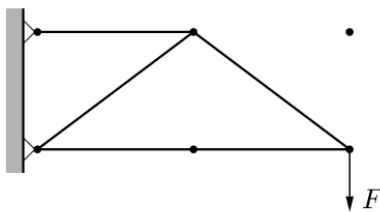
estructura, que satisface de mejor manera las condiciones del problema. Las características más distintivas son: (a) la propiedad elástica del material, como una función de su densidad, que puede variar a lo largo del espacio de diseño. (b) el material puede ser removido permanentemente del espacio de diseño. Este tipo de optimización se puede dividir en dos clases (Querin et al., 2017):

**3.2.3.1. Optimización topológica continua.** Busca encontrar la configuración óptima “solido-vacío” y la conectividad del material en un dominio de diseño dado. Es un método usado principalmente en la etapa conceptual del diseño (W. Zhang et al., 2016). En el caso bidimensional, por ejemplo para una lámina, los cambios en la topología se logran permitiendo que la variable que representa su espesor  $x$  pueda tomar el valor de cero. Si se optimizan características topológicas puras el valor óptimo del espesor puede tomar solo dos valores: 0 o un valor máximo definido que corresponde al espesor. En el caso tridimensional el mismo efecto puede ser logrado permitiendo que  $x$  sea una variable de densidad que puede tomar sólo valores de 0 y 1 (Christensen & Klarbring, 2009). Ejemplo de este tipo de optimización se presenta en la Tabla 2, ejemplo (a).

**3.2.3.2. Optimización topológica discreta.** En este caso la optimización topológica se logra tomando el área de las secciones transversales de los elementos como las variables de diseño  $x$ , permitiendo que puedan tomar el valor de cero, con lo cual pueden ser removidas. De esta forma la conectividad de los nodos es variable y se puede decir que la topología cambia (Christensen & Klarbring, 2009). Ejemplo de este tipo de optimización se presenta en la Tabla 2, ejemplo (b).

Tabla 2

*Ejemplos donde la optimización topológica puede ser aplicada*

Estructura	Diseño inicial	Diseño óptimo
(a)		
(b)		

Nota: Adaptado de Christensen & Klarbring (2009); Zhang et al. (2016)

En la Tabla 2, el caso (a) muestra una optimización topológica continua bidimensional (W. Zhang et al., 2016). En el caso (b) se representa la optimización topológica discreta de celosías (Christensen & Klarbring, 2009), donde barras del diseño inicial son removidas, permitiendo que las áreas de las secciones transversales tomen el valor de cero, generando una nueva estructura.

**3.2.4. Tipos de optimización estructural aplicables en estructuras de celosía.** Uno de los problemas clásicos en la optimización estructural es el diseño óptimo de estructuras de celosía, es decir, dadas las condiciones de soporte y las fuerzas externas, ¿Cómo debe ser la configuración óptima de la estructura? Cuando el método de Programación Lineal y los computadores aparecieron en los años 50s y 60s, Dorn, Gomory & Greenberg (1964) fueron pioneros en el desarrollo de una base para la solución de este problema de optimización estructural mediante un enfoque computacional (Rozvany, 1997).

De forma general, en estructuras de celosía pueden aplicarse tres tipos de optimización estructural (derivados de la clasificación de la Sección 0): tamaño (dimensión), forma (geometría) y topología discreta (Savsani et al., 2017; Tomšič & Duhovnik, 2014):

**3.2.4.1. Optimización de tamaño o dimensión.** Busca encontrar el valor óptimo de las áreas de las secciones transversales de los elementos estructurales, que contribuyen a optimizar una o varias funciones objetivo como el peso, costo, energía de deformación, entre otras, a partir de una configuración estructural inicial definida (conectividad de los elementos y coordenadas de los nodos). En este caso, las variables de diseño son las áreas de las secciones transversales en un espacio de búsqueda definido.

**3.2.4.2. Optimización de forma o geometría.** Optimiza la posición de los nodos a partir de una configuración estructural inicial definida (conectividad de los elementos y área de los elementos estructurales). Las variables de diseño son las coordenadas de los nodos.

**3.2.4.3. Optimización topológica discreta.** Busca el patrón de conectividad óptimo de los nodos o la secuencia espacial de los elementos de una estructura, es decir trata con la adición o remoción de elementos y nodos, a partir de una configuración inicial conocida como estructura base (con coordenadas de los nodos y conectividad de los elementos) que contiene todas las posibles soluciones. Las variables de diseño son las áreas de las secciones transversales, pero a diferencia de la optimización de forma pueden tomar un valor de cero o negativo (indicando remoción). Como el problema de optimización topológica permite usar las áreas como variables de diseño incluyendo la posibilidad de desaparecer, podría considerarse como un problema de optimización de tamaño (Bendsøe & Sigmund, 2004).

A continuación, en la Figura 10, se muestra de forma esquemática los tres casos de optimización aplicables a estructuras de celosía:

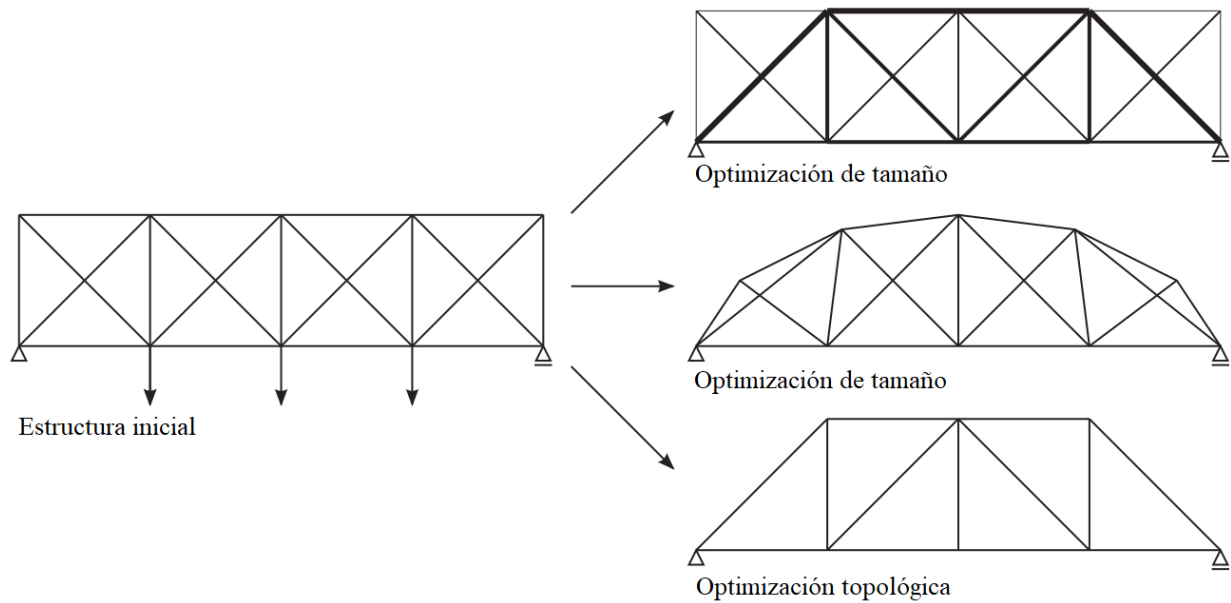


Figura 10. Problemas de optimización de tamaño, forma y topológica en celosías  
Adaptado: Descamps & Coelho (2013)

Los tres tipos de optimización se pueden ejecutar de forma simultánea para obtener los mejores resultados (aunque esto puede ocasionar una mayor complejidad del problema y un mayor costo computacional) (Miguel et al., 2013; Rozvany, 1997). El tipo de optimización empleado en este trabajo fue la optimización topológica discreta.

**3.2.5. Enfoque clásico de optimización topológica de estructuras de celosía.** El enfoque común usado en la optimización topológica de celosías es la técnica de la estructura base (ground structure). Está basado en un proceso de reducción, donde elementos y nodos son eliminados de una estructura base inicial altamente conectada (que contiene todos los nodos y elementos potenciales), hasta lograr una solución aceptable.

A partir del dominio de diseño inicial con las condiciones de carga, apoyos (ver Figura 11(a)), la técnica comienza con un número de puntos nodales  $N$  en el espacio de búsqueda (bidimensional o tridimensional) (ver Figura 11(b)). En algunos de estos nodos están aplicadas las fuerzas

externas, y en otros se definen las condiciones de apoyo. A continuación, se define una grilla con  $m$  potenciales conexiones (elementos base), con base en diferentes grados conectividad entre los nodos existentes, teniendo en cuenta que cada elemento está conectado a dos nodos. Esta grilla (incluyendo las cargas y las condiciones de apoyo) es lo que se denomina estructura base (ver Figura 11(c)). El objetivo de la optimización topológica en este caso, es encontrar una estructura óptima (ver Figura 11(d)) que cumpla con las condiciones de carga y apoyo y que los elementos resultantes sean un subconjunto de la estructura base inicial, es decir los elementos que forman la estructura solución están contenidos como una “subestructura” en el universo estructural definido en la estructura base (Rozvany, 1997).

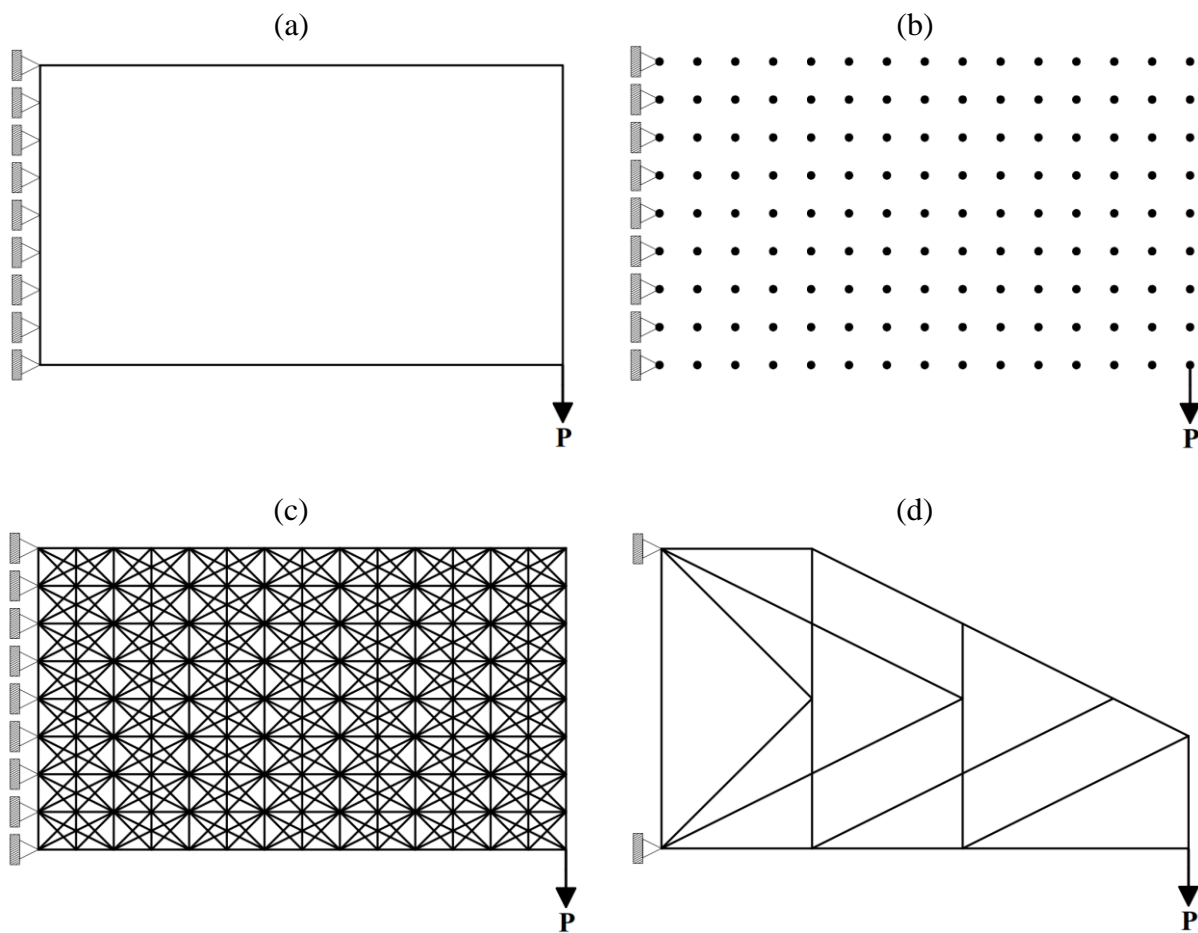


Figura 11. Optimización topológica de estructuras de celosía  
Adaptado: Rozvany (1997)

La posición de los nodos usualmente se mantiene fija para mantener el problema de optimización lo más simple posible. En este caso es necesario que la estructura base sea muy densa (alta cantidad de elementos en la conectividad inicial) para obtener una buena estructura solución. Por lo tanto, el número de elementos  $m$ , como función del número de nodos  $N$  se calcula con la Ecuación (3.3):

$$m \approx \frac{1}{2}N(N - 1) \quad (3.3)$$

En esta fórmula los elementos que traslapan los elementos más cortos usualmente no se tienen en cuenta. Como en el método de la estructura base se asume que la posición de los nodos se mantiene fija, no son usados como variables de diseño (Rozvany, 1997). Una desventaja de esta estrategia es que las topologías óptimas pueden ser muy sensitivas a la disposición de los nodos en el espacio, en especial si el número es relativamente bajo. Esto hace que pueda considerarse una extensión del método que incluya la optimización de los puntos nodales (optimización de forma) (Bendsøe & Sigmund, 2004). Cuando se permite la variación de las posiciones de los nodos, el problema de *topología* se mezcla con el problema de *forma*. Como las posiciones nodales tienen libertad de movimiento, puede usarse un menor número de nodos  $N$  en la estructura base, y por lo tanto un menor número de elementos potenciales  $m$ , ya que no es necesario simular las posiciones nodales variables con una malla densa de nodos. La desventaja de usar la optimización simultánea de forma y topología (como se mencionó anteriormente), es que el modelo matemático se vuelve mucho más complejo (Rozvany, 1997).

### **3.2.6. Métodos para la generación de la estructura base**

**3.2.6.1. Método de niveles de conectividad de nodos (NCN).** En este método una estructura base puede ser categorizada por niveles de conectividad. Iniciando desde un nivel 1, en el cual los

nodos solo se conectan con los nodos de la vecindad como se muestra en la Figura 12 (b). Un nivel 2 cuando los nodos se conectan hasta un nodo más allá de los nodos de la vecindad (ver Figura 12(c)). A medida que el nivel de conectividad va aumentando, cada nodo definido en la discretización inicial de la Figura 12(a), se conecta a un mayor número de nodos a su alrededor, hasta alcanzar la conectividad total entre cada uno de ellos, como sucede en el último nivel de la Figura 12(d) (X. S. Zhang, Maheshwari, Ramos, & Paulino, 2018). El último nivel es el que generalmente proporciona la mejor solución (X. Zhang, Maheshwari, Ramos, & Paulino, 2016).

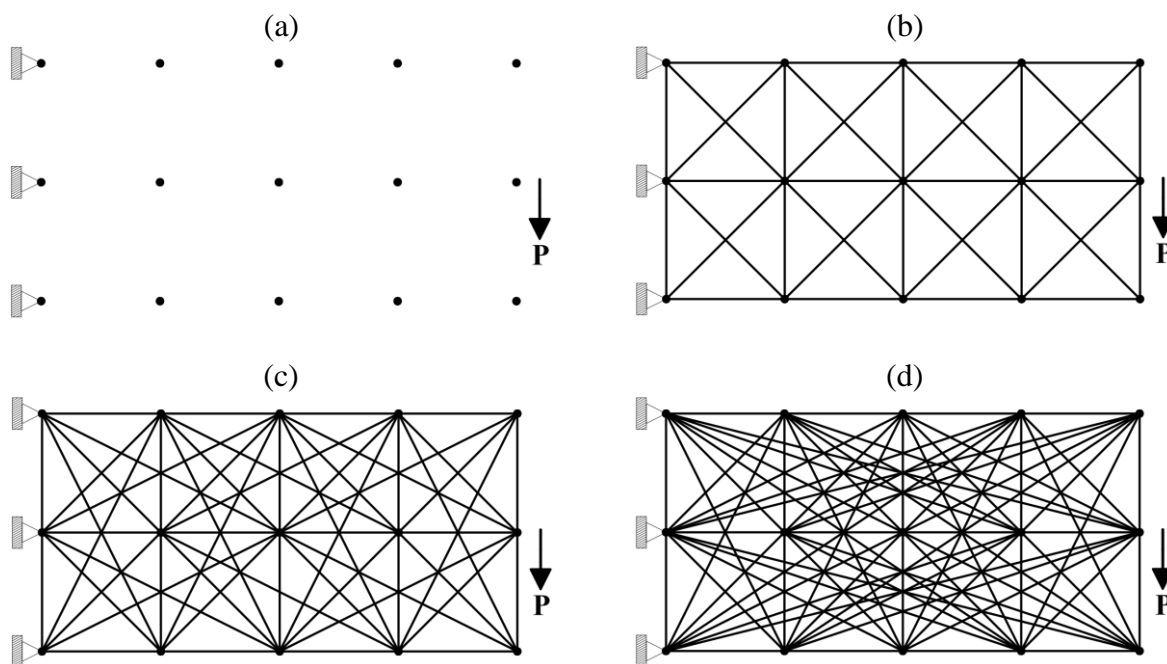


Figura 12. Niveles de conectividad para una estructura base en el método estándar.  
Adaptado: Bendsøe & Sigmund (2004)

**3.2.6.2. Método de macroelementos.** En el método de niveles de conectividad, los nodos pueden distribuirse inicialmente en el espacio de diseño y proporcionar una buena estructura base (si estos están distribuidos uniformemente). En el método de macroelementos, el uso de una malla inicial puede ayudar a realizar esta tarea, controlando esta distribución, de acuerdo a sus características geométricas. Pueden usarse mallas poligonales en forma de cuadrícula (ver Figura 13(a)) o basadas usando otros polígonos como los diagramas de Voronoi (ver Figura 13(b)). El

uso de un mallado basado en diagramas de voronoi, tiene ciertas ventajas con respecto a la cuadrícula ya que puede ser fácilmente discretizable para dominios no convexos y con presencia de espacios huecos, teniendo mayores posibilidades de proporcionar elementos base de las estructuras celosía en varias direcciones, debido a una distribución más aleatoria de los nodos (X. Zhang et al., 2016).

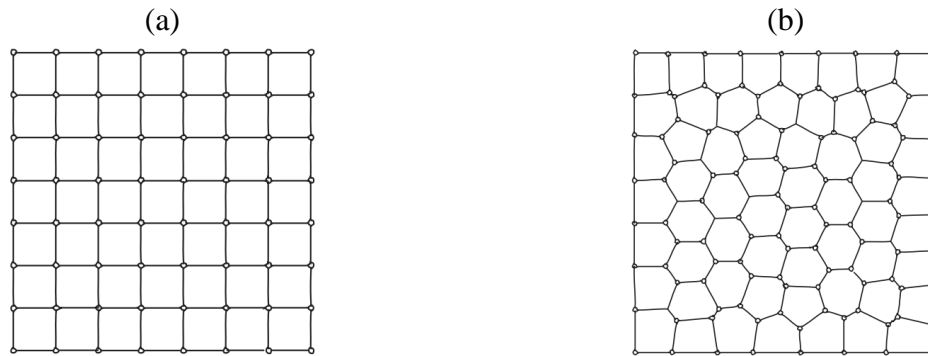
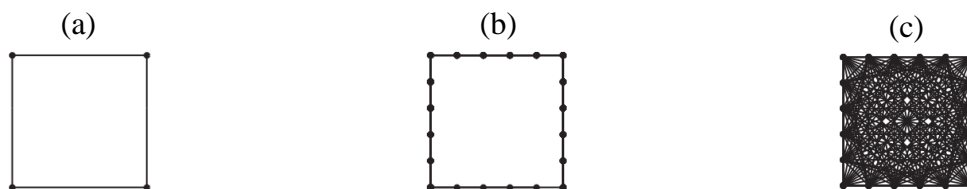


Figura 13. Tipos de malla para la estructura base usando el método de macroelementos  
Adaptado: X. Zhang et al. (2016)

La idea del *método del macroelemento* es insertar nodos igualmente espaciados en cada vértice de cada macroelemento (polígono) generado en la malla inicial, y después generar todas las posibles conexiones solo dentro de cada macroelemento (ver Figura 14). Mediante este método se logra una eficiente generación de las estructuras base iniciales, se reduce el número de elementos traslapados, las barras pueden ser generadas sin el paso adicional de detectar límites del espacio de diseño o verificar las conexiones posibles, se reduce el ancho de banda de la matriz de rigidez y permite nuevas formas en la topología final (X. Zhang et al., 2016; X. S. Zhang et al., 2018).



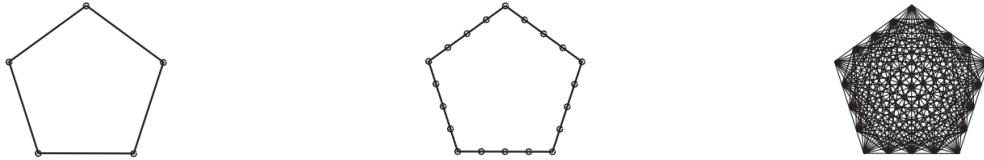


Figura 14. Macroelementos de malla cuadrada y voronoi

Adaptado: X. Zhang et al. (2016)

**3.2.6.3. Método de trayectorias de esfuerzo.** En este método de generación de la estructura base, los nodos son ubicados en la intersección de las trayectorias de esfuerzo (ver Figura 15). La distribución de estas trayectorias puede ser descrita con la ayuda de las ecuaciones de transformación de esfuerzo y el círculo Mohr, usando la información del estado de esfuerzo, mediante la solución del problema estático (ej. usando un software de elementos finitos) en todo el dominio de diseño (ver Figura 15(a)), considerando propiedades isotrópicas del material. Después de obtener los valores y dirección de los esfuerzos principales  $\sigma_1$  y  $\sigma_2$ , es posible dibujar las trayectorias de esfuerzo  $\sigma_1$  (en color azul) y  $\sigma_2$  (en color azul) (ver Figura 15(b)). En la selección de las trayectorias, se debe tener en cuenta que algunas de ellas deben pasar por los nodos esenciales (donde se ubican cargas externas y donde se encuentran ubicados los apoyos) (Gao, Li, Pan, Chen, & Liu, 2017; Gao, Liu, et al., 2017).

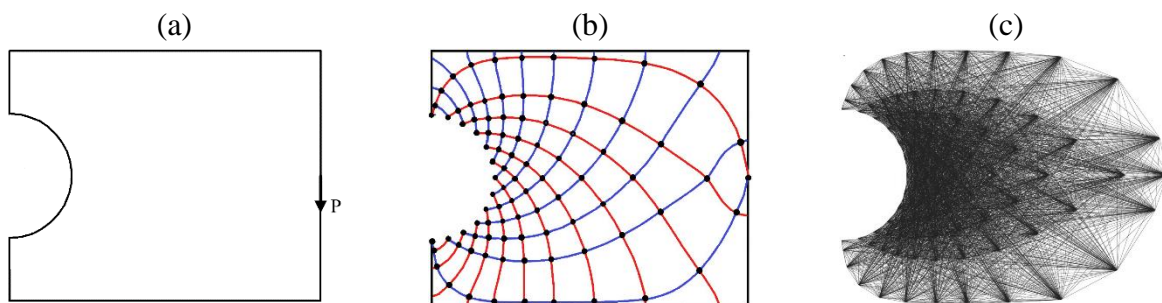


Figura 15. Estructura base usando el método de trayectorias de esfuerzo

Adaptado: Gao, Liu, et al. (2017)

Este método se adapta muy bien a dominios de diseño no convexos, con diferentes geometrías, y la forma como se ubican los nodos (en la intersección de los esfuerzos principales) para generar

la estructura base (ver Figura 15(c)), puede resultar en una buena solución, aplicando un proceso de optimización topológica (Gao, Liu, et al., 2017).

Teniendo en cuenta que la esencia general de la optimización topológica de estructuras de celosía es encontrar los caminos óptimos para la transmisión de las cargas a los apoyos (es decir las mejores conexiones de los nodos y barras) (Gao, Li, et al., 2017), la aplicación de este método de generación de la estructura base es adecuado, ya que para condiciones análogas de carga y soporte, las estructuras de celosía obtenidas con la optimización topológica coinciden (de forma discreta) con las trayectorias de esfuerzos principales de una estructura óptima continua (Rozvany, 1997).

En el Apéndice B, por ser de interés en este trabajo, se presenta una definición más amplia del concepto de trayectorias de esfuerzo, junto con la descripción de un algoritmo para calcularlas.

**3.2.7. Formulación matemática de un problema de optimización topológica multiobjetivo de estructuras de celosía.** Usando el método de la *estructura base* se generan las posibles combinaciones de los elementos disponibles en la estructura de celosía durante el proceso de optimización. Entonces un algoritmo de optimización (ej. clásico o metaheurístico) es aplicado para decidir si remover o mantener estos elementos, formando así una topología. En general un problema de optimización multiobjetivo de estructuras de celosía, busca encontrar el área de los elementos y la posición de los nodos, que minimicen funciones objetivo (peso de la estructura, energía de deformación, desplazamiento máximo) sujeto a restricciones estructurales estáticas (esfuerzo, desplazamiento, estabilidad cinemática, pandeo) y/o dinámicas (frecuencias naturales). La masa nodal y de los elementos puede ser incorporada en la función objetivo. El problema puede

expresarse por medio de la Ecuaciones (3.4), (3.5) y (3.6) de la siguiente manera (G. G. Tejani, Savsani, et al., 2018):

Encontrar las variables  $\mathbf{x} = \{A_1, A_2, \dots, A_m, \xi_1, \xi_2, \dots, \xi_n\}$ , para minimizar las funciones:

$$F(\mathbf{x}) = [f_1, f_2, f_3, \dots, f_{NOBJ}] \quad (3.4)$$

$$f_1 = W(\mathbf{x}) = \sum_{i=1}^m B_i A_i \rho_i L_i + \sum_{j=1}^n b_j$$

$$f_2 = E_s(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^m B_i \sigma_i \varepsilon_i A_i L_i$$

$$f_3 = \Delta_{max}(\mathbf{x}) = \max(\delta_j)$$

$$f_{NOBJ} = f(\mathbf{x})$$

$$B_i = \begin{cases} 0, & \text{si } A_i \leq A_{cr} \\ 1, & \text{si } A_i > A_{cr} \end{cases} \quad (3.5)$$

Sujeto a las restricciones:

$h_1(\mathbf{x})$ : Ecuación de equilibrio estático,  $\mathbf{ku} = \mathbf{f}$

$g_1(\mathbf{x})$ : Validez de la estructura

$g_2(\mathbf{x})$ : Estabilidad cinemática

$g_3(\mathbf{x})$ : Esfuerzo axial,  $|B_i \sigma_i| - \sigma_i^{max} \leq 0$

$g_4(\mathbf{x})$ : Desplazamiento,  $|\delta_j| - \delta_j^{max} \leq 0$

$g_5(\mathbf{x})$ : Esfuerzo de pandeo,  $|B_i \sigma_i^{comp}| - \sigma_i^{cr} \leq 0$  donde  $\sigma_i^{cr} = \frac{k_i A_i E_i}{L_i^2}$  (3.6)

$g_6(\mathbf{x})$ : Frecuencias naturales,  $f_r - f_r^{min} \geq 0$

$g_7(\mathbf{x})$ : Área de las secciones transversales,

- Para variables continuas  $A_i^{min} \leq A_i \leq A_i^{max}$
- Para variables discretas  $A_i \in D_i$ , donde  $D_i = \{d_{i,1}, d_{i,2}, \dots, d_{i,s}\}$

$$g_8: \text{Forma, } \xi_j^{\min} \leq \xi_j \leq \xi_j^{\max}$$

Donde,

$\mathbf{x}$  es el vector de las variables de diseño,  $W$ ,  $E_s$ ,  $\Delta_{max}$ ,  $\mathbf{f}$ ,  $\mathbf{u}$  y  $\mathbf{k}$  representan: el peso total de la estructura, la energía de deformación, la función de desplazamiento máximo, vector de fuerzas externas, vector de desplazamientos nodales y matriz de rigidez global de la estructura respectivamente.  $B_i$ , es un número binario (0 y 1 para eliminar o retener el  $i^{\text{ésimo}}$  elemento respectivamente).  $A_i$ ,  $\rho_i$ ,  $L_i$ ,  $E_i$ ,  $\sigma_i$  y  $\sigma_i^{cr}$ , son: la variable de diseño (área), densidad del material, longitud, módulo de elasticidad, esfuerzo y esfuerzo crítico de pandeo en el elemento  $i^{\text{ésimo}}$  respectivamente.  $\delta_j$ ,  $\xi_j$  y  $b_j$  son los valores del desplazamiento, la posición y el valor de la masa del nodo  $j^{\text{ésimo}}$  respectivamente.  $f_r$  es la frecuencia natural estructural del modo  $r^{\text{ésimo}}$ . Los exponentes *max*, *min* y *comp* significan máximo permitido, mínimo permitido y esfuerzo a compresión respectivamente.  $k_i$  es el coeficiente de pandeo de Euler.  $D_i$  es el conjunto de valores discretos para la variable de diseño  $A_i$  y  $s$  es el número de variables discretas disponibles para la variable de diseño  $i^{\text{ésima}}$  (G. G. Tejani, Savsani, et al., 2019).

El valor del área crítica  $A_{cr}$  determina la presencia o la ausencia de un elemento en la estructura base. Se compara el valor del área  $A_i$  de la sección transversal del elemento  $i^{\text{ésimo}}$ , con el valor del área crítica  $A_{cr}$  definido por el usuario, usualmente su valor es pequeño). Si el valor de  $A_i$  es menor o igual que  $A_{cr}$ , se asume que el elementos está ausente de la estructura, en el caso contrario, significa que el elemento se mantiene en la estructura (Kalyanmoy Deb & Gulati, 2001).

Con la restricción  $g_1$  se verifica la validez de la estructura, es decir que tenga todos los nodos esenciales (donde se ubican de cargas externas, apoyos y nodos predefinidos por el usuario).

La estabilidad cinemática  $g_2$  se incluye en las restricciones porque puede prevenir errores en el análisis estructural. Dos criterios para inestabilidad cinemática se tienen en cuenta (Assimi et al., 2017; Kalyanmoy Deb & Gulati, 2001):

a) El grado de libertad (DOF) de la estructura usando el criterio de Grubler ( $DOF = 2j - b - r$ ), el cual debe ser menor que 1 para ser una estructura, de lo contrario es un mecanismo.  $j$  es el número de nodos,  $b$  es el número de elementos y  $r$  denota el número de componentes de reacción en la estructura de celosía.

b) La matriz de rigidez global de la estructura  $\mathbf{k}$  debe ser estrictamente positiva-definida (es decir, tiene un valor de determinante mayor a cero y todos sus eigenvalores son positivos), asegurando que la estructura es estrictamente estable.

### 3.3. Métodos de solución de problemas de optimización

**3.3.1. Métodos de un solo objetivo.** Muchos métodos han sido aplicados en la solución de diversos problemas de optimización de un solo objetivo. En general pueden ser clasificados en dos categorías: algoritmos clásicos y algoritmos metaheurísticos.

**3.3.1.1. Algoritmos clásicos.** Son algoritmos de optimización determinísticos, es decir siguen un procedimiento riguroso, exacto y repetible. Para el mismo punto de inicio siguen el mismo camino y la solución final alcanzada es la misma cada vez que se ejecuta el algoritmo. Los algoritmos clásicos pueden ser de dos clases (Yang, 2010):

*3.3.1.1.1. Algoritmos basados en gradiente.* También llamados métodos derivativos. Son métodos de optimización de búsqueda local basados en el cálculo, que usan información de gradiente. Ofrecen soluciones precisas, eficientes y de rápida convergencia para problemas de optimización donde se asume continuidad y existencia de derivadas en las funciones objetivo, sin embargo, para problemas con discontinuidad y multimodalidad pueden presentar inconvenientes. En esta clasificación se encuentran: método del gradiente descendente (*Steepest descent*), método del gradiente conjugado, método de la variable métrica, método de Newton, método de Gauss-Newton (Du & Swamy, 2016; Yang, 2010). El aspecto más fuerte de estos métodos es que la teoría matemática de convergencia garantiza la calidad de la solución obtenida (Hare et al., 2013).

*3.3.1.1.2. Algoritmos libres de gradiente.* También llamados no derivativos o métodos de búsqueda directa. Son algoritmos que no usan información de derivadas o gradiente, solamente la evaluación del valor de la función objetivo. Ejemplos de este tipo de algoritmos incluyen: *Simplex search method*, *Nelder-Mead downhill simplex*, *Hook-Jeeves Pattern search method*, *Powell's method*, *Sequential simplex method*, *Rosenbrock's method*, *MADS (Mesh Adaptive Direct Search)*, *RAGS (Robust Approximate Gradient Sampling)* y *Trust regions methods* (Du & Swamy, 2016; Yang, 2010). Al igual que los basados en gradiente la teoría matemática de convergencia garantiza la calidad de la solución, siendo útiles para su aplicación como un paso final para asegurar la respuesta óptima. Sin embargo, pueden presentar inconvenientes cuando el número de variables incrementa (Hare et al., 2013).

*3.3.1.2. Algoritmos metaheurísticos.* La palabra “meta” significa “más allá, más alto o nivel superior” y la palabra “heurístico” significa “encontrar o descubrir nuevas estrategias por ensayo y error” (Yang, 2010). Una metaheurística se puede definir como el arte de descubrir una

metodología de alto nivel, independiente del problema que proporcione una serie de reglas para resolver problemas de optimización (G. G. Tejani, Savsani, et al., 2018).

Los métodos de optimización heurísticos y metaheurísticos son algoritmos de optimización probabilísticos (usan operadores aleatorios) y aproximados que no requieren información de gradiente (también pueden ser llamados libres de gradiente). La principal diferencia es que los algoritmos heurísticos son generalmente dependientes del problema y son diseñados para un problema en particular, mientras que los algoritmos metaheurísticos representan una aproximación más general aplicable, en principio, a una gran cantidad de problemas de optimización (Talbi, 2009). Aunque los términos heurístico o metaheurístico pueden usarse de forma indistinta, la tendencia es llamar a todos los algoritmos estocásticos con aleatorización y búsqueda local como metaheurísticos (Yang, 2010).

Los algoritmos metaheurísticos se basan en la abstracción de procesos biológicos, físicos, químicos y sociales inspirados en la naturaleza, construyendo modelos computacionales que integran principios de diferente nivel que pueden ser eficientes para solucionar problemas complejos (Du & Swamy, 2016; Gandomi, Yang, Talatahari, & Alavi, 2013).

Los algoritmos metaheurísticos tienen tres grandes propósitos: resolver problemas rápidamente, resolver problemas a gran escala y que sean robustos (Talbi, 2009). Los algoritmos metaheurísticos aplican reglas estocásticas que de forma iterativa convergen hacia una solución óptima o cercana de problemas complejos en un tiempo razonable (Talatahari & Kaveh, 2013). Se espera que estos algoritmos funcionen la mayoría de las veces, aunque no todas las veces. La idea es tener un algoritmo eficiente y práctico que funcione la mayoría de las veces y sea capaz de generar soluciones de buena calidad. Esto es suficiente cuando no necesariamente se quiere la mejor

solución, pero si buenas soluciones que estén muy cerca del óptimo y puedan alcanzarse fácilmente (Yang, 2010).

A pesar de que la teoría matemática de convergencia que garantice que la respuesta obtenida sea la óptima no es tan amplia (sólo existe para algunos de ellos), son algoritmos que por su fácil implementación, flexibilidad, versatilidad, robustez y habilidad para localizar óptimos globales o cercanos, son de gran utilidad en problemas que requieren el manejo de una gran cantidad de variables (continuas y discretas), con funciones objetivo altamente no lineales y con muchos locales óptimos, problemas en los cuales ha sido difícil o imposible encontrar una solución por medio de los algoritmos clásicos (Hare et al., 2013).

*3.3.1.2.1. Componentes de un algoritmo metaheurístico.* La intensificación y diversificación, son los principales componentes o características de un algoritmo metaheurístico (ver Figura 16). La fase de intensificación o de explotación busca los mejores candidatos o soluciones, enfocándose en una región del espacio de búsqueda local, explotando la información encontrada por las mejores soluciones en esta región. La fase de diversificación o de exploración, asegura que el algoritmo explore el espacio de búsqueda de manera más eficiente, a una escala global, usando técnicas aleatorias que aumentan la diversidad de las soluciones evitando que puedan quedar atrapadas en óptimos locales. La eficiencia global de un algoritmo depende de un buen balance entre estos dos componentes. El algoritmo puede quedar atrapado en óptimos locales si hay muy poca exploración o mucha explotación, haciendo que sea difícil lograr la convergencia. Lograr el balance de estos dos componentes es de gran importancia, ya que asegura que se pueda lograr una buena solución global (Gandomi et al., 2013).

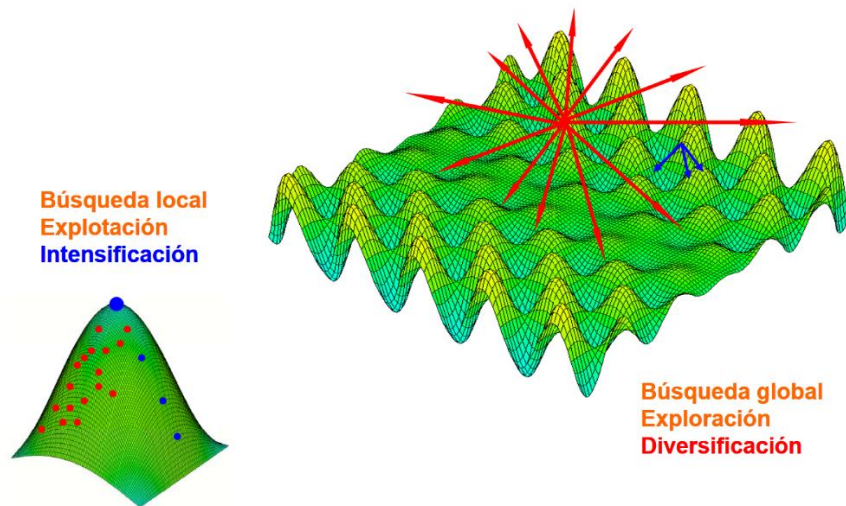


Figura 16. Componentes de un algoritmo metaheurístico  
Adaptado: <http://sci2s.ugr.es/otherCourses/Metaheuristics>

Durante la búsqueda de soluciones un apropiado mecanismo o criterio de selección, debe ser considerado en la selección de las mejores soluciones. Un criterio común es la supervivencia del más fuerte, el cual se basa en actualizar la mejor o las mejores soluciones encontradas en cada iteración. Esto se hace para verificar que las mejores soluciones se mantienen y son incluidas en las siguientes generaciones. Encontrar la solución óptima a un problema de optimización, depende de la elección y el uso correcto del algoritmo adecuado. La selección del algoritmo depende del tipo de problema, los algoritmos disponibles, los recursos computacionales y las restricciones de tiempo (Gandomi et al., 2013).

Los algoritmos metaheurísticos más conocidos pueden clasificarse de acuerdo a la fuente de inspiración en las siguientes categorías (Gandomi et al., 2013; Yang, 2010):

a) *Algoritmos evolutivos*: ES (Evolution Strategies), EP (Evolutionary Programming), GA (Genetic Algorithm), GP (Genetic Programming), DE (Differential Evolution), GE (Grammatical Evolution).

*b) Algoritmos físicos:* SA (Simulated Annealing), TS (Tabu Search), HS (Harmony Search), RO (Ray Optimization), BB-BC (Big Bang – Big Crunch), WWO (Water Wave Optimization), HTS (Heat Transfer Search).

*c) Algoritmos de enjambre:* ACO (Ant Colony Optimization), PSO (Particle Swarm Optimization), BFOA (Bacterial Foraging Optimization Algorithm), HBA (Honey Bee Algorithm), VBA (Virtual Bee Algorithm), ABC (Artificial Bee Colony), FA (Firefly Algorithm), CS (Cuckoo Search), BA (Bat Algorithm), CSS (Charged System Search), KH (Krill Herd), TLBO (Teaching Learning Based Optimization).

**3.3.2. Métodos multiobjetivo.** Los problemas de optimización multiobjetivo involucran varios objetivos conflictivos que deben ser optimizados simultáneamente. Por ejemplo, dos objetivos son conflictivos en un punto del espacio si su máxima mejoría es lograda en dirección contraria (Mitropoulou et al., 2013). El reto en este caso, es encontrar un conjunto óptimo de soluciones no dominadas que se aproximan al frente de Pareto verdadero y que estén uniformemente distribuidas a lo largo del frente de Pareto. Los métodos disponibles para su cálculo pueden ser de dos tipos: métodos clásicos multiobjetivo y algoritmos metaheurísticos multiobjetivo (Du & Swamy, 2016).

**3.3.2.1. Métodos clásicos multiobjetivo.** Los métodos clásicos multiobjetivo consisten en convertir el problema de optimización multiobjetivo en un problema de optimización de un solo objetivo, ya sea agregando (sumando) las funciones objetivo u optimizando uno de los objetivos y tratando los otros como restricciones. El problema entonces puede resolverse usando las técnicas tradicionales de optimización escalar. Estas técnicas están orientadas hacia la búsqueda de una única solución, la que mejor satisface el criterio y la información adicional previa (preferencias,

ya sea el orden de los objetivos de acuerdo a su importancia o alguna indicación de los valores que se quieren encontrar), requerida por la mayoría de estos métodos y que es proporcionada por el tomador de decisiones. La aproximación del frente de Pareto se obtiene con la ejecución repetida del método hasta obtener el número de soluciones requeridas. A continuación, se describen algunos de los métodos clásicos existentes para la solución de problemas multiobjetivo (Ngatchou et al., 2005):

*3.3.2.1.1. Método de agregación ponderada (Weighted Aggregation).* En el método de la agregación ponderada, la función objetivo que se va a aplicar, es diseñada por el tomador de decisiones de acuerdo a las preferencias. La función es una combinación lineal de las funciones originales factorizadas, de acuerdo a la Ecuación (3.7).

$$\begin{aligned} \text{Minimizar } Z &= \sum_{j=1}^M w_j * f_j(\mathbf{x}), \\ w_j &\geq 0 \text{ y } \sum_{j=1}^M w_j = 1 \end{aligned} \tag{3.7}$$

Donde,

$w_j$  es un factor de peso y representa el valor importancia relativa dado por el tomador de decisiones al objetivo  $f_j(\mathbf{x})$ . Los valores de los factores de peso deben determinarse antes de iniciar el proceso de optimización de la nueva función objetivo  $Z$ . La solución de este problema de optimización, resulta en un único resultado que depende del valor de los factores de peso seleccionados.

*3.3.2.1.2. Método de programación de objetivo (Goal programming).* También conocida como *Goal attainment*. Es una variación de la técnica de agregación ponderada, que busca minimizar las

desviaciones de los objetivos preestablecidos (*goals o targets*). su formulación se define con la Ecuación (3.8):

$$\text{Minimizar } Z = \sum_{j=1}^M w_j * |f_j(\mathbf{x}) - T_j| \quad (3.8)$$

Donde,

$T_j$  representa el valor de la función objetivo, o *target* definidos por el tomador de decisiones para  $f_j(\mathbf{x})$ . Los factores  $w_j$ 's representan ahora las prioridades. Como en el método de agregación ponderada la principal desventaja es la necesidad de información previa (prioridades y *target*).

*3.3.2.1.3.  $\varepsilon$ -Constraint.* Es un método diseñado para descubrir las soluciones óptimas del frente de Pareto, con base en la optimización de un solo objetivo y la consideración de los demás objetivos como restricciones, limitadas por algún rango permitido  $\varepsilon_i$ . El problema se resuelve en repetidas ocasiones para diferentes valores de  $\varepsilon_i$  para generar el frente de Pareto. Esta técnica es relativamente simple, aunque costosa computacionalmente. Además, las soluciones obtenidas no necesariamente son las soluciones no dominadas globales (Ngatchou et al., 2005).

En su forma general, lo que las técnicas clásicas buscan, es aproximarse al frente de Pareto, mediante la ejecución repetida del proceso de optimización, modificando los parámetros de agregación de las funciones (factores de peso y targets), hasta obtener un número significativo de soluciones en la aproximación del frente de Pareto. Este enfoque presenta algunas limitaciones (Ngatchou et al., 2005):

- a) Requiere tener un conocimiento previo de la importancia relativa de cada una de las funciones involucradas y de los límites de los objetivos que se convierten en restricciones.
- b) La función ponderada solo proporciona una solución.

c) El frente obtenido puede ser un frente local no dominado y no necesariamente una aproximación cercana del frente global no dominado que se está buscando.

d) La distribución de soluciones a lo largo del frente depende de la eficiencia del algoritmo de optimización

e) La compensación (trade-off) entre los objetivos no puede evaluarse fácilmente.

f) No puede encontrarse una solución, cuando el espacio de búsqueda no es convexo.

De acuerdo a lo anterior, aunque son técnicas relativamente simples en su implementación, la mayoría de ellas son ineficientes, y a veces sensitivas a la forma del frente de Pareto. por lo cual esta forma de optimización no es muy confiable cuando se trata con problemas que tienen múltiples objetivos conflictivos entre sí. Lo que se desea en un problema de estas características es conocer todas las soluciones posibles del problema propuesto, para todos los objetivos, de manera simultánea (Ngatchou et al., 2005).

**3.3.2.2. Algoritmos metaheurísticos multiobjetivo.** En contraste con las técnicas clásicas basadas en agregación, existen algunas técnicas de optimización multiobjetivo conocidas como algoritmos metaheurísticos multiobjetivo, y que están orientadas hacia el cálculo directo del frente de Pareto, optimizando simultáneamente los objetivos individuales. Los avances en computación y el desarrollo de los algoritmos metaheurísticos basados en la población, ha contribuido al crecimiento de estos métodos en los años recientes. Tienen la ventaja de evaluar múltiples soluciones potenciales en una sola iteración. Adicional a ello, ofrecen una mayor flexibilidad, principalmente en casos donde no se cuenta con información previa disponible, como sucede en la mayoría de los problemas multiobjetivo de la vida real, ya que proporcionan todas las posibles soluciones, o al menos un amplio rango de ellas, para tomar una decisión de manera informada. El

reto es, sin embargo, como guiar el proceso de búsqueda para obtener las soluciones óptimas del frente de Pareto y como mantener la diversidad de las soluciones, para prevenir una convergencia prematura (Ngatchou et al., 2005).

Generalmente, los algoritmos metaheurísticos multiobjetivo son versiones adaptadas a problemas multiobjetivo de los algoritmos metaheurísticos de un solo objetivo (ej. GA (Genetic Algorithm), ES (Evolution Strategy), SA (Simulated Annealing), PSO (Particle Swarm Optimization), ACO (Ant Colony Optimization), entre otros) y se han establecido como técnicas de optimización eficientes para la solución de este tipo de problemas (Du & Swamy, 2016). Están basados en el concepto de Pareto y generalmente adoptan los siguientes componentes (Du & Swamy, 2016; Talbi, 2009):

*a) Asignación de la calidad de la solución (fitness):* El principal rol de este componente es guiar la búsqueda del algoritmo hacia las soluciones Pareto-óptimas para una mejor convergencia. Se asigna un valor escalar de aptitud al vector de funciones objetivo.

*b) Una forma de clasificación de las soluciones conocida como Pareto Ranking.*

*c) Convergencia:* Debe tener un componente de convergencia para producir un frente de Pareto con un conjunto de soluciones cercanas al frente de Pareto-óptimo verdadero lo más pronto posible.

*d) Distribución y preservación de diversidad:* El énfasis en este componente es generar un conjunto de soluciones que sea diverso, en el espacio de decisión y/o en el espacio objetivo. Las soluciones deben distribuirse de forma uniforme a lo largo del frente de Pareto para proporcionar un mayor número de soluciones no dominadas.

*e) Elitismo:* La preservación y el uso de soluciones élite (ej. soluciones Pareto-óptimas que incluyen las mejores soluciones no dominadas encontradas durante el proceso de optimización y

registradas en un “archivo” o “repositorio”) permite un desempeño robusto, rápido y mejorado del algoritmo metaheurístico.

La lista de algoritmos metaheurísticos multiobjetivo representativos incluye (Du & Swamy, 2016; G. Zavala et al., 2016; G. R. Zavala et al., 2014):

NSGA-II (Non Dominated Sorting Genetic Algorithm II), SPEA2 (Strength Pareto Evolutionary Algorithm 2), MOEA/D (Multiobjective Evolutionary Algorithm Based on Decomposition), PAES (Pareto Archived Evolution Strategy) y PBIL (Population Based Incremental Learning), MOPSO (Multiobjective Particle Swarm Optimization), OMOPSO (Optimized Multiobjective Particle Swarm Optimization), SMPSO (Strength Constrained Multiobjective Particle Swarm Optimization) y AMOSA (Archived Multiobjective Simulated Annealing).

Con el uso de los algoritmos metaheurísticos multiobjetivo se ha logrado abordar las limitaciones de los algoritmos clásicos multiobjetivo, permitiendo la exploración (o la determinación) de diferentes puntos del frente de Pareto, generando múltiples soluciones en una sola ejecución, sin información previa relacionada con la importancia relativa de los objetivos, además de no ser susceptibles a la forma del frente de Pareto. La principal desventaja de estos algoritmos es una disminución en la calidad de los resultados del proceso de optimización a medida que aumenta el número de objetivos, debido a la falta de métodos computacionalmente eficientes para la clasificación de las soluciones, además de requerir del ajuste de parámetros adicionales relacionados con el funcionamiento de los algoritmos (Ngatchou et al., 2005).

Los algoritmos metaheurísticos multiobjetivo pueden tener una amplia aplicabilidad en diversos problemas presentes en la ingeniería estructural, como las siguientes áreas: optimización de tamaño, forma y topológica de estructuras de celosía, optimización de techos para la

minimización de área y luz solar, estructuras reticulares, diseño de fachadas, costo del ciclo de vida e impacto ambiental, eficiencia energética y costos en la construcción, optimización estructural morfológica para estructuras tipo cascarón, optimización acústica, diseño arquitectónico evolutivo, optimización del diseño arquitectónico, optimización de marcos de concreto reforzado, zonificación del uso de tierras, etc. Estos métodos pueden convertirse en un componente importante para apoyar las metodologías de diseño, en la creación de estructuras innovadoras, eficientes, estéticamente agradables y con un buen desempeño estructural (Moreno-De-Luca & Begambre Carrillo, 2013).

A continuación, se hace una descripción general de los tres algoritmos metaheurísticos multiobjetivo usados en este trabajo: NSGA-II, MOPSO y AMOSA.

*3.3.2.2.1. Algoritmo NSGA-II (Non-dominated Sorting Genetic Algorithm II).* El algoritmo NSGA-II fue propuesto por K. Deb, Pratap, Agarwal, & Meyarivan (2002) y es una versión multiobjetivo del algoritmo GA (Genetic Algorithm) (Goldberg, 1989), usado ampliamente para resolver problemas de optimización. Es un algoritmo evolutivo inspirado en los principios de genética natural, la selección natural y la evolución de las especies. El NSGA-II es uno de los algoritmos metaheurísticos evolutivos multiobjetivo más usados en la resolución de problemas de optimización multiobjetivo (Kalyanmoy Deb, 2001) y tiene las siguientes características (Branke, Deb, Miettinen, & Słowiński, 2008):

a) *Usa un principio elitista:* El mecanismo de elitismo se introduce de manera global en la población y asegura que la calidad de las mejores soluciones (fitness) de la población no se deteriore. Las buenas soluciones encontradas en las generaciones iniciales (élites), tienen una alta posibilidad de mantenerse, a menos que mejores soluciones sean encontradas. Además, la

presencia de élites aumenta la probabilidad de encontrar una mejor descendencia (Kalyanmoy Deb, 2001).

- b) *Usa un mecanismo explícito para preservar la diversidad:* Usa el concepto de comparación *crowded-distance* para mantener la diversidad entre los miembros de la población (K. Deb et al., 2002)
- c) *Se enfatiza en soluciones no dominadas:* Usa el enfoque *fast nondominated sorting* para la clasificación de una población en diferentes frentes no dominados, de acuerdo al nivel o rango de dominación de cada una de sus soluciones (K. Deb et al., 2002).

3.3.2.2.2. *Algoritmo MOPSO (Multiobjective Particle Swarm Optimization).* El algoritmo MOPSO fue presentado por C. A. Coello Coello & Lechuga (2002) y es una versión multiobjetivo del algoritmo PSO (Particle Swarm Optimization) (Kennedy & Eberhart, 1995), uno de los algoritmos metaheurísticos de optimización más populares, inspirado en el comportamiento social de poblaciones de especies animales como las aves y los peces. Cada individuo miembro de la población se conoce como partícula, y su comportamiento está afectado por su mejor ubicación local histórica  $p_{best}$ , y el mejor global  $g_{best}$  de todas las partículas, lo cual permite que se beneficien de las experiencias pasadas.

El algoritmo MOPSO usa el concepto de población (partículas) y una medida de desempeño similar al valor de fitness en los algoritmos evolutivos (Coello, Pulido, & Lechuga, 2004; C.A. Coello Coello & Lechuga, 2002). El algoritmo MOPSO se basa en la idea de tener un repositorio global, en el cual, cada partícula deposita sus experiencias después de cada iteración. El archivo histórico de las mejores soluciones encontradas por una partícula puede usarse para guardar las soluciones no dominadas generadas en el pasado (similar al concepto de elitismo en el NSGA-II).

El uso de mecanismos de atracción global, combinado con un archivo histórico que guarde las mejores soluciones no dominadas encontradas previamente, motiva la convergencia hacia las soluciones globales no dominadas. La actualización del repositorio se realiza considerando un sistema geográfico (coordenadas), en términos del valor de las funciones objetivo de cada individuo o partícula. Esta técnica está inspirada en el archivo externo usado en el algoritmo PAES (Pareto Archived Evolution Strategy) (Knowles & Corne, 2000). En el algoritmo MOPSO, el repositorio también es usado por las partículas para identificar un líder que guiará el proceso de búsqueda. El mecanismo implementado es tal, que cada partícula puede seleccionar una guía diferente. Este mecanismo está basado en la generación de hipercubos, los cuales se producen dividiendo el espacio de búsqueda explorado, en este caso el espacio objetivo (C.A. Coello Coello & Lechuga, 2002).

3.3.2.2.3. *Algoritmo AMOSA (Archived Multiobjective Simulated Annealing)*. El Algoritmo AMOSA fue desarrollado por Bandyopadhyay, Saha, Maulik, & Deb (2008) y es una versión multiobjetivo del algoritmo SA (Simulated Annealing) (Kirkpatrick, Gelatt, & Vecchi, 1983), uno de los algoritmos metaheurísticos de optimización más populares. Su fuente de inspiración es el principio físico del recocido (annealing) en metales, en el cual la temperatura inicialmente se incrementa y posteriormente, esta se reduce de forma gradual (asegurando un tiempo suficiente en cada valor de temperatura) hasta alcanzar un valor muy bajo ( $T_{min}$ ), produciendo estados estables de baja energía.

El algoritmo AMOSA incorpora el concepto de Archivo (similar al algoritmo MOPSO), donde se guardan las mejores soluciones no dominadas encontradas durante el proceso de optimización. El tamaño del Archivo es definido, ya que solo es necesario obtener un número limitado de

soluciones óptimas de Pareto bien distribuidas. Se establecen dos límites para el tamaño del archivo: un límite estricto denotado como *HL* (Hard Limit) y un límite flexible denotado como *SL* (Soft Limit).

Durante el proceso de optimización, las soluciones no dominadas se guardan en el Archivo a medida que se generan, hasta que el tamaño del Archivo alcanza el valor de *SL*, después de esto, si se generan más soluciones no dominadas, estas se añaden al archivo; como se ha superado el valor de *SL*, este se reduce hasta alcanzar el valor de *HL* usando una técnica para agrupar soluciones llamada clustering (usando el método Single Linkage Algorithm). El algoritmo AMOSA usa el concepto de cantidad de dominación para calcular la probabilidad de aceptación de una nueva solución en el Archivo.

Los pseudocódigos de los algoritmos NSGA-II, MOPSO y AMOSA empleados en este trabajo y los detalles adicionales de su implementación se presentan en el Apéndice A.

### **3.3.3. Técnicas para el manejo de restricciones en problemas de optimización**

#### ***3.3.3.1. Para problemas de optimización de un solo objetivo***

*3.3.3.1.1. Técnica de penalización.* Para el manejo de las restricciones de igualdad  $h(\mathbf{x})$  y desigualdad  $g(\mathbf{x})$  pueden aplicarse estrategias de penalización, en la cual las soluciones que no son factibles (no cumplen al menos una restricción) son consideradas durante el proceso de búsqueda, añadiendo una nueva función a la función objetivo (Talbi, 2009). Las estrategias de penalización convierten el problema de optimización original, restringido, en un problema sin restricciones, usando una función de transformación (Arora, 2017b), de acuerdo a la Ecuación (3.9):

$$F(\mathbf{x}, r) = f(\mathbf{x}) + P(h(\mathbf{x}), g(\mathbf{x}), r) \quad (3.9)$$

Donde  $f(\mathbf{x})$  representa la función objetivo original a optimizar (ej. peso o energía de deformación).  $P$  es la función de penalización.  $r$ ,  $h(\mathbf{x})$  y  $g(\mathbf{x})$  representan un parámetro de penalización escalar, y las restricciones de igualdad y desigualdad respectivamente.  $F(\mathbf{x}, r)$  representa la nueva función objetivo con la penalización aplicada. La idea de  $P$  es que si hay una violación de restricciones,  $f(\mathbf{x})$  sea penalizada añadiendo un valor positivo. La forma más popular para definir  $P$  es usando la función de pérdida cuadrática (Arora, 2017b), definida en la Ecuación (3.10) :

$$P(h(\mathbf{x}), g(\mathbf{x}), r) = r \left[ \sum [h_i(\mathbf{x})]^2 + \sum [g_i^+(\mathbf{x})]^2 \right] \quad (3.10)$$

$$g_i^+(\mathbf{x}) = \max(0, g_i(\mathbf{x}))$$

3.3.3.1.2. *Técnica de torneo restringido.* Para el manejo de las restricciones también puede aplicarse una estrategia eficiente (Kalyanmoy Deb, 2000), que no requiere ningún parámetro de penalización  $r$ , y que puede usarse especialmente cuando se usan algoritmos de optimización basados en población (ej. GA y PSO). En esta estrategia se propone el uso de un operador de selección basado en un torneo binario siguiendo los siguientes criterios:

- a) Entre dos soluciones factibles (cumplen todas las restricciones), se prefiere la que tenga un menor valor en la función objetivo
- b) Cuando se compara una solución factible con una no factible (no cumple al menos una restricción), se prefiere la factible.
- c) Entre dos soluciones no factibles, se prefiere la que tenga una menor violación general de restricción

Las restricciones de igualdad  $h(\mathbf{x})$  se convierten en una restricción de desigualdad  $g(\mathbf{x})$ , usando la Ecuación (3.11):

$$g(\mathbf{x}): |h(\mathbf{x})| - \mu \leq 0 \quad (3.11)$$

En la Ecuación (3.11)  $\mu$  es un pequeño valor positivo. De esta manera, la nueva función objetivo, considerando las restricciones se define con la Ecuación (3.12):

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{si } g_i(\mathbf{x}) \leq 0 \quad \forall i = 1, 2, 3, \dots, m \\ f_{max} + \sum_{i=1}^m g_i(\mathbf{x}) & \end{cases} \quad (3.12)$$

El parámetro  $f_{max}$  es el valor de la función objetivo de la peor solución factible en la población (Kalyanmoy Deb, 2000).

**3.3.3.2. Para problemas de optimización multiobjetivo.** En el contexto de optimización multiobjetivo, cuando se usan algoritmos metaheurísticos multiobjetivo, puede usarse la técnica de torneo restringido teniendo en cuenta las siguientes consideraciones (Kalyanmoy Deb, 2001):

Los casos b) y c) usados en los problemas de optimización de un objetivo pueden usarse de la misma manera en optimización multiobjetivo. Las dificultades aparecen es con el caso a), esto es porque ahora se tienen múltiples funciones objetivo y se tiene el dilema de cual función objetivo se debe considerar. En este caso el concepto de relación de dominación (ver Sección 3.1.2.1) es de gran utilidad. A continuación, se define la condición de dominación con restricciones para dos soluciones  $\mathbf{x}_1$  y  $\mathbf{x}_2$ :

Una solución  $\mathbf{x}_1$  restringe-domina una solución  $\mathbf{x}_2$ , si se cumplen las siguientes condiciones:

- a) La solución  $\mathbf{x}_1$  es factible y  $\mathbf{x}_2$  no lo es.
- b) Las soluciones  $\mathbf{x}_1$  y  $\mathbf{x}_2$  no son factibles, pero la solución  $\mathbf{x}_1$  tiene un menor valor de violación de restricción.
- c) Las soluciones  $\mathbf{x}_1$  y  $\mathbf{x}_2$  son factibles, pero la solución  $\mathbf{x}_1$  domina  $\mathbf{x}_2$ , siguiendo la definición formal de dominación de la sección 3.1.2.1.

**3.3.4. Medidas de desempeño de los algoritmos de optimización multiobjetivo.** Para valorar de forma cuantitativa el desempeño de un algoritmo de optimización multiobjetivo, se consideran los siguientes aspectos (Coello et al., 2004) :

a) Minimizar la distancia del frente de Pareto producido por el algoritmo con respecto al frente de Pareto verdadero (asumiendo que se conoce su localización previamente).

b) Maximizar la separación de las soluciones encontradas, para tener una distribución lo más uniforme posible a lo largo del frente de Pareto.

c) Maximizar el número de soluciones encontradas en el frente de Pareto.

d) El desempeño de un algoritmo de optimización multiobjetivo puede cuantificarse usando un número apropiado de funciones de prueba, de las cuales se conoce la localización exacta de las soluciones Pareto-óptimas (en el espacio de decisión y en el espacio objetivo), es decir el frente de Pareto verdadero.

**3.3.4.1. Distancia Generacional (Generational Distance - GD).** La distancia generacional es una métrica para medir la convergencia de un algoritmo multiobjetivo, introducida por Veldhuizen (Kalyanmoy Deb, 2001). Mide la distancia promedio de las soluciones del frente de Pareto calculado  $Q$  al frente de Pareto verdadero  $P^*$ , de acuerdo a la Ecuación (3.13).

$$GD = \frac{\left(\sum_{i=1}^{\|Q\|} d_i\right)^{1/p}}{\|Q\|} \quad (3.13)$$

Para  $p = 2$ , el parámetro  $d_i$  es la distancia euclidiana (en el espacio objetivo) entre la solución  $i \in Q$  y el miembro mas cercano de  $P^*$ , como se muestra en la Ecuación (3.14).

$$d_i = \min_{k=1}^{\|P^*\|} \sqrt{\sum_{m=1}^M \left(f_m^{(i)} - f_m^{*(k)}\right)^2} \quad (3.14)$$

Donde,

$f_m^{(i)}$  es el valor de la función objetivo para el miembro  $i$  de  $Q$  en el objetivo  $m$  y  $f_m^{*(k)}$  es el valor de la función objetivo para el miembro  $k$  de  $P^*$  en el objetivo  $m$ . Un valor de  $GD = 0$  indica que todas las soluciones de  $Q$  se encuentran sobre el frente de Pareto-óptimo verdadero.

**3.3.4.2. Propagación (Spread -  $\Delta$ ).** Es una métrica de diversidad presenta por Deb (Kalyanmoy Deb, 2001), definida en la Ecuación (3.15):

$$\Delta = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{\|Q\|} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + \|Q\| \bar{d}} \quad (3.15)$$

Donde,

$d_i$  es una medida de distancia entre las soluciones de la vecindad.  $\bar{d}$  es el valor promedio de las distancias  $d_i$ . El valor de  $d_i$  puede calcularse usando el operador *crowding distance* del NSGA-II (K. Deb et al., 2002). El parámetro  $d_m^e$  es la distancia entre las solución extrema del frente de Pareto verdadero  $P^*$  y la solución extrema del frente de Pareto calculado  $Q$ , para la función objetivo  $m$  (ver Figura 17).

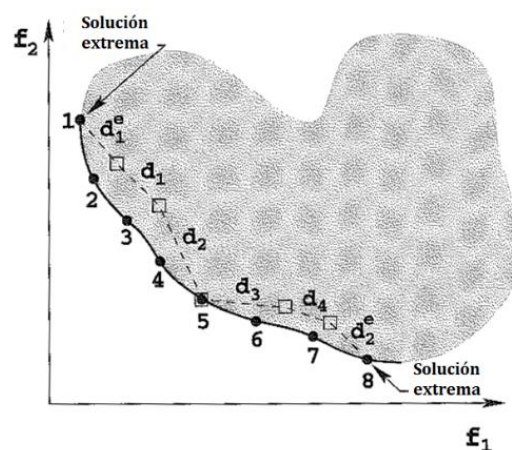


Figura 17. Distancia de las soluciones extremas  
Adaptado: Kalyanmoy Deb (2001)

Esta métrica toma el valor de cero  $\Delta = 0$  para una distribución ideal cuando  $d_j^e = 0$  y todos los valores  $d_i$  son iguales a  $\bar{d}$ . La primera condición indica que en el frente calculado  $Q$  existen soluciones óptimas del frente verdadero  $P^*$ . La segunda condición indica que la distribución de las soluciones intermedias es uniforme.

**3.3.4.3. Hipervolumen (HV).** Es una métrica que permite evaluar de forma cualitativa la convergencia y diversidad de un algoritmo forma combinada (Kalyanmoy Deb, 2001). Para problemas donde no se conoce el frente de Pareto verdadero  $P^*$  es una métrica muy popular (Du & Swamy, 2016). El hipervolumen mide el volumen (en el espacio objetivo) cubierto por los elementos del frente calculado  $Q$  (la región sombreada de la Figura 18), en problemas donde todos los objetivo son minimizados. Matemáticamente para cada solución  $i \in Q$  un hipercubo  $v_i$  se construye con un punto de referencia  $W$  y la solución  $i$  como las esquinas diagonales de hipercubo. El punto de referencia puede encontrarse construyendo un vector con los peores valores de las funciones objetivo. Después de esto la unión de todos los hipercubos calculados y su hipercubo  $HV$  es calculado:

$$HV = \text{volumen}(\cup_{i=1}^{|Q|} v_i) \quad (3.16)$$

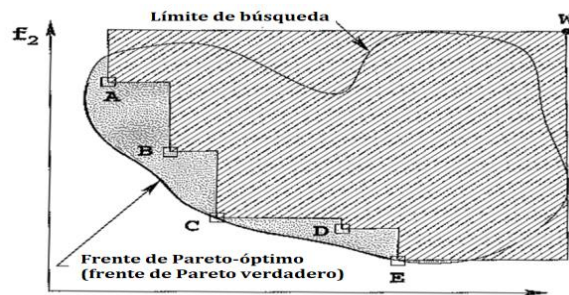


Figura 18. Hipervolumen encerrado por las soluciones no dominadas de  $Q = \{A, B, C, D, E\}$   
Adaptado: Kalyanmoy Deb (2001)

Es deseable que un algoritmo tenga un alto valor de  $HV$ . Esta métrica no es independiente de una escala arbitraria de las funciones objetivo, por lo cual se recomienda su evaluación usando una escala normalizada de los valores de las funciones objetivo.

**3.3.4.4. Suma de rangos de Wilcoxon.** Para comparar los resultados de desempeño entre dos algoritmos de optimización multiobjetivo usando las métricas de desempeño (ej. GD,  $\Delta$  y HV), se verifica si existe una diferencia estadísticamente significativa. Para ello puede usarse la prueba no paramétrica de suma de rangos de Wilcoxon (Navidi, 2015), siguiendo las recomendaciones de G. Zavala et al. (2016). La prueba permite comparar mediciones entre dos muestras y analizar la significancia de los datos obtenidos. Se usa un nivel de significancia del 5% ( $\alpha = 0.05$ ) y se establecen las hipótesis de la Ecuación (3.17):

$$\begin{aligned} H_0: \mu_1 &\geq \mu_2 \text{ Hipótesis nula} \\ H_1: \mu_1 &< \mu_2 \text{ Hipótesis alternativa} \end{aligned} \tag{3.17}$$

Donde,

$\mu_1$ : Media de las mediciones del algoritmo 1

$\mu_2$ : Media de las mediciones del algoritmo 2

Teniendo en cuenta lo anterior, una vez realizada la prueba, el valor de las métricas (GD,  $\Delta$  y HV) para dos algoritmos se compara siguiendo la siguiente convención:

a) Para la métrica GD si el valor de P (P-value) es mayor que  $\alpha$ , significa que el algoritmo 2 tuvo un mejor desempeño que el algoritmo 1.

b) Para la métrica GD si el valor de P (P-value) es menor que  $\alpha$ , significa que el algoritmo 1 tuvo un mejor desempeño que el algoritmo 2.

c) Para la métrica  $\Delta$  si el valor de P (P-value) es mayor que  $\alpha$ , significa que el algoritmo 2 tuvo un mejor desempeño que el algoritmo 1.

d) Para la métrica  $\Delta$  si el valor de P (P-value) es menor que  $\alpha$ , significa que el algoritmo 1 tuvo un mejor desempeño que el algoritmo 2.

e) Para la métrica HV si el valor de P (P-value) es mayor que  $\alpha$ , significa que el algoritmo 1 tuvo un mejor desempeño que el algoritmo 2.

f) Para la métrica HV si el valor de P (P-value) es menor que  $\alpha$ , significa que el algoritmo 2 tuvo un mejor desempeño que el algoritmo 1.

#### **4. Algoritmo de optimización topológica multiobjetivo híbrido (AOTMOH) propuesto**

Teniendo en cuenta lo expuesto en los dos capítulos anteriores, en este trabajo se desarrolló un proceso de optimización topológica multiobjetivo aplicable a estructuras de celosía planas, basado en tres de los algoritmos metaheurísticos multiobjetivo más populares, en términos del número de citas encontradas, reportados en la literatura. Estos algoritmos son: NSGA II (*Non-Dominated Sorting Genetic Algorithm*), MOPSO (*Multiobjective Particle Swarm Optimization*) y AMOSA (*Archived Multi-Objective Simulated Annealing*). La estrategia aquí planteada busca combinar las fortalezas de estos algoritmos para mejorar los resultados de convergencia y diversidad. La estrategia usada para la generación de la *estructura base*, toma como referencia los métodos descritos en la Sección 3.2.6. Se incorpora una interfaz gráfica para visualizar de manera interactiva el proceso de optimización. La implementación del proceso de optimización se lleva a cabo en el software MATLAB (The Mathworks Inc., 2017), con el apoyo de los softwares ANSYS Workbench (ANSYS Inc., 2018) y Rhinoceros-Grasshopper (Robert McNeel & Associates, 2019a, 2019b). A continuación, se hace una descripción detallada del proceso planteado:

#### 4.1. Descripción de la formulación matemática aplicada al problema de optimización topológica multiobjetivo

El proceso de optimización topológica multiobjetivo desarrollado en este trabajo se considera como un problema de optimización matemático que busca encontrar (a partir de una estructura base) el valor de las áreas de las secciones transversales de los elementos (variables de diseño), de acuerdo a la Sección 3.2.4.3, para minimizar las funciones objetivo peso  $W$  y energía de deformación  $E_s$ . La formulación matemática aplicada se deriva de la formulación general descrita en la Sección 3.2.7), con algunas simplificaciones en las restricciones y se presenta a continuación en las Ecuaciones (4.1), (4.2) y (4.3):

Encontrar las variables de diseño,  $\mathbf{x} = \{A_1, A_2, \dots, A_m\}$ , para minimizar simultáneamente las funciones objetivo  $F = [f_1, f_2]$ :

$$f_1 = W(\mathbf{x}) = \sum_{i=1}^m B_i A_i \rho_i L_i \quad (4.1)$$

$$f_2 = E_s(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^m B_i \sigma_i \varepsilon_i A_i L_i$$

$$B_i = \begin{cases} 0, & \text{si } A_i \leq A_{cr} \\ 1, & \text{si } A_i > A_{cr} \end{cases} \quad (4.2)$$

Sujeto a las restricciones:

$$h_1(\mathbf{x}): \text{Ecuación de equilibrio estático, } \mathbf{ku} = \mathbf{f}$$

$$g_1(\mathbf{x}): \text{Validez de la estructura}$$

$$g_2(\mathbf{x}): \text{Estabilidad cinemática} \quad (4.3)$$

$$g_3(\mathbf{x}): \text{Esfuerzo axial de los elementos, } |B_i \sigma_i| - \sigma_i^{max} \leq 0$$

$$g_4(\mathbf{x}): \text{Desplazamiento de los nodos en } x \text{ y } y, \quad |\delta_j| - \delta_j^{max} \leq 0$$

$g_5(\mathbf{x})$ : Esfuerzo de pandeo elementos a compresión,  $|B_i \sigma_i^{comp}| - \sigma_i^{cr} \leq 0$

$g_6(\mathbf{x})$ : Área de las secciones transversales de los elementos,

- Para variables continuas  $A_i^{min} \leq A_i \leq A_i^{max}$
- Para variables discretas  $A_i \in D_i$ , donde  $D_i = \{d_{i,1}, d_{i,2}, \dots, d_{i,s}\}$

En la formulación no se considera el peso ni el diseño de las conexiones, ni el diseño de las estructuras de soporte (apoyos).

*Nota:* En los problemas de optimización donde se tiene en cuenta la especificación ASI/AISC 360-10 (AISC, 2016) y la norma NSR-10 Título F (AIS, 2010), se aplican los requisitos de diseño del Apéndice C.2 para calcular los valores límites  $\sigma_i^{max}$ ,  $\delta_j^{max}$  y  $\sigma_i^{cr}$  de las restricciones  $g_3$ ,  $g_4$  y  $g_5$ . Además, en este caso se considera que las variables son discretas y se usa un catálogo de perfiles estructurales de acero  $D_i$  tipo L (AISC, 2017).

#### 4.2. Descripción de la estrategia para el manejo de las restricciones y funciones objetivo

Para el manejo de la restricción  $h_1(\mathbf{x})$ :  $\mathbf{ku} = \mathbf{f}$ , correspondiente al equilibrio estático, se usa el método de la rigidez directa (ver Apéndice C.1.1). Este método constituye la herramienta básica de análisis estructural para calcular los datos de esfuerzos en los elementos y desplazamientos de los nodos para la evaluación de las restricciones  $g_3$  a  $g_5$ . Su implementación se realizó en el software MATLAB (The Mathworks Inc., 2017), validando su funcionamiento con los resultados del software SAP2000 (CSI, 2018).

Las restricciones  $g_1$  a  $g_2$ , se manejan de acuerdo a la Sección 3.2.7. Para mejorar el desempeño de los algoritmos de optimización (NSGA-II, MOPSO y AMOSA) en el manejo de  $g_1$  y  $g_2$ , se usa el enfoque de reparación cinemática (*Kinematic Stability Repair*) (Richardson et al., 2012), el cual

permite generar un mayor número de estructuras estables usando operaciones de reparación (añadir y/o remover elementos en las estructura) durante el proceso de optimización topológica.

Para evaluar las restricciones  $g_3$  a  $g_5$  se usa la estrategia de torneo restringido de la Sección

3.3.3.1.2, en su versión multiobjetivo de acuerdo a la Sección 3.3.3.2. Para que tengan la misma importancia, cada una de ellas se normaliza usando la Ecuación (4.4):

$$\begin{aligned} g_3(\mathbf{x}): \quad & \frac{|B_i \sigma_i|}{\sigma_i^{max}} - 1 \leq 0 \\ g_4(\mathbf{x}): \quad & \frac{|\delta_j|}{\delta_j^{max}} - 1 \leq 0 \\ g_5(\mathbf{x}): \quad & \frac{|B_i \sigma_i^{comp}|}{\sigma_i^{cr}} - 1 \leq 0 \end{aligned} \quad (4.4)$$

La restricción  $g_6$  se cumple automáticamente durante el proceso de optimización y no es necesario aplicar ninguna estrategia de evaluación de esta restricción, ya que desde un inicio se define el espacio de búsqueda del algoritmo.

El valor de asignación del vector de funciones objetivo  $F = [f_1, f_2]$ , depende del cumplimiento de cada una de las restricciones  $g_1$  a  $g_5$ , de la siguiente manera (Kalyanmoy Deb & Gulati, 2001):

A) Si no se cumple  $g_1$ :

$$F = [10^9, 10^9] \quad (4.5)$$

B) Si no se cumple el criterio a) en  $g_2$ :

$$F = [10^8, 10^8] \quad (4.6)$$

C) Si no se cumple el criterio b) en  $g_2$ :

$$F = [10^7, 10^7] \quad (4.7)$$

D) Si se cumple  $g_1$  y  $g_2$ , pero no se cumple mínimo un  $g_i$ :

$$F = \left[ f_{1_{max}} + \sum_{i=3}^5 g_i, f_{2_{max}} + \sum_{i=3}^5 g_i \right] \quad (4.8)$$

E) Si se cumplen todas las restricciones  $g_1$  a  $g_5$ :

$$F = [W(\mathbf{x}), E_s(\mathbf{x})] \quad (4.9)$$

El valor de  $f_{1_{max}}$  y  $f_{2_{max}}$  corresponde al valor de las funciones objetivo de  $W(\mathbf{x})$  y  $E_s(\mathbf{x})$  de la peor solución factible (que cumple con las restricciones  $g_1$  a  $g_5$ ) en la población (Kalyanmoy Deb, 2000). Si no existe una solución factible, el valor de  $f_{1_{max}}$  y  $f_{2_{max}}$  es cero, hasta encontrar los valores correspondientes a la primera solución factible, durante el proceso de optimización.

### 4.3. Descripción de la estrategia de generación de la estructura base

En este trabajo, la estrategia utilizada para generar la estructura base (a partir de la cual se inicia el proceso de optimización topológica multiobjetivo), consiste en el uso de dos posibles métodos de generación:

**4.3.1. Método de niveles de conectividad de nodos (NCN).** Consiste en usar estructuras base definidas con el método descrito en la Sección 3.2.6.1. En la Figura 19 se ilustra la generación de una estructura base usando un nivel de conectividad completo entre los nodos distribuidos de manera uniforme en el dominio de diseño inicial continuo.



(a) Dominio de diseño inicial continuo

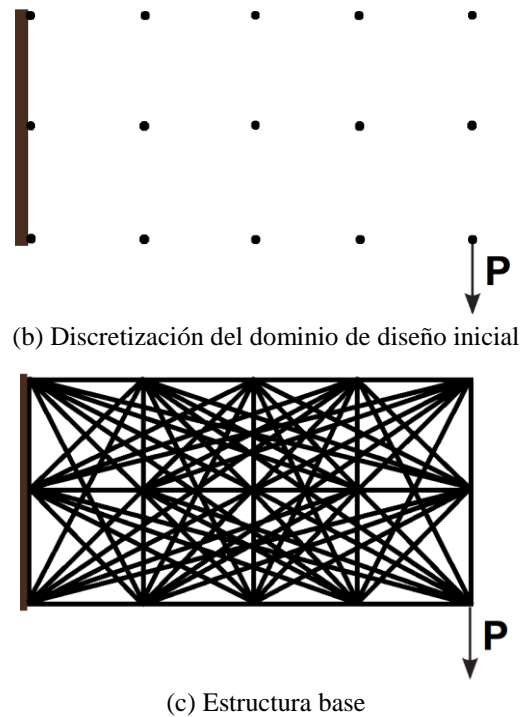


Figura 19. Representación de la estructura base usando el método NCN

**4.3.2. Método de trayectorias de esfuerzo y macroelementos (TEM).** Es la propuesta desarrollada en este trabajo y consiste en la combinación de características del *método de macroelementos* (Sección 3.2.6.2) y el *método de esfuerzos principales* (Sección 3.2.6.3). La estrategia se describe a continuación:

**4.3.2.1. Solución del problema estático.** Inicialmente se tiene un problema de análisis estático para un dominio de diseño plano y continuo con unas condiciones de borde dadas (cargas externas y condiciones de apoyo), considerando un material homogéneo, isotrópico y con comportamiento elástico de acuerdo a la Figura 20(a). Posteriormente, se usa el software de elementos finitos ANSYS (ANSYS Inc., 2018) para resolver el problema estático plano y continuo (ver Figura 20(b)). Con la ayuda de este software se obtiene un archivo de texto (.txt) (ver Figura 20(d)) con la información numérica correspondiente a la identificación, localización (coordenadas) y el estado

de esfuerzo plano ( $\sigma_x$ ,  $\sigma_y$  y  $\tau_{xy}$ ) asociado a cada uno de los nodos de los elementos finitos usados en la discretización del dominio de diseño plano en la Figura 20(c).

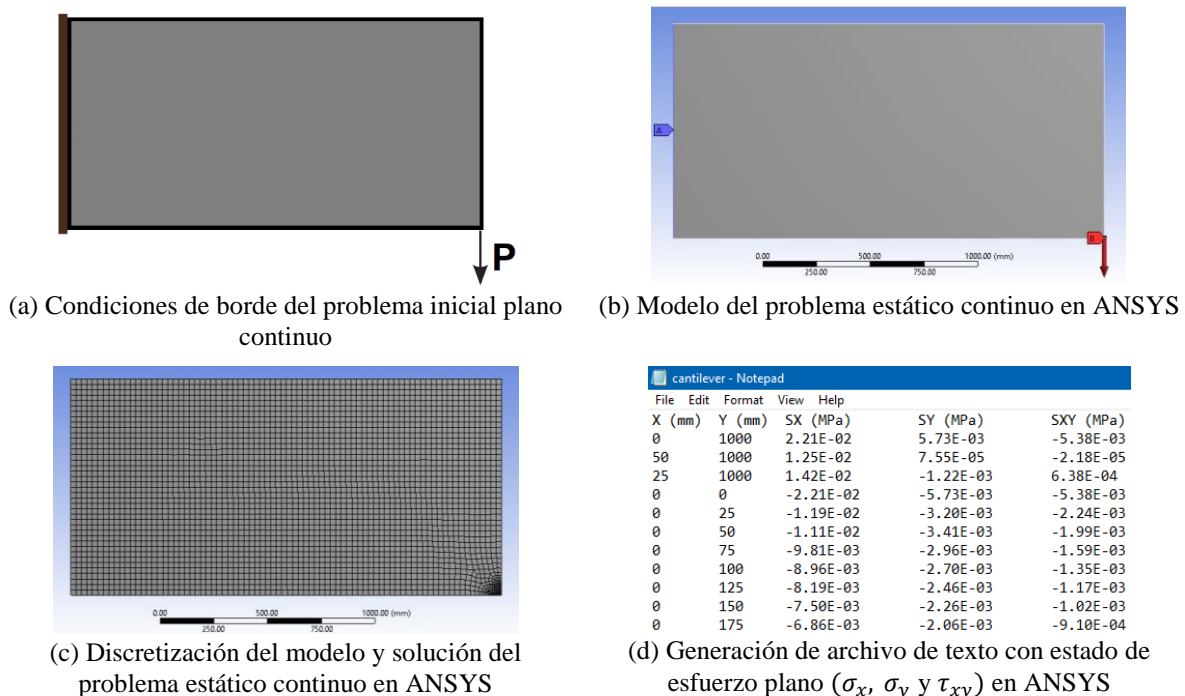


Figura 20. Cálculo del estado de esfuerzo para un problema plano y continuo

**4.3.2.2. Cálculo de las trayectorias de esfuerzo.** La información numérica del archivo de texto (.txt) de la Figura 20(d), se procesa en el software MATLAB (The Mathworks Inc., 2017), usando el método numérico descrito en el Apéndice B.2), para calcular de forma aproximada las trayectorias de esfuerzo  $\sigma_1$  (color rojo) y  $\sigma_2$  (color azul) de acuerdo a la Figura 21.

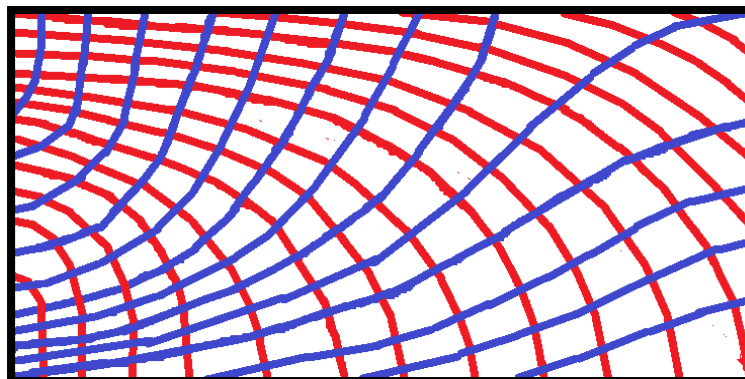


Figura 21. Trayectorias de esfuerzo

Con el método numérico del Apéndice B.2 se calcula un número de trayectorias de esfuerzo  $Num\sigma_1$  y  $Num\sigma_2$  que sea representativo y permita identificar como es la distribución global de esfuerzos principales en el dominio de diseño plano dado. Esto se logra con la interacción del usuario, quien indica un valor arbitrario de separación entre trayectorias  $a$ .

La selección de todas trayectorias de esfuerzo  $Num\sigma_1$  y  $Num\sigma_2$  de la Figura 21 (las cuales se calcularon para identificar la distribución de esfuerzos), para generar la estructura base definitiva (a partir de la cual se inicia el proceso de optimización topológica multiobjetivo), puede resultar impráctico, teniendo en cuenta que pueden generarse un número elevado de nodos y elementos (al igual que en el método estándar de la Sección 3.2.6.1 cuando se alcanza el máximo nivel de conectividad). De acuerdo a lo anterior, se plantea el uso de una estrategia que permite usar un número reducido de trayectorias de esfuerzo para la generación de la estructura base, la cual se describe a continuación:

La estrategia consiste en usar el concepto de costo de trayectoria (Gao, Li, et al., 2017). Aplicando este concepto se pueden seleccionar las trayectorias que transmiten un mayor valor de esfuerzo (lo que se traduce en mayores cargas). El concepto de costo tiene una relación directa con el objetivo general de la optimización topológica, el cual es determinar la distribución óptima de material en un dominio de diseño, dando prioridad a las zonas donde se transmiten las cargas y se presentan los mayores esfuerzos (Gao, Liu, et al., 2017; Rozvany, 1997). Los pasos abordados para implementar la estrategia de selección de las trayectorias más costosas son:

1) *Medir el costo de cada trayectoria*: Para cada una de las trayectorias de esfuerzo  $\sigma_1$  y  $\sigma_2$ , se calcula el esfuerzo principal promedio de la trayectoria ( $\overline{\sigma_1}$  o  $\overline{\sigma_2}$ , según corresponda). Como cada trayectoria es un conjunto de puntos discretos en el espacio siguiendo la dirección del esfuerzo principal ( $\sigma_1$  o  $\sigma_2$ ), el costo de la trayectoria es el resultado de calcular el valor promedio de

esfuerzo principal de los puntos que la componen. Lo anterior permite medir de forma aproximada el costo total de transmisión de cargas por dicha trayectoria.

2) *Registro de clasificación de trayectorias*: Se crea un registro en el cual se clasifican las trayectorias de cada familia ( $\sigma_1$  y  $\sigma_2$ ), de acuerdo al valor del costo. Para las trayectorias  $\sigma_1$  el registro consiste en ordenar los valores de costo en orden descendente, teniendo en cuenta que son trayectorias de esfuerzo principal máximo (o de tensión). Para las trayectorias  $\sigma_2$  se ordenan los valores en orden ascendente teniendo en cuenta que son trayectorias de esfuerzo principal mínimo (o de compresión).

3) *Identificar trayectorias prioritarias*: Para cada uno de los nodos esenciales (nodos en los cuales se ubican cargas y apoyos), se identifica la trayectoria ( $\sigma_1$  o  $\sigma_2$ ) que se origina o es derivada de dicho nodo y tiene el mayor costo, de acuerdo al registro de clasificación. De acuerdo a los cálculos preliminares realizados en este trabajo y a la literatura consultada (Gao, Li, et al., 2017), las trayectorias de mayor costo generalmente corresponden a las trayectorias derivadas de los nodos esenciales. Estas trayectorias son prioritarias y deben estar presentes (siempre) en la estructura base definitiva.

4) *Identificar trayectorias adicionales*: Para la construcción definitiva de la estructura base es necesario incluir algunas trayectorias de esfuerzo que no son derivadas de los nodos esenciales. Para ello se aplica un filtro de selección usando el registro de clasificación del paso 2). En este caso el usuario define un valor arbitrario de *porcentaje PT* (cuyo valor oscila entre 0% y 100%) de las trayectorias originales (ver Figura 21) de cada familia ( $\sigma_1$  y  $\sigma_2$ ) que quiere incluir en la estructura base definitiva. El número de trayectorias adicionales se seleccionan siguiendo el orden del registro. De esta manera es posible obtener una estructura base con un número reducido de trayectorias clave (ver Figura 22) que representan el mayor costo.

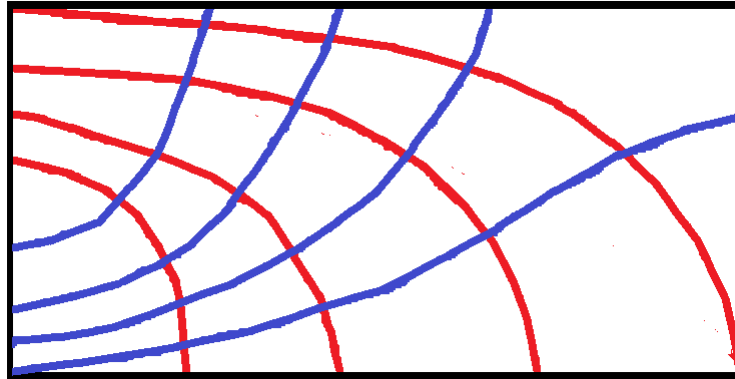


Figura 22. Trayectorias de esfuerzo clave

**4.3.2.3. Cálculo de la estructura de macroelementos.** Con la información de las trayectorias de esfuerzo clave de la Figura 22, se generan los posibles macroelementos (X. Zhang et al., 2016). Para ello se generan las uniones por medio de líneas rectas entre los puntos de intersección de las trayectorias calculadas, formando una estructura de polígonos (macroelementos) como se muestra en la Figura 23. Se usa el software MATLAB (The Mathworks Inc., 2017) .

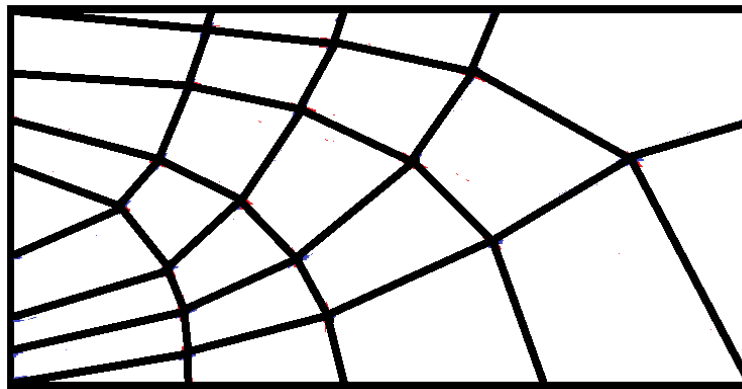


Figura 23. Estructura de macroelementos

**4.3.2.4. Cálculo de la estructura base.** A partir de la estructura de macroelementos de la Figura 23 se genera la estructura base definitiva, creando nuevas conexiones dentro de cada macroelemento cuya forma poligonal sea diferente a un triángulo, para generar una estructura con elementos triangulares que sea estable, como se muestra en la Figura 24. Se usa el software

MATLAB (The Mathworks Inc., 2017) para obtener la información de los nodos (identificación y coordenadas) y la conectividad de los elementos de la estructura base.

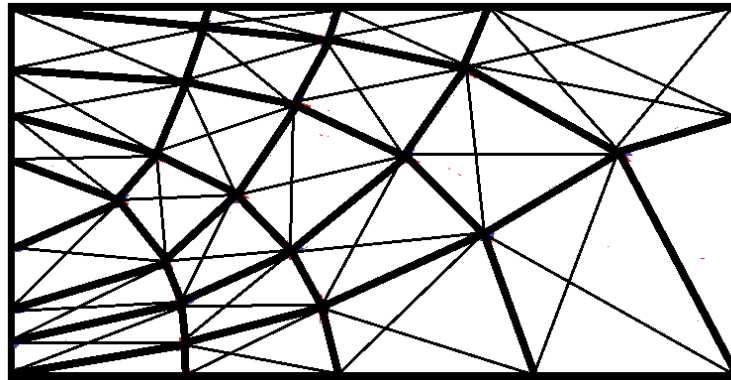


Figura 24. Estructura base usando el método TEM

#### **4.4. Descripción de la estrategia de solución del problema de optimización topológica multiobjetivo**

A partir de la estructura base de la Figura 19 o la Figura 24, se resuelve el problema matemático de optimización topológica multiobjetivo planteado en la Sección 4.1, usando los algoritmos metaheurísticos multiobjetivo NSGA-II, MOPSO y AMOSA (descritos en las Secciones 3.3.2.2.1, 3.3.2.2.2 y 3.3.2.2.3 y en el Apéndice A) como métodos de optimización. Se aplican las estrategias de la Sección 4.2 (para el manejo de las restricciones y las funciones objetivo) y Sección 4.3 (para remover o mantener elementos).

La respuesta obtenida con cada uno de los algoritmos es un frente de Pareto con las soluciones del problema de optimización distribuidas a lo largo de los dos objetivos (peso y energía de deformación), como se ilustra conceptualmente en la Figura 25.

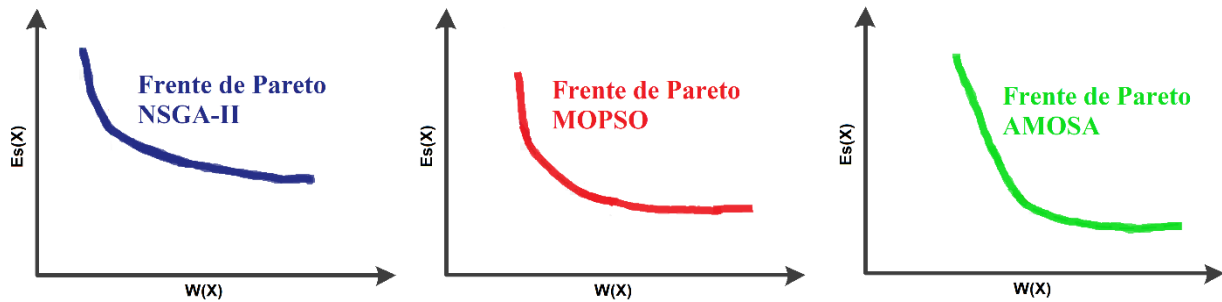


Figura 25. Representación de frentes de Pareto para el problema de optimización topológica multiobjetivo calculados con los algoritmos NSGA.II (izquierda), MOPSO (centro) y AMOSA (derecha)

Los tres frentes de Pareto de la Figura 25 se representan en una sola gráfica (ver Figura 26 izquierda) y se calculan las soluciones no dominadas resultantes de esta gráfica (Figura 26 centro). De esta manera se forma un único frente de Pareto, que en este trabajo se denomina **frente de Pareto envolvente** (Figura 26 derecha), y que contiene el conjunto de las mejores soluciones del proceso de optimización topológica multiobjetivo obtenidas con los algoritmos NSGA-II, MOPSO y AMOSA.

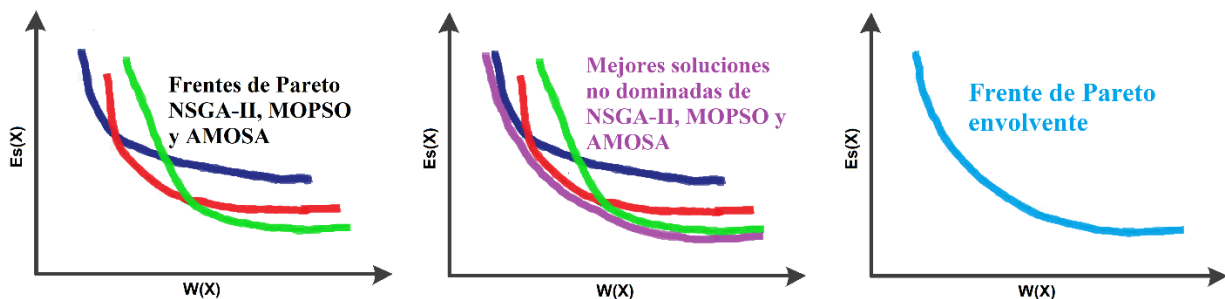


Figura 26. Representación del proceso de cálculo del frente de Pareto envolvente

La representación gráfica de la geometría (topología) óptima para algunas de las soluciones del frente de Pareto envolvente, se muestra en la Figura 27 (izquierda) para la estructura base generada con el método NCN (ver Figura 19) y en la Figura 27 (derecha) para la estructura base generada a partir del método TEM (ver Figura 24).

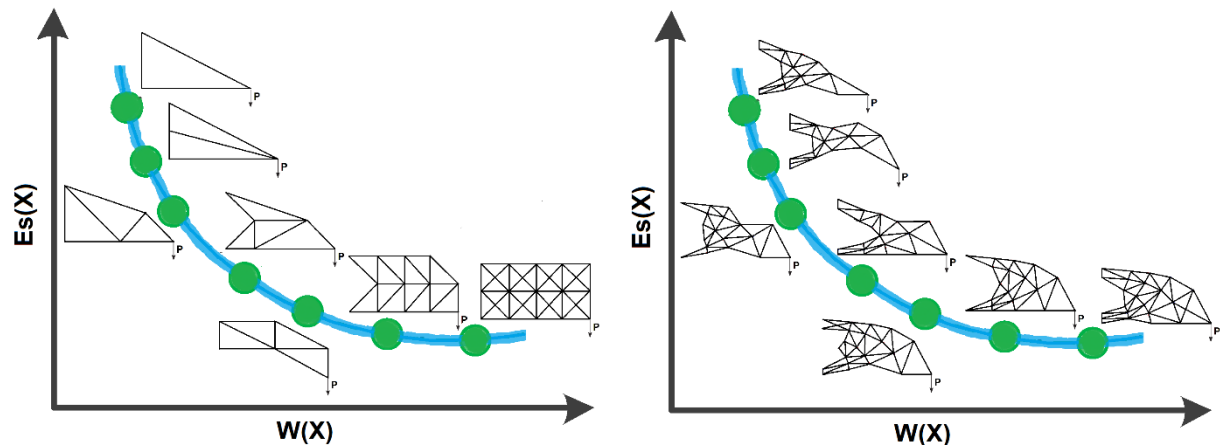


Figura 27. Representación del frente de Pareto envolvente con las soluciones óptimas de la estructura base generada con el método NCN (izquierda) y TEM (derecha)

Todas las soluciones de la Figura 27 son óptimas. El diseñador es quien toma la decisión de definir cual es la solución más adecuada del frente de Pareto envolvente, dependiendo información previa relacionada con la prioridad de las funciones objetivo, sus preferencias, experiencia, factores estéticos, ambientales, constructivos, económicos, entre otros.

#### 4.5. Descripción de la estrategia para remover o mantener elementos de la estructura base durante el proceso de optimización

En el proceso de optimización topológica de una estructura de celosía, los nodos pueden clasificarse en dos grupos: nodos importantes (lo cuales deben existir en cualquier diseño que sea factible) y nodos opcionales (se añaden para la transferencia de las cargas). Los nodos importantes, también conocidos como nodos “básicos” son los que soportan las cargas externas y donde se encuentran los apoyos. La información de estos nodos es proporcionada por el usuario y su condición de importancia debe cumplirse durante el proceso de optimización. Los nodos opcionales, también conocidos como nodos “no básicos” son los que ayudan a distribuir los esfuerzos de una mejor manera en la estructura de celosía (Kalyanmoy Deb & Gulati, 2001). De

esta manera en este trabajo, el objetivo de la optimización topológica, es encontrar cuales nodos opcionales y elementos son necesarios, y cuáles son las secciones transversales de estos elementos, para minimizar el peso y la energía de deformación. Lo anterior cumpliendo las restricciones de diseño definidas en la formulación matemática de la Sección 4.1.

La estrategia para representar la ausencia o presencia de nodos y elementos durante el proceso de optimización consiste en la comparación de las áreas de las secciones transversales de la estructura base con un pequeño valor de un área crítica  $A_{cr}$  definida por el usuario (Kalyanmoy Deb & Gulati, 2001). Si el área de un elemento es menor o igual que  $A_{cr}$ , se asume que este elemento está ausente, de lo contrario se encuentra presente. De esta manera pueden obtenerse diferentes topologías a partir de la estructura base. Para que todos los elementos tengan la misma probabilidad de aparecer o desaparecer de la estructura base, el rango inicial de variables de diseño ( $A_{min}, A_{max}$ ), se modifica definiendo el límite inferior de las variables como  $A_{min} = -A_{max}$ . Si un elemento es absolutamente esencial, los algoritmos de optimización guiarán el proceso de búsqueda para hacer que el valor del área sea un valor positivo (Kalyanmoy Deb & Gulati, 2001). Adicional a lo anterior, también se aplica el método de reparación cinemática KSR (Richardson et al., 2012) para reparar estructuras (añadiendo y/o removiendo elementos) durante el proceso de optimización (con base en un valor de probabilidad), lo cual permite aumentar el número de estructuras estables.

#### **4.6. Interfaz gráfica**

Para representar de forma gráfica la evolución de la geometría de la estructura durante el proceso de optimización topológica, se incorpora una interfaz gráfica que se basa en la integración del software MATLAB (The Mathworks Inc., 2017), con el software de diseño paramétrico

Rhinoceros-Grasshopper (Robert McNeel & Associates, 2019a, 2019b). Mediante un algoritmo de programación visual escrito Grasshopper (ver Figura 28 izquierda) se procesa la información de texto (.txt) relacionada con las coordenadas de los nodos y conectividad de los elementos (obtenida de MATLAB (The Mathworks Inc., 2017)) y se interpreta visualmente en Rhinoceros (Robert McNeel & Associates, 2019b) (ver Figura 28 derecha):

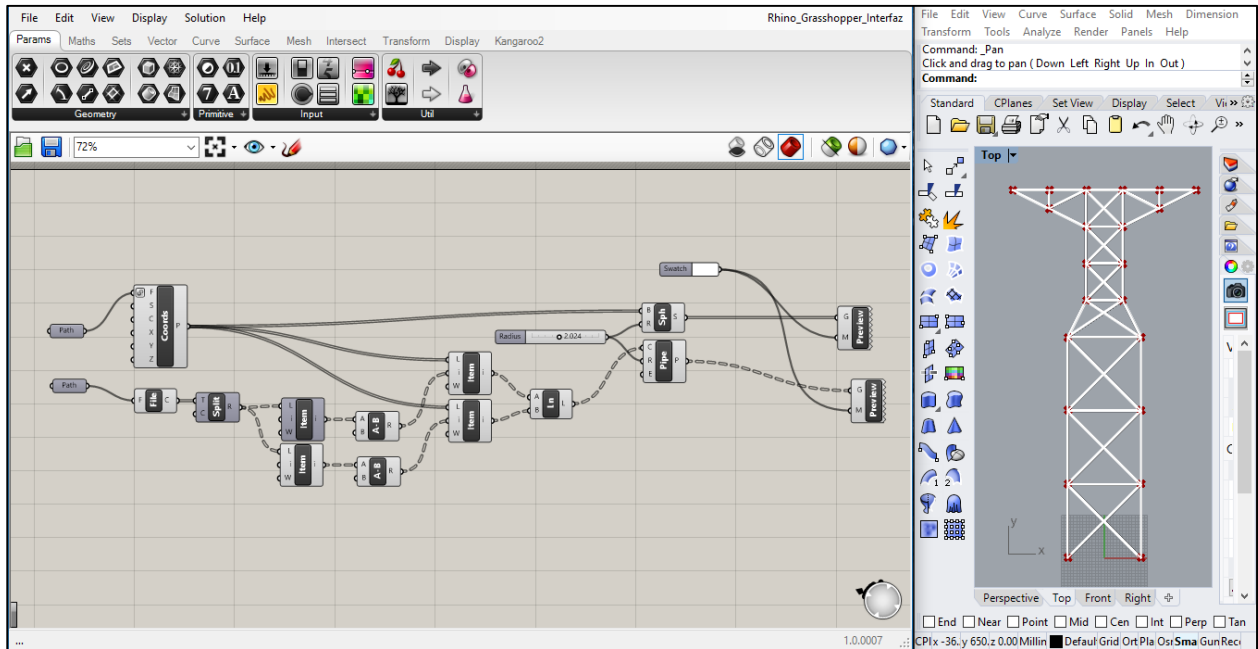


Figura 28. Interfaz gráfica en el software Rhinoceros-Grasshopper

El proceso descrito en las Secciones 4.1, 4.2, 4.3, 4.4, 4.5 y 4.6 constituye el Algoritmo de Optimización Topológica Multiobjetivo Híbrido (AOTMOH) propuesto y desarrollado en este trabajo, el cual se resume en el diagrama de flujo de la Figura 29.

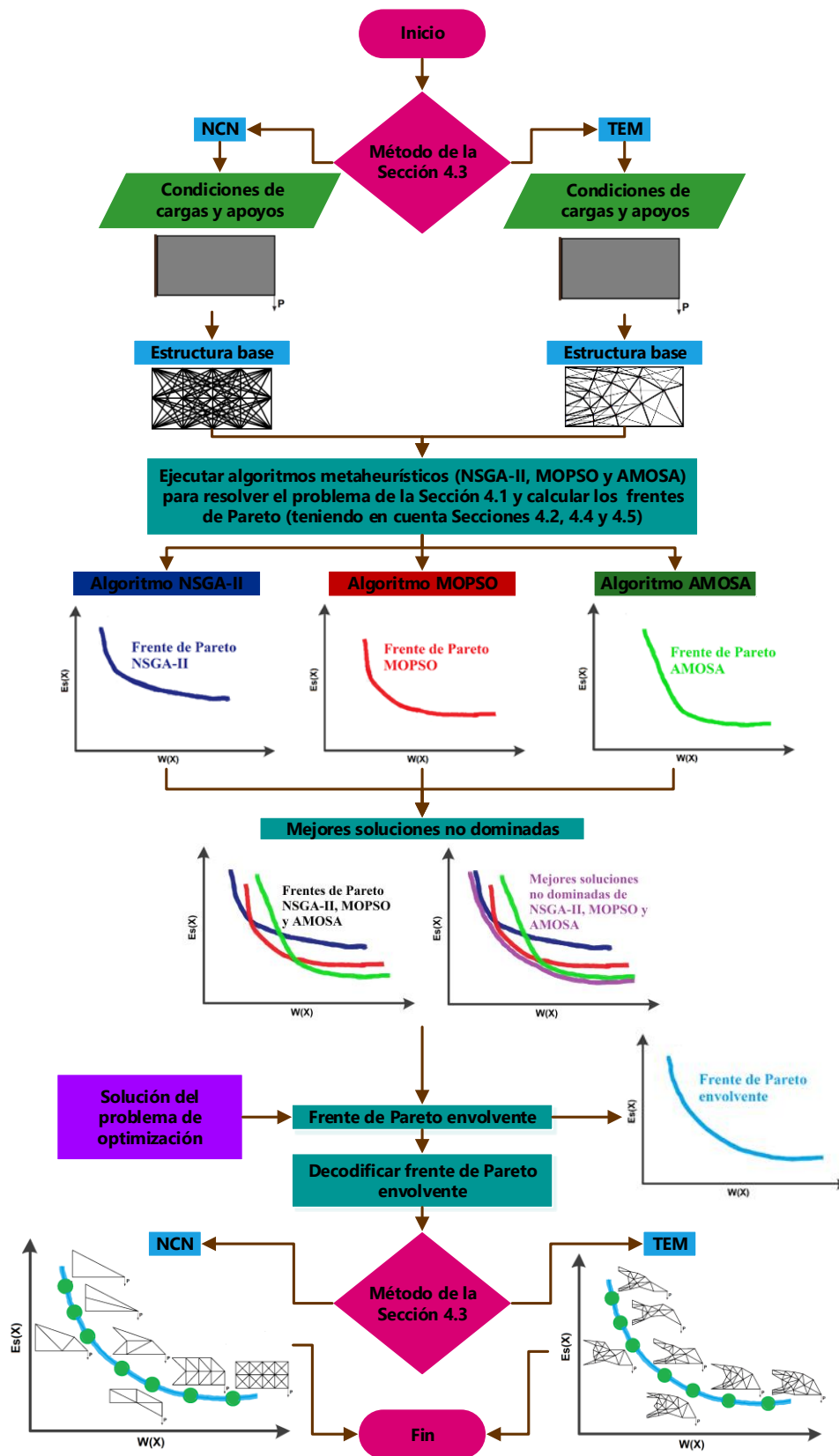


Figura 29. Diagrama de flujo del algoritmo AOTMOH

## 5. Validación de los algoritmos programados

En este capítulo se evalúa el desempeño de los algoritmos que fueron empleados en este estudio y que hacen parte del AOTMOH: Algoritmos metaheurísticos de optimización multiobjetivo (NSGA-II, MOPSO y AMOSA), procedimiento para generar la estructura base y método de análisis matricial de las estructuras de celosías planas.

### 5.1. Validación de los algoritmos metaheurísticos de optimización multiobjetivo NSGA-II, MOPSO y AMOSA

Teniendo en cuenta que el algoritmo AOTMOH se basa en los algoritmos NSGA-II, MOPSO y AMOSA, se hace necesario un estudio de validación de su desempeño. Se abordaron 14 problemas de referencia de la literatura: 8 problemas de funciones matemáticas sin restricciones y 6 problemas de funciones matemáticas con restricciones, realizando una comparación con los resultados reportados en la literatura. La implementación de los algoritmos se llevó a cabo en el software MATLAB (The Mathworks Inc., 2017), teniendo en cuenta las recomendaciones y parámetros definidos por los autores en los estudios originales (Bandyopadhyay et al., 2008; Coello et al., 2004; C.A. Coello Coello & Lechuga, 2002; K. Deb et al., 2002). Estos parámetros se presentan en la Tabla 3, Tabla 4 y Tabla 5 y se emplearon en todos los 14 problemas (a excepción de los problemas ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, en los cuales número de generaciones para NSGA-II fue de 500, y el número de iteraciones para MOPSO y AMOSA fue de 500). Todos los problemas se ejecutaron 20 veces (run) con cada algoritmo.

Tabla 3.

*Parámetros empleados en el algoritmo NSGA-II*

Parámetro	Valor
Tamaño de la población	100
Máximo número de generaciones	250
Índice de distribución para SBX $\eta_c$	20
Índice de distribución para PBMO $\eta_m$	20
Probabilidad de cruzamiento $p_c$	0.9
Probabilidad de mutación $p_m$	1/número de variables
Número de evaluaciones del vector de funciones objetivo (por cada run)	25000

Nota: Adaptado de K. Deb et al. (2002)

Tabla 4.

*Parámetros empleados en el algoritmo MOPSO*

Parámetro	Valor
Número de partículas	100
Máximo número de iteraciones	250
Tamaño del repositorio	100
Tamaño de división del repositorio	7
Coficiente de aceleración cognitivo $c_1$	2
Coficiente de aceleración social $c_2$	2
Operador de mutación	0.1
Inercia con decremento lineal	$w_{max} = 0.9$ $w_{min} = 0.4$
Número de evaluaciones del vector de funciones objetivo (por cada run)	25000

Nota: Adaptado de Coello et al. (2004)

Tabla 5.

*Parámetros empleados en el algoritmo AMOSA*

Parámetro	Valor
Número máximo de iteraciones	250
Número de iteraciones por cada temperatura	100
Número de iteraciones <i>Hill-Climbing</i>	10
Temperatura máxima	200
Temperatura mínima	1E-06
Tamaño máximo del archivo final HL	100
Tamaño máximo del archivo antes de aplicar <i>clustering</i> SL	150
Parámetro gamma $\gamma$	2
Decremento de temperatura $T = \alpha T$	$\alpha = 0.8$
Número de evaluaciones del vector de funciones objetivo (por cada run)	25000

Nota: Adaptado de Bandyopadhyay et al. (2008)

Para todas las pruebas realizadas, el equipo de cómputo usado en la optimización tiene las siguientes características: Sistema operativo Windows 10, Procesador Intel ® Core ™ i5-3230 CPU 2.60GHz y RAM: 8.0 GB.

**5.1.1. Problemas de funciones matemáticas sin restricciones.** Los problemas de funciones matemáticas sin restricciones consideradas para la evaluación de desempeño de los algoritmos se presentan en la Tabla 6 (Kalyanmoy Deb, 2001):

Tabla 6.

*Problemas de funciones matemáticas sin restricciones*

Problema	Funciones objetivo	Soluciones óptimas
(a) SCH1	$f_1(x) = x^2$ $f_2(x) = (x - 2)^2$ $-10^5 \leq x \leq 10^5$	$x \in [0,2]$
(b) SCH2	$f_1(x) = \begin{cases} -x & \text{si } x \leq 1 \\ x - 2 & \text{si } 1 < x \leq 3 \\ 4 - x & \text{si } 3 < x \leq 4 \\ x - 4 & \text{si } x > 4 \end{cases}$	$x \in \{[1,2] \cup [4,5]\}$
	$f_2(x) = (x - 5)^2$ $-5 \leq x \leq 10$	
(c) KUR	$f_1(x) = \sum_{i=1}^2 -10 \exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)$ $f_2(x) = \sum_{i=1}^3  x_i ^{0.8} + 5\sin(x_i^3)$ $-5 \leq x_i \leq 5, \quad i = 1,2,3$	Ver (Carlos A. Coello Coello, 2019)
(d) ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x) \left(1 - \sqrt{\frac{x_1}{g(x)}}\right)$ $g(x) = \frac{9}{n-1} \sum_{i=2}^n x_i$ $0 \leq x_i \leq 1, \quad i = 1, \dots, n \quad n = 30$	$x_1 \in [0,1]$ $x_i = 0$ $i = 2, \dots, 30$
(e) ZDT2	$f_1(x) = x_1$	$x_1 \in [0,1]$ $x_i = 0$

$$f_2(x) = g(x) \left( 1 - \left( \frac{x_1}{g(x)} \right)^2 \right) \quad i = 2, \dots, 30$$

$$g(x) = \frac{9}{n-1} \sum_{i=2}^n x_i$$

$$0 \leq x_i \leq 1, \quad i = 1, \dots, n \quad n = 30$$

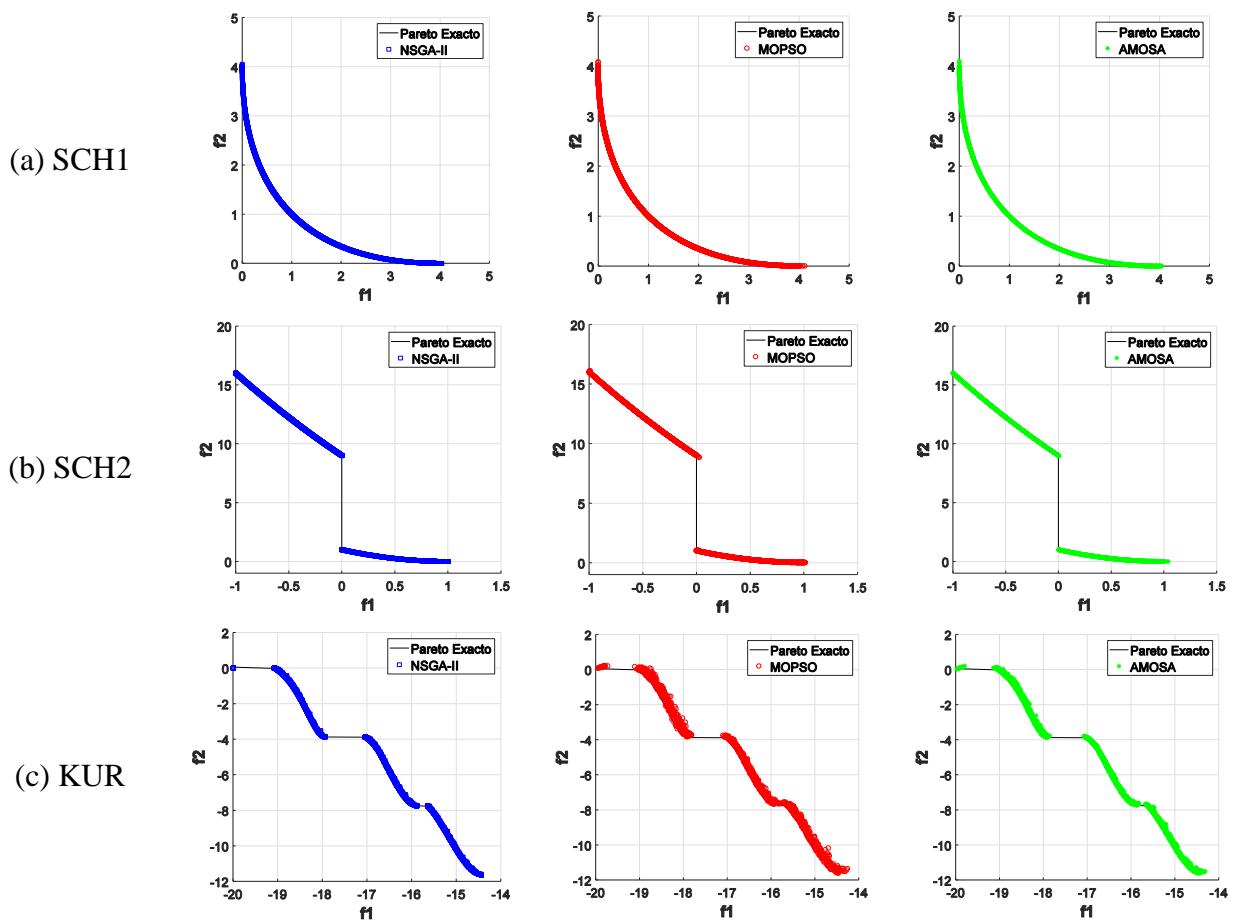
Tabla 6.

Problemas de funciones matemáticas sin restricciones (continuación)

Problema	Funciones objetivo	Soluciones óptimas
(f) ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x) \left( 1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)} \sin(10\pi x_1) \right)$ $g(x) = \frac{9}{n-1} \sum_{i=2}^n x_i$ $0 \leq x_i \leq 1, \quad i = 1, \dots, n \quad n = 30$	$x_1 \in [0,1]$ $x_i = 0$ $i = 2, \dots, 30$
(g) ZDT4	$f_1(x) = x_1$ $f_2(x) = g(x) \left( 1 - \sqrt{\frac{x_1}{g(x)}} \right)$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^n [x_i^2 - 10\cos(4\pi x_i)]$ $0 \leq x_1 \leq 1 \quad -5 \leq x_i \leq 5, \quad i = 2, \dots, n \quad n = 10$	$x_1 \in [0,1]$ $x_i = 0$ $i = 2, \dots, 10$
(h) ZDT6	$f_1(x) = 1 - \exp(-4x_1) [\sin^6(6\pi x_1)]$ $f_2(x) = g(x) \left( 1 - \left( \frac{f_1(x)}{g(x)} \right)^2 \right)$ $g(x) = 1 + 9 + \left( \frac{\sum_{i=2}^n x_i}{n-1} \right)^{0.25}$ $0 \leq x_i \leq 1 \quad i = 1, \dots, n, \quad n = 10$	$x_1 \in [0,1]$ $x_i = 0$ $i = 2, \dots, 10$

### 5.1.1.1. Resultados de desempeño en problemas de funciones matemáticas sin restricciones.

Los resultados gráficos de desempeño de los algoritmos NSGAII, MOPSO y AMOSA para cada uno de los problemas de funciones matemáticas sin restricciones, se presentan por medio de las gráficas de los 20 frentes de Pareto obtenidos por cada algoritmo, de acuerdo a la Figura 30:



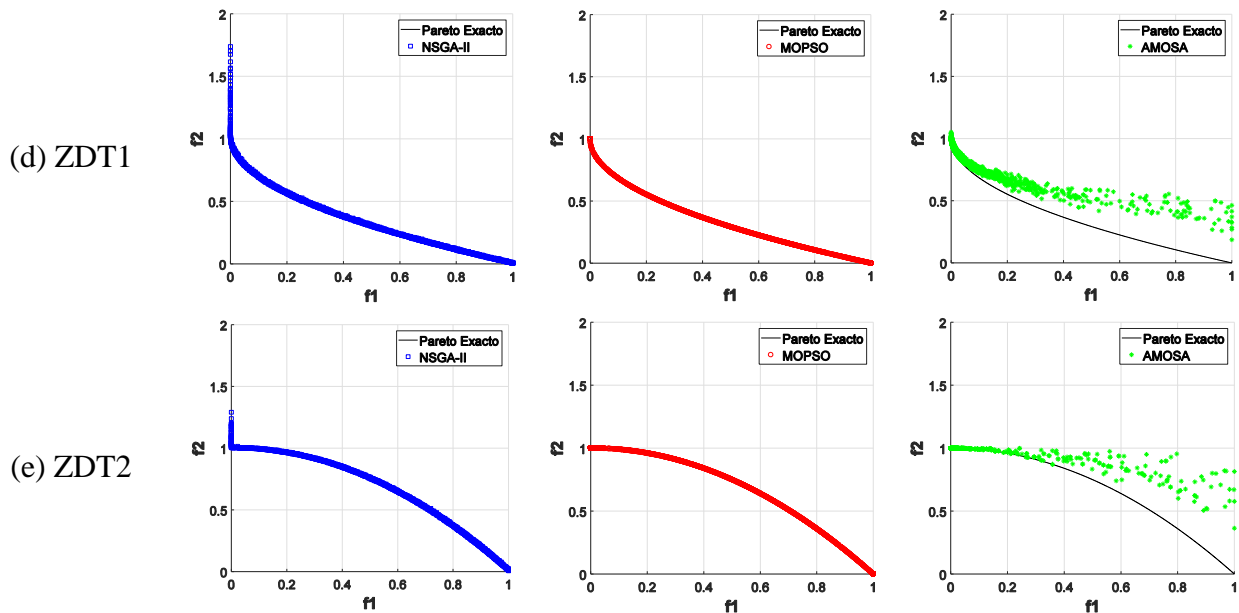


Figura 30. Frentes de Pareto obtenidos en cada una de las 20 ejecuciones, para cada algoritmo y para cada una de los 8 problemas de funciones matemáticas sin restricciones

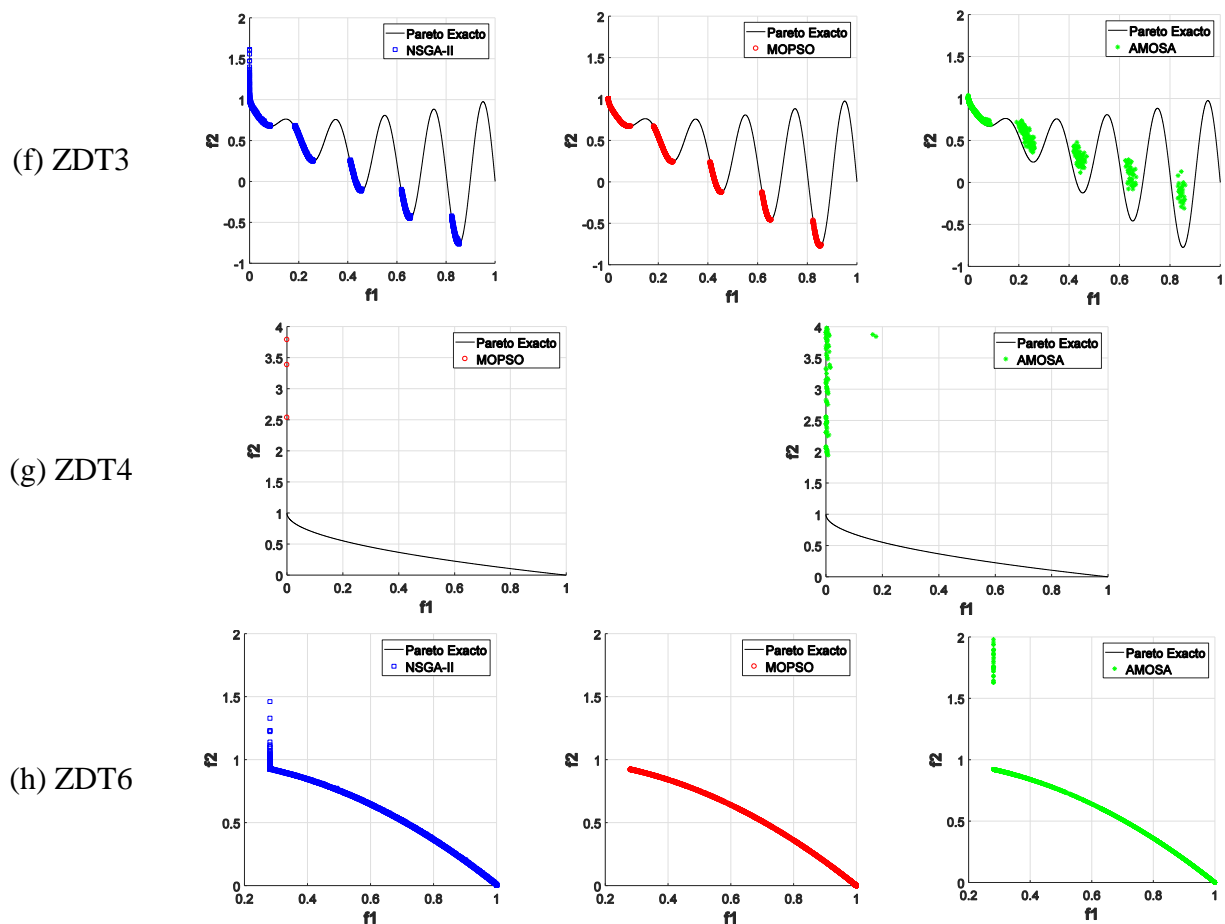
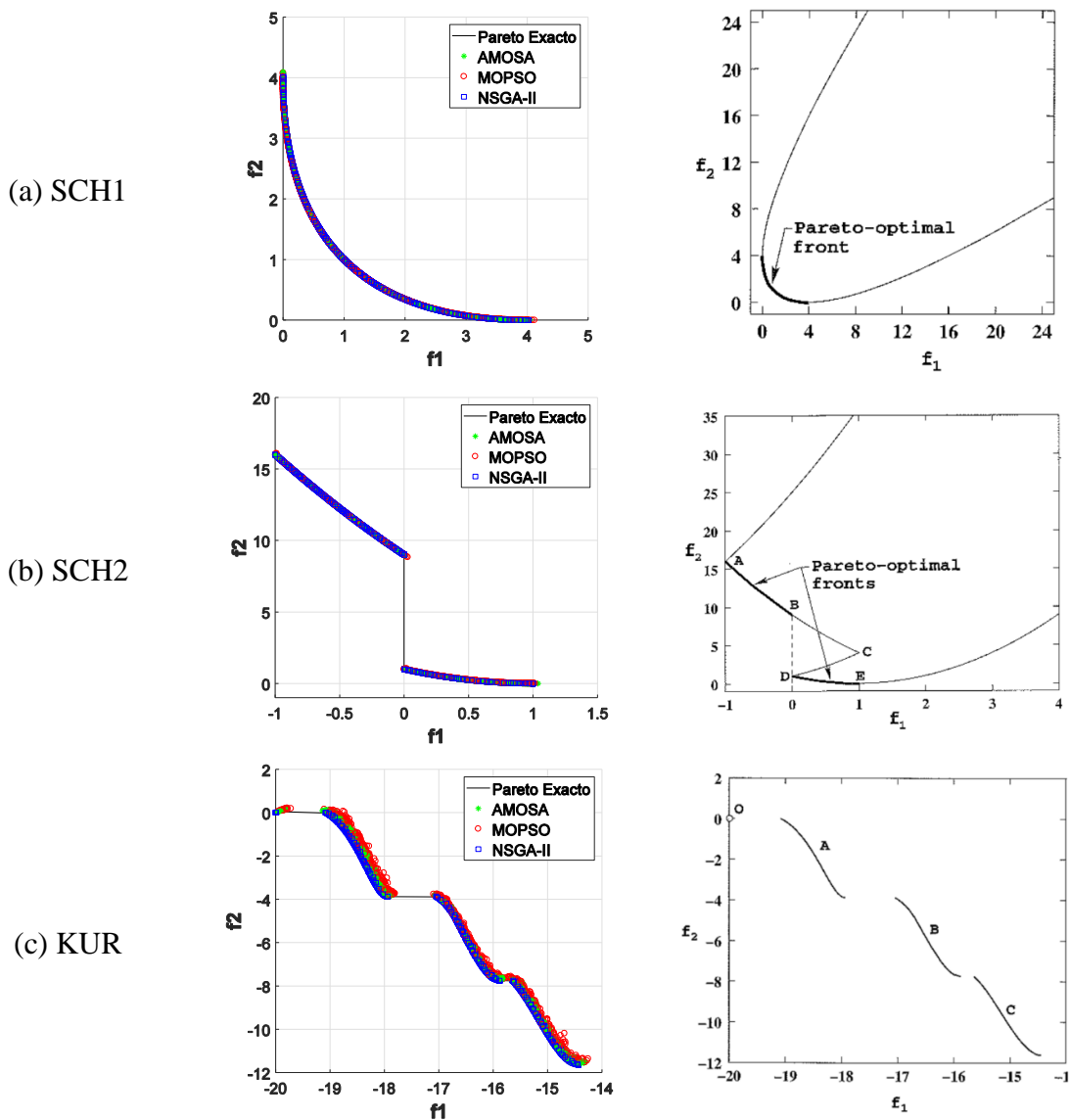


Figura 30. Frentes de Pareto obtenidos en cada una de las 20 ejecuciones, para cada algoritmo y para cada una de los 8 problemas de funciones matemáticas sin restricciones (continuación)

A continuación, en la Figura 31 (izquierda) se muestran los frentes de Pareto de los tres algoritmos en una sola gráfica (para cada uno de los 8 problemas de la Figura 30) y se comparan con las soluciones de referencia de la Figura 31 (derecha) (Kalyanmoy Deb, 2001):



(d) ZDT1

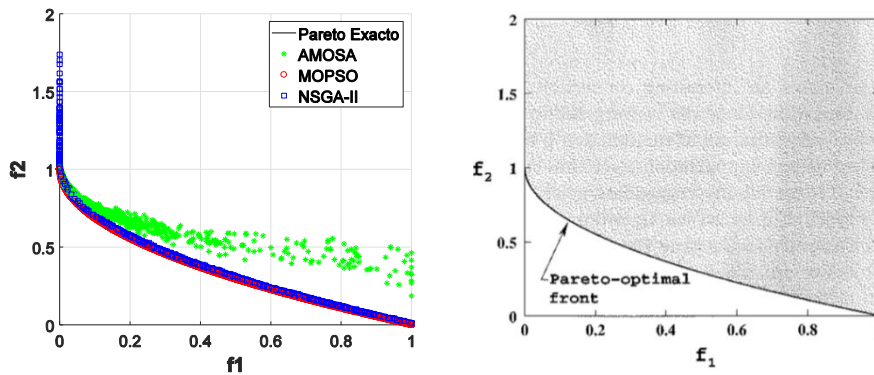
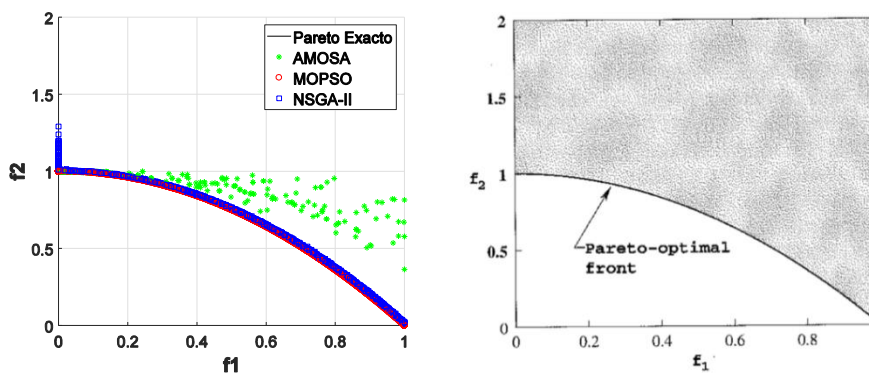
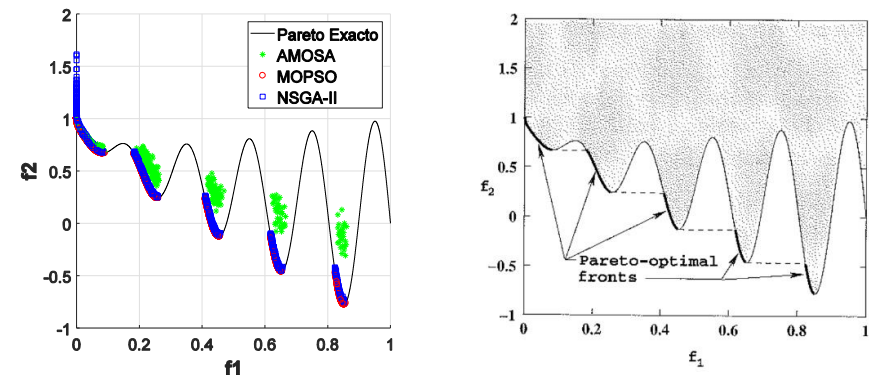


Figura 31. Comparación de todos los frentes de Pareto de los tres algoritmos (izquierda) con las soluciones de referencia (derecha), para los 8 problemas de funciones matemáticas sin restricciones

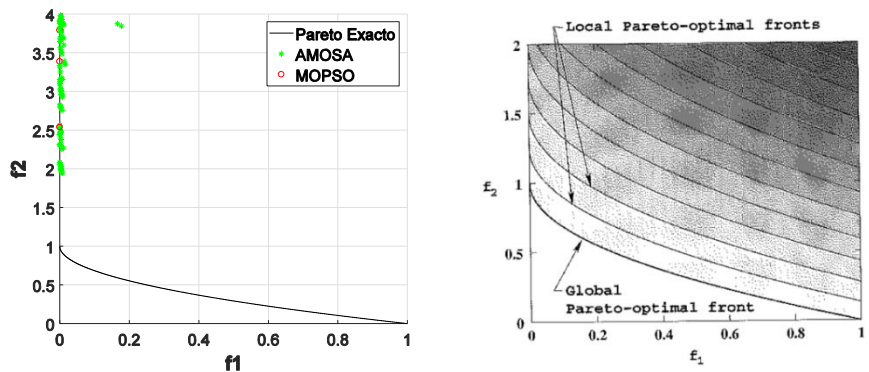
(e) ZDT2



(f) ZDT3



(g) ZDT4



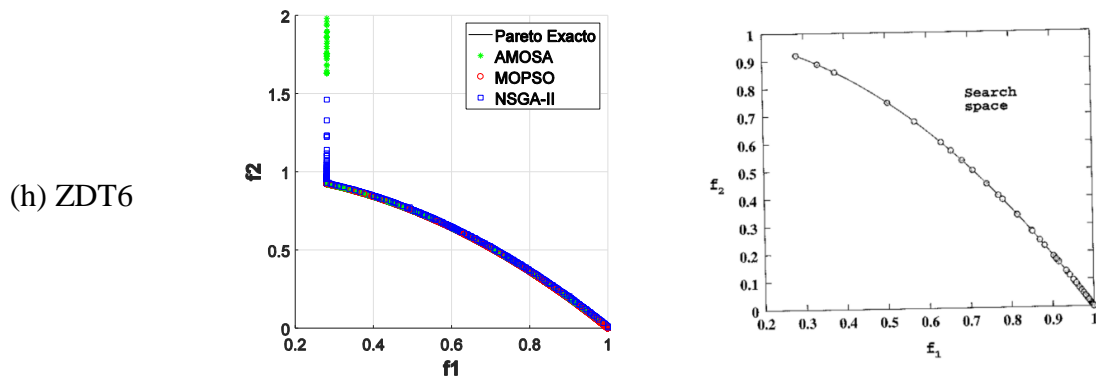


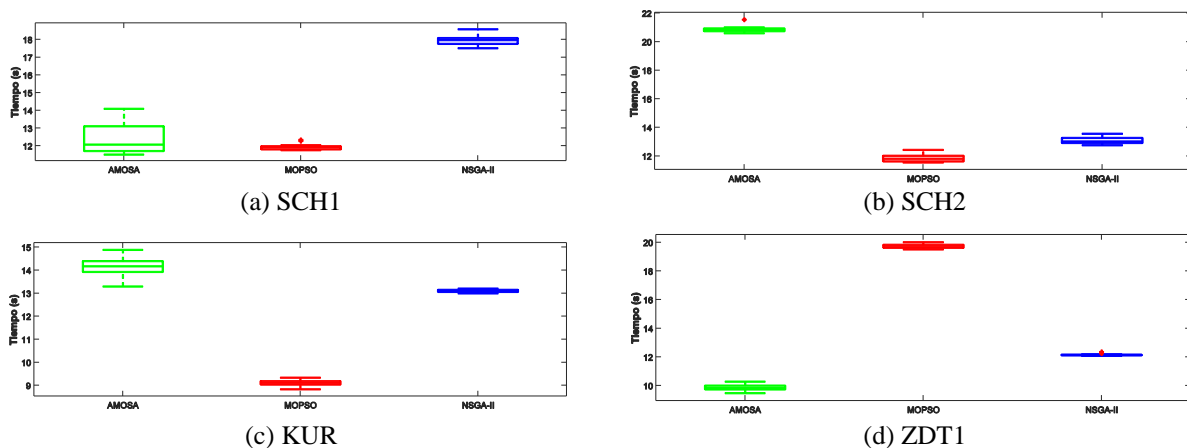
Figura 31. Comparación de todos los frentes de Pareto de los tres algoritmos (izquierda) con las soluciones de referencia (derecha), para los 8 problemas de funciones matemáticas sin restricciones (continuación)

Al comparar de forma gráfica las soluciones numéricas de la Figura 31 (izquierda) con las soluciones exactas de la Figura 31 (derecha), se observa que en los problemas SCH1, SCH2 y KUR los 3 algoritmos (NSGA-II, MOPSO y AMOSA) presentan una buena convergencia y diversidad con relación al frente de Pareto exacto. Para las funciones ZDT1, ZDT2, ZDT3, en las cuales el número de variables aumenta de forma considerable (30 variables) los algoritmos que exhiben el mejor desempeño son el NSGA-II y el MOPSO, ya que presentan soluciones cercanas al Pareto exacto (ver Figura 31 derecha), es decir buena convergencia y, además, se distribuyen de forma uniforme a lo largo del Pareto (ver Figura 31 izquierda). Para el AMOSA, en estos problemas se obtienen algunas soluciones satisfactorias, pero en general, las soluciones se concentran en algunas regiones específicas del frente de Pareto (falta de diversidad y convergencia).

El desempeño de los tres algoritmos en el problema ZDT4, muestra que estos tienen dificultad para aproximarse al frente de Pareto óptimo. El NSGA-II solo converge de forma cercana muy pocas veces cada vez que se ejecuta el programa (razón por la cual no se muestran resultados gráficos para este algoritmo). Para el MOPSO muy pocas veces converge a una solución satisfactoria, y es necesario aumentar el número de las partículas (100 mínimo) y el número de

iteraciones (mínimo 10000) para obtener una aproximación del Pareto óptimo (en algunas ocasiones). El AMOSA presenta dificultades para obtener una solución cercana. Para el problema ZDT6, los tres algoritmos presentan un buen desempeño, es decir buena convergencia y diversidad, ya que la respuesta de aproximación al frente de Pareto exacto es muy cercana (ver Figura 31).

A continuación, se muestran los resultados de evaluación de desempeño cuantitativo de los algoritmos. Los factores evaluados incluyen: costo computacional, convergencia y diversidad. Para evaluar el costo computacional se registra el tiempo empleado por cada algoritmo durante cada ejecución (run) del proceso de optimización. Como la solución analítica (Pareto exacto) es conocida, se usa la métrica de distancia generacional (GD) para medir la convergencia y la métrica de propagación ( $\Delta$ ) para medir la diversidad. Para garantizar que cada función objetivo tiene la misma importancia cuando se evalúa la calidad de la solución, su valor fue normalizado para evitar problemas de escala que pueden presentarse en algunos problemas. Como cada algoritmo se ejecutó 20 veces, los resultados de evaluación de tiempo, GD y  $\Delta$  se presentan usando los diagramas de cajas (boxplots), como se muestra en la Figura 32, Figura 33 y Figura 34:



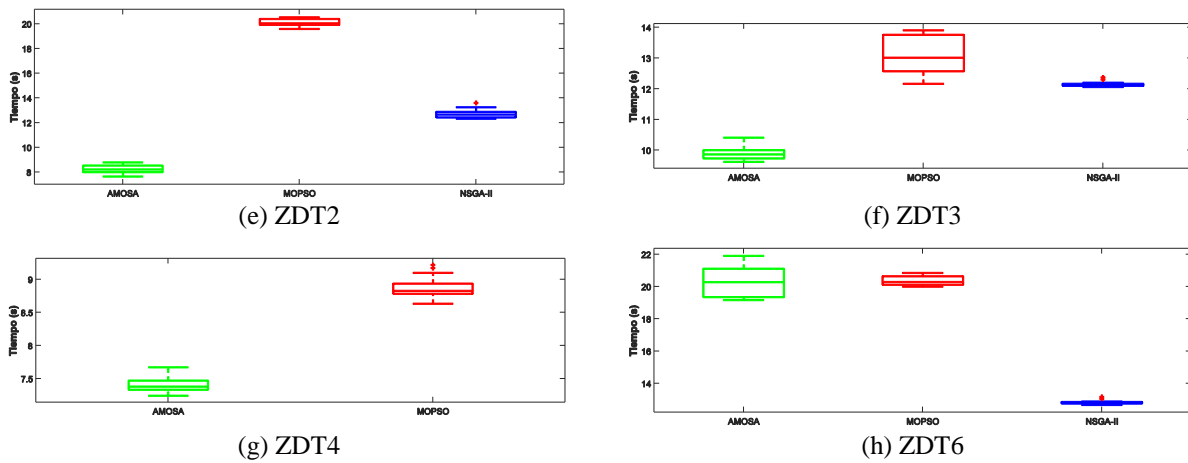


Figura 32. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de funciones matemáticas sin restricciones

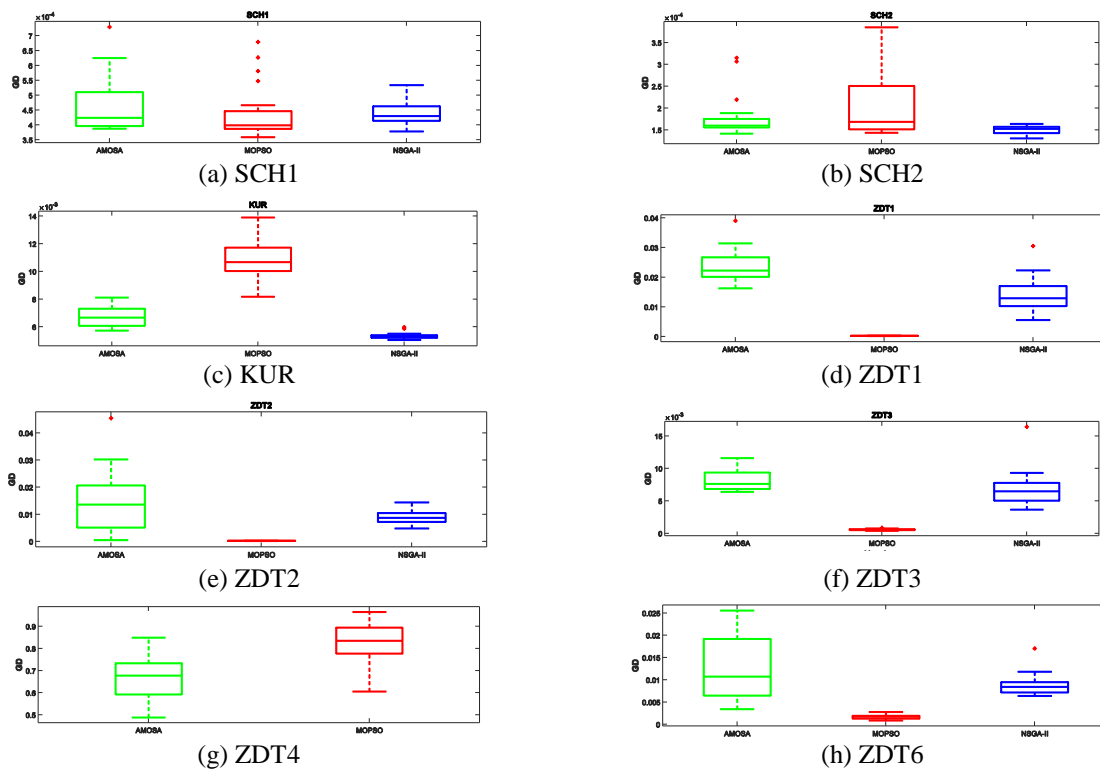
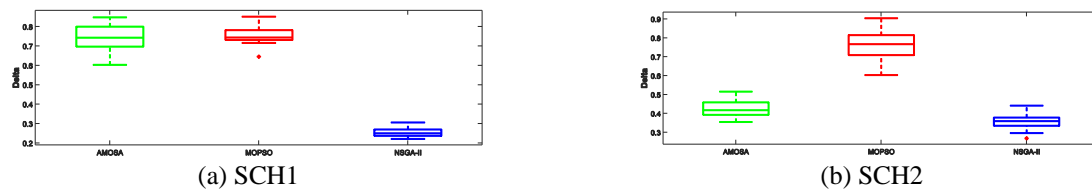


Figura 33. Diagramas de cajas para distancia generacional (GD) de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de funciones matemáticas sin restricciones



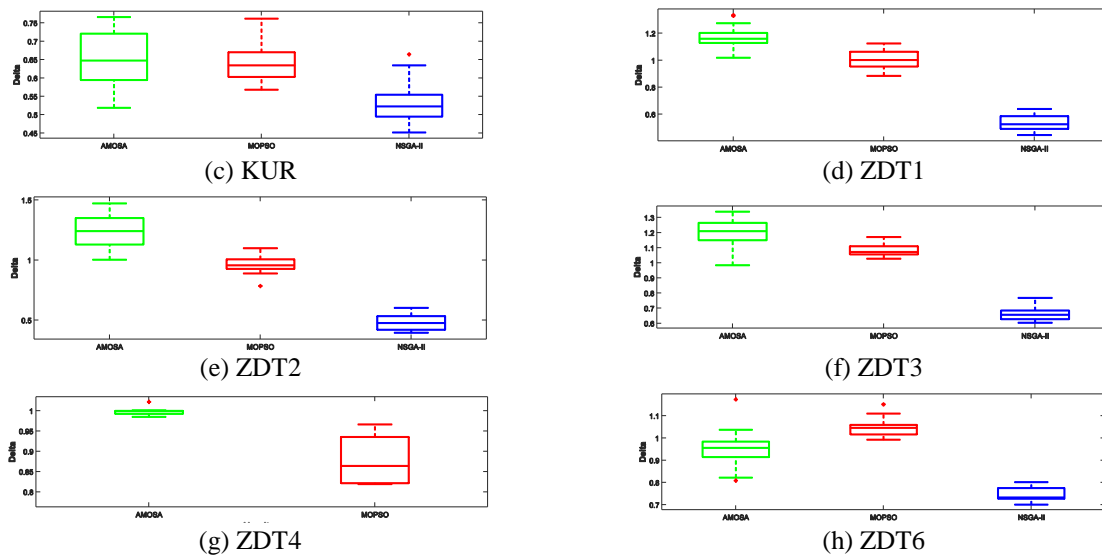


Figura 34. Diagramas de cajas para propagación ( $\Delta$ ) de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de funciones matemáticas sin restricciones

A continuación, se muestran los resultados de evaluación de desempeño cuantitativo de los algoritmos usando el test de Wilcoxon. (ver Tabla 7). Los algoritmos se compararon por medio de los 3 pares de pruebas posibles (NSGAII vs MOPSO, NSGAII vs AMOSA y MOPSO vs AMOSA) para determinar estadísticamente cual es mejor, de acuerdo a los valores de las métricas de distancia generacional (GD) y propagación ( $\Delta$ ) en las 20 ejecuciones. La convención tomada para presentar los resultados del test es: A, si el algoritmo AMOSA es estadísticamente mejor en la prueba realizada; M, el algoritmo MOPSO es estadísticamente mejor en la prueba realizada; N el algoritmo NSGA-II es estadísticamente mejor en la prueba realizada; (-) estadísticamente no hay diferencia o no hay resultados consistentes

Tabla 7.

*Criterio estadístico de Wilcoxon para GD y  $\Delta$  para problemas de funciones matemáticas sin restricciones*

Problema	NSGAII vs MOPSO		NSGAII vs AMOSA		MOPSO vs AMOSA	
	GD	$\Delta$	GD	$\Delta$	GD	$\Delta$
SCH1	M	N	A	N	M	A
SCH2	N	N	N	N	A	A
KUR	N	N	N	N	A	A
ZDT1	M	N	N	N	M	M
ZDT2	M	N	A	N	M	M
ZDT3	M	N	N	N	M	M

ZDT4	(-)	(-)	(-)	(-)	A	M
ZDT6	M	N	A	N	M	A

Con relación al costo computacional de cada algoritmo (ver Figura 32) puede observarse que, en general, el tiempo empleado en cada ejecución oscila en el intervalo de 9 a 20 segundos. Lo anterior muestra que se pueden obtener resultados en un tiempo razonable.

Al evaluar la convergencia usando la métrica de distancia generacional ( $GD$ ), se busca obtener un valor que sea lo más cercano posible a cero, el cual indica que las soluciones obtenidas por el algoritmo son cercanas a la solución analítica. En este caso, a partir de la Figura 33, para los problemas SCH1 y SCH2 los tres algoritmos tienen un valor normalizado es muy cercano a cero, lo que indica un desempeño aceptable y existe muy poca variación en cada una de las ejecuciones. Para la función KUR el NSGA-II es el que presentó una mejor convergencia ya que el valor de ( $GD$ ) fue más cercano a cero, seguido de este, el AMOSA tiene el segundo mejor desempeño y finalmente MOPSO, el que presentó el mayor valor de  $GD$ . En los problemas ZDT1, ZDT2, ZDT3 y ZDT6, el algoritmo que exhibió un valor de convergencia muy cercano a cero es el MOPSO, en comparación al NSGA-II y AMOSA respectivamente.

Cuando se evalúa la diversidad de soluciones usando la métrica de propagación ( $\Delta$ ), se busca obtener un valor que sea el mínimo posible, lo cual indica que las soluciones abarcan de manera uniforme diferentes regiones del frente de Pareto sin concentrarse en una región específica. A partir de la Figura 34, para las funciones SCH1, SCH2 y KUR el algoritmo que presenta una mejor diversidad de soluciones es el NSGA-II, seguido por el AMOSA y MOPSO respectivamente. Para las funciones ZDT1, ZDT2 y ZDT3, el algoritmo que presentó el mejor desempeño fue el NSGA-II, seguido por el MOPSO y AMOSA, sin embargo, el valor del AMOSA estuvo afectado por las soluciones lejanas al frente de Pareto que incrementaron el valor de propagación. Para la función ZDT6 el mejor  $\Delta$  lo tuvo el NSGA-II, seguido por el AMOSA y MOPSO.

De acuerdo a la Tabla 7 el algoritmo que presenta una mayor convergencia es el MOPSO (sumatoria de M en las pruebas GD igual a 10), seguido por NSGA-II y AMOSA (ambos con sumatoria de N y A en las pruebas GD igual a 6). Por otro lado, el algoritmo con mejor diversidad es el NSGA-II (sumatoria de N en las pruebas de  $\Delta$  igual a 14), seguido por MOPSO (ambos con sumatoria de M y A igual a 4).

**5.1.1.2. Conclusión de desempeño de los algoritmos en los problemas de funciones sin restricciones.** De forma general, acuerdo con la Figura 30, Figura 31, Figura 33, Figura 34 y la Tabla 7, el algoritmo que presentó la mejor evaluación de desempeño, con base en los resultados gráficos, las métricas de convergencia y diversidad fue el MOPSO (presentó la mejor convergencia), seguido de cerca por el NSGA-II (presentó la mejor diversidad) y finalmente AMOSA (presentó problemas con las funciones ZDT1, ZDT2 y ZDT3, y buenos resultados en SCH1, SCH2, KUR y ZDT6). Para la evaluación de desempeño de los algoritmos, se consideraron en conjunto los resultados gráficos y las medidas cuantitativas de convergencia y diversidad.

**5.1.2. Problemas de funciones matemáticas con restricciones.** Los problemas de funciones matemáticas con restricciones consideradas durante las pruebas se presentan en la Tabla 8 (Kalyanmoy Deb, 2001; Kalyanmoy Deb & Sundar, 2006):

Tabla 8.

*Problemas de funciones matemáticas con restricciones*

Problema	Funciones objetivo y restricciones
(a) CONSTR-EX	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = \frac{1 + x_2}{x_1}$ Sujeto a: $g_1(\mathbf{x}) \equiv x_2 + 9x_1 \geq 6$ $g_2(\mathbf{x}) \equiv -x_2 + 9x_1 \geq 1$

	$0.1 \leq x_1 \leq 1 \quad 0 \leq x_2 \leq 5$
<b>(b) SRN</b>	$f_1(\mathbf{x}) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2$
	$f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2$
	Sujeto a:
	$g_1(\mathbf{x}) \equiv x_1^2 + x_2^2 \leq 225$
	$g_2(\mathbf{x}) \equiv x_1 - 3x_2 + 10 \leq 0$ $-20 \leq x_i \leq 20 \quad i = 1,2$
<b>(c) TNK</b>	$f_1(\mathbf{x}) = x_1$
	$f_2(\mathbf{x}) = x_2$
	Sujeto a:
	$g_1(\mathbf{x}) \equiv x_1^2 + x_2^2 - 1 - 0.1 \cos\left(16 \arctan \frac{x_1}{x_2}\right) \geq 0$
	$g_2(\mathbf{x}) \equiv (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5$ $0 \leq x_i \leq \pi \quad i = 1,2$
<b>(d) BNH</b>	$f_1(\mathbf{x}) = 4x_1^2 + 4x_2^2$
	$f_2(\mathbf{x}) = (x_1 - 5)^2 + (x_2 - 5)^2$
	Sujeto a:
	$g_1(\mathbf{x}) \equiv (x_1 - 5)^2 + x_2^2 \leq 25$
	$g_2(\mathbf{x}) \equiv (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7.7$ $0 \leq x_1 \leq 5 \quad 0 \leq x_2 \leq 3$

Tabla 8.

*Problemas de funciones matemáticas con restricciones (continuación)*

Problema	Funciones objetivo y restricciones
<b>(d) OSY</b>	$f_1(\mathbf{x}) = -[25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2]$
	$f_2(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2$
	Sujeto a:
	$g_1(\mathbf{x}) \equiv x_1 + x_2 - 2 \geq 0$
	$g_2(\mathbf{x}) \equiv 6 - x_1 - x_2 \geq 0$
	$g_3(\mathbf{x}) \equiv 2 - x_2 + x_1 \geq 0$
	$g_4(\mathbf{x}) \equiv 2 - x_1 + 3x_2 \geq 0$
	$g_5(\mathbf{x}) \equiv 4 - (x_3 - 3)^2 - x_4 \geq 0$
	$g_6(\mathbf{x}) \equiv (x_5 - 3)^2 + x_6 - 4 \geq 0$
	$0 \leq x_1, x_2, x_6 \leq 10$ $1 \leq x_3, x_5 \leq 5$ $0 \leq x_4 \leq 6$
<b>(e) Viga soldada</b>	$w(\mathbf{x}) = 1.10471h^2l + 0.04811tb(14 + l)$
	$\delta(\mathbf{x}) = \frac{2.1952}{t^3b}$
	$\mathbf{x} = (h, l, t, b)$
	Sujeto a, $g_1(\mathbf{x}) \equiv 13600 - \tau(\mathbf{x}) \geq 0$

$$\begin{aligned}
 g_2(\mathbf{x}) &\equiv 30000 - \sigma(\mathbf{x}) \geq 0 \\
 g_3(\mathbf{x}) &\equiv b - h \geq 0 \\
 g_4(\mathbf{x}) &\equiv P_c(\mathbf{x}) - 6000 \geq 0 \\
 &0.125 \leq h, b \leq 5 \\
 &0.1 \leq l, t \leq 10
 \end{aligned}$$

Los términos  $\tau(\mathbf{x})$ ,  $\sigma(\mathbf{x})$  y  $P_c(\mathbf{x})$  son:

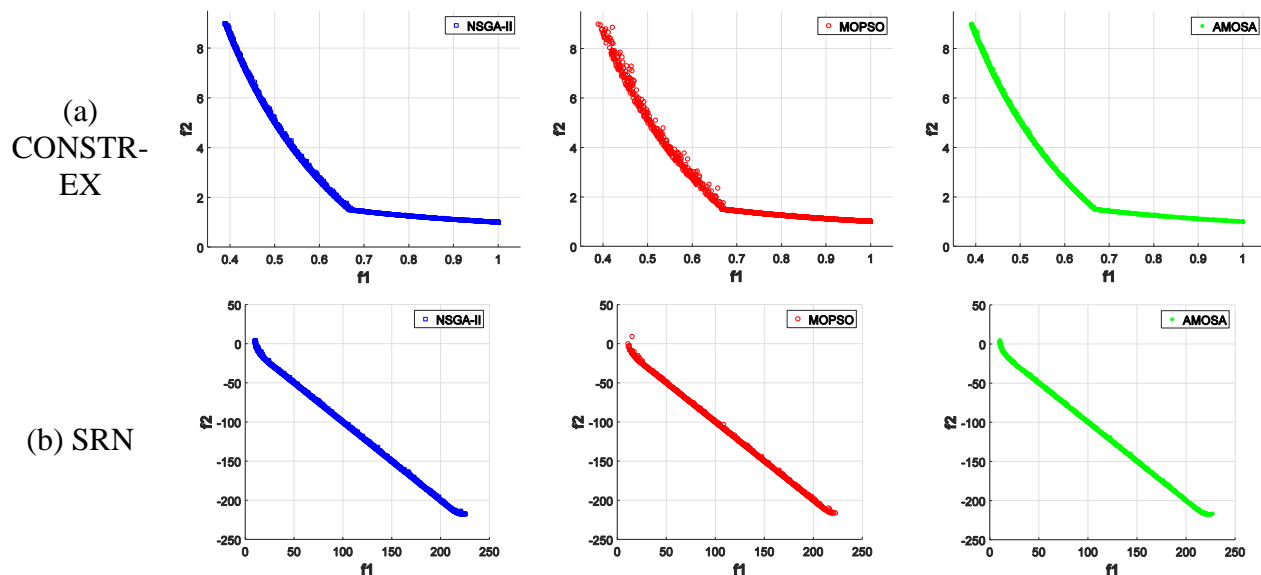
$$\begin{aligned}
 \tau(\mathbf{x}) &= \sqrt{(\tau')^2 + (\tau'')^2 + \frac{l\tau'\tau''}{\sqrt{0.25(l^2 + (h+t)^2)}}} \\
 \sigma(\mathbf{x}) &= \frac{504000}{t^2b} \\
 P_c(\mathbf{x}) &= 64746.022(1 - 0.0282346t)tb^3
 \end{aligned}$$

Donde:

$$\begin{aligned}
 \tau' &= \frac{6000}{\sqrt{2}hl} \\
 \tau'' &= \frac{6000(14 + 0.5l)\sqrt{0.25(l^2 + (h+t)^2)}}{2\left(0.707hl\left(\frac{l^2}{12} + 0.25(h+t)^2\right)\right)}
 \end{aligned}$$

### 5.1.2.1. Resultados de desempeño en problemas de funciones matemáticas con restricciones.

Los resultados gráficos de desempeño de los algoritmos NSGAII, MOPSO y AMOSA en los problemas de funciones matemáticas con restricciones, se presentan por medio de las gráficas de los frentes de Pareto obtenidos en las 20 veces que se ejecutaron, de acuerdo a la Figura 35:



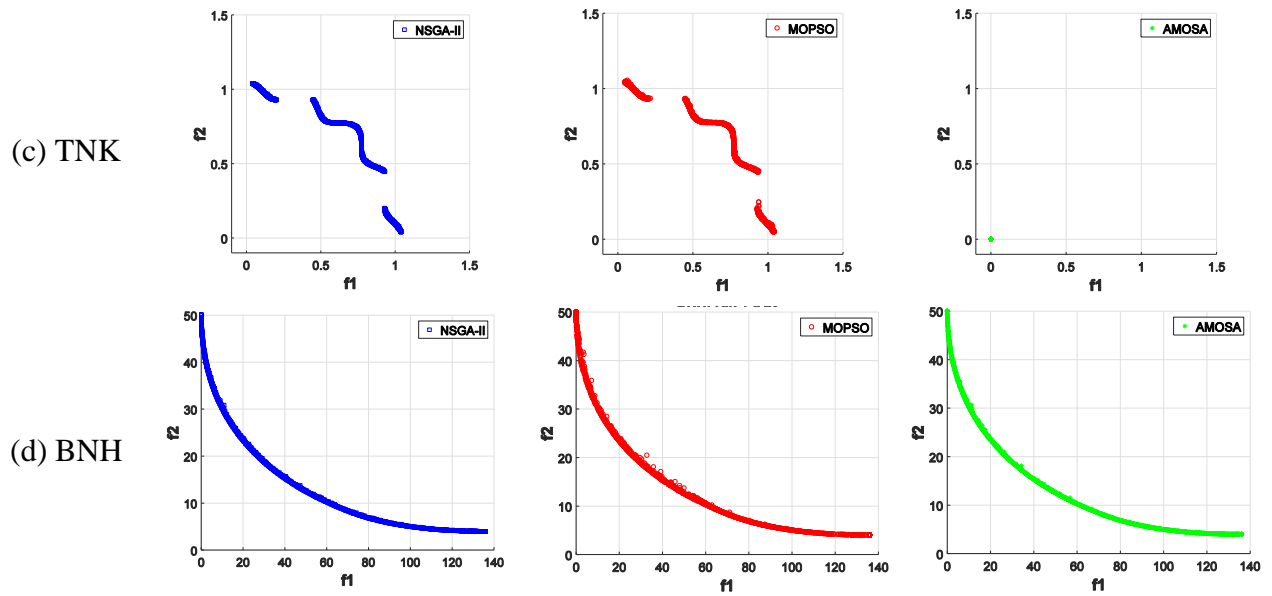


Figura 35. Frentes de Pareto obtenidos en cada una de las 20 ejecuciones, para cada algoritmo y para cada una de los 6 problemas de funciones matemáticas sin restricciones

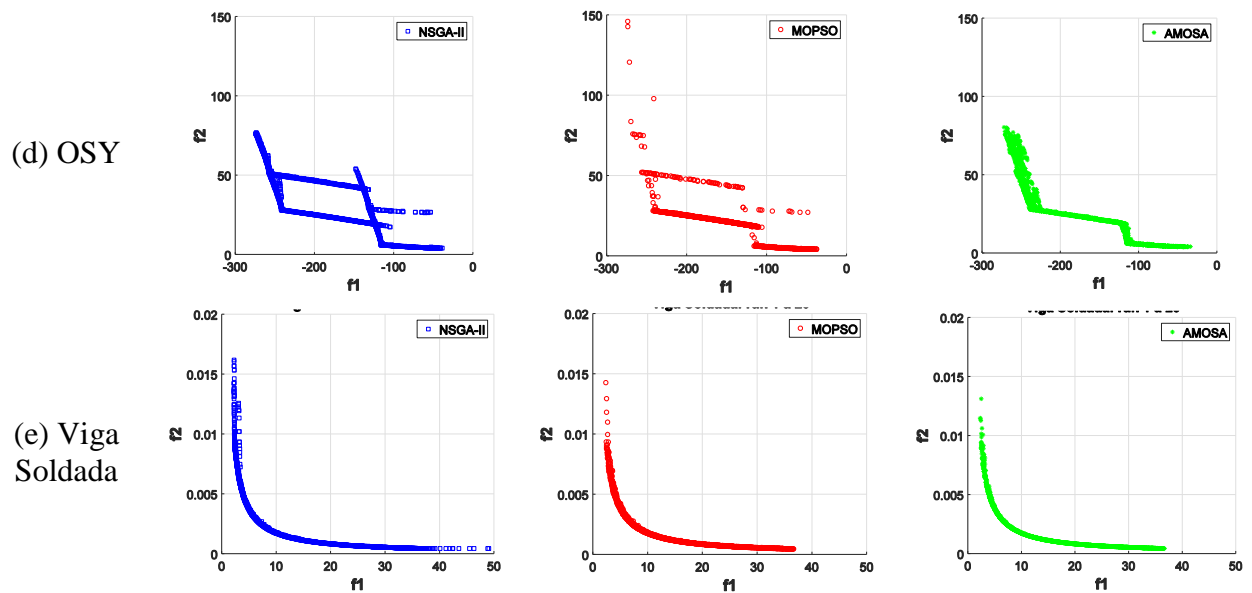


Figura 35. Frentes de Pareto obtenidos en cada una de las 20 ejecuciones, para cada algoritmo y para cada una de los 6 problemas de funciones matemáticas sin restricciones (continuación)

En la Figura 36 (izquierda) se muestran los frentes de Pareto de los tres algoritmos en una sola gráfica (para cada uno de los 6 problemas Figura 35) y se comparan con las soluciones de referencia (Figura 36 derecha) (Kalyanmoy Deb, 2001; Kalyanmoy Deb & Sundar, 2006):

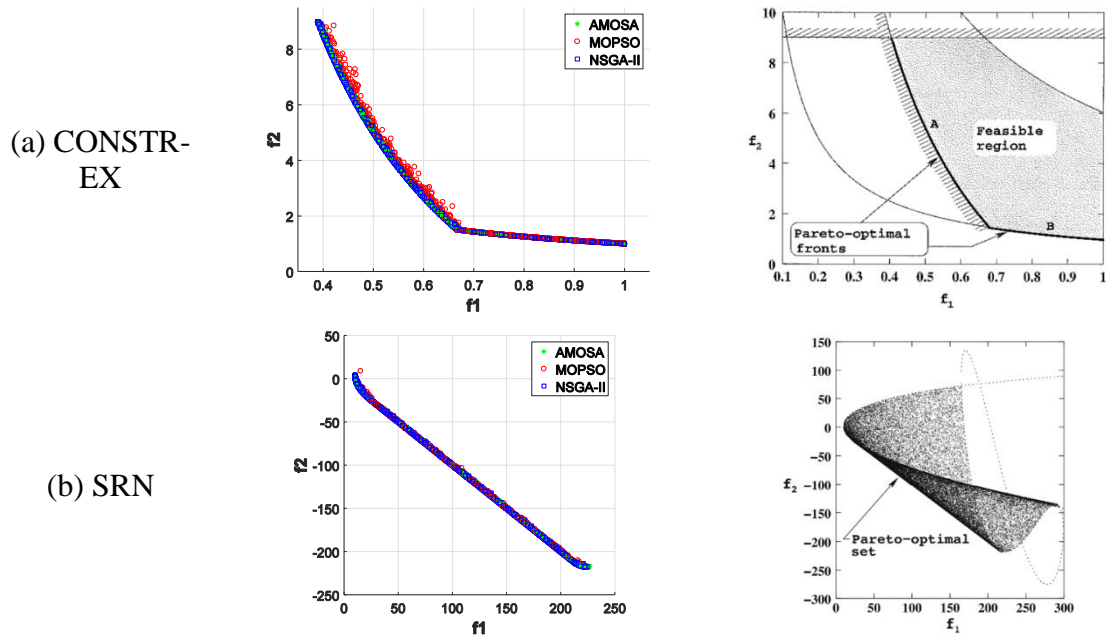
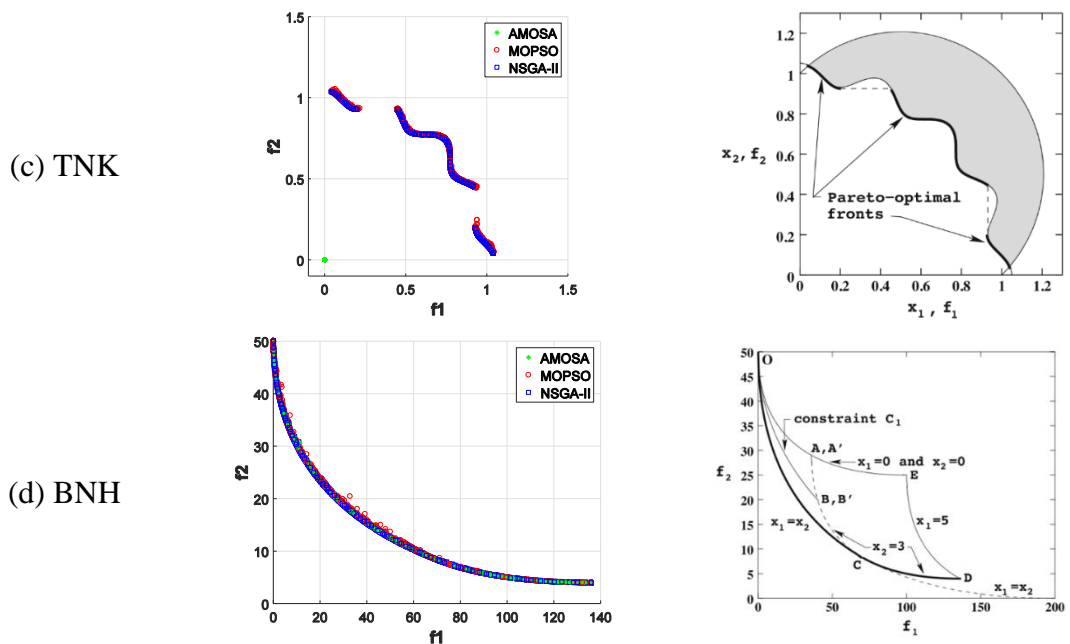
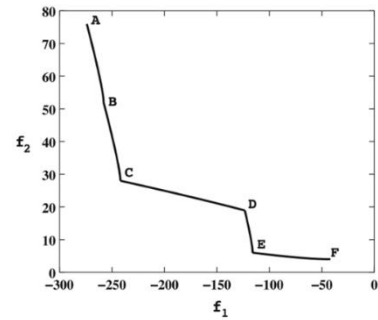
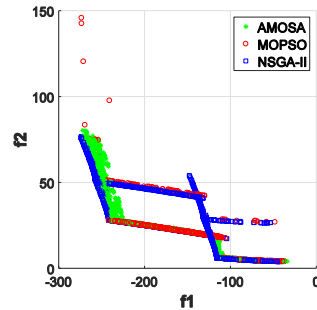


Figura 36. Comparación de todos los frentes de Pareto de los tres algoritmos (izquierda) con las soluciones de referencia (derecha), para los 6 problemas de funciones matemáticas con restricciones



(e) OSY



(f) Viga Soldada

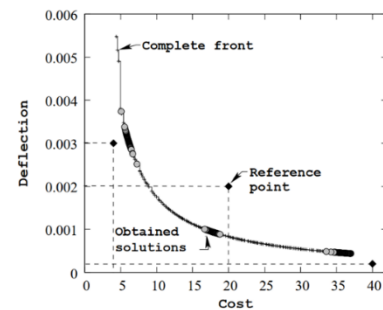
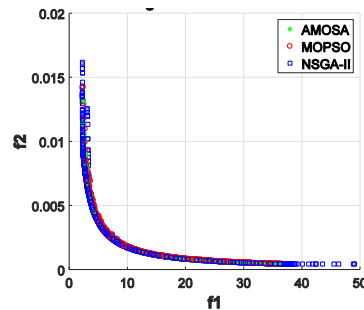


Figura 36. Comparación de todos los frentes de Pareto de los tres algoritmos (izquierda) con las soluciones de referencia (derecha), para los 6 problemas de funciones matemáticas con restricciones (continuación)

Al comparar de forma gráfica las soluciones numéricas de la Figura 36 (izquierda) con las soluciones de referencia de la Figura 36 (derecha), los tres algoritmos presentaron una buena convergencia y diversidad en los problemas CONSTR-EX, SRN y BNH, ya que las respuestas son similares entre sí y son cercanas a la referencia de estos problemas. Para el problema TNK el algoritmo NSGA-II es el que presenta el mejor desempeño, ya que la convergencia y la diversidad del frente del Pareto obtenido, es muy similar a la respuesta de referencia. El algoritmo MOPSO también converge de forma aproximada a la referencia, aunque con una menor diversidad en comparación con el NSGA-II. En este problema el algoritmo AMOSA es el que menor desempeño presenta, ya que queda atrapado en un único punto desde las iteraciones iniciales, lo que no le permite aproximarse a la respuesta de referencia.

Para el problema OSY, que presenta un mayor número de variables y restricciones, en comparación con los problemas anteriores, los tres algoritmos muestran una convergencia satisfactoria especialmente en algunas zonas del Pareto. Los algoritmos que cubren la mayor parte

de las regiones son el AMOSA y el NSGAII, es decir presentan una mejor diversidad en comparación con el MOPSO.

Para el problema de diseño de la viga soldada, el cual presenta características similares a los problemas de optimización estructural tratados en el presente proyecto (en lo que se refiere a la naturaleza de funciones objetivo y las restricciones consideradas), los tres algoritmos presentaron soluciones cercanas al frente de Pareto de referencia (Kalyanmoy Deb & Sundar, 2006).

A continuación, se muestran los resultados de evaluación de desempeño cuantitativo de los algoritmos. Los factores evaluados, al igual que en los problemas de funciones sin restricciones, incluyen costo computacional (registrando el tiempo empleado por cada algoritmo en cada ejecución), y la convergencia y diversidad (usando la métrica de hipervolumen HV). De igual manera, se normalizaron los valores de las funciones objetivo para evitar problemas de escala. Como cada algoritmo también se ejecutó 20 veces, los resultados de evaluación se presentan usando los diagramas de cajas (boxplots) como se muestra en la Figura 37 y Figura 38.

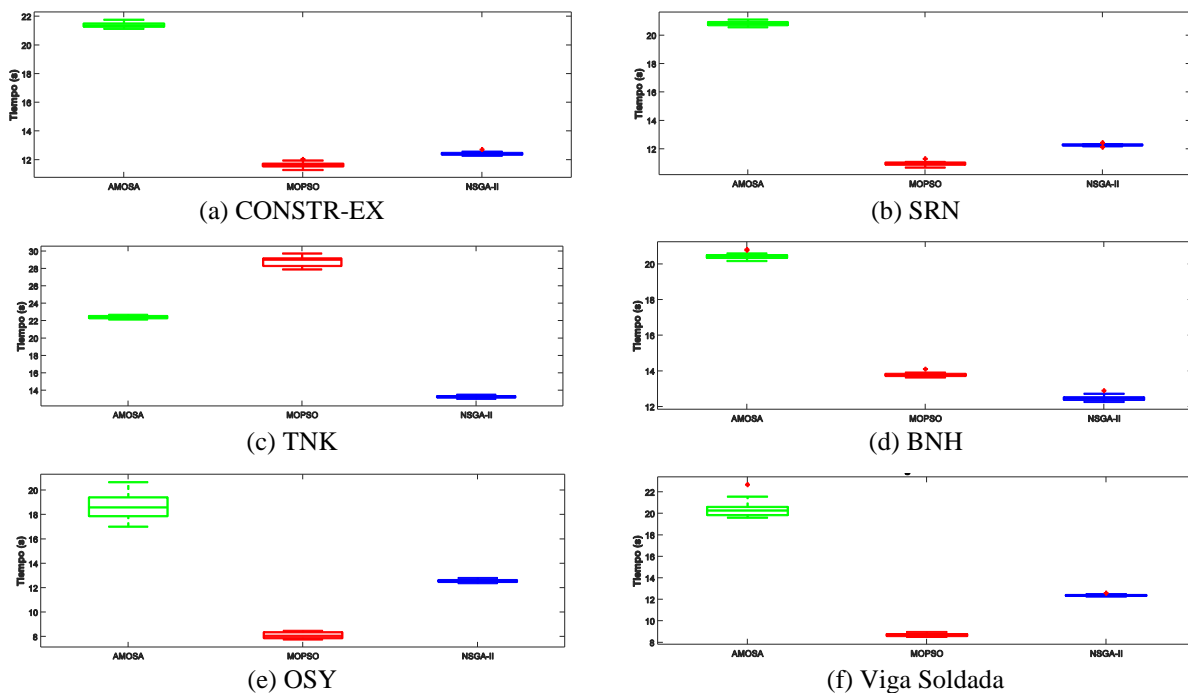


Figura 37. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de funciones matemáticas con restricciones

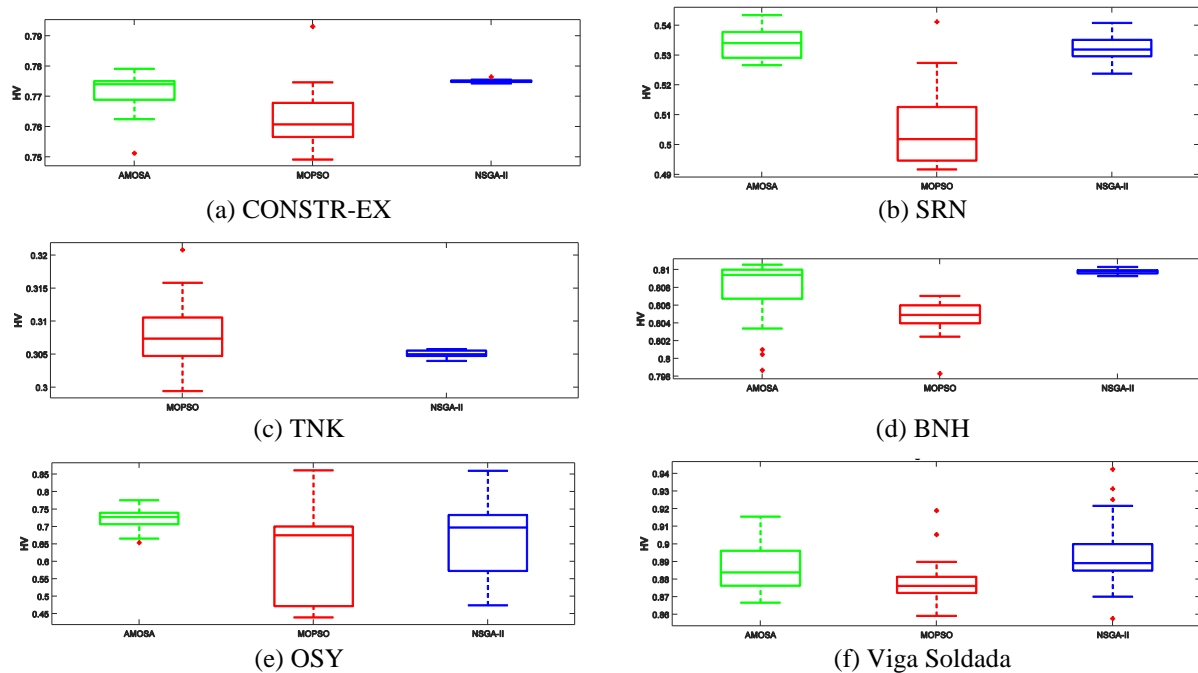


Figura 38. Diagramas de cajas para hipervolumen (HV) de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de funciones matemáticas con restricciones

A continuación, se muestran los resultados de evaluación de desempeño cuantitativo de los algoritmos usando el test de Wilcoxon (ver Tabla 9).

Tabla 9.

*Criterio estadístico de Wilcoxon para HV para problemas de funciones matemáticas con restricciones*

	NSGAI vs MOPSO	NSGAI vs AMOSA	MOPSO vs AMOSA
Problema			
CONSTR-EX	N	N	A
SRN	N	N	A
TNK	M	(-)	(-)
BNH	N	N	A
OSY	N	A	A
Viga Soldada	N	N	A

Al observar la Figura 37 se puede afirmar que el tiempo empleado por cada algoritmo en una ejecución oscila entre 9 y 20 segundos, lo cual muestra que las restricciones no tienen una influencia significativa en el proceso de optimización y también se pueden obtener resultados en un tiempo corto (al igual que en los problemas de funciones sin restricciones).

Cuando se evalúa la convergencia y diversidad usando la métrica de hipervolumen (HV), las funciones objetivo deben ser minimizadas, y lo que se busca es obtener un valor que sea el máximo posible, lo cual indica que el frente de Pareto obtenido es cercano a los valores mínimos de las funciones objetivo y abarca un volumen máximo en la dirección de los objetivos. Para los problemas CONSTR-EX, SRN y BNH (Figura 38) los 3 algoritmos presentan un valor parecido de hipervolumen. El mejor algoritmo, con un valor ligeramente mayor es el NSGA-II para las funciones CONSTR-EX y BNH, seguido por el algoritmo AMOSA y MOPSO. Para la función SRN el desempeño es levemente mayor para el algoritmo AMOSA, seguido por el NSGA-II y MOPSO. En general los 3 algoritmos presentan una convergencia y diversidad similar. En el problema TNK el valor del hipervolumen es muy similar para los algoritmos NSGA-II y MOPSO, siendo ligeramente superior para el MOPSO. Para el algoritmo AMOSA, no fue posible calcular este valor, ya que las soluciones se concentraron en un único punto y no existe una distribución de soluciones en diversas regiones del frente de Pareto. Para el problema OSY, el cual presenta un mayor número de variables y restricciones, en comparación con los problemas anteriores, el mejor algoritmo es el AMOSA, ya que presenta un mayor valor de hipervolumen. Para los algoritmos NSGA-II y MOPSO el desempeño es muy similar siendo ligeramente superior para el NSGA-II (ver Figura 38) y finalmente en el caso de diseño de la viga soldada, los 3 algoritmos mostraron resultados de desempeño muy cercanos, ya que el valor del hipervolumen es similar (HV entre 0.85 y 0.92), siendo ligeramente superior para el algoritmo NSGA-II. Los siguientes de acuerdo al orden de desempeño son el AMOSA y el MOPSO.

De acuerdo a la Tabla 9 el algoritmo que presenta una mayor convergencia y diversidad es el NSGA-II (sumatoria de N en la prueba HV igual a 9), seguido por AMOSA (sumatoria de A en la prueba GD igual a 6) y MOPSO (sumatoria de M en la prueba GD igual a 1).

**5.1.2.2. Conclusión de desempeño de los algoritmos en los problemas de funciones con restricciones.** De forma general, de acuerdo a la Figura 36, Figura 37, Figura 38 y a la Tabla 9, los tres algoritmos tuvieron una evaluación de desempleo similar, siendo ligeramente superior el NSGA-II, en relación al AMOSA y MOPSO. El algoritmo AMOSA solo presenta dificultades en el problema TNK, ya que, en los demás problemas, incluyendo el problema de diseño de la viga soldada su desempeño es satisfactorio y similar a NSGA-II y MOPSO.

**5.1.3. Conclusión general de desempeño de los algoritmos NSGA-II, MOPSO y AMOSA en problemas sin restricciones y con restricciones.** Teniendo en cuenta las conclusiones de pruebas realizadas en los problemas de funciones con restricciones y sin restricciones (Sección 5.1.1.2 y Sección 5.1.2.2), el algoritmo NSGA-II es el más conveniente para ser utilizado en el proceso de optimización estructural, seguido por el MOPSO. Sin embargo, en los problemas con restricciones se observa que el algoritmo AMOSA presenta un buen comportamiento (a diferencia de algunos problemas sin restricciones). De esta manera, con base en la información presentada en la Secciones 5.1.1 y 5.1.2 se tomó la decisión de usar los tres algoritmos en el proceso de optimización topológica multiobjetivo planteado en este trabajo (a diferencia de un único algoritmo como se había planteado en uno de los objetivos específicos de la Sección 1.2), con el fin de aprovechar las fortalezas de cada uno de ellos y evaluar cómo es su comportamiento en este tipo de problemas, ya que, además, como se observó en el problema de la viga soldada (el cual es similar al problema de optimización topológica, en relación a las funciones objetivo, restricciones y forma del frente de Pareto) los tres algoritmos tuvieron un desempeño satisfactorio, lo que indica que su uso es adecuado. Además, estos tres algoritmos se encuentran dentro de los más populares del área, en términos número de citas (como se mencionó en la Sección 2.5).

## 5.2. Validación de la estrategia de generación de la estructura base usando el método TEM

En esta sección se presentan los resultados de validación de la estrategia para generar la estructura base usando el método TEM (presentado en la Sección 4.3.2). Se abordaron 5 problemas de referencia desarrollados por Gao, Li, et al. (2017); Gao, Liu, et al. (2017) para validar el funcionamiento del algoritmo programado en MATLAB (The Mathworks Inc., 2017), realizando una comparación con las soluciones reportadas (en este caso las trayectorias de esfuerzo). Los problemas son:

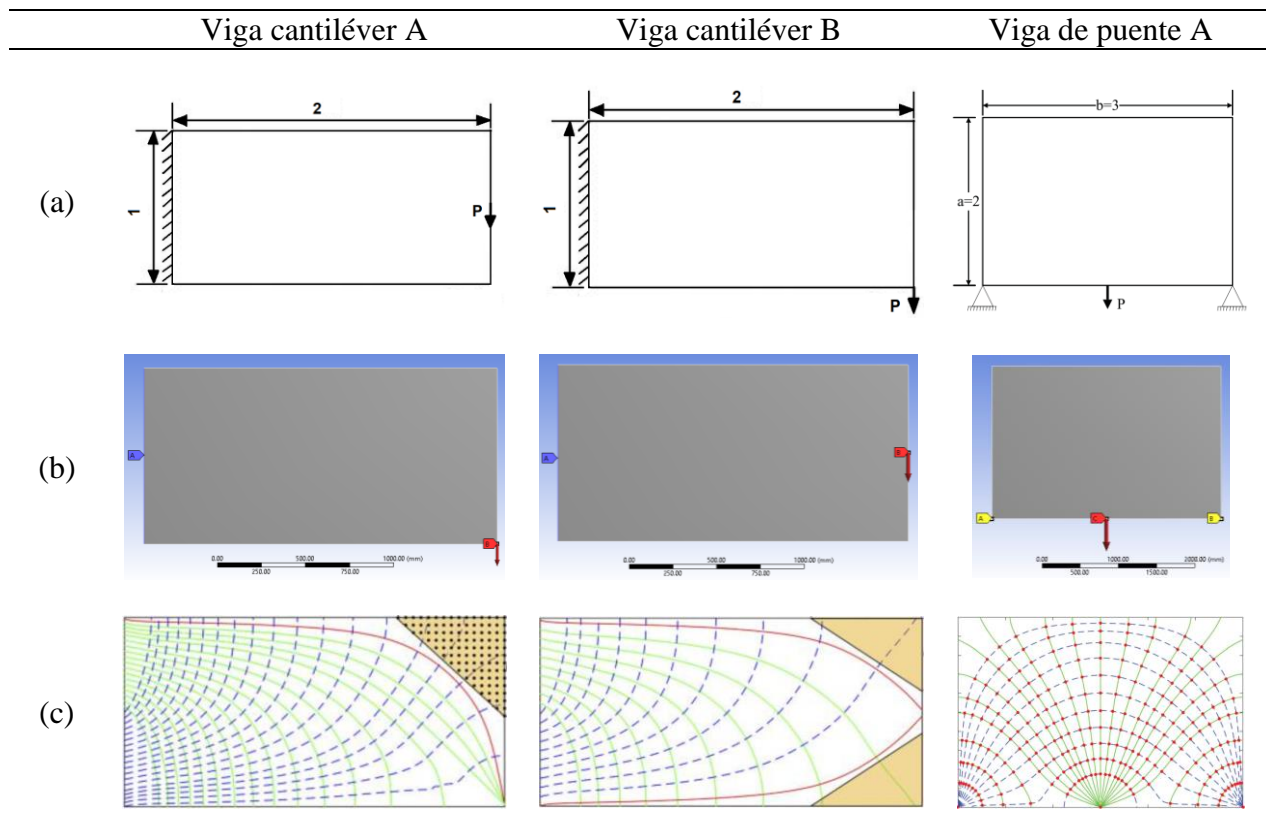
- 1) Problema de la viga cantiléver A
- 2) Problema de la viga cantiléver B
- 3) Problema de la viga de puente A
- 4) Problema de la viga de puente B
- 5) Problema de la viga de puente C

Para los 5 problemas el módulo de elasticidad es  $E = 2 \times 10^{11} Pa$ , el coeficiente de Poisson es  $\nu = 0.33$ , el valor de la carga es  $P = 1 N$  y las distancias se miden en metros. Los resultados se presentan en la Tabla 10 de acuerdo a la siguiente convención:

- En (a) se muestran las condiciones de borde y el espacio de diseño inicial plano continuo del problema
- En (b) se muestra el modelo continuo del problema, desarrollado en el software de elementos finitos ANSYS (ANSYS Inc., 2018).
- En (c) se muestra la solución de referencia para las trayectorias de esfuerzo: trayectorias  $\sigma_1$  con línea continua en color verde (—) y trayectorias  $\sigma_2$  con línea punteada en color azul (- - -).

- En (d) se muestra la solución obtenida en MATLAB (The Mathworks Inc., 2017) para las trayectorias de esfuerzo usando el método numérico descrito en el Apéndice B.2: trayectorias  $\sigma_1$  con línea continua en color rojo (—) y trayectorias  $\sigma_2$  con línea continua en color azul (—).
- En (e) se muestra la estructura de macroelementos obtenida en MATLAB (The Mathworks Inc., 2017), a partir de las trayectorias de esfuerzo calculadas en la parte (d).
- En (f) se muestra la estructura base definitiva obtenida en MATLAB (The Mathworks Inc., 2017) a partir de la estructura de macroelementos de la parte (e).

Tabla 10.

*Problemas de generación de la estructura base*

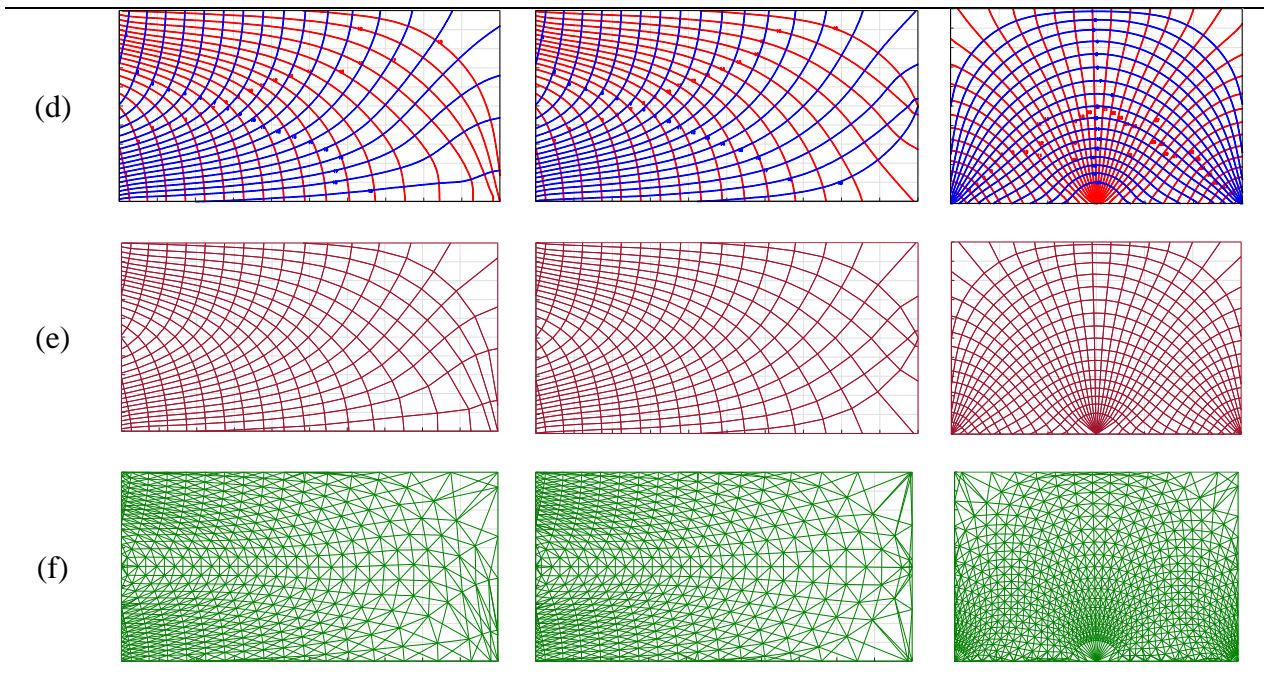
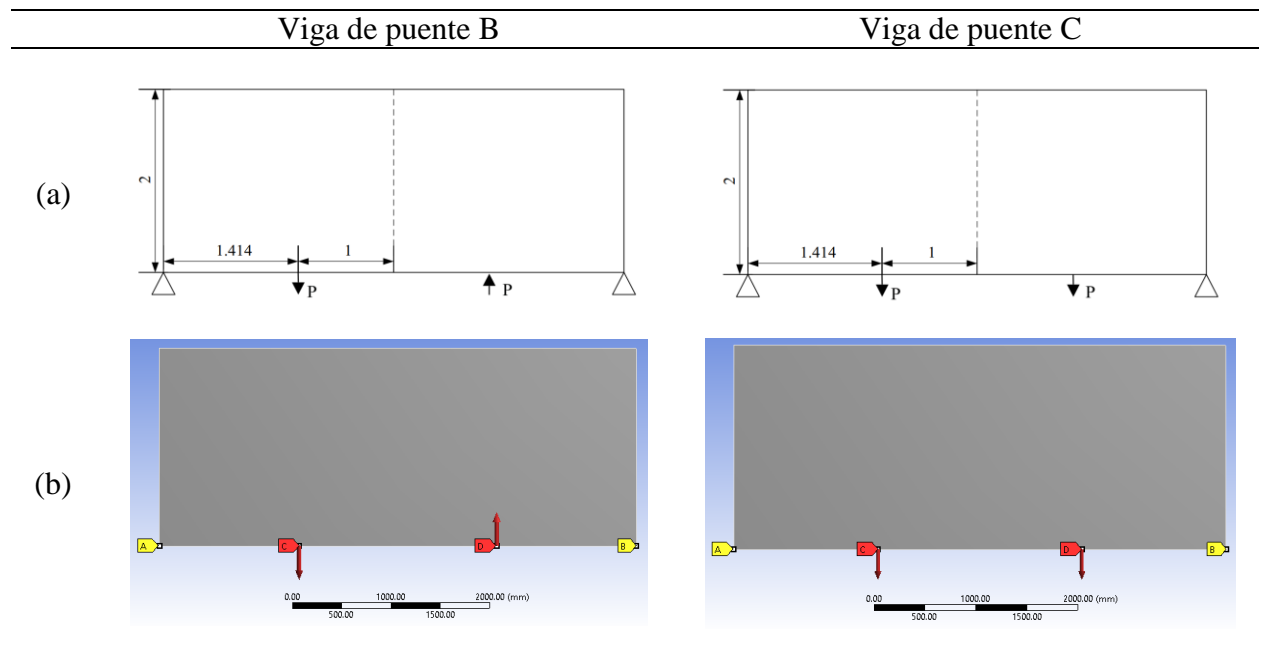
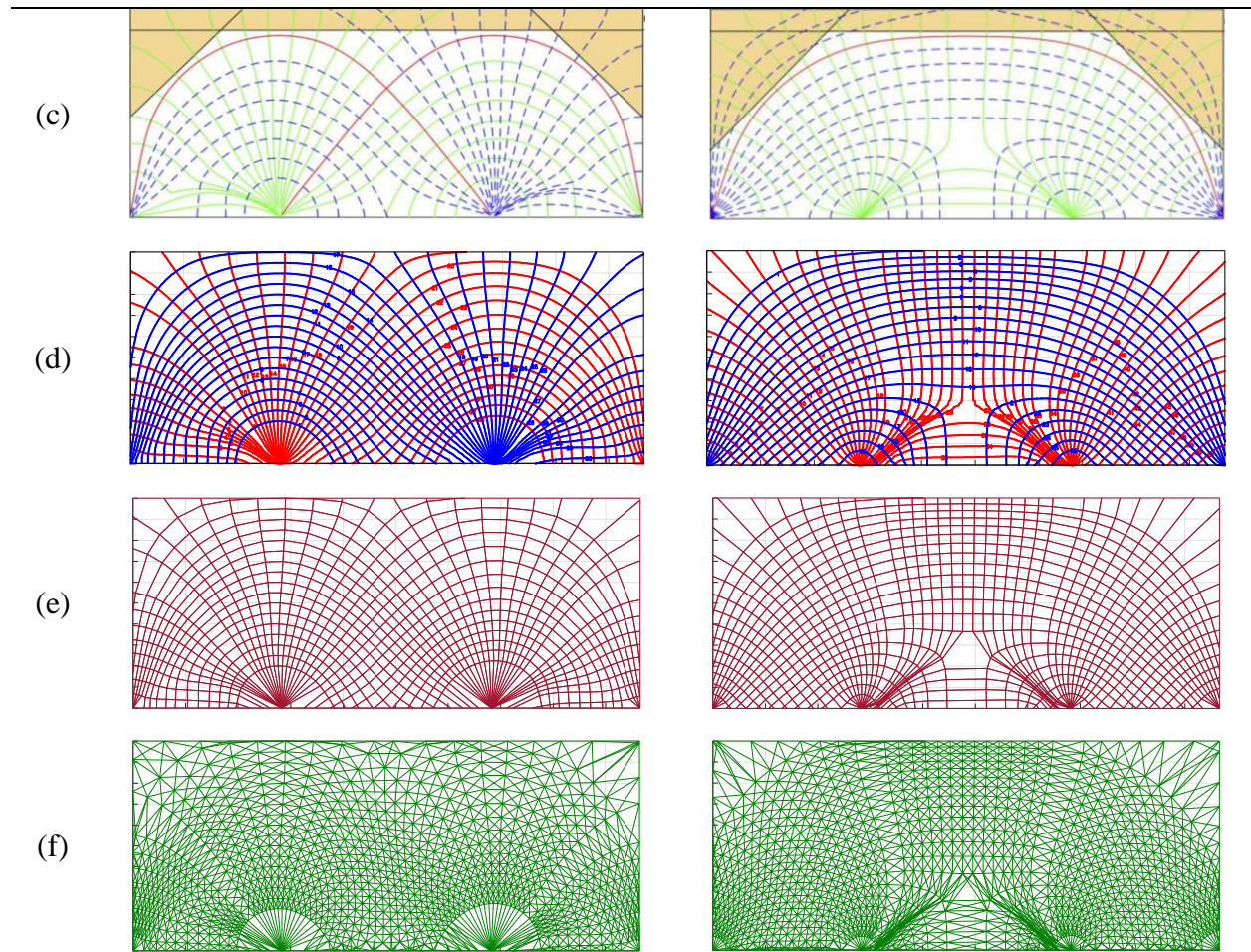


Tabla 10.

*Problemas de generación de la estructura base (continuación)*





Para los problemas 1 a 5, las trayectorias de esfuerzo calculadas con el algoritmo programado en MATLAB (The Mathworks Inc., 2017) (ver Tabla 10(d)) presentan características muy similares a las soluciones de referencia mostradas en la Tabla 10(c). Las pequeñas diferencias radican con respecto al número y separación entre trayectorias, ya que este es un dato definido por el usuario antes de ejecutar el algoritmo, de acuerdo a la densidad de trayectorias requeridas en la solución.

Teniendo en cuenta que las soluciones de referencia solo proporcionan una solución gráfica, solo fue posible realizar una comparación cualitativa de los resultados. De forma general, los resultados del análisis cualitativo indican que el algoritmo programado proporciona soluciones de

trayectorias de esfuerzo confiables, para unas condiciones de borde (cargas y apoyos) en un dominio de diseño plano dado, validando su funcionamiento.

De igual manera, con las trayectorias de esfuerzo calculadas, para cada uno de los problemas se calculó la estructura de macroelementos (ver Tabla 10(e)), a partir de la cual se generó la estructura base en la Tabla 10(f). De forma general, los resultados son satisfactorios y corresponden a la estrategia descrita en el método TEM de la Sección 4.3.2,. Con la información de la estructura base (identificación y coordenadas de los nodos y conectividad de los elementos) se inicia el proceso de optimización topológica multiobjetivo usando los algoritmos NSGA-II, MOPSO y AMOSA.

### **5.3. Validación del algoritmo de análisis estructural**

A continuación, se presentan los resultados de validación del algoritmo que realiza el análisis estructural lineal estático de las estructuras de celosía (armaduras) planas durante el proceso de optimización topológica multiobjetivo, usando el método matricial o método de la rigidez directa (descrito en el Apéndice C.1.1). El algoritmo fue programado en MATLAB (The Mathworks Inc., 2017), y es aplicable para una configuración geométrica (coordenadas de los nodos y conectividad de los elementos), condiciones de carga, apoyos, materiales y secciones definidas por el usuario. Se abordaron dos problemas de análisis estructural de armaduras planas de 7 elementos y 26 elementos respectivamente, y se realizó una comparación entre los resultados del algoritmo programado y los resultados del software SAP2000 (CSI, 2018). Los valores comparados corresponden a las reacciones en los apoyos, los desplazamientos nodales y las fuerzas internas en los elementos.

**5.3.1. Armadura de 7 elementos.** La armadura de acero se muestra en la Figura 39. En paréntesis se indica el valor del área de cada uno de los elementos. Las coordenadas de los nodos y la conectividad de los elementos se presentan en la Tabla 11 y Tabla 12 respectivamente.

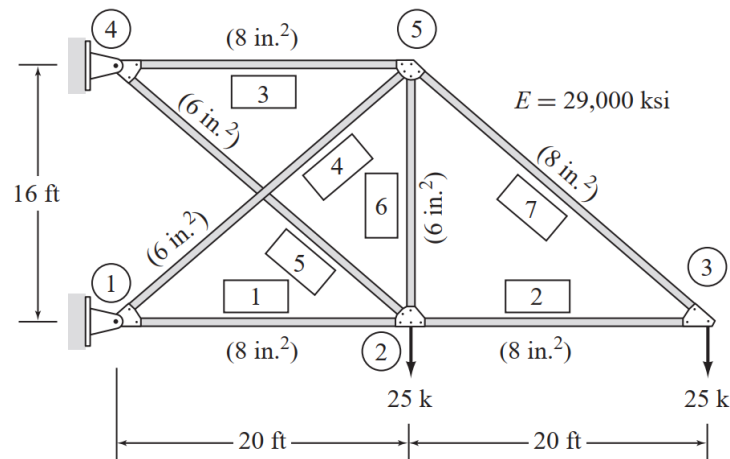


Figura 39. Armadura de 7 elementos  
Adaptado: Kassimali (2012)

Tabla 11.

*Coordenadas*

Nodo	X (in)	Y (in)
1	0	0
2	240	0
3	480	0
4	0	192
5	240	192

Tabla 12.

*Conectividad*

Elemento	Nodo cercano	Nodo lejano
1	1	2
2	2	3
3	4	5
4	1	5
5	2	4
6	2	5
7	3	5

A continuación, en la Tabla 13, Tabla 14 y

Tabla 15 se presentan los resultados comparativos correspondientes a las reacciones, desplazamientos de los nodos y fuerzas internas en los elementos:

Tabla 13.

*Reacciones de la armadura de 7 elementos*

Nodo	MATLAB		SAP2000		Error Fx (%)	Error Fy (%)
	Fx (kip)	Fy (kip)	Fx (kip)	Fy (kip)		
1	93.75	25	93.75	25	0	0

4	-93.75	25	-93.75	25	0	0
---	--------	----	--------	----	---	---

Tabla 14. Desplazamientos de la armadura de 7 elementos

Nodo	MATLAB		SAP2000		Error Dx (%)	Error Dy (%)
	Dx (in)	Dy (in)	Dx (in)	Dy (in)		
1	0	0	0	0	0	0
2	-0.064655	-0.193978	-0.064655	-0.193978	0.000267	0.000192
3	-0.096983	-0.480894	-0.096983	-0.480894	0.000249	0.000008
4	0	0	0	0	0	0
5	0.064655	-0.193978	0.064655	-0.193978	0.000267	0.000192

Tabla 15.

Fuerzas internas de la armadura de 7 elementos

Elemento	MATLAB	SAP2000	Error (%)
	Fuerza axial (kip)	Fuerza axial (kip)	
1	-62.5	-62.5	0
2	-31.25	-31.25	0
3	62.5	62.5	0
4	-40.0195	-40.02	0.001183
5	40.0195	40.02	0.001183
6	-2.84E-14	0	0
7	40.0195	40.02	0.001183

**5.3.2. Armadura de 26 elementos.** La armadura de aluminio se muestra en la Figura 40. Todos los elementos tienen el mismo valor de área. Las coordenadas de los nodos y la conectividad de los elementos se presentan en la Tabla 16 y Tabla 17 respectivamente.

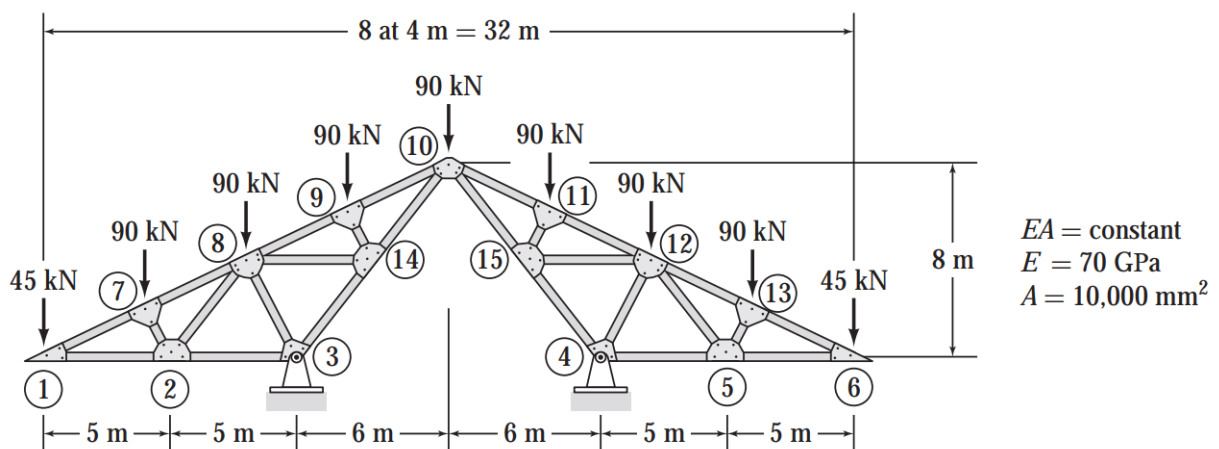


Figura 40. Armadura de 26 elementos  
Adaptado: Kassimali (2012)

Tabla 16.

*Coordenadas de los nodos*

Nodo	X (mm)	Y (mm)
1	0	0
2	5000	0
3	10000	0
4	22000	0
5	27000	0
6	32000	0
7	4000	2000
8	8000	4000
9	12000	6000
10	16000	8000
11	20000	6000
12	24000	4000
13	28000	2000
14	13000	4000
15	19000	4000

Tabla 17.

*Conectividad de los elementos*

Elemento	NC	NL	Elemento	NC	NL
1	1	2	14	12	11
2	2	3	15	13	12
3	5	4	16	6	13
4	6	5	17	2	7
5	3	14	18	8	2
6	14	10	19	3	8
7	4	15	20	8	14
8	15	10	21	14	9
9	1	7	22	15	11
10	7	8	23	12	15
11	8	9	24	4	12
12	9	10	25	12	5
13	11	10	26	5	13

Los resultados comparativos correspondientes a las reacciones, desplazamientos de los nodos y fuerzas internas se muestran en la Tabla 18, Tabla 19 y Tabla 20 respectivamente.

Tabla 18.

*Reacciones de la armadura de 26 elementos*

Nodo	MATLAB		SAP2000		Error Fx (%)	Error Fy (%)
	Fx (kN)	Fy (kN)	Fx (kN)	Fy (kN)		
3	-90	360	-90	360	0	0
4	90	360	90	360	0	0

Tabla 19.

*Desplazamientos de la armadura de 26 elementos*

Nodo	MATLAB		SAP2000		Error Dx (%)	Error Dy (%)
	Dx (mm)	Dy (mm)	Dx (mm)	Dy (mm)		
1	1.928571	-17.074823	1.928571	-17.074820	0	0
2	1.285714	-7.226719	1.285714	-7.226719	0.000022	0.000002
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	-1.285714	-7.226719	-1.285714	-7.226719	0.000022	0.000002
6	-1.928571	-17.074823	-1.928571	-17.07482	0.000022	0.000003
7	-1.449255	-8.881698	-1.449255	-8.881698	0.000024	0.000001

8	-3.447107	-2.873531	-3.447107	-2.873531	0.000011	0.000017
9	-3.194960	-0.790376	-3.194960	-0.790376	0.000002	0.000000
10	1.65E-14	-4.017857	-1.82E-14	-4.017857	0	0.000004
11	3.194960	-0.790376	3.194960	-0.790376	0.000002	0.000000
12	3.447107	-2.873531	3.447107	-2.873531	0.000011	0.000017
13	1.449255	-8.881698	1.449255	-8.881698	0.000024	0.000001
14	-2.804250	-0.307527	-2.804250	-0.307527	0.000008	0.000124
15	2.804250	-0.307527	2.804250	-0.307527	0.000008	0.000124

Tabla 20.

*Fuerzas internas de la armadura de 26 elementos*

MATLAB			SAP2000			MATLAB		SAP2000	
Elem	Fuerza axial (kN)	Fuerza axial (kN)	Error (%)	Elem	Fuerza axial (kN)	Fuerza axial (kN)	Error (%)		
1	-90	-90	0	14	181.1215	181.122	0.000273		
2	-180	-180	0	15	140.8723	140.872	0.000201		
3	-180	-180	0	16	100.6231	100.623	5.86E-05		
4	-90	-90	0	17	-80.4984	-80.498	0.000556		
5	-270	-270	0	18	90	90	0		
6	-180	-180	0	19	-160.9969	-160.997	6.56E-05		
7	-270	-270	0	20	90	90	0		
8	-180	-180	0	21	-80.4984	-80.498	0.000556		
9	100.6231	100.623	5.86E-05	22	-80.4984	-80.498	0.000556		
10	140.8723	140.872	0.000201	23	90	90	0		
11	181.1215	181.122	0.000273	24	-160.9969	-160.997	6.56E-05		
12	221.3707	221.371	0.000122	25	90	90	0		
13	221.3707	221.371	0.000122	26	-80.4984	-80.498	0.000556		

Al comparar los resultados de análisis estructural del algoritmo programado en MATLAB (The Mathworks Inc., 2017) con los resultados de SAP2000 (CSI, 2018), para reacciones, desplazamientos y fuerzas internas (para las dos armaduras), puede observarse que los resultados son muy cercanos entre sí con errores muy cercanos al 0%. Estos resultados validan el funcionamiento del algoritmo programado e indican que es adecuado para su aplicación en el algoritmo de optimización topológica multiobjetivo híbrido (AOTMOH) propuesto.

## 6. Aplicación del algoritmo de optimización topológica multiobjetivo híbrido (AOTMOH)

En este capítulo se presentan los resultados de aplicación del algoritmo AOTMOH propuesto en el Capítulo 4. Para ello fue necesario integrar cada uno de los algoritmos de optimización metaheurísticos AMOSA, MOPSO y NSGAI (validados en la Sección 5.1) con el algoritmo de generación de la estructura base (usando el método NCN o el método TEM validado en la Sección 5.2) y con el algoritmo de análisis estructural (validado en la Sección 5.3).

Para todos los problemas abordados en este capítulo, las soluciones extremas de los frentes de Pareto obtenidos (ver Figura 41), es decir la solución  $P_1$  (con mínimo valor de la función objetivo **A** y máximo valor de la función objetivo **B**) y la solución  $P_2$  (con máximo valor de la función objetivo **A** y mínimo valor de la función objetivo **B**), se identifican con el valor de las coordenadas en el plano **AB** definido por las dos funciones objetivo.

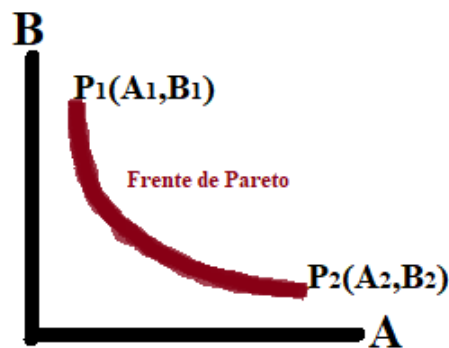


Figura 41. Convención utilizada para referenciar las soluciones extremas del frente de Pareto

### 6.1. Validación del algoritmo AOTMOH

Inicialmente se evalúa y se valida el desempeño del algoritmo AOTMOH en la solución de problemas de optimización multiobjetivo de estructuras de referencia (benchmark) tipo celosía (armadura) en 2D, realizando una comparación con las soluciones reportadas. En este caso, cada uno de los algoritmos metaheurísticos (NSGA-II, MOPSO y AMOSA) se ejecutaron 20 veces (run) y los parámetros de cada uno de ellos son los mismos que se usaron para los problemas de

funciones matemáticas sin restricciones y con restricciones (ver Tabla 3, Tabla 4 y Tabla 5), a excepción del número de generaciones para NSGA-II que fue de 500 y el número de iteraciones para MOPSO y AMOSA, también de 500. Se abordaron cuatro problemas de validación:

- a) Problema de optimización de tamaño multiobjetivo de la armadura de 10 elementos.
- b) Problema de optimización topológica multiobjetivo de la estructura base de 10 elementos
- c) Problema de optimización topológica multiobjetivo de la estructura base de 14 elementos
- d) Problema de optimización topológica multiobjetivo de la estructura base de 15 elementos

### 6.1.1. Problema de optimización de tamaño multiobjetivo de la armadura de 10 elementos.

La armadura de 10 elementos mostrada en la Figura 42 fue estudiada por G. G. Tejani, Pholdee, Bureerat, & Prayogo (2018) usando el algoritmo MOASOS (Multiobjective Adaptive Symbiotic Organisms Search) para ejecutar el proceso de optimización de tamaño (el algoritmo AOTMOH también funciona en problemas de optimización de tamaño, realizando un pequeño cambio en la estrategia de la Sección 4.5). Las propiedades del material son: densidad  $\rho = 0.1 \text{ lb/in}^2$  y módulo de elasticidad  $E = 10000 \text{ ksi}$ . Los nodos 2 y 4 están sujetos a cargas verticales de **100 kips**. El esfuerzo axial de tensión y compresión de los elementos está limitado a  $|\sigma_{max}^{\pm}| = 25 \text{ ksi}$ . Las secciones transversales de los elementos son seleccionadas de un conjunto de 42 valores discretos: {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, 11.5, 13.5, 13.9, 14.2, 15.5, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.5, 30.00, 33.5}  $\text{in}^2$ . Las funciones objetivo son:

- 1) Minimizar el peso de la estructura:  $f_1(A) = \sum_{i=1}^m \rho A_i L_i$

- 2) Minimizar el máximo desplazamiento nodal en las direcciones  $x$  y  $y$ :  $f_2(A) = \max(|\delta_j|)$

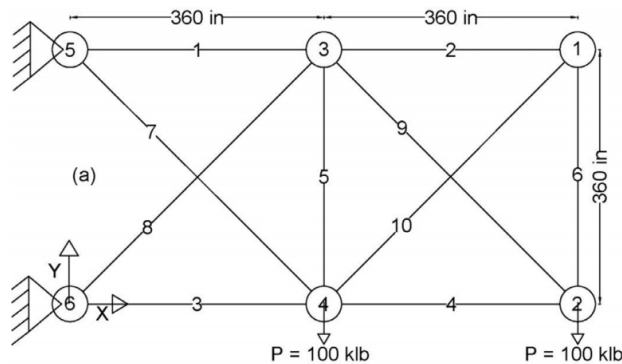


Figura 42. Armadura de 10 elementos para optimización de tamaño multiobjetivo  
Adaptado: G. G. Tejani, Pholdee, et al. (2018)

Los resultados gráficos de desempeño de los algoritmos NSGAII, MOPSO y AMOSA en este problema, se presentan por medio de los frentes de Pareto obtenidos en las 20 veces que se ejecutaron, de acuerdo a la Figura 43:

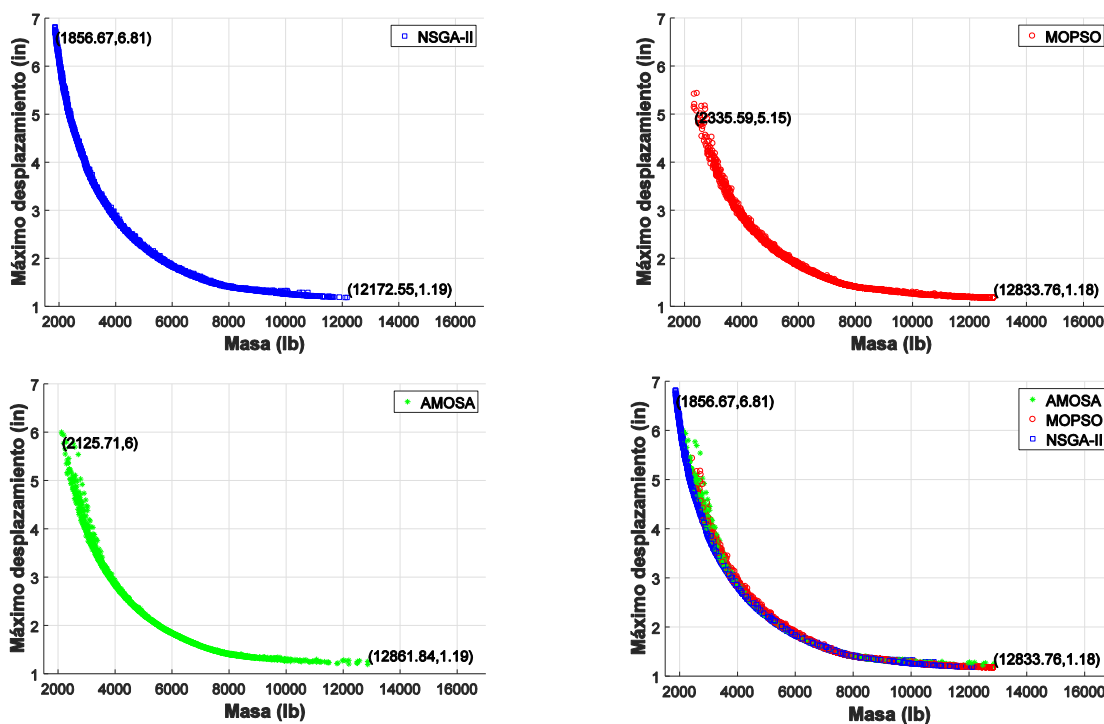


Figura 43. Frentes de Pareto en cada una de las 20 ejecuciones, para cada algoritmo en problema de optimización de tamaño multiobjetivo armadura de 10 elementos

Ahora, a partir de todos los frentes de Pareto de los tres algoritmos (ver Figura 43 inferior derecha) se obtiene el frente de Pareto envolvente (definido en la Sección 4.4) con las mejores

soluciones no dominadas del proceso de optimización (Figura 44 izquierda), el cual se compara con la solución de referencia (Figura 44 derecha):

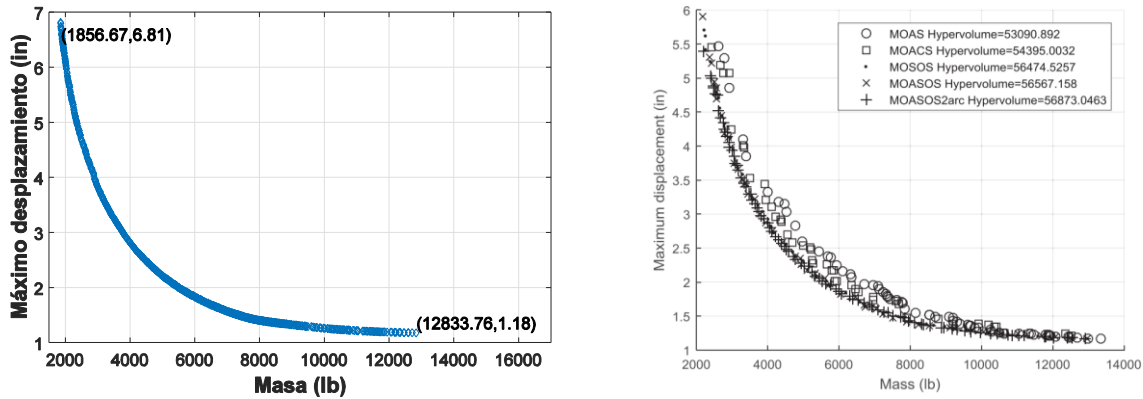


Figura 44. Comparación de frente de Pareto envolvente (este trabajo) con la solución de referencia (G. G. Tejani, Pholdee, et al., 2018) en el problema de optimización de tamaño multiobjetivo de la armadura de 10 elementos.

### 6.1.2. Problema de optimización topológica multiobjetivo de la estructura base de 10

**elementos.** La estructura base de la armadura de 10 elementos mostrada en la Figura 45 fue estudiada por Su, Wang, Gui, & Fan (2011) usando el algoritmo AMISS-MOP (Adaptive Multi-island Search Strategy for Multi-objective Optimization Problem) durante la optimización topológica multiobjetivo. En esta estructura  $\mathbf{a} = 9.14 \text{ m}$  y  $\mathbf{P} = 444.5 \text{ kN}$ . La densidad del material es  $\rho = 2768 \text{ kg/m}^3$  y el módulo de elasticidad es  $\mathbf{E} = 68.9 \text{ GPa}$ . El esfuerzo axial de tensión y compresión de los elementos está limitado a  $|\sigma_{max}^{\pm}| = 172 \text{ MPa}$ . Las secciones transversales de los elementos son seleccionadas de un conjunto de 32 valores discretos: {1.05, 1.16, 1.54, 1.69, 1.86, 1.99, 2.02, 2.18, 2.34, 2.48, 2.50, 2.70, 2.90, 3.10, 3.21, 3.30, 3.70, 4.66, 5.14, 7.42, 8.71, 8.97, 9.16, 10.00, 10.32, 12.13, 12.84, 14.19, 14.77, 17.10, 19.35, 21.61}  $\times 10^{-3} \text{ m}^2$ . Las funciones objetivo son: minimizar el peso de la estructura  $f_1(\mathbf{A}) = \sum_{i=1}^m \rho \mathbf{A}_i L_i$  y minimizar el máximo desplazamiento nodal en las direcciones  $x$  y  $y$ :  $f_2(\mathbf{A}) = \max(|\delta_j|)$ .

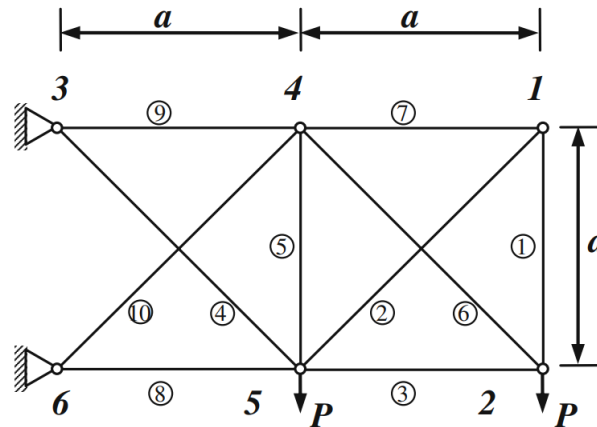


Figura 45. Estructura base de 10 elementos para optimización topológica multiobjetivo  
Adaptado: Su et al. (2011)

Los resultados gráficos de desempeño de los algoritmos NSGAI, MOPSO y AMOSA, también se presentan por medio de los frentes de Pareto obtenidos en las 20 veces que se ejecutaron, de acuerdo a la Figura 46:

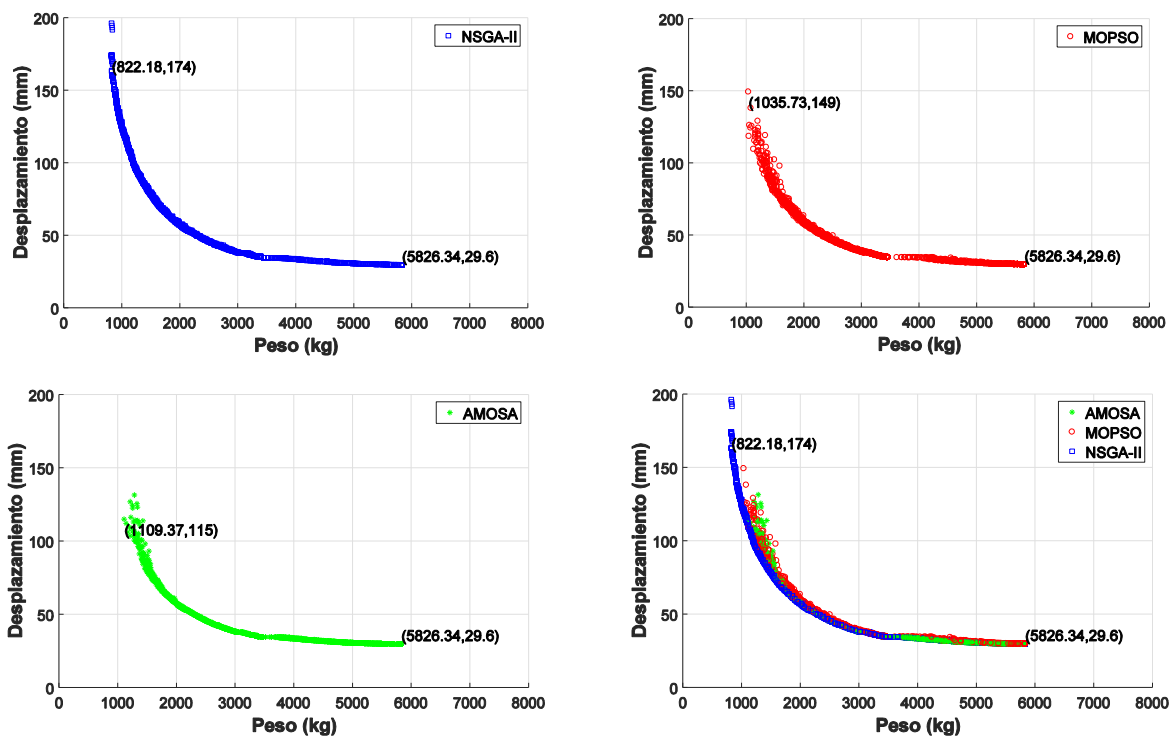


Figura 46. Frentes de Pareto en cada una de las 20 ejecuciones, para cada algoritmo en problema de optimización topológica multiobjetivo estructura base de 10 elementos

A continuación se presenta la comparación del frente de Pareto envolvente (ver Figura 47 izquierda) con la solución de referencia (Figura 47 derecha):

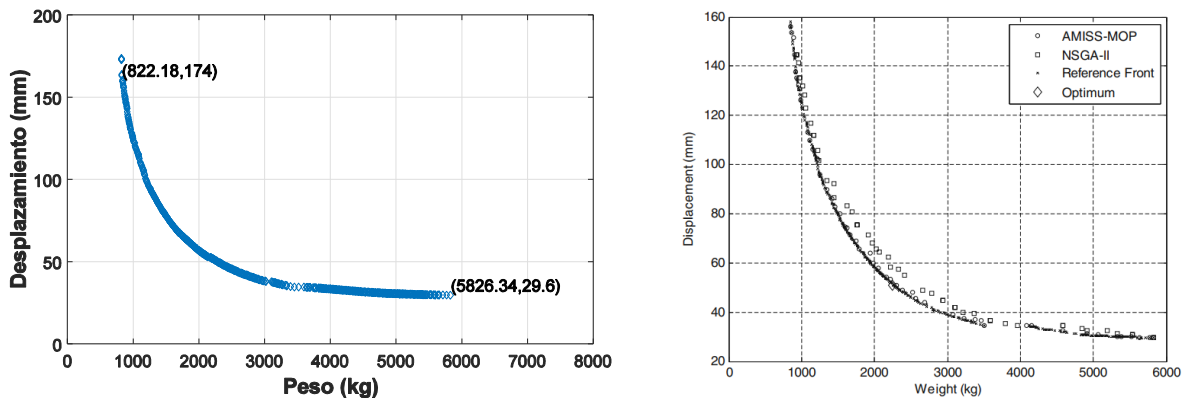


Figura 47. Comparación de frente de Pareto envolvente (este trabajo) con la solución de referencia (Su et al., 2011) en el problema de optimización topológica multiobjetivo de la estructura base de 10 elementos

Algunas de las topologías seleccionadas del frente de Pareto de la envolvente se muestran en la

Figura 48:

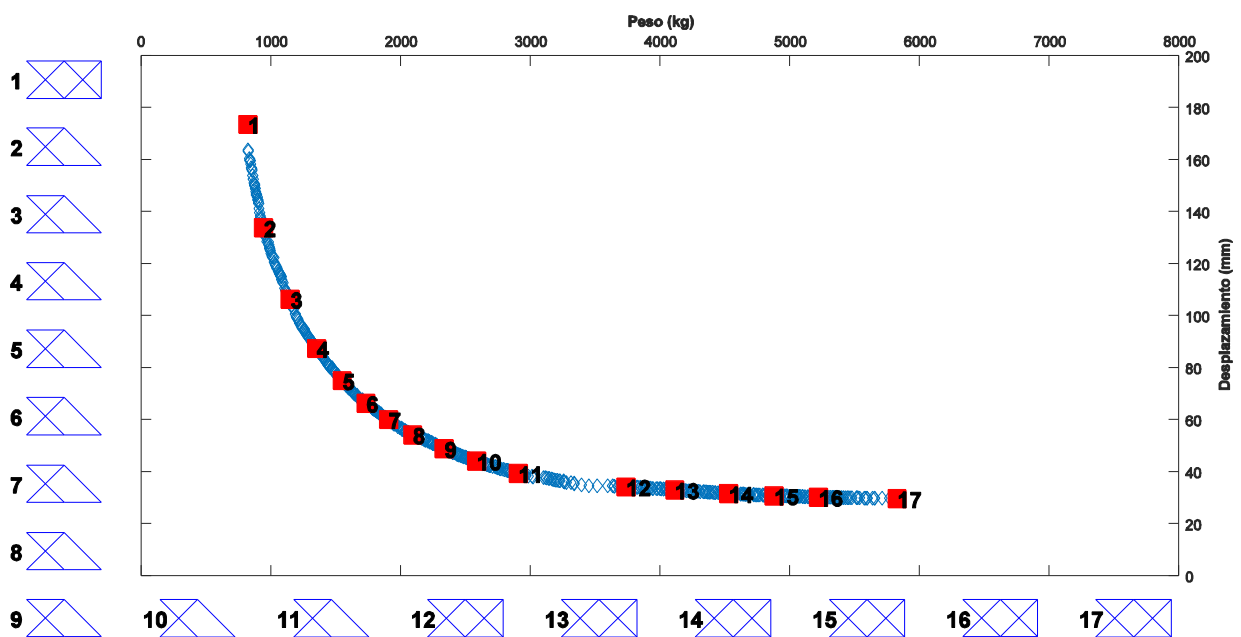


Figura 48. Topologías del frente de Pareto envolvente para problema de optimización topológica multiobjetivo estructura base de 10 elementos

En la Figura 48 se representa gráficamente 17 soluciones del problema de optimización topológica multiobjetivo de la estructura base de 10 elementos. Estas soluciones fueron seleccionadas de forma uniforme a lo largo del frente en el frente de Pareto envolvente a partir del conjunto total de soluciones óptimas disponibles.

Estas soluciones incluyen dos tipos de topologías (ej. solución 8 con 6 elementos y solución 15 con 9 elementos) generadas a partir de la estructura base de 10 elementos. Las soluciones 1 a 11 son soluciones de menor peso, pero con mayor desplazamiento (lo cual se ve reflejado en un menor número de elementos, excepto para la solución 1), en comparación con las soluciones 12 a 17 que tienen un mayor peso, pero un menor desplazamiento (y presentan un mayor número de elementos). Las soluciones extremas 1 y 17 presentan la misma topología, sin embargo, la solución 1 es una estructura de mínimo peso (y máximo desplazamiento) y la solución 17 es una estructura de mínimo desplazamiento (máximo peso).

**6.1.3. Problema de optimización topológica multiobjetivo de la estructura base de 14 elementos.** La estructura base de la armadura de 14 elementos se muestra en la Figura 49 fue resuelta por Richardson et al. (2012), quienes usaron el algoritmo MOGA (Multiobjective Genetic Algorithm) para resolver el problema de optimización topológica multiobjetivo, mejorando su desempeño con el enfoque KSR (Kinematic Stability Repair). En la Figura 49,  $l = 9.144 \text{ m}$  y  $P = 448.2 \text{ kN}$ . La densidad del material es  $\rho = 2768 \text{ kg/m}^3$  y el módulo de elasticidad es  $E = 68.95 \text{ GPa}$ . El esfuerzo axial de tensión y compresión de los elementos está limitado a  $|\sigma_{max}^{\pm}| = 172.4 \text{ MPa}$ . Las secciones transversales de los elementos solo pueden tomar el valor de  $1.419352 \times 10^{-2} \text{ m}^2$ . Las funciones objetivo son:

1) Minimizar el peso de la estructura:  $f_1(A) = \sum_{i=1}^m \rho A_i L_i$

2) Minimizar el máximo desplazamiento nodal en las direcciones  $x$  y  $y$ :  $f_2(A) = \max(|\delta_j|)$

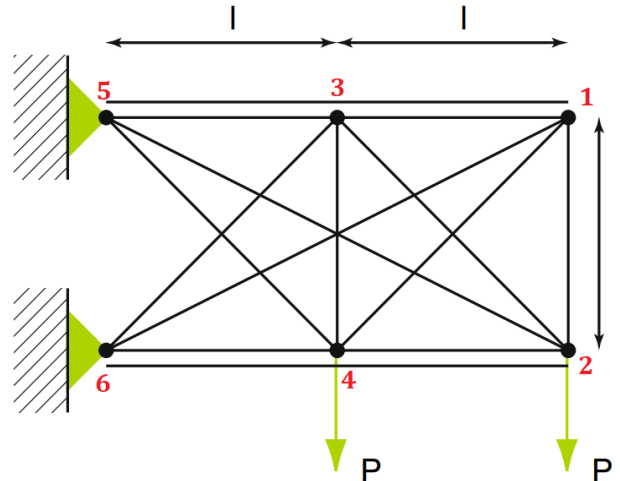


Figura 49. Estructura base de 14 elementos para optimización topológica multiobjetivo  
Adaptado: Richardson et al. (2012)

Los resultados gráficos de desempeño de los algoritmos (NSGAI, MOPSO y AMOSA) por medio de los frentes de Pareto obtenidos en las 20 veces que se ejecutaron, se presentan en la

Figura 50:

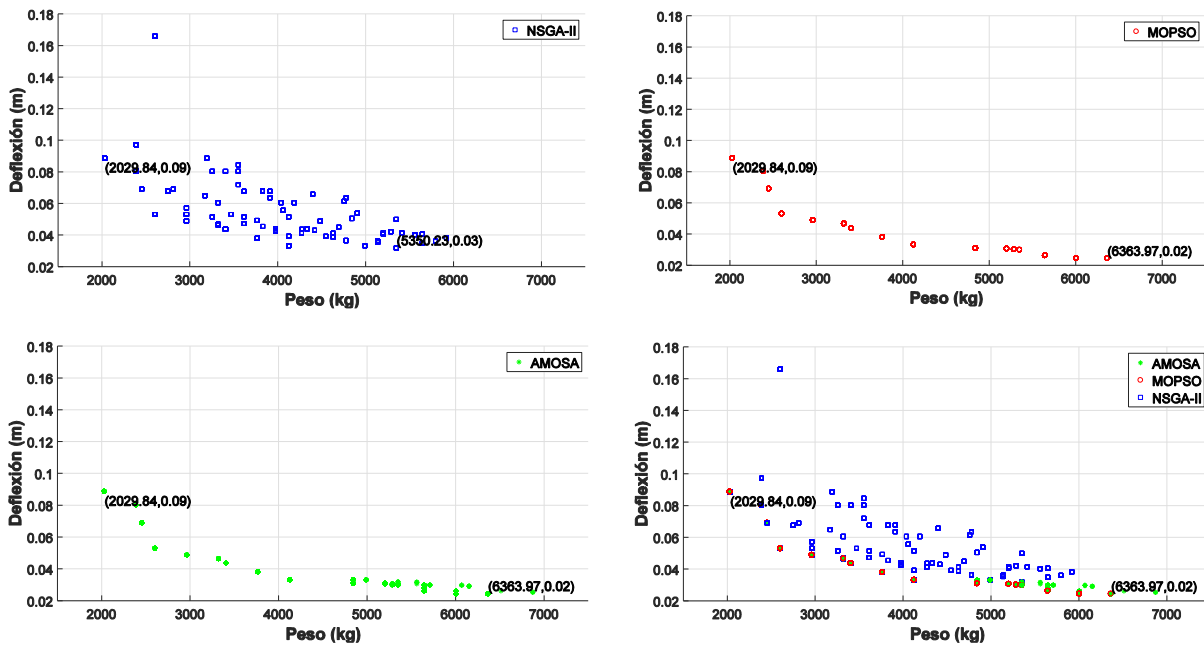


Figura 50. Frentes de Pareto en cada una de las 20 ejecuciones, para cada algoritmo en problema de optimización topológica multiobjetivo estructura base de 14 elementos

El frente de Pareto envolvente (Figura 51 izquierda), es comparado con la solución de referencia (Figura 51 derecha):

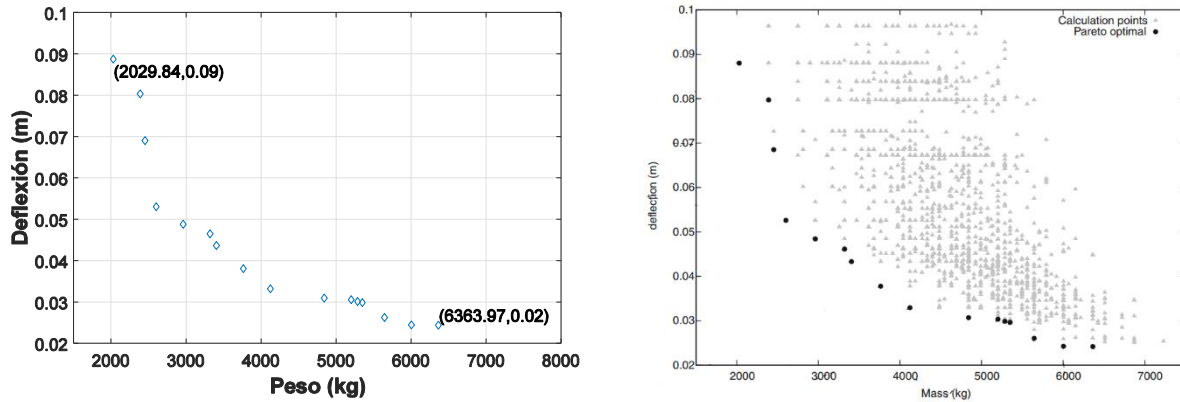


Figura 51. Comparación de frente de Pareto envolvente (este trabajo) con la solución de referencia (Richardson et al., 2012) en el problema de optimización topológica multiobjetivo estructura base de 14 elementos

A continuación se muestran las topologías decodificadas del frente de Pareto envolvente y del Pareto de referencia en la Figura 52 (arriba), y Figura 52 (abajo) respectivamente.

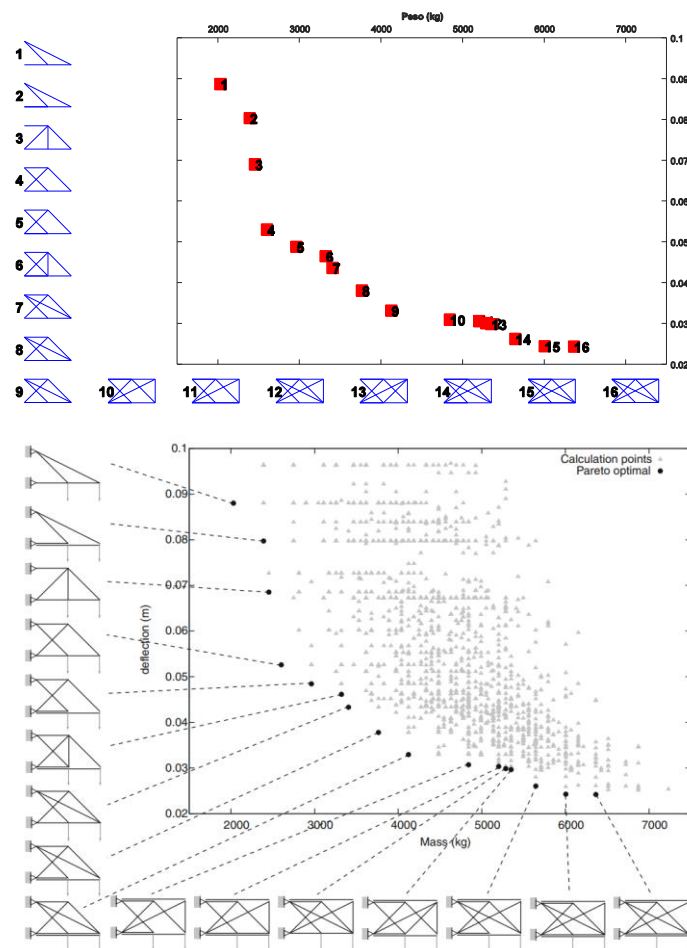


Figura 52. Topologías del frente de Pareto envolvente (este trabajo) y del frente de Pareto de referencia (Richardson et al., 2012) para problema de optimización topológica multiobjetivo estructura base de 14 elementos

**6.1.4. Problema de optimización topológica multiobjetivo de la estructura base de 15 elementos.** La estructura base de la armadura de 15 elementos se muestra en la Figura 53 y fue estudiada por Su et al. (2011) usando el algoritmo AMISS-MOP (Adaptive Multi-island Search Strategy for Multi-objective Optimization Problem). Donde  $a = 1.016 \text{ m}$ ,  $b = 0.762 \text{ m}$  y  $P = 89 \text{ kN}$ . El módulo de elasticidad es  $E = 207 \text{ GPa}$ . El esfuerzo axial de tensión y compresión de los elementos está limitado a  $|\sigma_{max}^{\pm}| = 345 \text{ MPa}$ . Las secciones transversales de los elementos son seleccionadas de un conjunto de 16 valores discretos uniformemente distribuidos entre  $0.645$  y  $10.323 \text{ cm}^2$ :  $\{0.645, 1.2902, 1.9354, 2.5806, 3.2258, 3.871, 4.5162, 5.1614, 5.8066, 6.4518, 7.097, 7.7422, 8.3874, 9.0326, 9.6778, 10.323\} \text{ cm}^2$ . Las funciones objetivo son:

- 1) Minimizar el volumen de la estructura:  $f_1(A) = \sum_{i=1}^m A_i L_i$
- 2) Minimizar el máximo desplazamiento nodal en las direcciones  $x$  y  $y$ :  $f_2(A) = \max(|\delta_j|)$

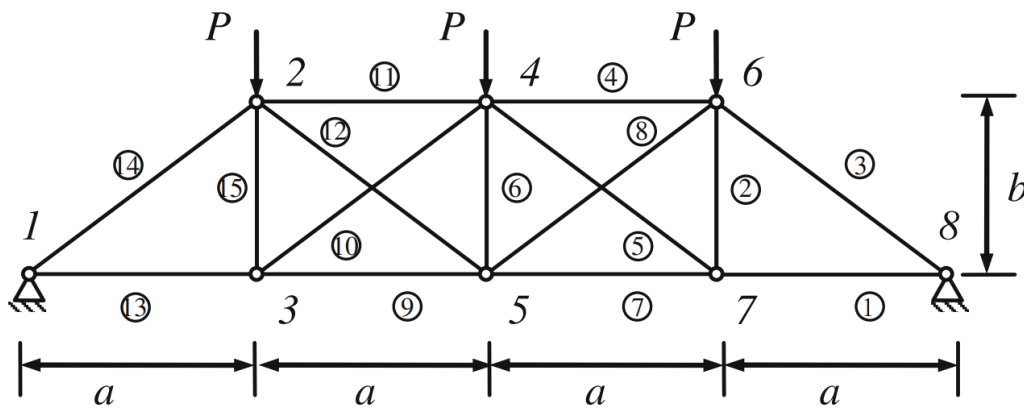


Figura 53. Estructura base de 15 elementos para optimización topológica multiobjetivo  
Adaptado: Su et al. (2011)

Los resultados gráficos de desempeño de los algoritmos NSGAI, MOPSO y AMOSA se presentan usando los frentes de Pareto obtenidos en las 20 veces que se ejecutaron, de acuerdo a la Figura 54:

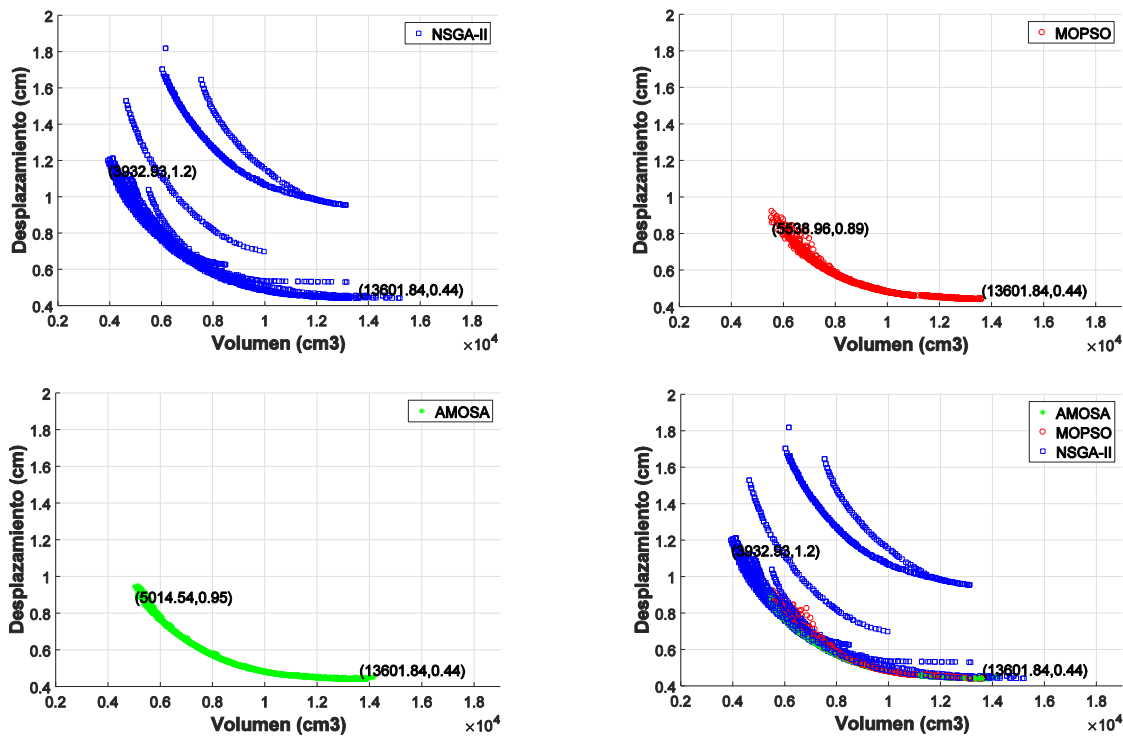


Figura 54. Frentes de Pareto en cada una de las 20 ejecuciones, para cada algoritmo en problema de optimización topológica multiobjetivo estructura base de 14 elementos

El frente de Pareto envolvente (Figura 55 izquierda), se compara con la solución de referencia (Figura 55 derecha):

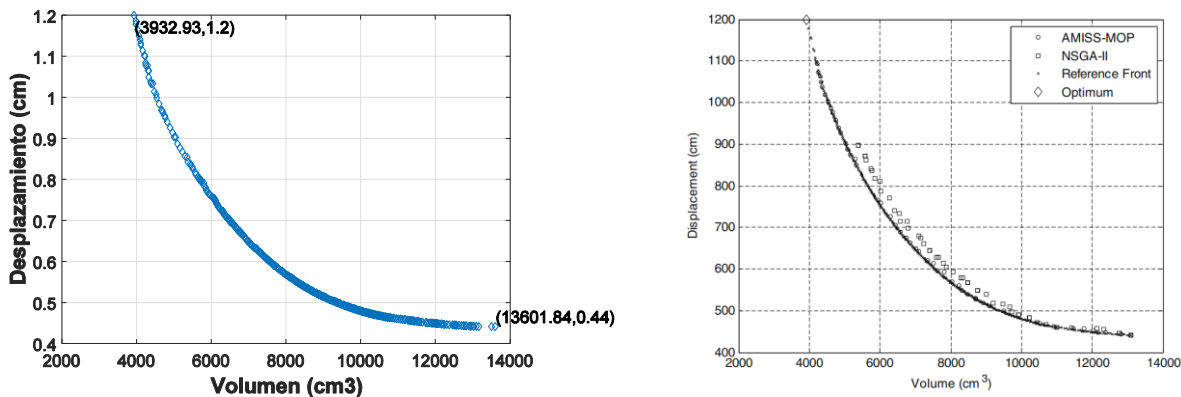


Figura 55. Comparación de frente de Pareto envolvente (este trabajo) con la solución de referencia (Su et al., 2011) en el problema de optimización topológica multiobjetivo de la estructura base de 15 elementos

Algunas de las topologías seleccionadas del frente de Pareto envolvente se muestran en la Figura 56:

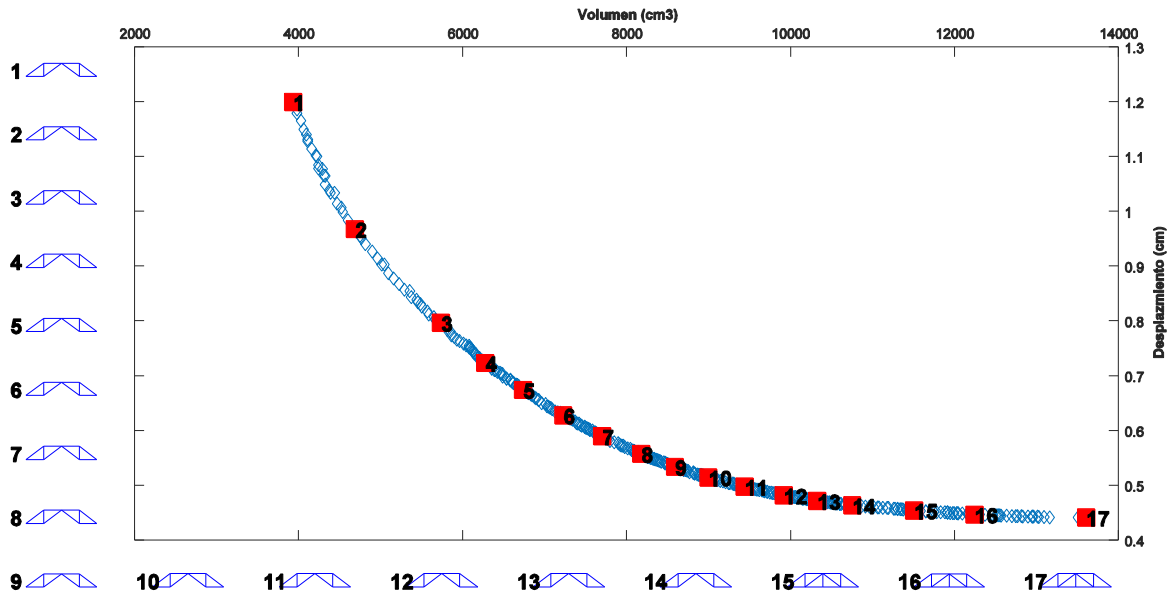


Figura 56. Topologías del frente de Pareto envolvente para problema de optimización topológica multiobjetivo estructura base de 15 elementos

**6.1.5. Análisis de resultados de desempeño en problemas de optimización multiobjetivo de estructuras de celosía 2D.** El desempeño de los algoritmos metaheurísticos NSGA-II, MOPSO y AMOSA en los problemas de optimización multiobjetivo de celosías 2D fue satisfactorio.

Para el problema de optimización de tamaño de la armadura de 10 elementos (ver Figura 42) los 3 algoritmos (ver Figura 43) presentan resultados similares a la referencia consultada (ver Figura 44 derecha).

Para el problema de optimización topológica de la estructura base de 10 elementos (ver Figura 45) los 3 algoritmos (ver Figura 46) logran aproximarse al frente de Pareto de la referencia consultada (ver Figura 47 derecha). El mejor desempeño en este caso fue para el NSGA-II, ya que logró obtener soluciones en una zona más amplia del frente de Pareto (en dirección del peso mínimo), en comparación con MOPSO y AMOSA respectivamente (ver Figura 46). De igual manera con la construcción del frente de Pareto envolvente (Figura 47 izquierda) se puede lograr

el mejor conjunto de soluciones óptimas cuyas topologías se representan gráficamente en la Figura 48.

En el problema de la estructura base de 14 elementos (ver Figura 49) los mejores algoritmos fueron el MOPSO y el AMOSA, ya que los resultados del NSGA-II fueron más variables (ver Figura 50). La cantidad de soluciones obtenidas para los 3 algoritmos fueron más dispersas en comparación con los problemas anteriores, teniendo en cuenta la restricción existente en las variables discretas (solo había una única opción posible). Al graficar el frente de Pareto envolvente (ver Figura 51 izquierda), y compararlo con la referencia (Figura 51 derecha) se observa que la distribución de las soluciones y sus topologías (representadas gráficamente en la Figura 52) son muy similares.

Finalmente, para el problema de la estructura base de 15 elementos (ver Figura 53) el algoritmo con el mejor desempeño fue el NSGA-II (aunque en algunas ejecuciones presenta variaciones en los resultados, mostrando una falta de consistencia para este caso) ya que logró encontrar soluciones en una zona más amplia del frente de Pareto a lo largo de los dos objetivos (ver Figura 54), en comparación con el MOPSO y AMOSA, los cuales cubren una menor región del frente de Pareto. Sin embargo, a diferencia del NSGA-II, son algoritmos más consistentes (no presentan variación de resultados en cada una de las ejecuciones). Al comparar el frente de Pareto envolvente (Figura 55 izquierda) con la referencia consultada (Figura 55 derecha), los resultados son muy similares (en la función de desplazamiento existe una diferencia en la escala de 1000, ya que el valor obtenido es de 1.2 cm y la referencia indica un valor máximo de 1200 cm excediendo la dimensión  $a = 1.016 m$  de la estructura original, por lo cual se plantea que puede ser un error de tipografía), y la representación gráfica de las topologías de las soluciones óptimas se muestran en la Figura 56.

A continuación, se muestran los resultados de evaluación de desempeño cuantitativo de los algoritmos NSGA-II, MOPSO y AMOSA en los 4 problemas analizados en esta sección. Los factores evaluados incluyen costo computacional (registrando el tiempo empleado por cada algoritmo en cada ejecución), y la convergencia y diversidad (usando la métrica de hipervolumen  $HV$ ). Se normalizaron los valores de las funciones objetivo para evitar problemas de escala. Como cada algoritmo se ejecutó 20 veces, los resultados de evaluación se presentan usando los diagramas de cajas (boxplots) como se muestra en la Figura 57 y Figura 58.

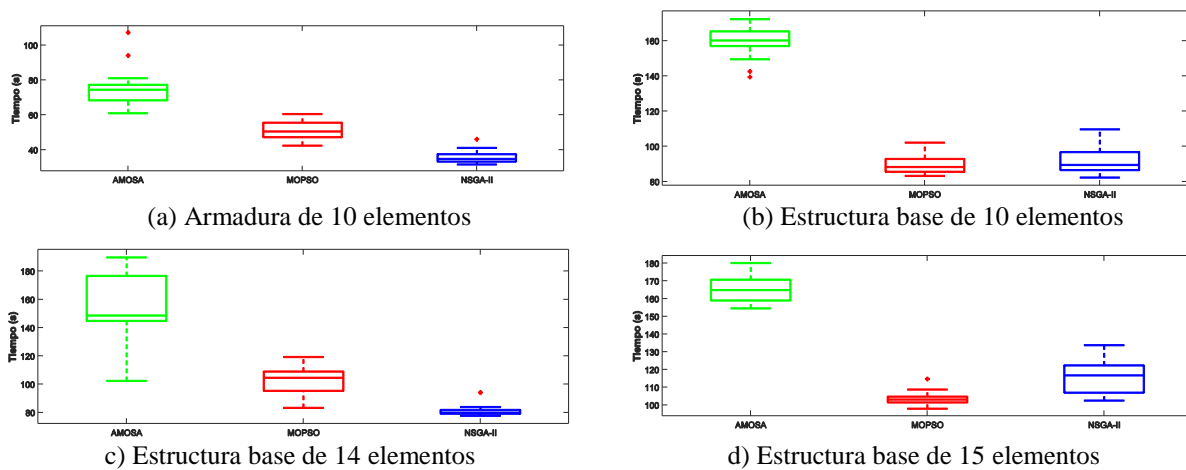


Figura 57. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de optimización multiobjetivo de celosías

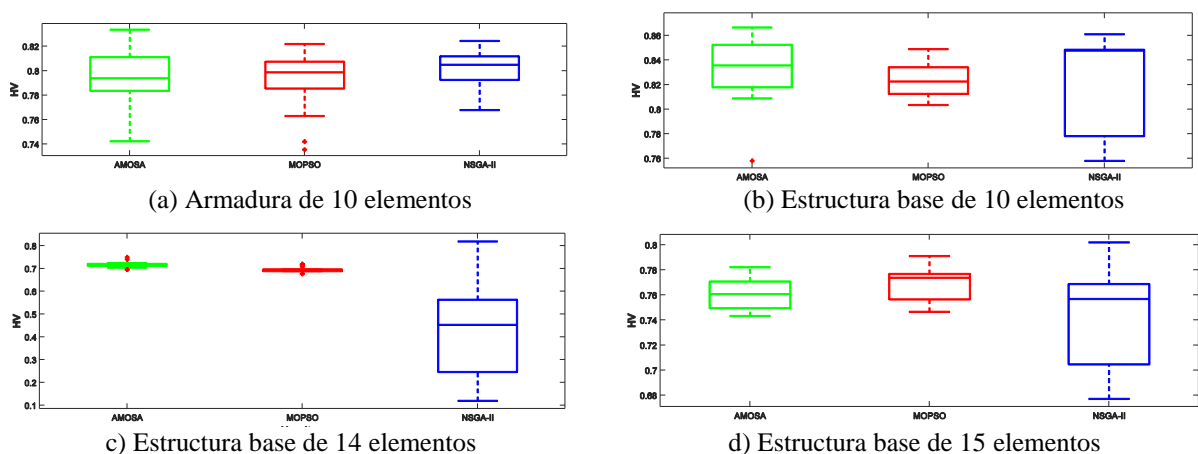


Figura 58. Diagramas de cajas para hipervolumen (HV) de los algoritmos AMOSA, MOPSO y NSGA-II en los problemas de optimización multiobjetivo de celosías

A partir de la Figura 57, el tiempo empleado por los algoritmos en cada ejecución para resolver el problema de optimización de tamaño de la armadura de 10 elementos fue de 40 a 80 s, siendo el más rápido el NSGA-II con 30 a 40 s, seguido por el MOPSO con 40 a 60 s y el AMOSA con 60 a 80 s oscila en el intervalo de 40 y 180 segundos. Para los 3 problemas de optimización topológica, el tiempo de los algoritmos aumentó y fue de 80 a 180 s teniendo en cuenta un aumento en la complejidad del problema (quitar elementos de la estructura base). En este caso los algoritmos NSGA-II y MOPSO fueron los más rápidos con un tiempo similar de 80 a 120 s y el algoritmo con mayor tiempo fue el AMOSA en el intervalo de 140 a 180 s. Como cada algoritmo metaheurístico (NSGA-II, MOPSO y AMOSA) que hace parte del algoritmo AOTMOH se ejecutó 20 veces, de manera secuencial (en este proyecto) para generar el frente de Pareto envolvente, el tiempo empleado en el proceso de optimización por el algoritmo AOTMOH para las 4 estructuras analizadas es cercano a los 10800 s (3 horas), considerando que cada uno de los algoritmos metaheurísticos (NSGA-II, MOPSO y AMOSA) se demora 180 s en cada ejecución.

Por otro lado, al analizar la convergencia y diversidad (ver Figura 58), se observa que los 3 algoritmos metaheurísticos presentan un valor similar en el primer problema con un valor de HV cercano a 0.8 y el valor de HV para el frente de Pareto envolvente calculado (ver Figura 44 izquierda) es de 0.8409. En el caso de los 3 problemas de optimización topológica, en especial para las estructuras base de 14 y 15 elementos se observa que existe una mayor variación de los valores de HV del algoritmo NSGA-II, en comparación con los algoritmos MOPSO y AMOSA (los cuales son cercanos entre sí), lo cual corresponde a los resultados gráficos de la Figura 50 y Figura 54, donde el algoritmo NSGA-II presenta problemas de consistencia en algunas ejecuciones y donde los algoritmos MOPSO y AMOSA fueron consistentes en todas las ejecuciones. El valor de HV para el frente de Pareto envolvente fue  $HV=0.8650$  para la estructura base de 10 elementos

(ver Figura 47 izquierda),  $HV=0.7201$  para la estructura base de 14 elementos (ver Figura 51 izquierda) y  $HV=0.7749$  para la estructura base de 15 elementos (ver Figura 55 izquierda).

Para los 4 problemas analizados, el valor de HV calculado con el frente de Pareto envolvente fue mejor (más cercano a 1) que el valor obtenido por cada algoritmo metaheurísticos (NSGA-II, MOPSO y AMOSA) de forma individual en cada ejecución, lo cual muestra la utilidad de combinar los resultados en un único frente de Pareto envolvente como solución del problema de optimización.

Finalmente, con base en los resultados presentados en esta sección se puede concluir que se valida el funcionamiento del Algoritmo de Optimización Topológica Multiobjetivo Híbrido (AOTMOH), el cual, como se dijo anteriormente, es la integración: los algoritmos metaheurísticos (NSGA-II, MOPSO y AMOSA), el algoritmo de análisis estructural y el algoritmo de la estrategia de generación de la estructura base.

## **6.2. Problemas de optimización topológica multiobjetivo con el método NCN**

En esta sección se presentan los resultados de aplicación del algoritmo AOTMOH en cuatro problemas, usando el método NCN (ver Sección 4.3.1). El proceso de optimización se ejecutó 20 veces con cada uno de los algoritmos metaheurísticos NSGA-II, MOPSO y AMOSA que componen el AOTMOH, para minimizar el peso  $W$  y la energía de deformación  $E_s$ . Las tres estructuras base de 10, 17 y 39 elementos fueron estudiadas por Assimi et al. (2017) usando el algoritmo GP (Genetic Programming) para resolver el problema de optimización topológica de un solo objetivo (minimizar el peso estructural). La cuarta estructura base de 54 elementos fue estudiada por Richardson et al. (2012), quienes usaron el algoritmo MOGA (Multiobjective

Genetic Algorithm) y el enfoque KSR (Kinematic Stability Repair) para resolver el problema de optimización topológica para minimizar el peso estructural y la primera frecuencia natural.

**6.2.1. Estructura base de 10 elementos y 6 nodos.** El primer problema se muestra en la Figura 59 y tiene la misma geometría y condiciones de borde (apoyos y cargas) que los problemas de la Sección 6.1.1 y 6.1.2, excepto que se cambia una función objetivo (ahora se usa la función energía de deformación en la lugar de la función desplazamiento), se agregan restricciones (la función desplazamiento se convierte en restricción) y cambia el tipo de variables utilizado (ahora son variables continuas). El material tiene una densidad  $\rho = 0.1 \text{ lb/in}^3$  y un módulo de elasticidad  $E = 10000 \text{ ksi}$ . La estructura tiene una restricción de esfuerzo en los elementos a tensión y compresión de  $|\sigma^\pm| \leq 25 \text{ ksi}$  y una restricción de desplazamiento en todos los nodos en las direcciones  $x$  y  $y$  de  $|\delta| \leq 2.0 \text{ in}$ . El área permitida de los elementos puede tomar valores continuos en el intervalo  $A_{lim} = [1, 35] \text{ in}^2$ . El valor de la carga  $P = 100 \text{ kips}$  y la longitud  $L = 360 \text{ in}$ .

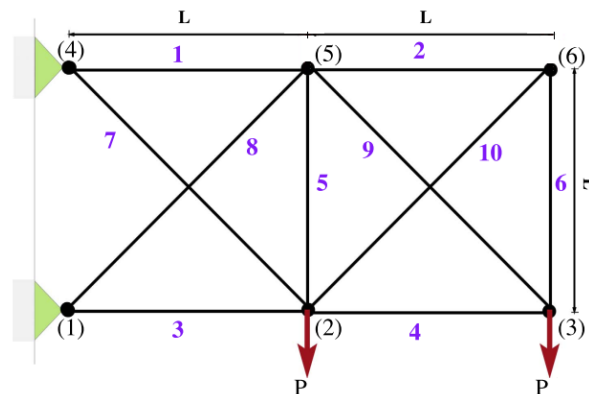


Figura 59. Estructura base de 10 elementos y 6 nodos  
Adaptado: Assimi et al. (2017)

**6.2.2. Estructura base de 17 elementos y 9 nodos.** El segundo problema se muestra en la Figura 60. El material tiene una densidad  $\rho = 0.268 \text{ lb/in}^3$  y un módulo de elasticidad  $E = 30000 \text{ ksi}$ . La estructura tiene una restricción de esfuerzo en los elementos a tensión y

compresión  $|\sigma^\pm| \leq 50 \text{ ksi}$ , una restricción de desplazamiento en todos los nodos en las direcciones  $x$  y  $y$  de  $|\delta| \leq 2.0 \text{ in}$ . El área permitida de los elementos puede tomar valores continuos en el intervalo  $A_{lim} = [0.1, 20] \text{ in}^2$ . El valor de la carga  $P = 100 \text{ kips}$  y la longitud  $L = 100 \text{ in}$ .

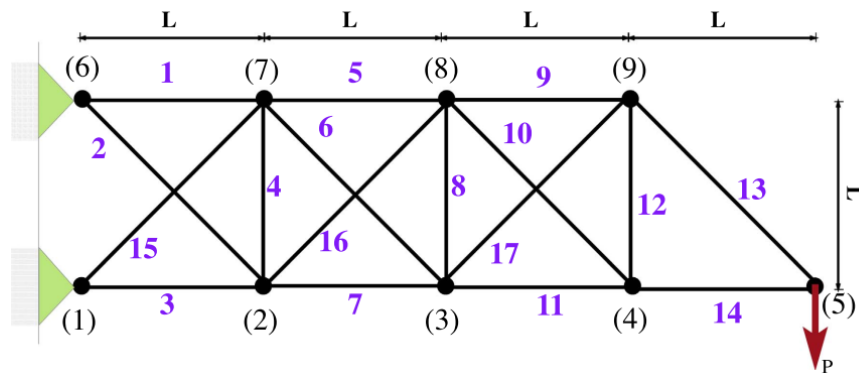


Figura 60. Estructura base de 17 elementos y 9 nodos  
Adaptado: Assimi et al. (2017)

**6.2.3. Estructura base 39 elementos y 12 nodos.** El tercer problema se muestra en la Figura 61. El material tiene una densidad  $\rho = 0.1 \text{ lb/in}^3$  y un módulo de elasticidad  $E = 10000 \text{ ksi}$ . La estructura tiene una restricción de esfuerzo en los elementos a tensión y compresión de  $|\sigma^\pm| \leq 20 \text{ ksi}$ , una restricción de desplazamiento en todos los nodos en las direcciones  $x$  y  $y$  de  $|\delta| \leq 2.0 \text{ in}$ . El área permitida de los elementos puede tomar valores continuos en el intervalo  $A_{lim} = [0.05, 2.5] \text{ in}^2$ . El valor de la carga  $P = 100 \text{ kips}$  y la longitud  $L = 100 \text{ in}$ .

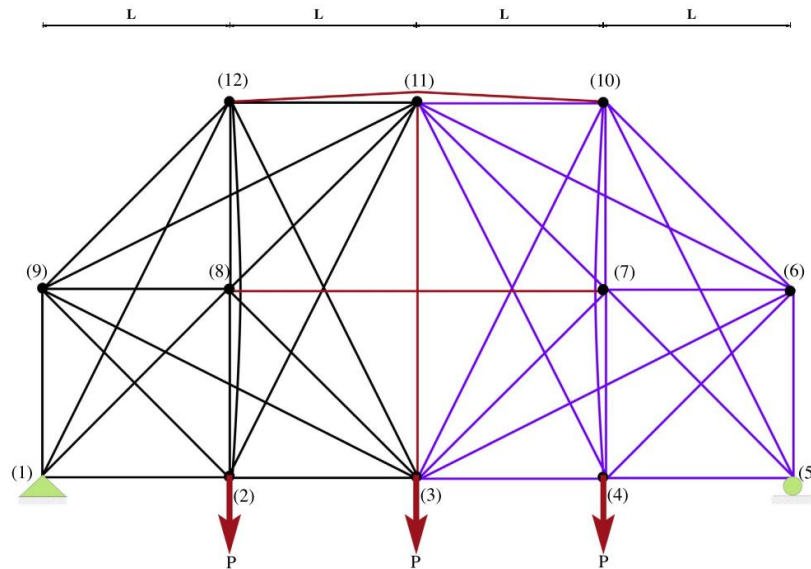


Figura 61. Estructura base de 39 elementos y 12 nodos  
Adaptado: Assimi et al. (2017)

**6.2.4. Estructura base 54 elementos y 15 nodos.** Para el cuarto problema (ver Figura 62),  $\rho = 0.1 \text{ lb/in}^3$ ,  $E = 10000 \text{ ksi}$ ,  $P = 22.5 \text{ kips}$  y  $l = 40 \text{ in}$ . La estructura está sujeta a las restricciones:  $|\sigma^\pm| \leq 25 \text{ ksi}$ ,  $|\delta| \leq 2.0 \text{ in}$  y esfuerzo de pandeo en los elementos a compresión de  $|\sigma^-| \leq (3.96EA_i)/L_i^2$ . El área de los elementos se toma de un conjunto de 64 valores discretos  $D = \{0.111, 0.141, 0.25, 0.307, 0.391, 0.442, 0.563, 0.602, 0.766, 0.785, 0.994, 1, 1.228, 1.266, 1.457, 1.563, 1.62, 1.8, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.8, 4.97, 5.12, 5.74, 7.22, 7.97, 8.53, 9.3, 10.85, 11.5, 13.5, 13.9, 14.2, 15.5, 16, 16.9, 18.8, 19.9, 22, 22.9, 24.5, 26.5, 28, 30, 33.5\} \text{ in}^2$ .

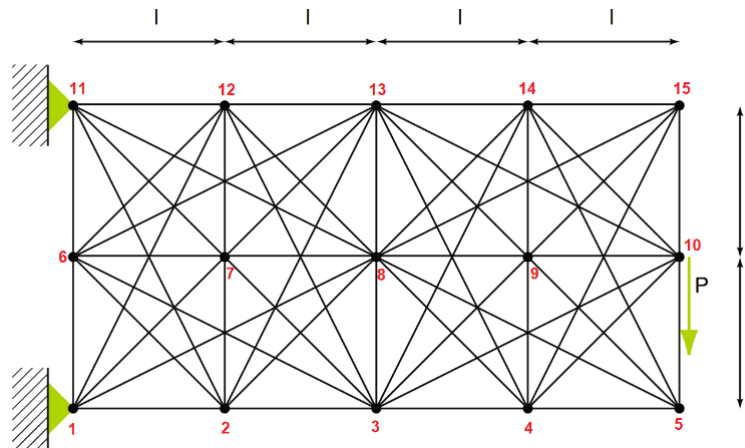


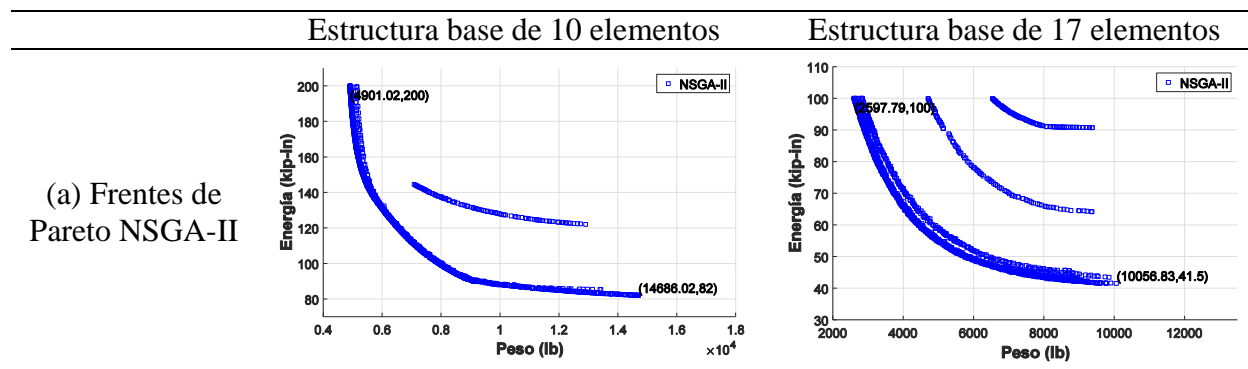
Figura 62. Estructura base de 54 elementos y 15 nodos  
Adaptado: Richardson et al. (2012)

**6.2.5. Resultados de desempeño de los algoritmos en las estructuras base generadas usando el método NCN.** Para cada problema los resultados gráficos reportados son:

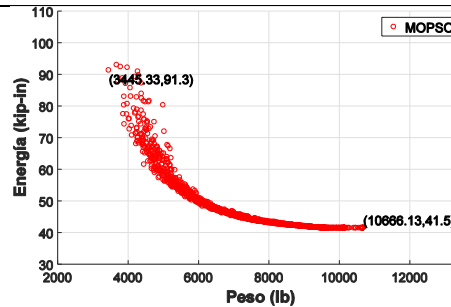
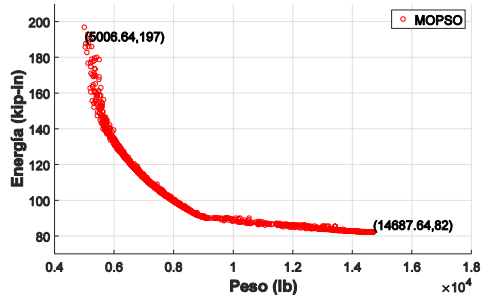
- Los 20 frentes de Pareto de cada uno de los algoritmos metaheurísticos (NSGA-II, MOPSO y AMOSA) y la construcción del frente de Pareto envolvente: Tabla 21
- Representación de las soluciones óptimas decodificadas del frente de Pareto envolvente: Figura 63.
- Diagramas de cajas (boxplots) con el costo computacional y la métrica de Hipervolumen (HV) para cada algoritmo metaheurístico (NSGA-II, MOPSO y AMOSA): Figura 64 y Figura 65.

Tabla 21.

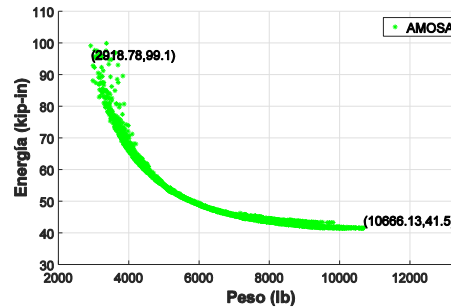
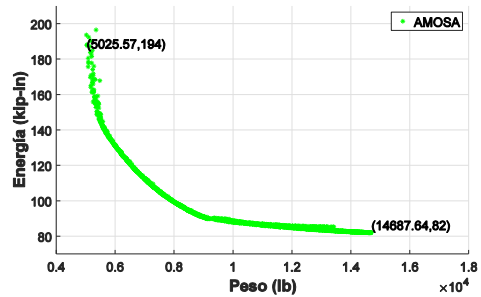
*Desempeño del algoritmo AOTMOH en las 4 estructuras base generadas usando el método NCN*



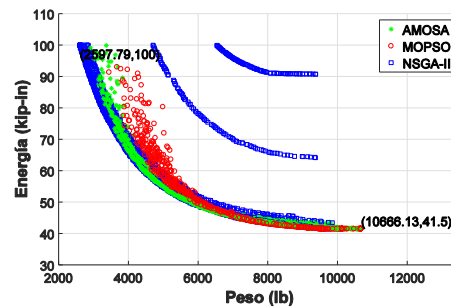
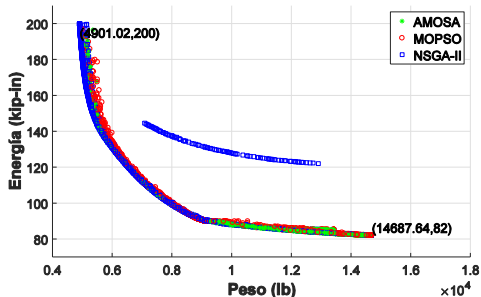
(b) Frentes de Pareto MOPSO



(c) Frentes de Pareto AMOSA



(d) Frentes de Pareto NSGA-II, MOPSO y AMOSA



(e) Frente de Pareto envolvente

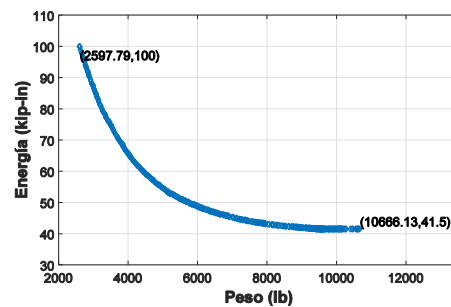
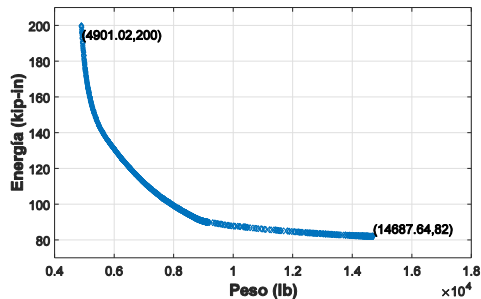
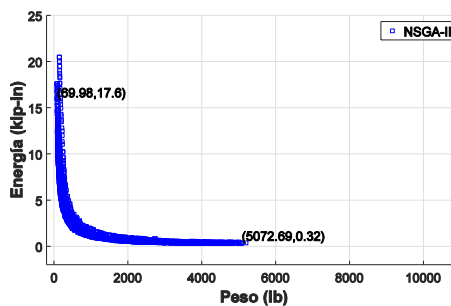
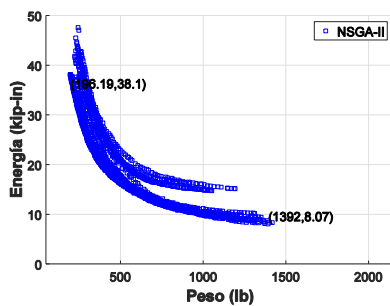


Tabla 21. Desempeño de los algoritmos NSGA-II, MOPSO y AMOSA en las 4 estructuras base generadas usando el método NCN (continuación)

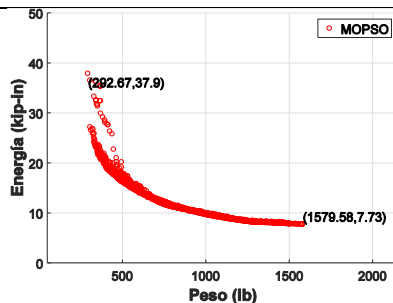
Estructura base de 39 elementos

Estructura base de 54 elementos

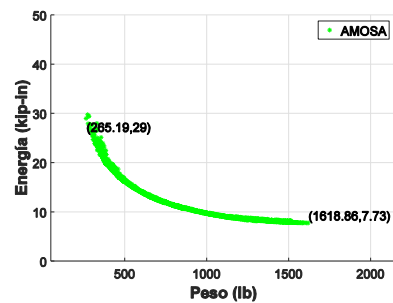
(a) Frentes de Pareto NSGA-II



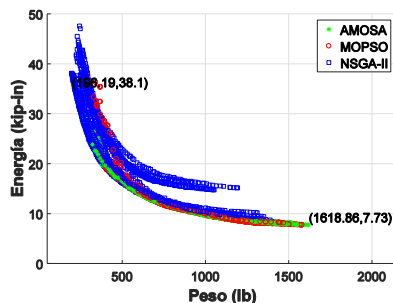
(b) Frentes de Pareto MOPSO



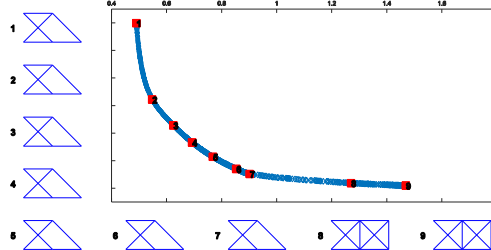
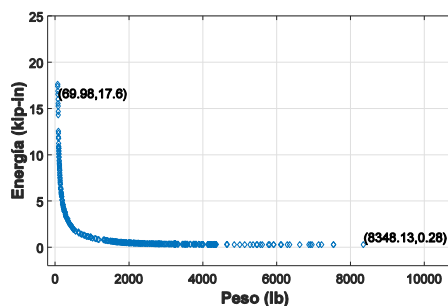
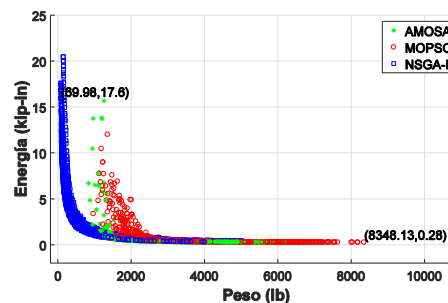
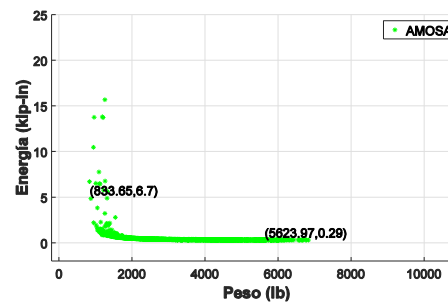
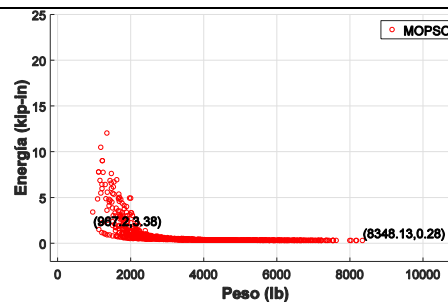
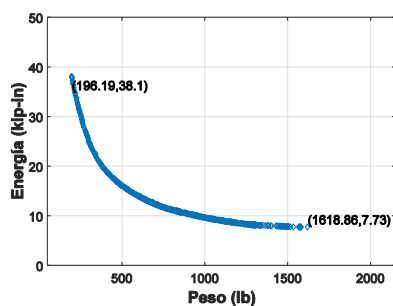
(c) Frentes de Pareto AMOSA



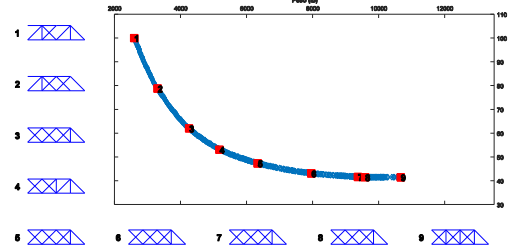
(d) Frentes de Pareto NSGA-II, MOPSO y AMOSA



(e) Frente de Pareto envolvente



(a) Soluciones de la estructura base de 10 elementos



(b) Soluciones de la estructura base de 17 elementos

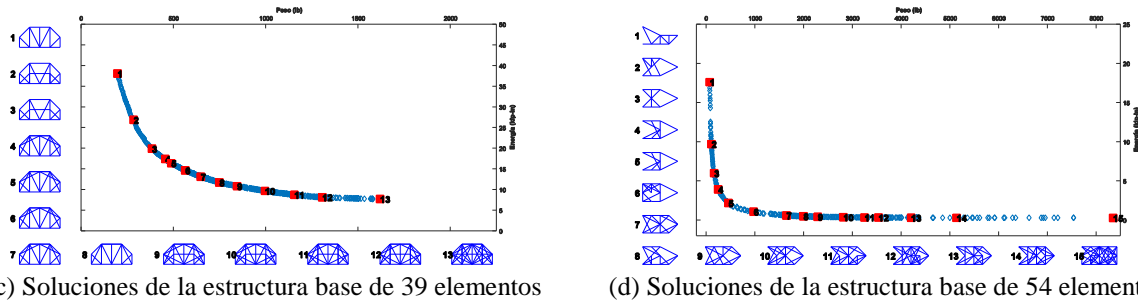


Figura 63. Soluciones decodificadas del Frente de Pareto envolvente para las 4 estructuras base con el método NCN

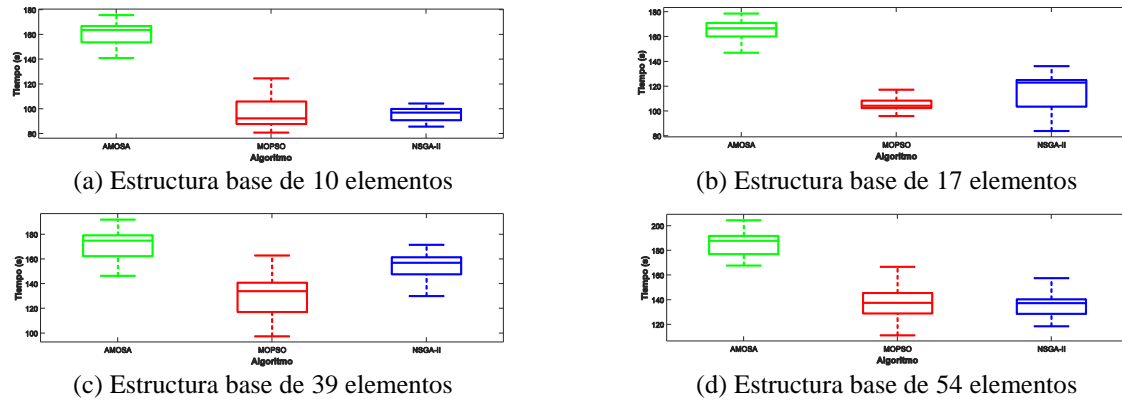


Figura 64. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en las 4 estructuras base generadas con el método NCN

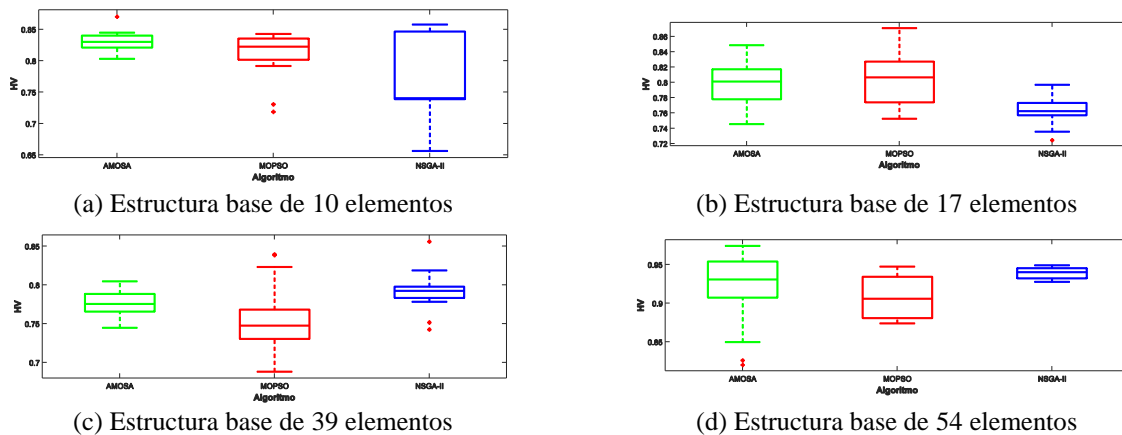


Figura 65. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en las 4 estructuras base generadas con el método NCN

**6.2.6. Análisis de resultados de desempeño en las estructuras base generadas con el método NCN.** En esta sección se aplicó el algoritmo AOTMOH con el método NCN (ver Sección 4.3.1), ya que las estructuras base estudiadas estaban definidas inicialmente empleando este método. A partir de la Tabla 21 puede observarse que el algoritmo NSGA-II, en general presenta

buenos resultados de convergencia y diversidad para los cuatro problemas (aunque presentó inconsistencias en algunas de las ejecuciones para los 3 primeros problemas). Los algoritmos MOPSO y AMOSA presentaron resultados de convergencia y diversidad cercanos al NSGA-II, y fueron consistentes en todas las ejecuciones de los 4 problemas, ya que no hay variación en la forma del frente de Pareto.

El uso conjunto de los algoritmos NSGA-II, MOPSO y AMOSA, fue apropiada en estos problemas, ya que fue posible obtener un frente de Pareto envolvente (para cada problema) que se extiende en una mayor región más amplia a lo largo de los dos objetivos, teniendo en cuenta que el algoritmo NSGA-II ofrece mejores soluciones en la zona de peso mínimo (energía de deformación máxima) y los algoritmos MOPSO y AMOSA presentan mejores soluciones en la zona de energía de deformación mínima (peso máximo), como se observa especialmente en la estructura base de 39 elementos (problema 3) y la estructura base de 54 elementos (problema 4).

En la Figura 63, se observa que las soluciones decodificadas de los frentes de Pareto envolvente (para cada uno de los 4 problemas) ubicadas en la zona de mínimo peso tienen pocos elementos, en comparación con la estructura base inicial a partir de la cual fueron generadas. En cambio, a medida que se avanza hacia la zona intermedia y hacia la zona de mínima energía de deformación, las soluciones presentan un mayor número de elementos, ya que para cumplir con este objetivo se requieren estructuras más rígidas (más elementos). En este caso las soluciones se parecen más a la estructura base inicial.

Para el tiempo computacional, en la Figura 64 se observa que el tiempo empleado por los 3 algoritmos en cada ejecución (en la que cada uno de ellos realiza 50000 pruebas de análisis y diseño estructural), para cada problema, oscila entre 100 y 180 segundos. El algoritmo más rápido es el MOPSO, seguido de cerca por el NSGA-II. En este caso el algoritmo AMOSA es el más

lento, aunque la diferencia de tiempo no es muy amplia (20 segundos en promedio en relación al NSGA-II y MOPSO). El tiempo empleado por el algoritmo AOTMOH es cercano a los 10800 s (3 horas), considerando que cada uno de los algoritmos metaheurísticos (NSGA-II, MOPSO y AMOSA) ejecutados de forma secuencial se demora 180 s en cada ejecución.

En la evaluación cuantitativa de desempeño de acuerdo a la Figura 65, los tres algoritmos (NSGA-II, MOPSO y AMOSA) presentaron resultados de convergencia y diversidad (usando la métrica de Hipervolumen (HV)) que son cercanos entre sí, y que tienen valores que oscilan entre 0.7 y 0.85 para cada uno de ellos (acercándose al valor ideal de 1). Estos resultados corresponden a los resultados gráficos de la Tabla 21, en la cual se muestra que la convergencia y diversidad de los algoritmos fueron similares. Los valores de HV para los frentes de Pareto envolvente de la Tabla 21(e) fueron: HV=0.8531 (para la estructura base de 10 elementos), HV=0.8055 (para la estructura base de 17 elementos), HV=0.8285 (para la estructura base de 39 elementos) y HV=0.9781 (para la estructura base de 54 elementos). Estos valores indican que se obtiene una mejor convergencia y diversidad de forma cuantitativa (un mayor valor de HV), cuando se combinan los resultados individuales de optimización de los tres algoritmos metaheurísticos (NSGA-II, MOPSO y AMOSA) que hacen parte del algoritmo AOTMOH en un único frente de Pareto envolvente con las mejores soluciones no dominadas.

A continuación, usando los resultados del proceso de optimización de la Tabla 21 para la estructura base de 17 elementos se analiza la validez de la hipótesis establecida en este trabajo: *“La aplicación de un proceso de optimización topológica multiobjetivo, influye de manera significativa en el desempeño estructural (medido con relación a un diseño inicial satisfactorio) y en la automatización del procedimiento de diseño de estructuras de celosía.*

Considerando como diseños iniciales satisfactorios todas las soluciones del frente de Pareto obtenido en la ejecución número 13 por el algoritmo NSGA-II (señalado en la Figura 66 izquierda), las cuales cumplen con todas las restricciones del problema, puede observarse que el proceso de optimización topológica multiobjetivo aplicado influyó de manera significativa en el desempeño estructural (en relación a los diseños iniciales) de las estructuras de celosía, ya que se logró obtener un frente de Pareto envolvente (Figura 66 derecha) con un conjunto de diseños óptimos finales que son mejores que los iniciales (es decir los dominan para los dos objetivos planteados) y que fueron obtenidos de forma automática usando los algoritmos de optimización. De esta manera, puede concluirse que con la aplicación del algoritmo AOTMOH, se está validando el cumplimiento de la hipótesis planteada.

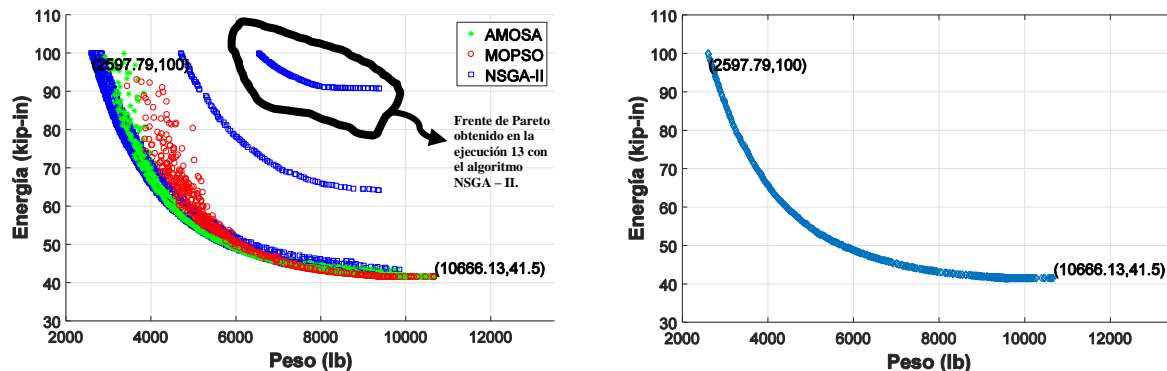


Figura 66. Frentes de Pareto de los algoritmos metaheurísticos (NSGA-II, MOPSO y AMOSA) y frente de Pareto envolvente para el problema de la estructura base de 17 elementos

### 6.3. Problemas de optimización topológica multiobjetivo con el método TEM

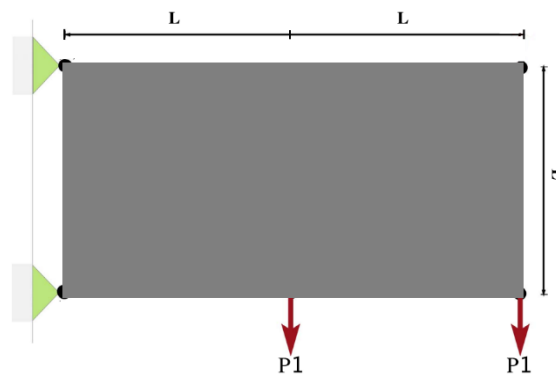
En esta sección se presentan los resultados de aplicación del algoritmo AOTMOH, empleando el método TEM (ver Sección 4.3.2) para generar la estructura base. El proceso de optimización se ejecutó 5 veces con cada uno de los algoritmos metaheurísticos NSGA-II, MOPSO y AMOSA para minimizar el peso  $W$  y la energía de deformación  $E_s$ . Se abordaron los cuatro problemas de

la Sección 6.2, excepto que el espacio de diseño inicial se asume continuo. Los datos conocidos en este caso son las condiciones de cargas, apoyos y propiedades de material.

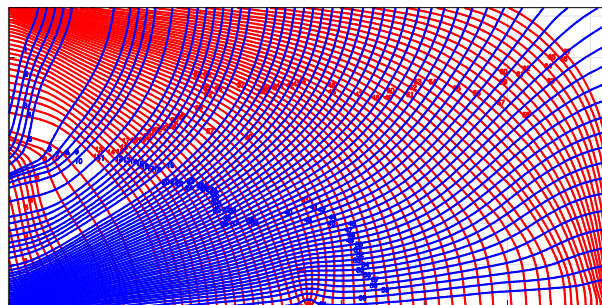
En cada problema se seleccionaron dos valores arbitrarios de porcentaje de trayectorias de esfuerzo,  $PT = 5\%$  y  $PT = 1\%$  (de acuerdo a la descripción de la Sección 4.3.2.2), para generar dos estructuras base. Para cada problema los resultados gráficos reportados incluyen:

- Generación de las 2 estructuras base: Figura 67, Figura 71, Figura 75 y Figura 79.
- Los 5 frentes de Pareto de cada uno de los algoritmos NSGA-II, MOPSO y AMOSA y la construcción del frente de Pareto envolvente para las 2 estructuras base:
- Tabla 22, Tabla 23, Tabla 24 y Tabla 25.
- Representación de las soluciones óptimas decodificadas del frente de Pareto envolvente para las 2 estructuras base: Figura 68, Figura 72, Figura 76 y Figura 80.
- Diagramas de cajas (boxplots) con el costo computacional para cada algoritmo: Figura 69, Figura 73, Figura 77 y Figura 81.
- Diagramas de cajas (boxplots) con la métrica de Hipervolumen (HV) para cada algoritmo: Figura 70, Figura 74, Figura 78 y Figura 82.

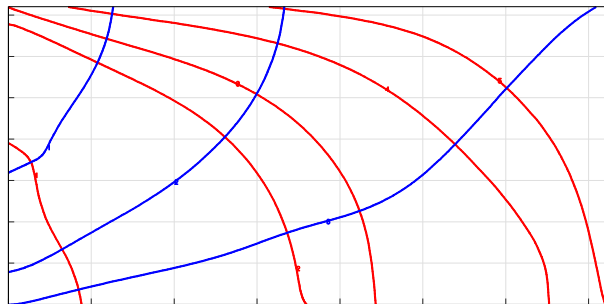
**6.3.1. Viga cantiléver 1.** Es la versión continua del problema presentado en la Sección 6.2.1.



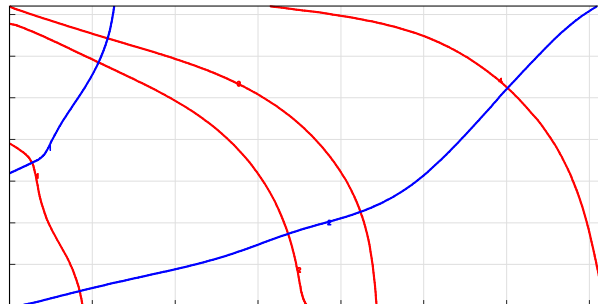
(a) Dominio de diseño inicial continuo



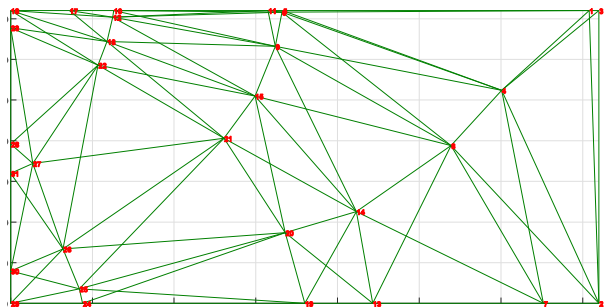
(b) Trayectorias de esfuerzo



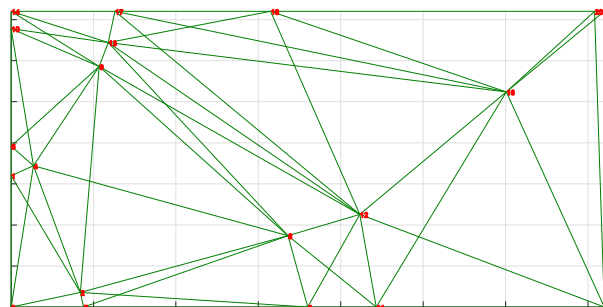
(c) PT=5% de las trayectorias de esfuerzo



(d) PT=1% de las trayectorias de esfuerzo



(d) Estructura base de 94 elementos

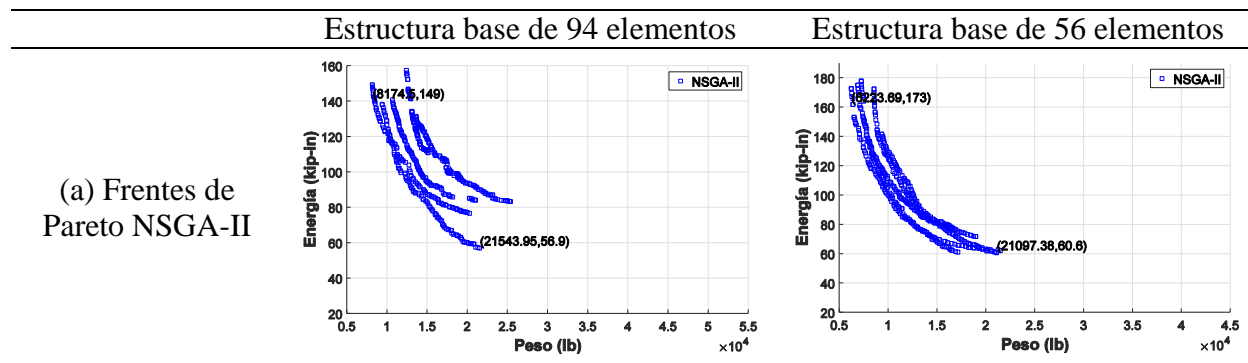


(e) Estructura base de 56 elementos

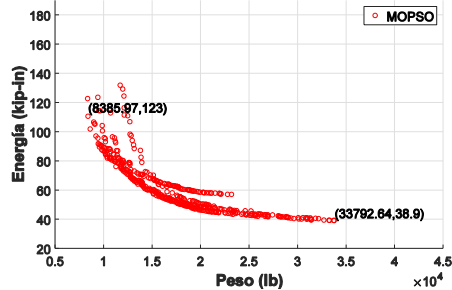
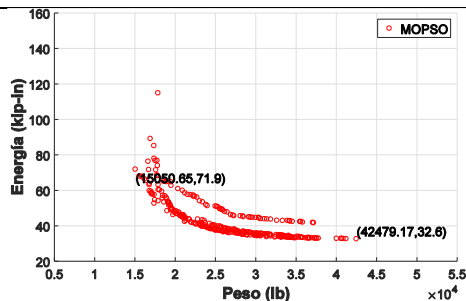
Figura 67. Generación de las estructuras base para la viga cantiléver 1

Tabla 22.

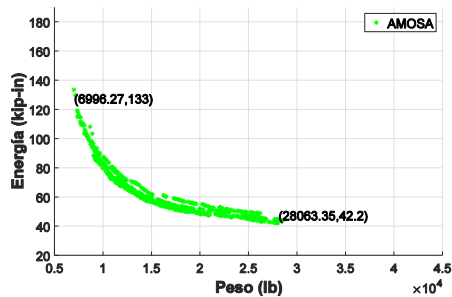
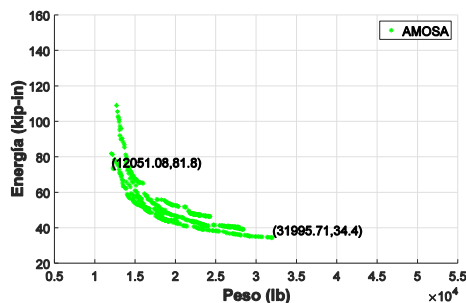
Desempeño del algoritmo AOTMOH en la viga cantiléver 1



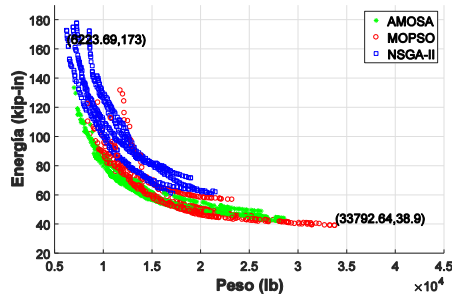
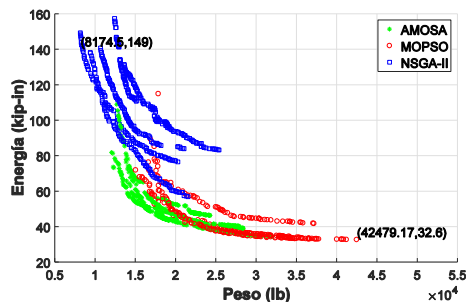
(b) Frentes de Pareto MOPSO



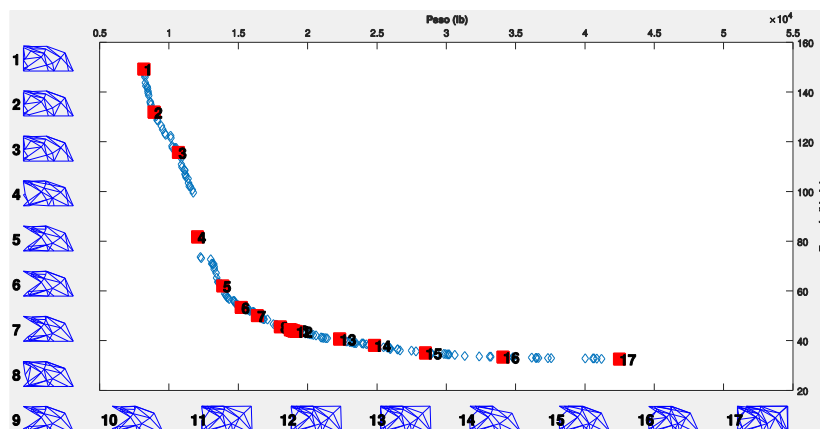
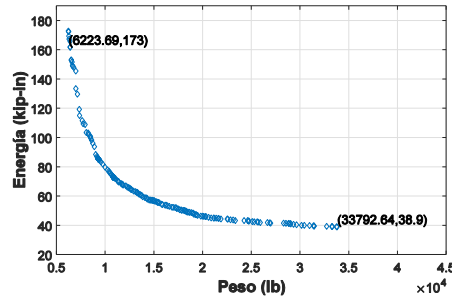
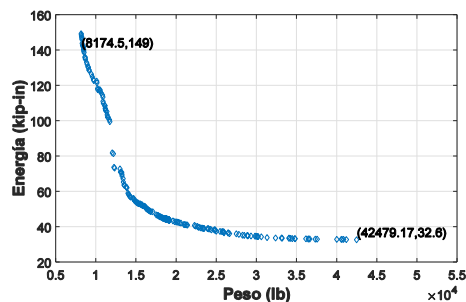
(c) Frentes de Pareto AMOSA



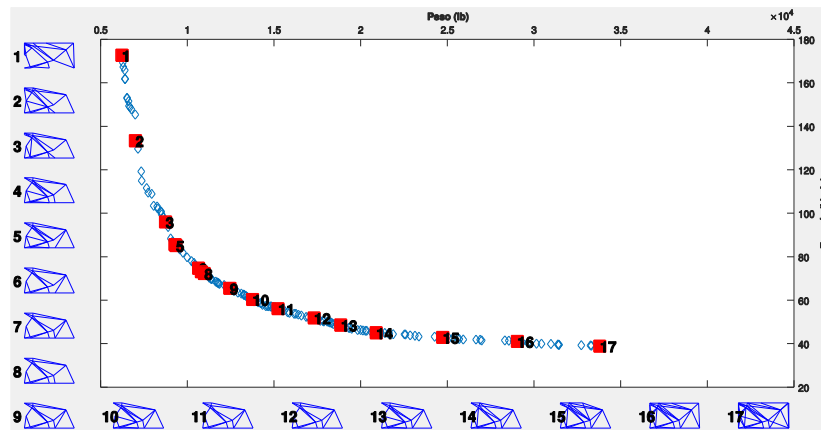
(d) Frentes de Pareto NSGA-II, MOPSO y AMOSA



(e) Frente de Pareto envolvente

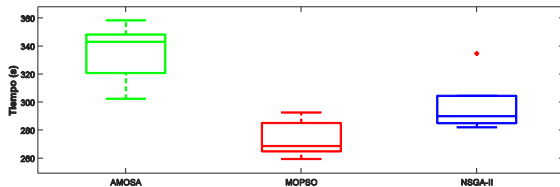


(a) Soluciones de la estructura base de 94 elementos

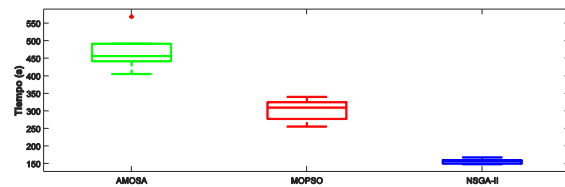


(b) Soluciones de la estructura base de 56 elementos

Figura 68. Soluciones decodificadas del Frente de Pareto envolvente de las dos estructuras base para la viga cantiléver 1

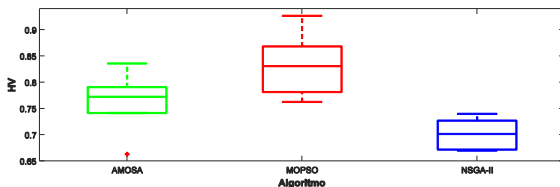


(a) Estructura base de 94 elementos

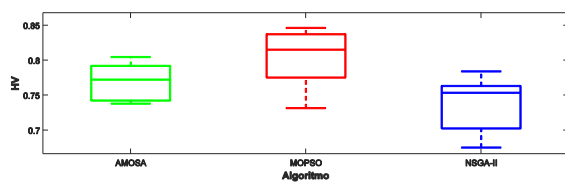


(b) Estructura base de 56 elementos

Figura 69. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 1



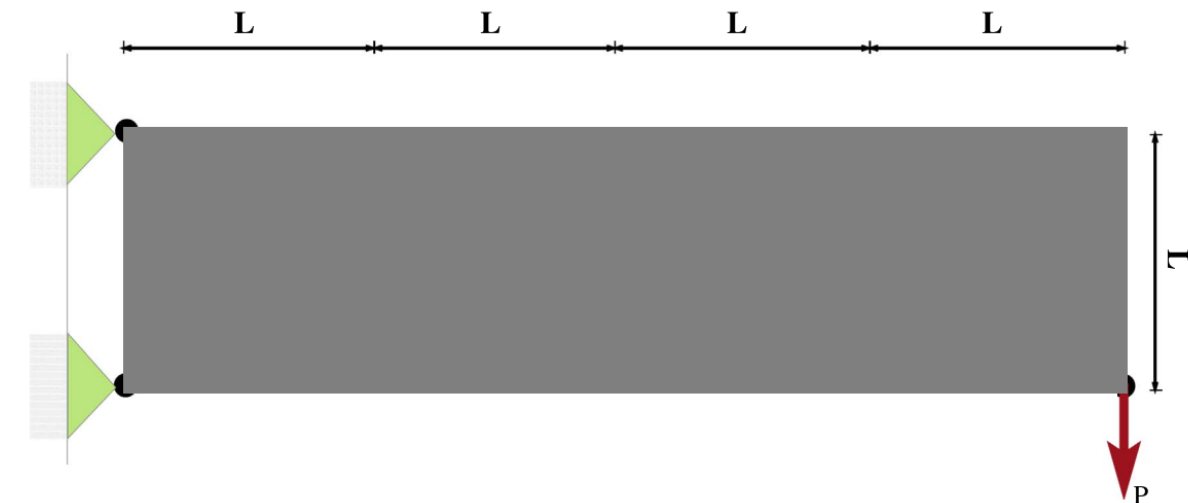
(a) Estructura base de 94 elementos



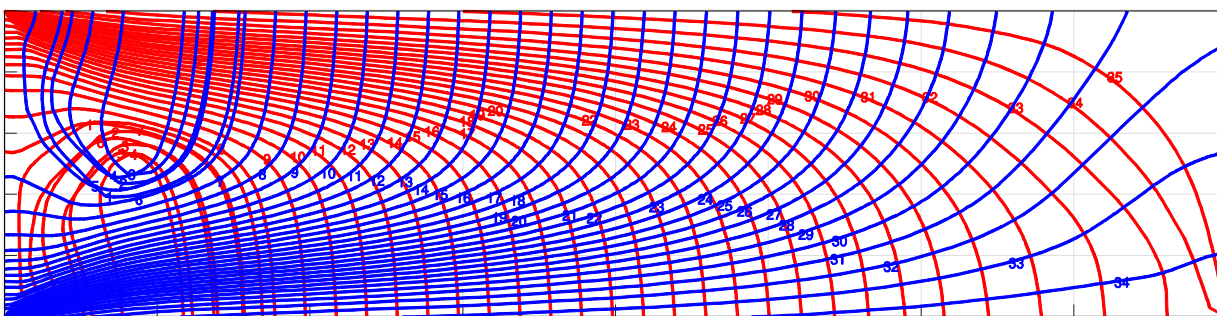
(b) Estructura base de 56 elementos

Figura 70. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 1

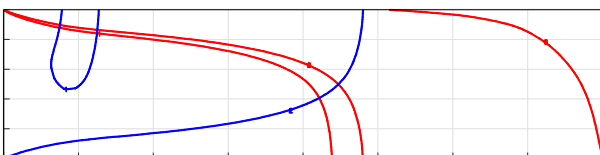
**6.3.2. Viga cantiléver 2.** Es la versión continua del problema presentado en la Sección 6.2.2.



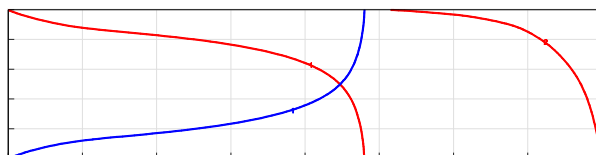
(a) Dominio de diseño inicial continuo



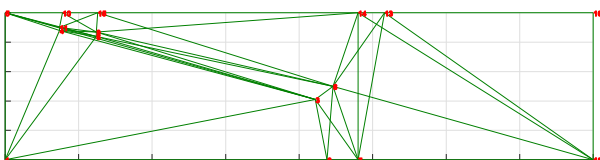
(b) Trayectorias de esfuerzo



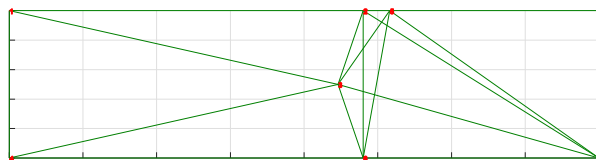
(c) PT=5% de las trayectorias de esfuerzo



(d) PT=1% de las trayectorias de esfuerzo



(e) Estructura base de 46 elementos



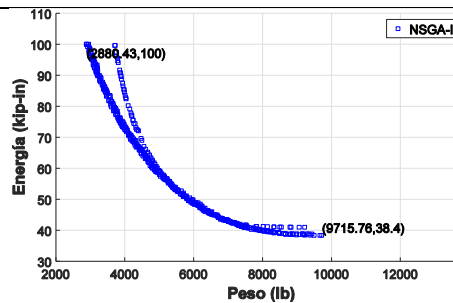
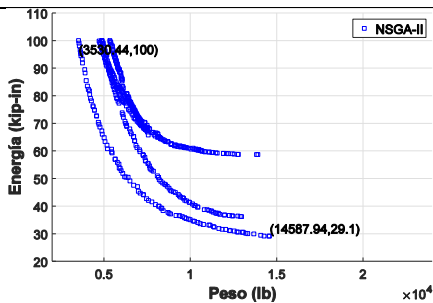
(f) Estructura base de 17 elementos

Figura 71. Generación de las estructuras base para la viga cantiléver 2

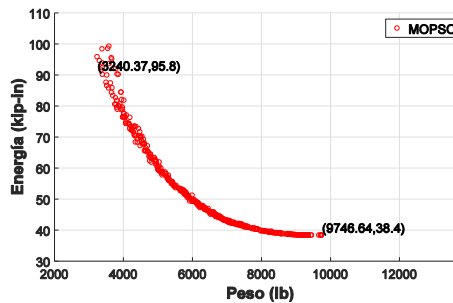
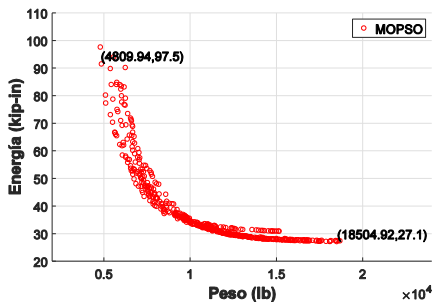
Tabla 23. Desempeño del algoritmo AOTMOH en la viga cantiléver 2

Estructura base de 46 elementos	Estructura base de 17 elementos
---------------------------------	---------------------------------

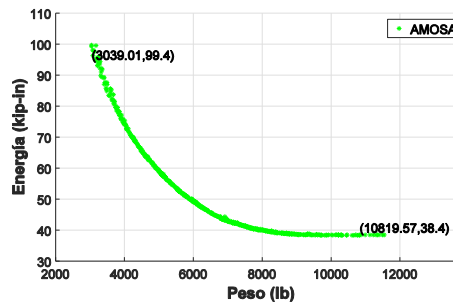
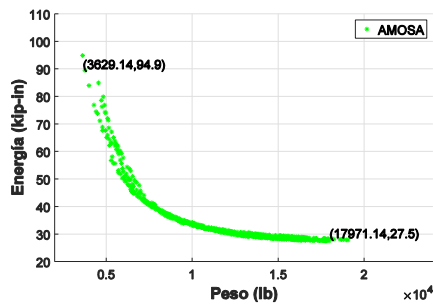
(a) Frentes de Pareto NSGA-II



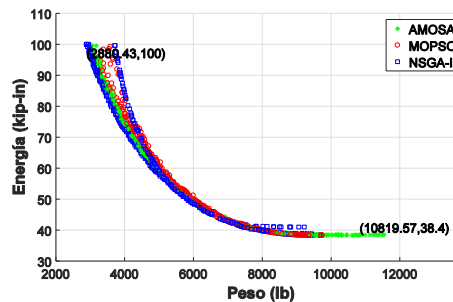
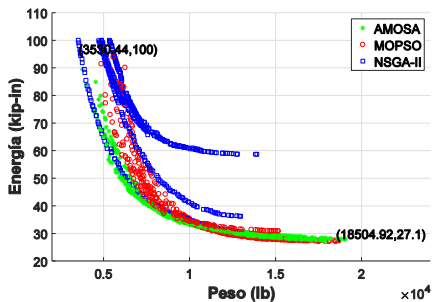
(b) Frentes de Pareto MOPSO



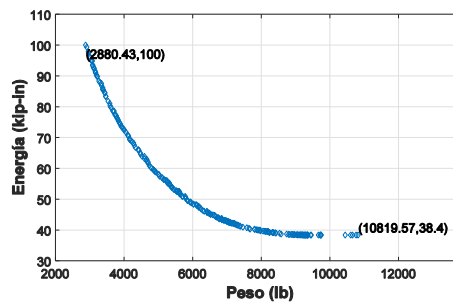
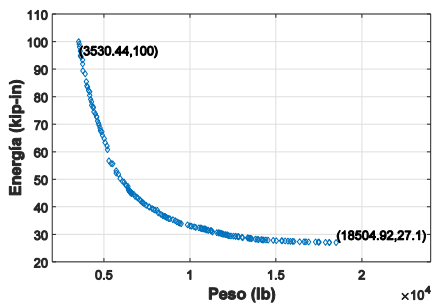
(c) Frentes de Pareto AMOSA

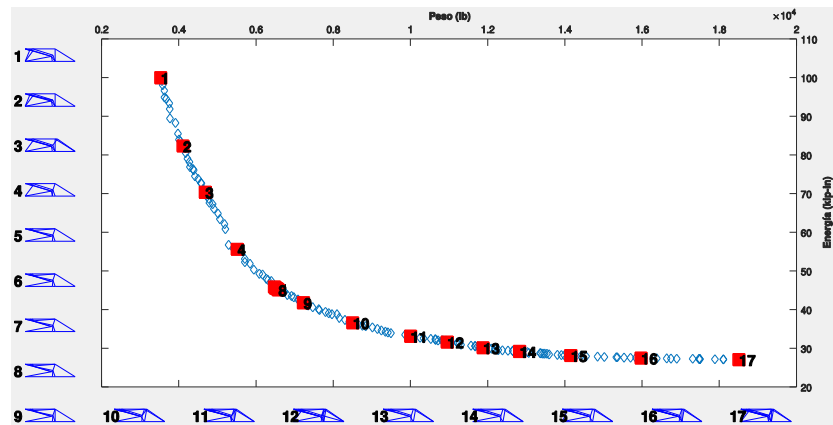


(d) Superposición de frentes de Pareto NSGA-II, MOPSO y AMOSA

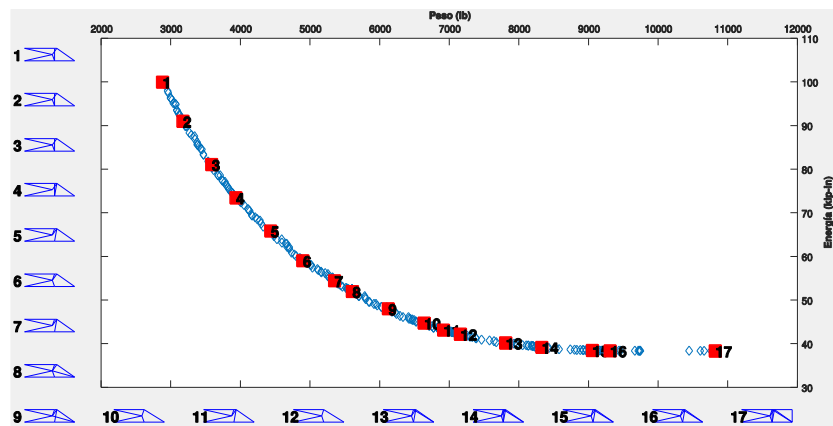


(e) Frente de Pareto envolvente



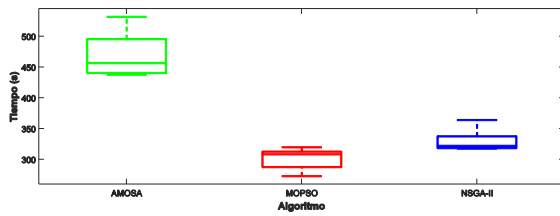


(a) Soluciones de la estructura base de 46 elementos

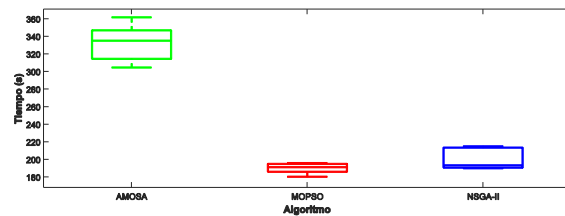


(b) Soluciones de la estructura base de 17 elementos

Figura 72. Soluciones decodificadas del Frente de Pareto envolvente de las dos estructuras base para la viga cantiléver 2

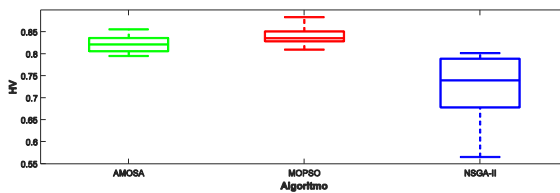


(a) Estructura base de 46 elementos

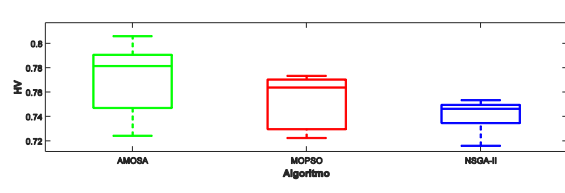


(b) Estructura base de 17 elementos

Figura 73. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 2



(a) Estructura base de 46 elementos



(b) Estructura base de 17 elementos

Figura 74. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 2

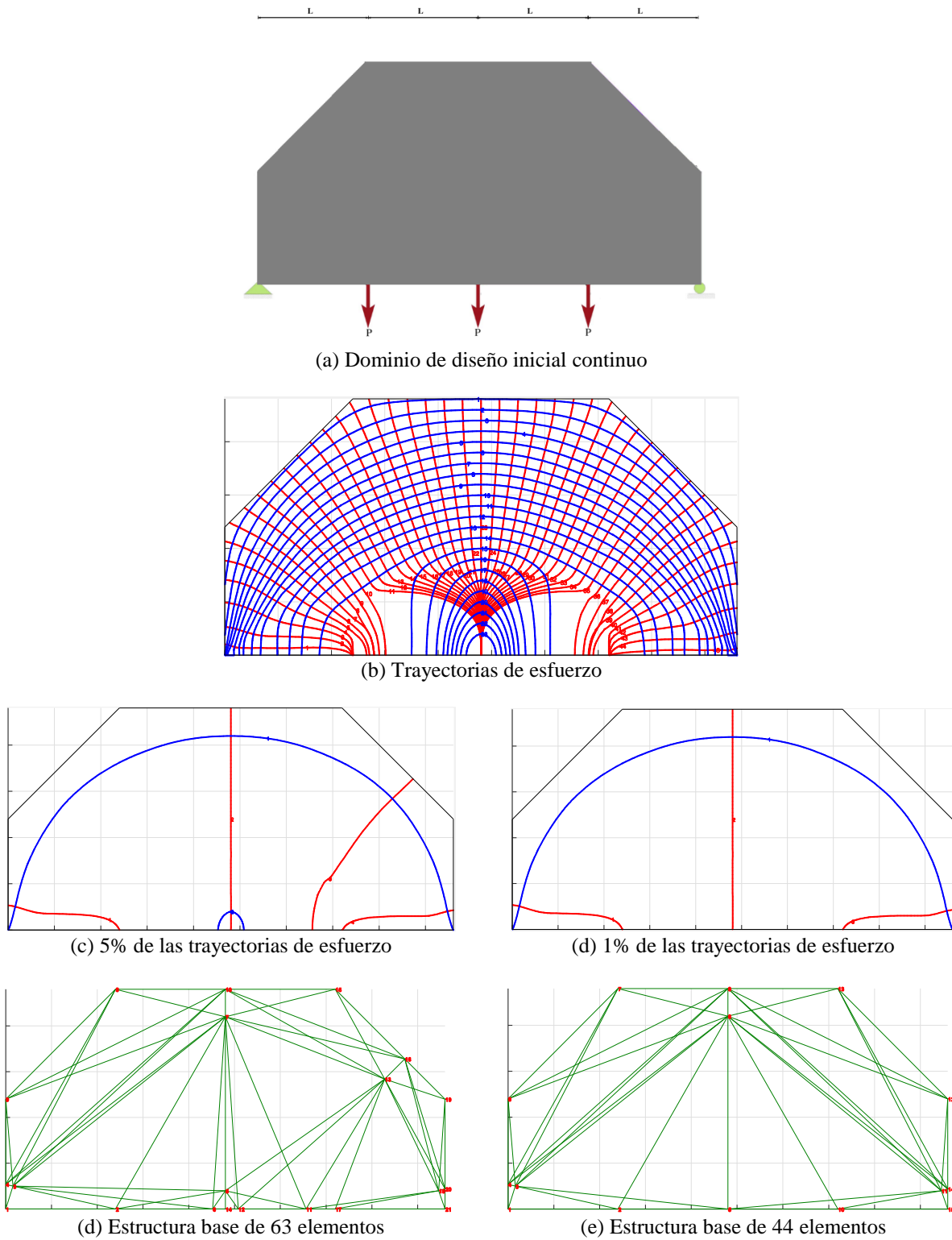
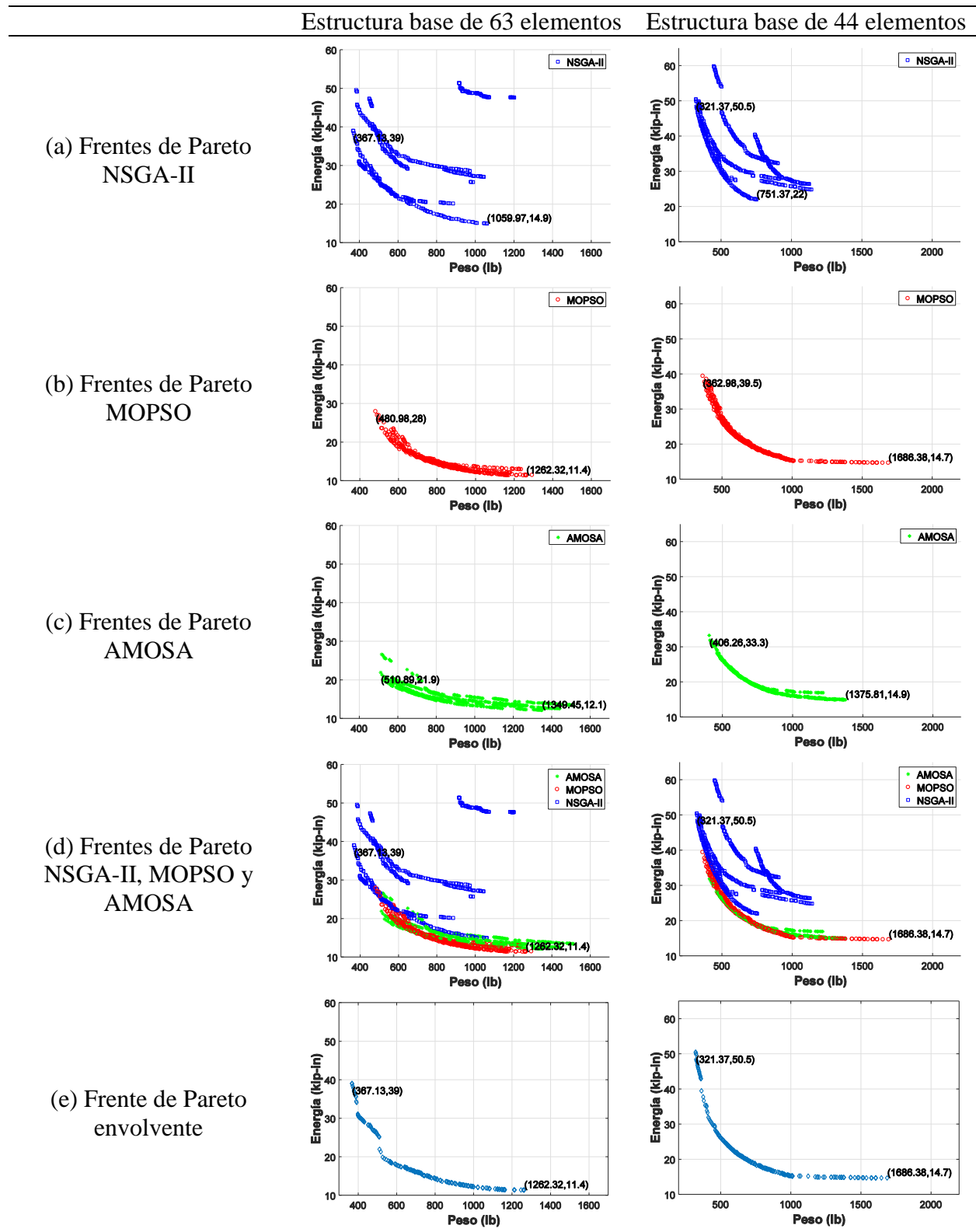
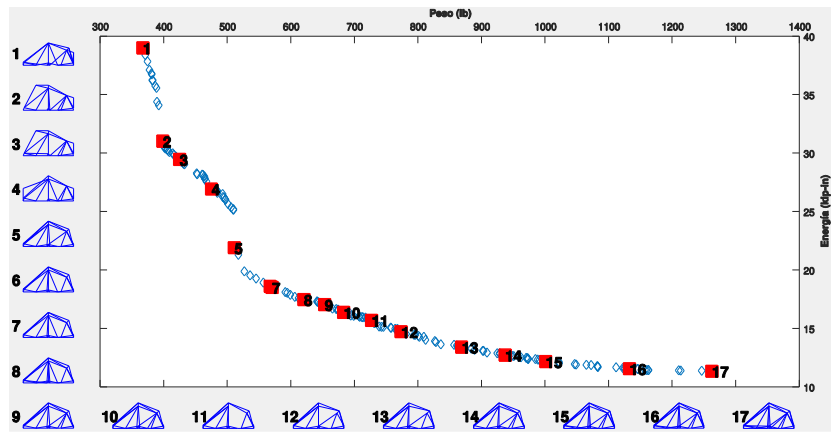
**6.3.3. Viga simplemente apoyada.** Es la versión continua del problema de la Sección 6.2.3.

Figura 75. Generación de las estructuras base para la viga simplemente apoyada

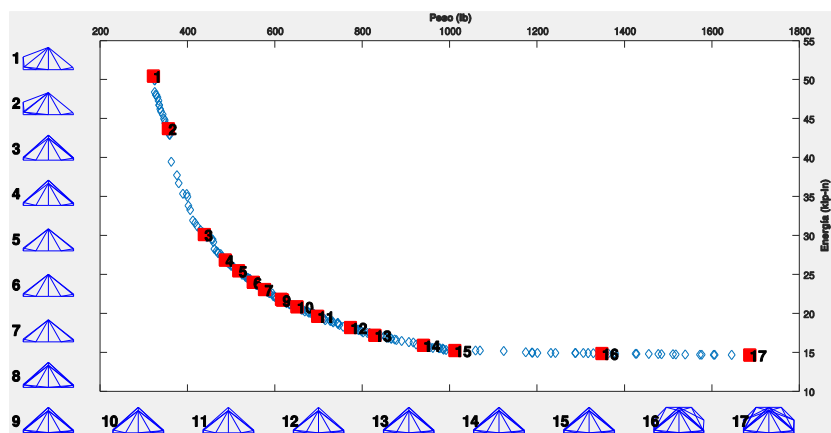
Tabla 24.

Desempeño del algoritmo AOTMOH en la viga simplemente apoyada



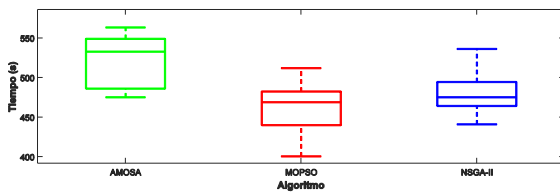


(a) Soluciones de la estructura base de 63 elementos

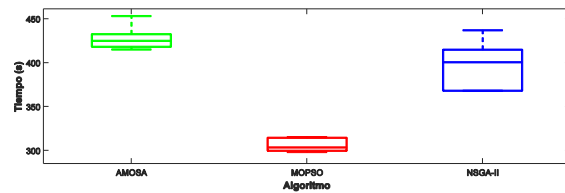


(b) Soluciones de la estructura base de 44 elementos

Figura 76. Soluciones decodificadas del Frente de Pareto envolvente de las dos estructuras base para la viga simplemente apoyada

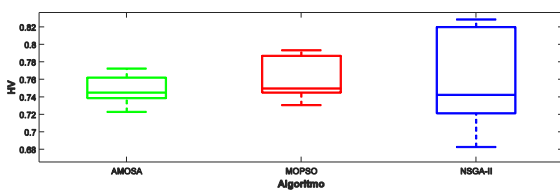


(a) Estructura base de 63 elementos

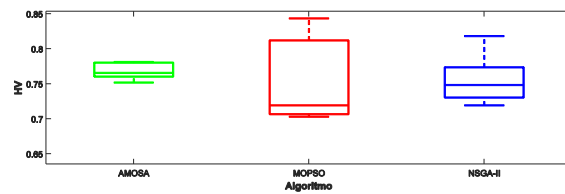


(b) Estructura base de 44 elementos

Figura 77. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la viga simplemente apoyada



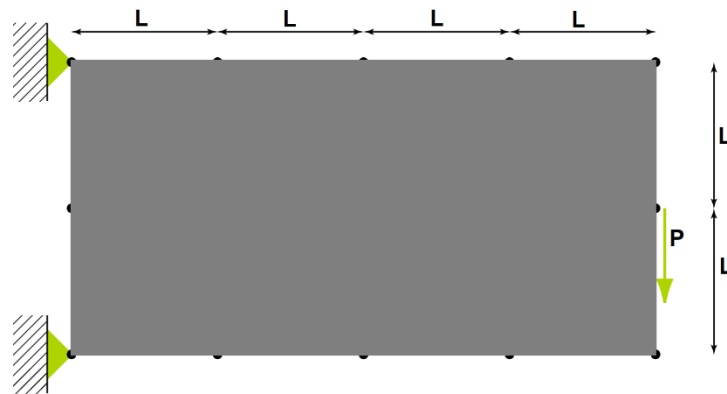
(a) Estructura base de 63 elementos



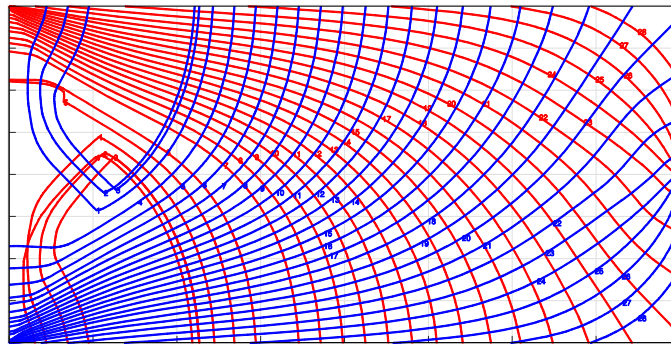
(b) Estructura base de 44 elementos

Figura 78. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver simplemente apoyada

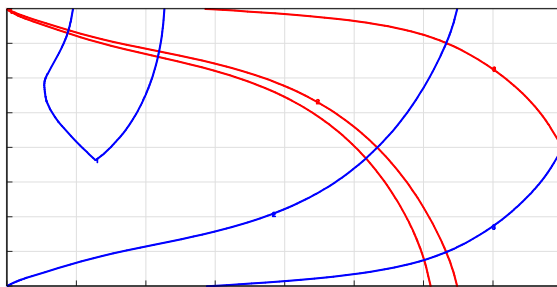
**6.3.4. Viga cantiléver 3.** Es la versión continua del problema presentado en la Sección 6.2.4.



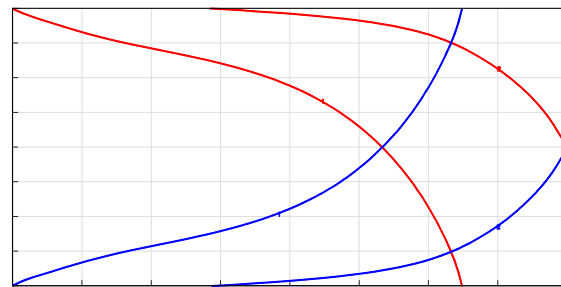
(a) Dominio de diseño inicial continuo



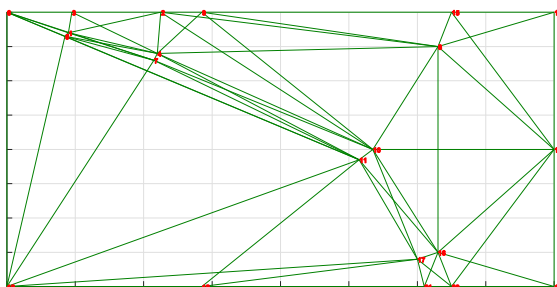
(b) Trayectorias de esfuerzo



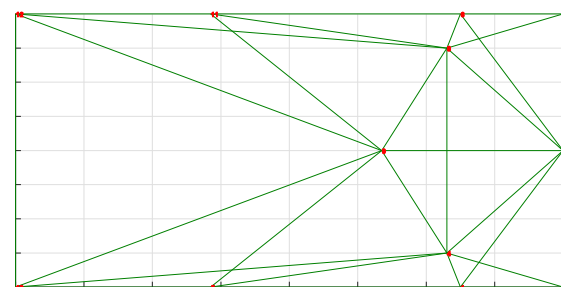
(c) PT=5% de las trayectorias de esfuerzo



(d) PT=1% de las trayectorias de esfuerzo



(d) Estructura base de 63 elementos

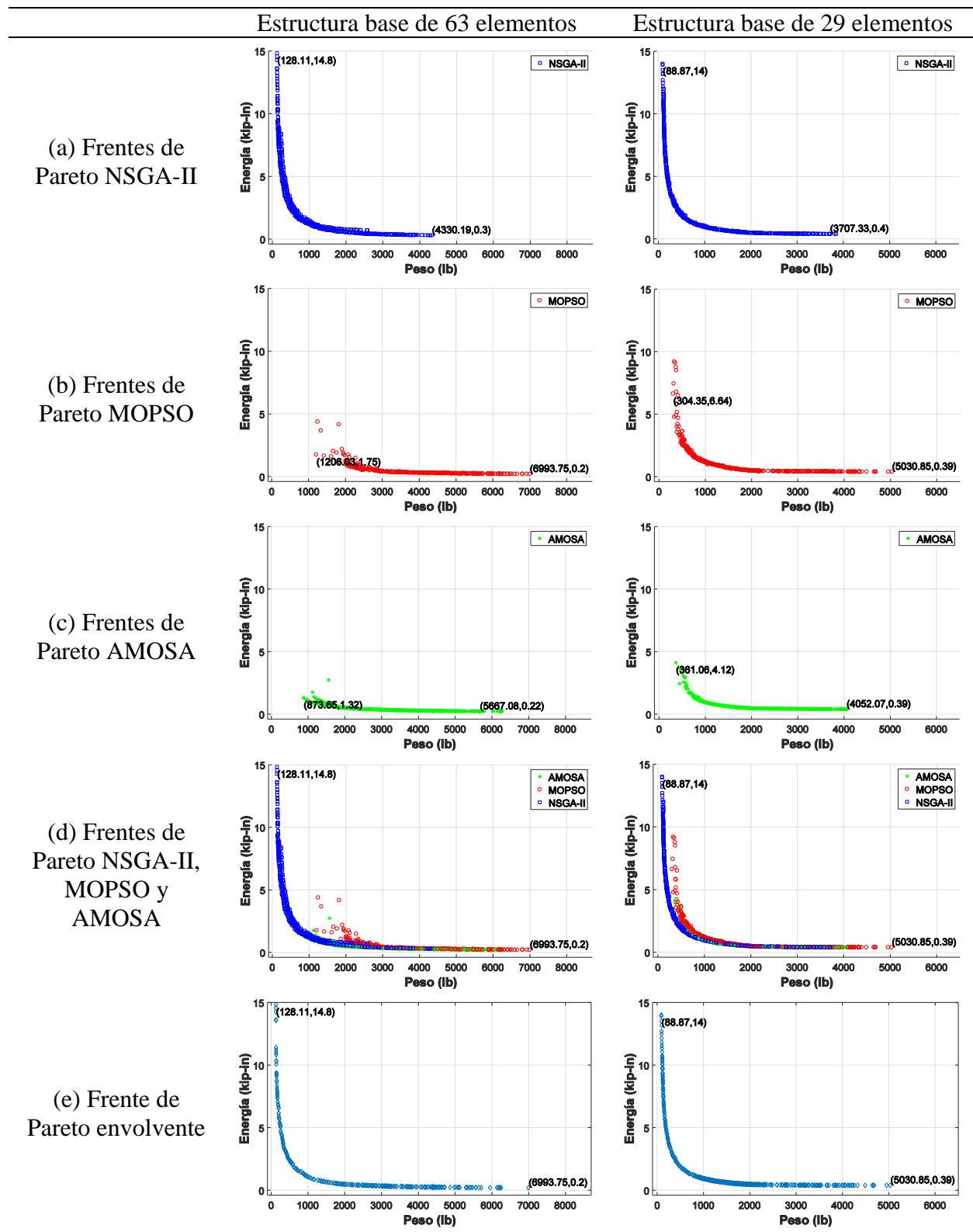


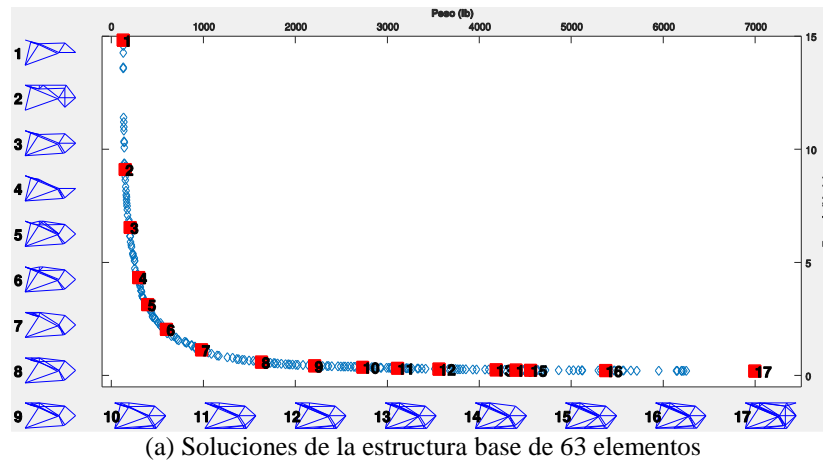
(e) Estructura base de 29 elementos

Figura 79. Generación de las estructuras base para la viga cantiléver 3

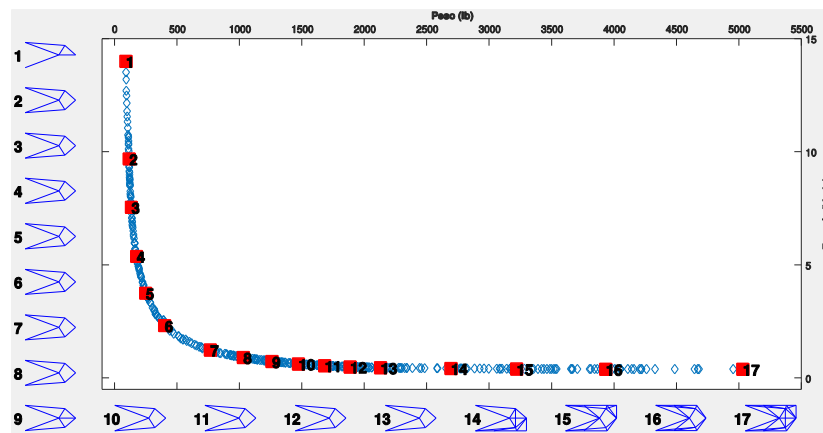
Tabla 25.

Desempeño del algoritmo AOTMOH en la viga cantiléver 3



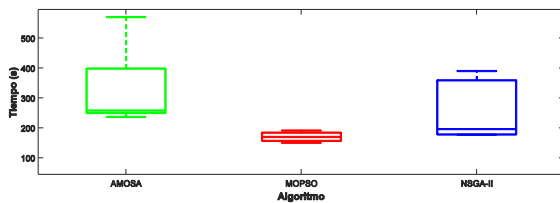


(a) Soluciones de la estructura base de 63 elementos

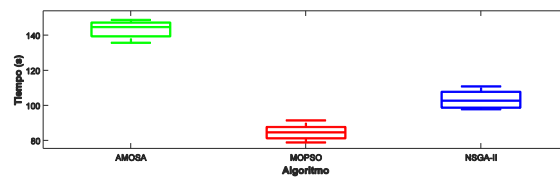


(b) Soluciones de la estructura base de 29 elementos

Figura 80. Soluciones decodificadas del Frente de Pareto envolvente de las dos estructuras base para la viga cantiléver 3

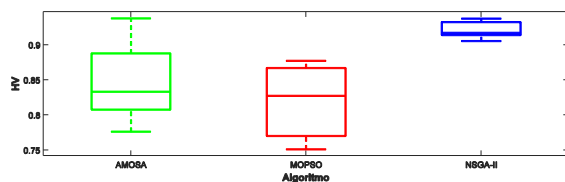


(a) Estructura base de 63 elementos

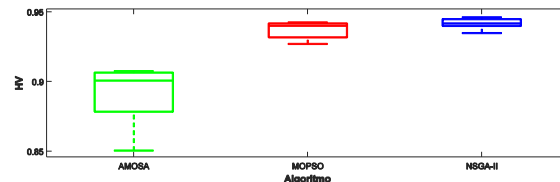


(b) Estructura base de 29 elementos

Figura 81. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 3



(a) Estructura base de 63 elementos



(b) Estructura base de 29 elementos

Figura 82. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la viga cantiléver 3

**6.3.5. Análisis de resultados de desempeño en las estructuras base generadas con el método TEM.** En esta sección se empleó el algoritmo AOTMOH para resolver los 4 problemas de la Sección 6.2, aplicando el método TEM (ver Sección 4.3.2) en la generación de la estructura base. Para cada uno de los 4 problemas se generaron las dos estructuras base correspondientes (ver Figura 67, Figura 71, Figura 75 y Figura 79), de acuerdo a los valores de porcentaje de trayectorias  $PT = 5\%$  y  $PT = 1\%$ , los cuales permiten seleccionar las trayectorias de esfuerzo más representativas (mayor valor de esfuerzo promedio), del total disponible.

A partir de la

Tabla 22, Tabla 23, Tabla 24 y Tabla 25, donde para cada problema se presentan los resultados de convergencia de cada uno de los algoritmos (NSGA-II, MOPSO y AMOSA) y se genera el frente de Pareto envolvente con las mejores soluciones, para las dos estructuras base, puede observarse que para las estructuras base generadas con un valor de  $PT = 1\%$  se obtiene un frente de Pareto con una mayor convergencia hacia el valor mínimo de la función objetivo peso mínimo, teniendo en cuenta que el número de elementos generados es menor en comparación con la estructura generada con un valor  $PT = 5\%$ . Para el caso de la función objetivo energía de deformación los resultados son contrarios, ya que la energía fue menor en la estructura base con  $PT = 5\%$  (al tener un mayor número de elementos).

Con respecto a las soluciones decodificadas del frente de Pareto envolvente para cada problema, de acuerdo a la Figura 68, Figura 72, Figura 76 y Figura 80 respectivamente, la geometría de las soluciones generadas es similar para las dos estructuras base ( $PT=5\%$  y  $PT=1\%$ ), sin embargo para la estructura base con menos elementos ( $PT=1\%$ ), los algoritmos NSGA-II, MOPSO y AMOSA lograron quitar un mayor número de elementos, teniendo en cuenta que la complejidad

del problema disminuye al tener un menor número de variables y un espacio de búsqueda más pequeño en comparación a la estructura base generada con  $PT=5\%$ .

Al analizar el tiempo computacional de los algoritmos, como se muestra en la Figura 69, Figura 73, Figura 77 y Figura 81, el algoritmo más rápido sigue siendo el MOPSO, seguido por el NSGA-II y AMOSA. Como el número de elementos de las estructuras base fue mayor que en los problemas de la Sección 6.2, en general el costo computacional aumentó (ahora los algoritmos se demoran entre 200 y 500 s por cada ejecución), teniendo en cuenta el esfuerzo adicional que deben hacer para ejecutar un proceso de optimización con más variables (lo que se traduce en un espacio de búsqueda más amplio y un análisis y diseño estructural con más elementos). Como cada algoritmo (NSGA-II, MOPSO y AMOSA) se ejecutó 5 veces de forma secuencial y considerando un tiempo por ejecución de 500 s, el costo total del algoritmo AOTMOH fue de 7500 s (2.11 horas) para los problemas abordados en esta sección, lo cual es razonable con los recursos computacionales disponibles.

Para los resultados de desempeño cuantitativo de los algoritmos, en los problemas 1, 2 y 3 (Figura 70, Figura 74 y Figura 78 respectivamente) el algoritmo MOPSO obtuvo el mejor desempeño, con un mayor valor de HV, seguido por AMOSA y NSGA-II. En estos problemas, como se observa gráficamente en la

Tabla 22, Tabla 23 y Tabla 24, el NSGA-II presentó inconsistencias, ya que en cada ejecución los frentes de Pareto obtenidos son un poco lejanos entre sí (aunque se concentran en la misma región, la zona de peso mínimo). Los algoritmos MOPSO y AMOSA si mostraron resultados consistentes en todas las ejecuciones (es decir frentes de Pareto cercanos entre sí), con las soluciones concentradas principalmente en la región intermedia y en la región de mínima energía de deformación.

Para el problema 4 (Figura 82) el mejor HV fue ahora para el NSGA-II, seguido por MOPSO y AMOSA. Estos resultados corresponden a las gráficas reportadas en la Tabla 25 donde se ve que el algoritmo NSGA-II es consistente en todas sus ejecuciones y es el que mejor soluciones proporciona en la región del Pareto correspondiente a peso mínimo y en la región intermedia. Para este problema Los algoritmos MOPSO y AMOSA proporcionan las mejores soluciones principalmente en la región de mínima energía de deformación

Durante la evaluación de los algoritmos en los 4 problemas, se observa la importancia y la utilidad del frente de Pareto envolvente, ya que permite cubrir una región más amplia de soluciones a lo largo de los dos objetivos y un mayor valor de convergencia y diversidad (calculado con la métrica de Hipervolumen HV), combinando las mejores soluciones en la región de peso mínimo (proporcionadas por el NSGA-II) y en la región intermedia y de energía de deformación mínima (proporcionadas por el MOPSO y el AMOSA). Los valores cuantitativos de HV calculados para los frentes de Pareto envolvente de la parte (e) de la

Tabla 22, Tabla 23, Tabla 24 y Tabla 25 fueron:

- Estructura base de 94 elementos (
- Tabla 22 (e) izquierda) HV=0.8559 y estructura base de 56 elementos (
- Tabla 22 (e) derecha) HV=0.8595.
- Estructura base de 46 elementos (Tabla 23 (e) izquierda) HV=0.8415 y estructura base de 17 elementos (Tabla 23 (e) derecha) HV=0.7459.
- Estructura base de 63 elementos (Tabla 24 (e) izquierda) HV=0.8066 y estructura base de 44 elementos (Tabla 24 (e) derecha) HV=0.8748.
- Estructura base de 63 elementos (Tabla 25 (e) izquierda) HV=0.9673 y estructura base de 29 elementos (Tabla 25 (e) derecha) HV=0.9652.

Finalmente, se realiza una comparación de los mejores resultados de los 4 problemas abordados en esta Sección 6.3, con los mejores resultados de los 4 problemas de la Sección 6.2, teniendo en cuenta que la única diferencia entre ellos es el método de generación de la estructura base. Para los problemas de la Sección 6.2, la estructura base ya era conocida inicialmente (y fue generada usando el método NCN), en cambio para los problemas de la Sección 6.3, la estructura base no era conocida inicialmente y fue construida empleando el método TEM (lo cual constituye uno de los principales aportes de este trabajo) a partir del espacio de diseño inicial continuo. Los resultados comparados incluyen las soluciones extremas del frente de Pareto envolvente para los problemas de la Sección 6.2 y las soluciones extremas del frente de Pareto envolvente para los problemas de la Sección 6.3 (con la estructura base construida con el valor de porcentaje de trayectorias  $PT=1\%$ ).

Tabla 26.

*Comparación de las soluciones extremas de las Secciones 6.2 y 6.3*

# Problema	Solución extrema 1 (mínimo peso $W$ y máxima energía de deformación $E_s$ )				Solución extrema 2 (máximo peso $W$ y mínima energía de deformación $E_s$ )			
	Sección 6.2		Sección 6.3		Sección 6.2		Sección 6.3	
	$W$ (lb)	$E_s$ (kip-in)	$W$ (lb)	$E_s$ (kip-in)	$W$ (lb)	$E_s$ (kip-in)	$W$ (lb)	$E_s$ (kip-in)
A: [6.2.1 y 6.3.1]	4901.02	200	6223.69	173	14687.64	82	33792.64	38.9
B: [6.2.2 y 6.3.2]	2597.79	100	2880.43	100	10666.13	41.5	10819.57	38.4
C: [6.2.3 y 6.3.3]	196.19	38.1	321.37	50.5	1618.86	7.73	1686.38	14.7
D: [6.2.4 y 6.3.4]	69.98	17.6	88.87	14	8348.13	0.28	5030.85	0.39

A Partir de la Tabla 26 se puede observar que las soluciones extremas presentan una mayor variación entre sí para los problemas A: [6.2.1 y 6.3.1] y C: [6.2.3 y 6.3.3], lo cual puede ser causado por el mayor número de elementos presentes en la estructura base usando el espacio de diseño continuo. Para los problemas B: [6.2.2 y 6.3.2] y [6.2.4 y 6.3.4], las soluciones extremas si son muy cercanas entre sí, ya que en este caso el número de elementos presentes en la estructura base es muy similar para ambos problemas.

Los resultados de la Tabla 26 indican la utilidad y el potencial uso que tiene el método TEM como estrategia de generación de la estructura base en el proceso de optimización topológica

multiobjetivo, ya que es posible obtener estructuras óptimas discretas a partir de una estructura continua, que son similares (desde el punto de vista del valor de las funciones objetivo en el frente de Pareto) a las estructuras generadas con el método clásico NCN. Cuando se genera una estructura base con pocos elementos pueden obtenerse los mejores resultados (ej. problemas B: [6.2.2 y 6.3.2] y [6.2.4 y 6.3.4]), por lo cual es importante que durante la construcción de la estructura base se pueda seleccionar el menor número de trayectorias de esfuerzo posibles. Lo anterior, teniendo en cuenta que este es un proceso que depende del usuario, ya que es el encargado de definir el parámetro de selección de las trayectorias más relevantes (porcentaje de trayectorias PT).

#### **6.4. Problema de diseño óptimo de una torre de transmisión de energía eléctrica plana**

En esta sección se presentan los resultados de desempeño del algoritmo AOTMOH en el diseño de una torre de transmisión de energía eléctrica plana empleando el método TEM (Sección 4.3.2) para generar la estructura base. Cada uno de los algoritmos que componen el AOTMOH (es decir el NSGA-II, MOPSO y AMOSA) se ejecutaron 5 veces usando los mismos parámetros usados en los problemas de la Sección 6.3. El problema abordado fue estudiado por Geem (2009) usando el algoritmo HS (Harmony Search) y por Kaveh & Ilchi Ghazaan (2018) usando el algoritmo ECBO (Enhanced Colliding Bodies Optimization). En ambos estudios se aplicó un proceso de optimización de tamaño (ver Sección 3.2.4.1) y se consideró una sola función objetivo: minimizar el peso de la estructura. Los detalles del problema se describen a continuación:

##### **6.4.1. Descripción del problema de diseño de la torre plana**

**6.4.1.1. Geometría, condiciones de carga, apoyos y variables.** La torre tiene 47 elementos y 22 nodos (ver Figura 83). La estructura es simétrica alrededor del eje  $Y$ . Todos los elementos están

hechos de acero con densidad  $\rho = 0.3 \text{ lb/in}^3$  y módulo de elasticidad  $E = 30000 \text{ ksi}$ . La torre es diseñada para 3 casos de carga: (1) **6 kips** en la dirección positiva de  $X$  y **14 kips** en la dirección negativa de  $Y$ , para los nodos 16 y 22. (2) **6 kips** en la dirección positiva de  $X$  y **14 kips** en la dirección negativa de  $Y$ , para el nodo 16. (3) **6 kips** en la dirección positiva de  $X$  y **14 kips** en la dirección negativa de  $Y$ , para el nodo 22. El primer caso representa las cargas impuestas por dos líneas de transmisión atadas a la torre a un ángulo dado y los casos 2 y 3 representan condiciones que ocurren cuando una de las 2 líneas se rompe. Las áreas de los elementos fueron categorizadas en 27 grupos de variables, de acuerdo a la **Referencia.:**

Tabla 27. Grupos de variables

#	Áreas
1	$A_1, A_3$
2	$A_2, A_4$
3	$A_5, A_6$
4	$A_7$
5	$A_8, A_9$
6	$A_{10}$
7	$A_{11}, A_{12}$
8	$A_{13}, A_{14}$
9	$A_{15}, A_{16}$
10	$A_{17}, A_{18}$
11	$A_{19}, A_{20}$
12	$A_{21}, A_{22}$
13	$A_{23}, A_{24}$
14	$A_{25}, A_{26}$
15	$A_{27}$
16	$A_{28}$
17	$A_{29}, A_{30}$
18	$A_{31}, A_{32}$
19	$A_{33}$
20	$A_{34}, A_{35}$
21	$A_{36}, A_{37}$

---

22	$A_{38}$
23	$A_{39}, A_{40}$
24	$A_{41}, A_{42}$
25	$A_{43}$
26	$A_{44}, A_{45}$
27	$A_{46}, A_{47}$

---

Nota: Adaptado de  
Geem (2009)

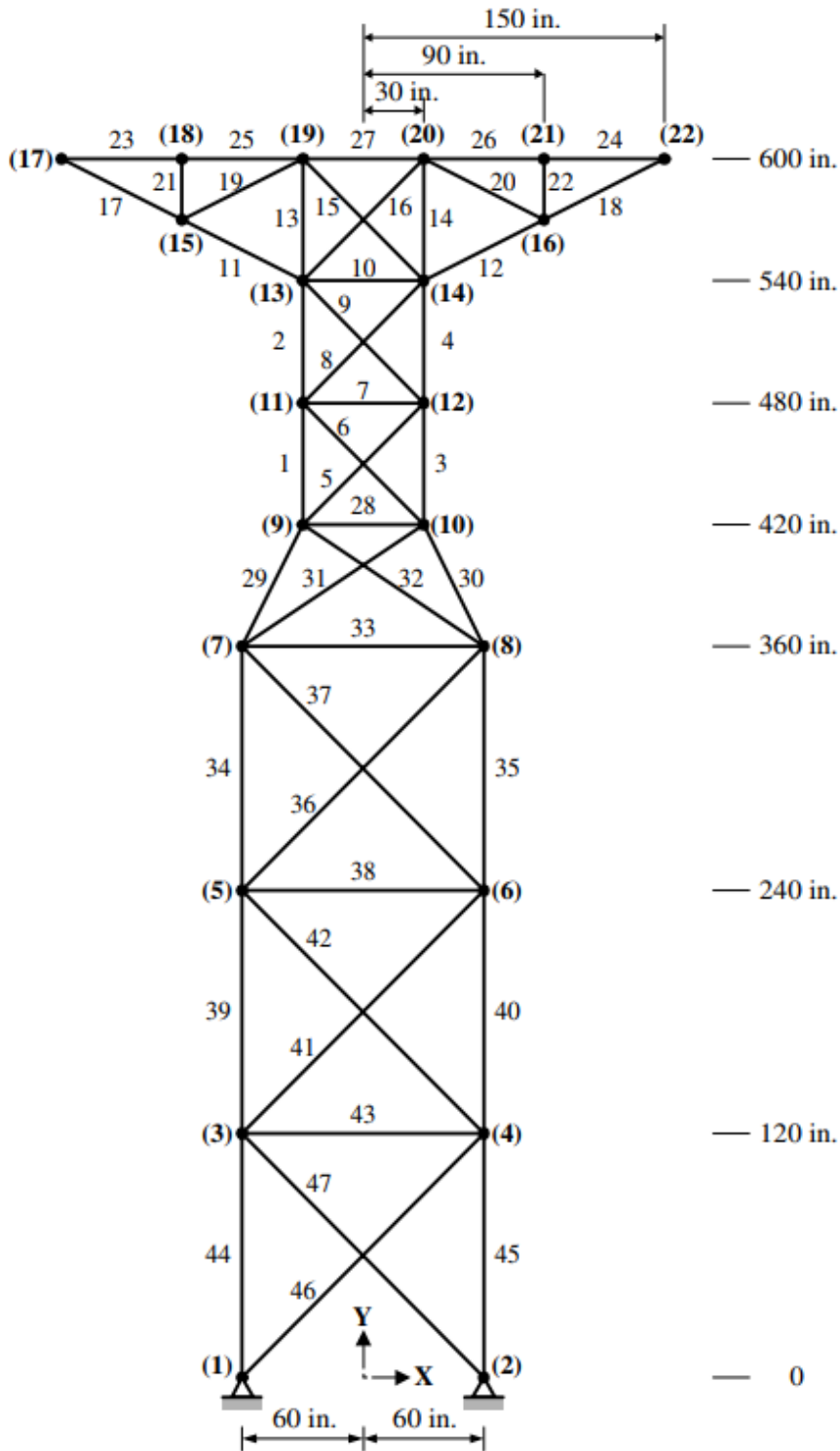


Figura 83. Torre de transmisión plana de 47 barras  
Adaptado: Geem (2009)

6.4.1.2. Restricciones

6.4.1.2.1. *Restricciones tipo A.* Son las restricciones originales del problema (Geem, 2009; Ali Kaveh & Ilchi Ghazaan, 2018). La estructura está sujeta a restricciones de esfuerzo axial y esfuerzo de pandeo. La restricción de esfuerzo a tensión es de 20 ksi ( $\sigma^+ \leq 20$ ) y para esfuerzo a compresión es 15 ksi ( $|\sigma^-| \leq 15$ ). El esfuerzo de pandeo de Euler para los elementos a compresión está limitado a:

$$|\sigma_{cr}^-| \leq \frac{KEA_i}{L_i^2} \quad (i = 1, 2, 3, \dots, 47)$$

El valor de  $E$  corresponde al módulo de elasticidad,  $L_i$  es la longitud del elemento y  $A_i$  el área de la sección transversal.  $K$  es la constante de pandeo (para este estudio su valor es  $K = 3.96$ ). Durante el proceso de optimización, el área de las secciones transversales de los elementos será seleccionada de un conjunto de 64 valores discretos  $D = \{0.111, 0.141, 0.25, 0.307, 0.391, 0.442, 0.563, 0.602, 0.766, 0.785, 0.994, 1, 1.228, 1.266, 1.457, 1.563, 1.62, 1.8, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.8, 4.97, 5.12, 5.74, 7.22, 7.97, 8.53, 9.3, 10.85, 11.5, 13.5, 13.9, 14.2, 15.5, 16, 16.9, 18.8, 19.9, 22, 22.9, 24.5, 26.5, 28, 30, 33.5\}$  in<sup>2</sup>.

6.4.1.2.2. *Restricciones tipo B.* Son las restricciones propuestas en este trabajo. La estructura está sujeta a restricciones de esfuerzo axial y esfuerzo de pandeo definidas en la norma ANSI/AISC 360-16 (AISC, 2016) y en el Reglamento Colombiano de Construcción Sismo Resistente NSR-10 Título F (AIS, 2010) (ver Apéndice C.2.). Se asume acero estructural ASTM A36 ( $F_y = 36$  ksi y  $F_u = 58$  ksi). Durante el proceso de optimización, el área de las secciones transversales de los elementos será seleccionada del conjunto de 60 valores discretos de perfiles estructurales tipo L definidos en la Tabla C.1 del Apéndice C.2.

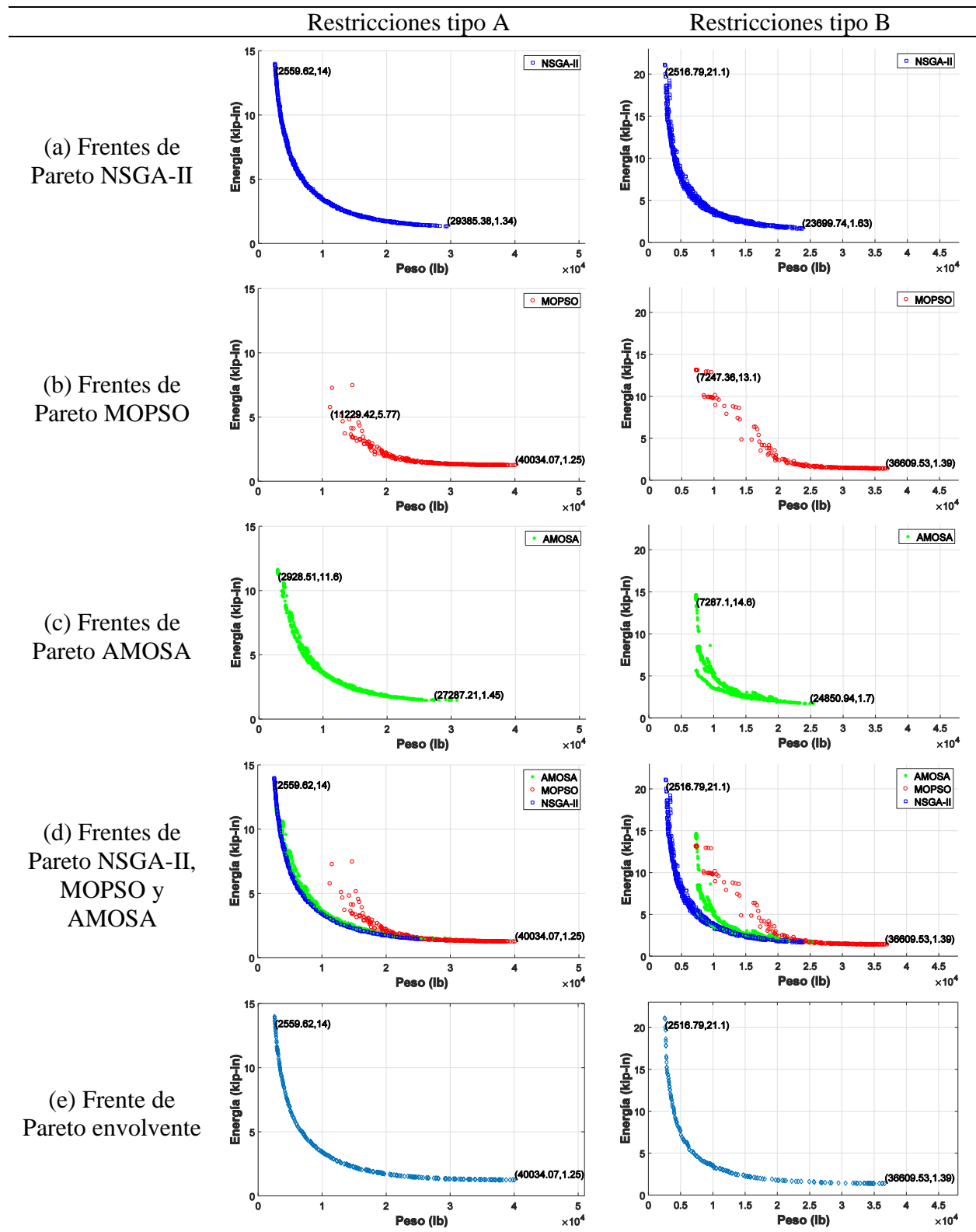
**6.4.2. Optimización de tamaño multiobjetivo de la torre plana.** Inicialmente se usó el algoritmo AOTMOH para realizar el proceso de optimización de tamaño de la estructura mostrada en la Figura 83. Lo anterior, con el fin de verificar si los resultados son consistentes, en comparación con las referencias consultadas. El problema se resolvió por separado para los dos tipos de restricciones A y B (las originales y las propuestas en este trabajo), usando la geometría, condiciones de carga, apoyos y variables definidas en la Sección 6.4.1.1.

Teniendo en cuenta que el problema de optimización en las referencias (Geem, 2009; Ali Kaveh & Ilchi Ghazaan, 2018) fue solucionado con una sola función objetivo (el peso estructural) y considerando las restricciones tipo A; y el algoritmo AOTMOH propuesto en este trabajo es multiobjetivo, para comparar los resultados obtenidos se considera la solución extrema del frente de Pareto envolvente para las restricciones tipo A, que tiene peso mínimo y energía de deformación máxima.

**6.4.2.1. Resultados de comparación optimización de tamaño.** A continuación, en la Tabla 28 se muestran los resultados gráficos de aplicación de los algoritmos NSGA-II, MOPSO y AMOSA en el problema de optimización de tamaño de la torre plana de 47 elementos, comparando los resultados obtenidos con cada tipo de restricción. De igual manera en la Figura 84 y Figura 85 se muestran los diagramas de cajas para el tiempo computacional y la métrica de Hipervolumen (HV) para los algoritmos NSGA-II, MOPSO y AMOSA.

Tabla 28.

Desempeño del algoritmo AOTMOH en la optimización de tamaño de la torre plana de 47 elementos



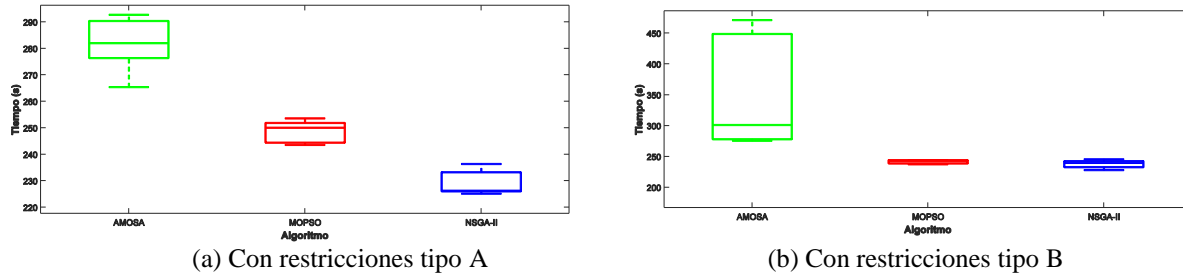


Figura 84. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización de forma de la torre plana de 47 elementos

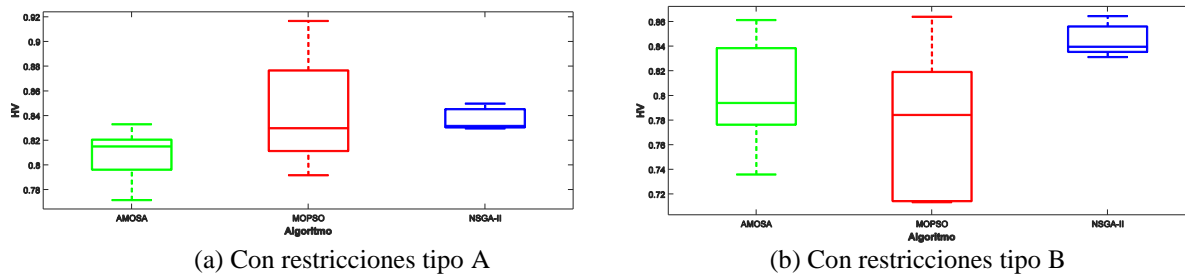


Figura 85. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización de forma de la torre plana de 47 elementos

En la Tabla 28, de forma gráfica puede observarse que el algoritmo que tuvo el mejor desempeño en los dos tipos de restricciones fue el NSGA-II, seguido por el AMOSA y MOPSO. Esto se ve reflejado de forma cuantitativa en la Figura 85, donde el diagrama de cajas del NSGA-II presenta un menor rango de variación y se acerca más al valor de 1, en comparación con AMOSA y MOPSO. Los valores de HV calculados para los frentes de Pareto envolvente de la Tabla 28(e) fueron  $HV=0.8924$  (para las restricciones tipo A) y  $HV=0.9168$  (para las restricciones tipo B).

Para el tiempo computacional (ver Figura 84), se sigue presentando la tendencia de los problemas abordados en secciones anteriores, ya que los algoritmos más rápidos siguen siendo el NSGA-II y MOPSO con tiempos entre 200 y 250 s, en comparación con el AMOSA que demora entre 250 y 450 s para resolver el problema. Las cinco ejecuciones secuenciales de los tres algoritmos metaheurísticos (considerando un tiempo por ejecución de 450 s), hicieron que el costo computacional de algoritmo AOTMOH fuera de 6750 s (1.875 horas) en promedio.

A continuación, en la Tabla 29 se presenta la comparación con las referencias consultadas (Geem, 2009; Ali Kaveh & Ilchi Ghazaan, 2018) de las 2 soluciones extremas con peso mínimo de los frentes de Pareto envolvente de la Tabla 28(e), (usando las restricciones tipo A y B).

Tabla 29.

Comparación de desempeño para el problema de optimización de tamaño de la torre plana de 47 elementos

VARIABLES (Áreas en in <sup>2</sup> )	HS (Geem, 2009), con restricciones tipo A	ECBO (Ali Kaveh & Ilchi Ghazaan, 2018), con restricciones tipo A	AOTMOH (En este trabajo), con restricciones tipo A	AOTMOH (En este trabajo), con restricciones tipo B
1 ( $A_1 = A_3$ )	3.84	3.84	4.22	2.26
2 ( $A_2 = A_4$ )	3.38	3.38	3.38	3.25
3 ( $A_5 = A_6$ )	0.766	0.766	0.785	0.944
4 ( $A_7$ )	0.141	0.111	0.141	1.7
5 ( $A_8 = A_9$ )	0.785	0.785	1.266	2.26
6 ( $A_{10}$ )	1.99	1.99	1.8	2.5
7 ( $A_{11} = A_{12}$ )	2.13	2.13	2.62	2.5
8 ( $A_{13} = A_{14}$ )	1.228	1.228	1.228	1.44
9 ( $A_{15} = A_{16}$ )	1.563	1.563	1.8	1.46
10 ( $A_{17} = A_{18}$ )	2.13	2.13	2.13	2.26
11 ( $A_{19} = A_{20}$ )	0.111	0.111	0.602	1.16
12 ( $A_{21} = A_{22}$ )	0.111	0.141	0.141	1.16
13 ( $A_{23} = A_{24}$ )	1.8	1.8	1.8	1.44
14 ( $A_{25} = A_{26}$ )	1.8	1.8	2.13	3.75
15 ( $A_{27}$ )	1.457	1.457	1.62	1.73
16 ( $A_{28}$ )	0.442	0.442	0.391	23
17 ( $A_{29} = A_{30}$ )	3.63	3.63	4.59	2.26
18 ( $A_{31} = A_{32}$ )	1.457	1.457	1.563	1.19
19 ( $A_{33}$ )	0.391	0.307	0.602	0.491
20 ( $A_{34} = A_{35}$ )	3.09	3.09	3.09	1.46
21 ( $A_{36} = A_{37}$ )	1.457	1.266	1.457	2.11
22 ( $A_{38}$ )	0.196	0.307	0.111	0.491
23 ( $A_{39} = A_{40}$ )	3.84	3.84	3.87	2.5
24 ( $A_{41} = A_{42}$ )	1.563	1.563	1.563	1.44
25 ( $A_{43}$ )	0.196	0.111	0.141	0.722
26 ( $A_{44} = A_{45}$ )	4.59	4.59	4.59	2.1
27 ( $A_{46} = A_{47}$ )	1.457	1.457	1.563	2.11
<b>Peso (lb)</b>	<b>2396.8</b>	<b>2375.35</b>	<b>2559.62</b>	<b>2516.79</b>

Los resultados de la Tabla 29, muestran que los resultados obtenidos con el algoritmo AOTMOH para la función objetivo peso y para las variables de diseño (para las restricciones de

tipo A), son cercanos a las referencias consultadas (7% de diferencia en promedio), lo cual indica que son consistentes y no existe una variación muy amplia. Por otro lado, con el algoritmo AOTMOH, el valor del peso mínimo para las restricciones tipo B es similar (1% de diferencia) al obtenido con las restricciones tipo A, aun existiendo diferencias con las variables de diseño, lo cual muestra que existen diferentes posibilidades de combinación de las secciones de los elementos para obtener pesos similares, que pueden ser evaluadas por el algoritmo AOTMOH.

**6.4.3. Optimización topológica multiobjetivo de la torre plana con el método TEM.** En esta sección el algoritmo AOTMOH se aplica en el problema de optimización topológica de diseño de la torre plana de la Figura 83, asumiendo el espacio de diseño continuo, es decir se aplica el método TEM de la Sección 4.3.2 para generar la estructura base.

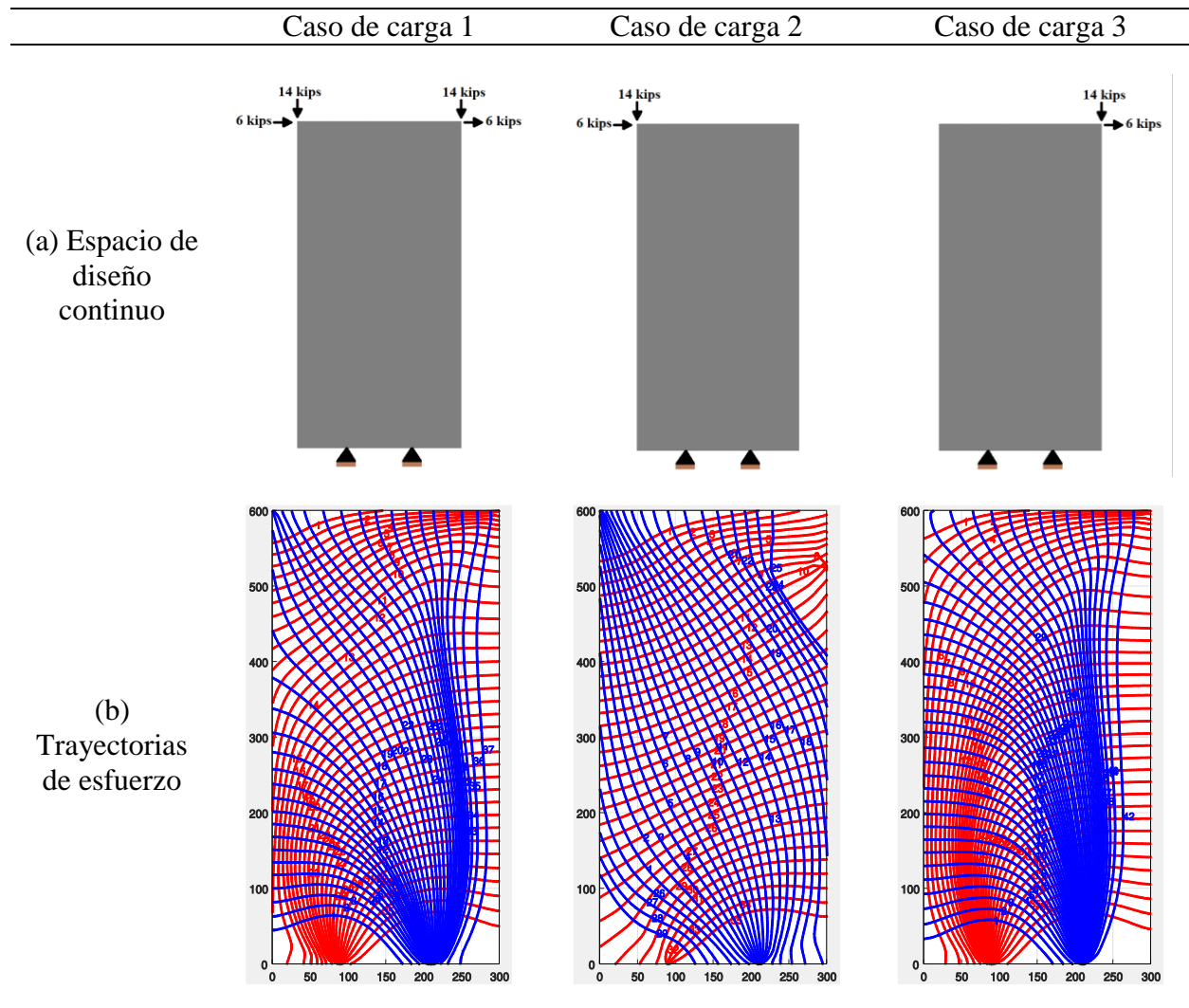
Para cada uno de los casos de carga, se calcularon las trayectorias de esfuerzo ( $\sigma_1$  en color rojo, y  $\sigma_2$  en color azul) y se construyeron las tres estructuras base posibles, usando un valor arbitrario de porcentaje de trayectorias  $PT=10\%$  (valor seleccionado de acuerdo a pruebas preliminares). En la Tabla 30 se muestran los resultados de este proceso, teniendo en cuenta la siguiente convención.

- En (a) se muestra el espacio de diseño inicial plano continuo para cada caso de carga
- En (b) se muestran las trayectorias de esfuerzo calculadas para cada caso de carga.
- En (c) se muestran las trayectorias de esfuerzo de mayor costo utilizadas para construir la estructura base de cada caso de carga, calculadas a partir de (b) usando el valor  $PT=10\%$ .
- En (d) se muestra la estructura de macroelementos de cada caso de carga, construida a partir de las trayectorias de esfuerzo representativas de (c).
- En (e) se muestra la estructura base definitiva generada para cada caso de carga, a partir de la estructura de macroelementos de la parte (d).

- En (f) se muestra el número de elementos que tiene la estructura base generada para cada caso de carga.

Tabla 30.

*Construcción de la estructura base para los 3 casos de carga de la torre plana con el método TEM*



(c)  
Trayectorias  
de mayor  
costo  
(PT=10%)

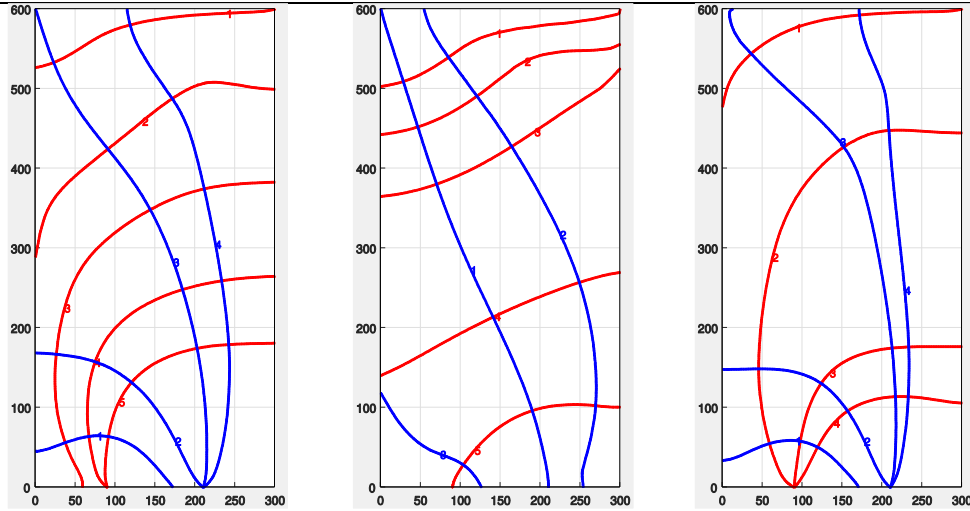
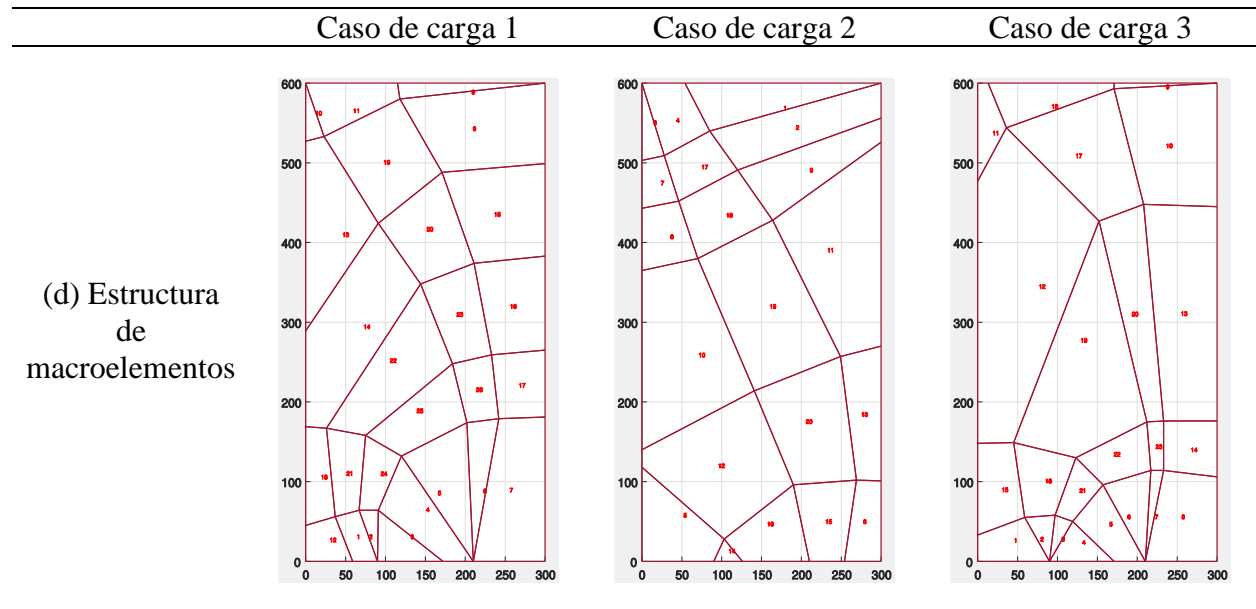
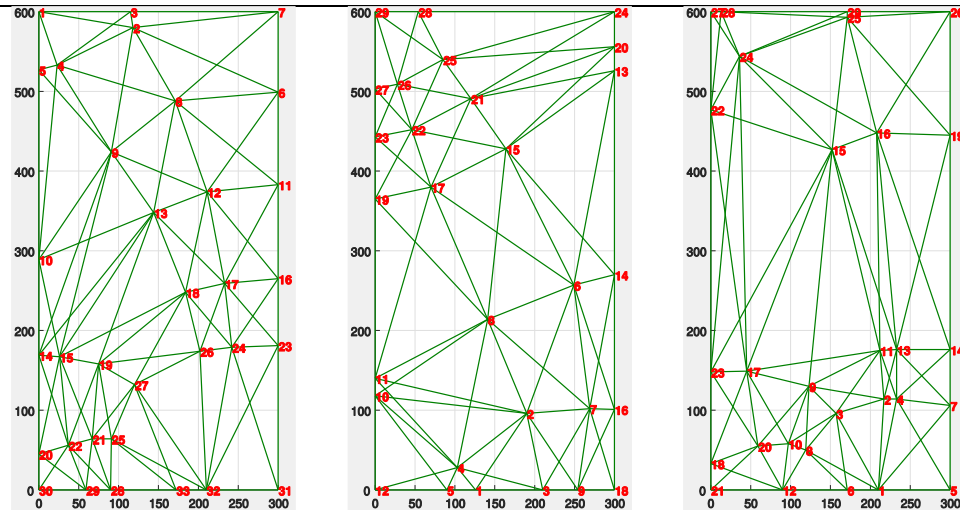


Tabla 30.

*Construcción de la estructura base para los 3 casos de carga de la torre plana con el método TEM (continuación)*



(e) Estructura base



(f) Número de elementos de la estructura base generada

101

85

88

Ahora, cada estructura base de la Tabla 30(e), se optimizó por separado para los dos tipos de restricciones A y B. Las condiciones de carga fueron las mismas establecidas en la Sección 6.4.1.1 (es decir, aunque cada estructura base se generó a partir de uno de los 3 casos de carga, cuando se va a optimizar se considera que está sujeta a los tres casos de carga de manera simultánea).

Para cada estructura base los resultados gráficos reportados incluyen:

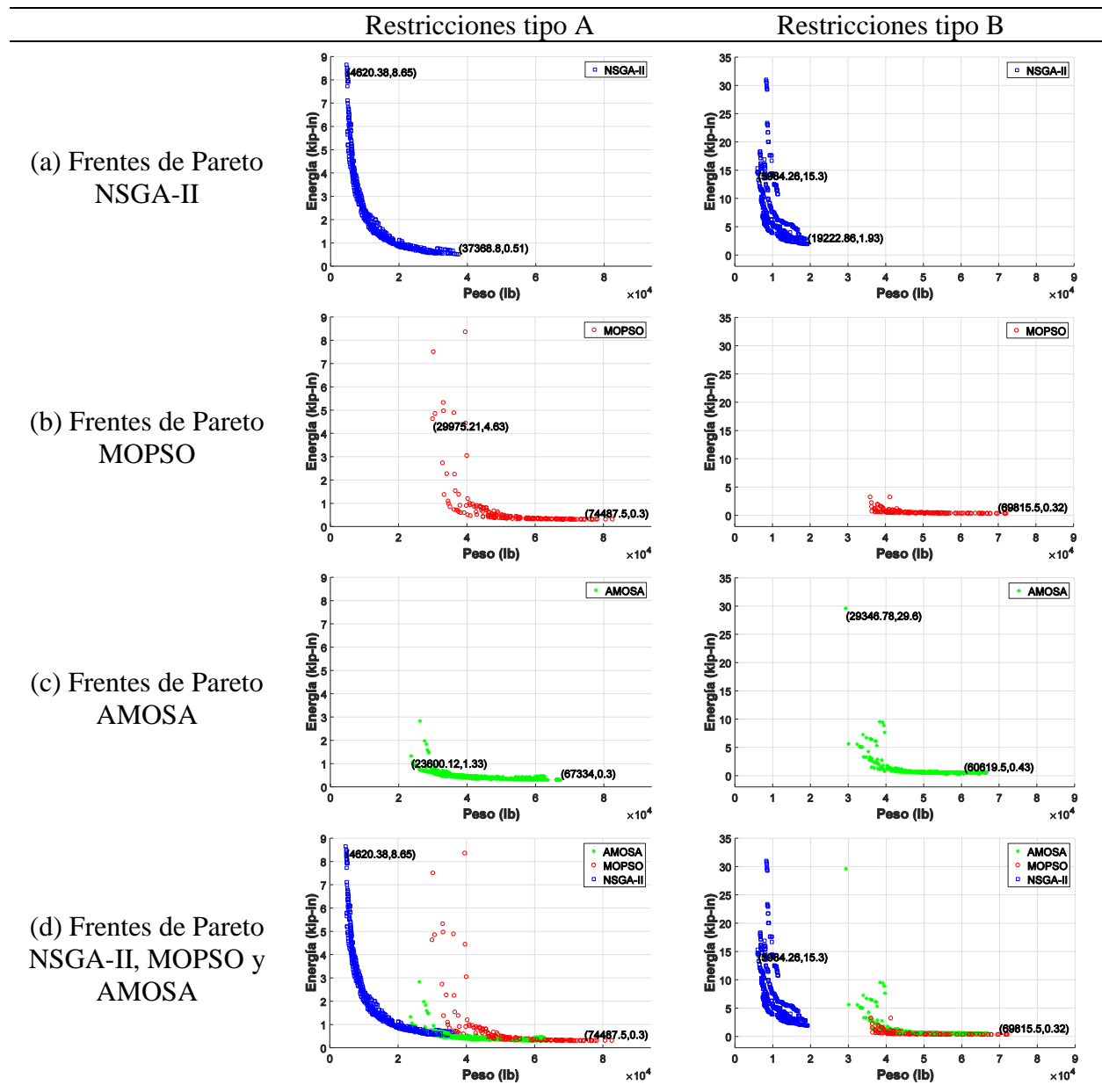
- Frentes de Pareto con el desempeño de los algoritmos NSGA-II, MOPSO y AMOSA y construcción del frente de Pareto envolvente para los 2 tipos de restricciones A y B: Tabla 31, Tabla 32 y Tabla 33
- Representación de las soluciones óptimas decodificadas del frente de Pareto envolvente para los 2 tipos de restricciones A y B: Figura 86, Figura 89 y Figura 92
- Diagramas de cajas (boxplots) con el costo computacional para cada algoritmo: Figura 87, Figura 90, Figura 93

- Diagramas de cajas (boxplots) con la métrica de Hipervolumen (HV) para cada algoritmo:,  
 Figura 88, Figura 91 y Figura 94

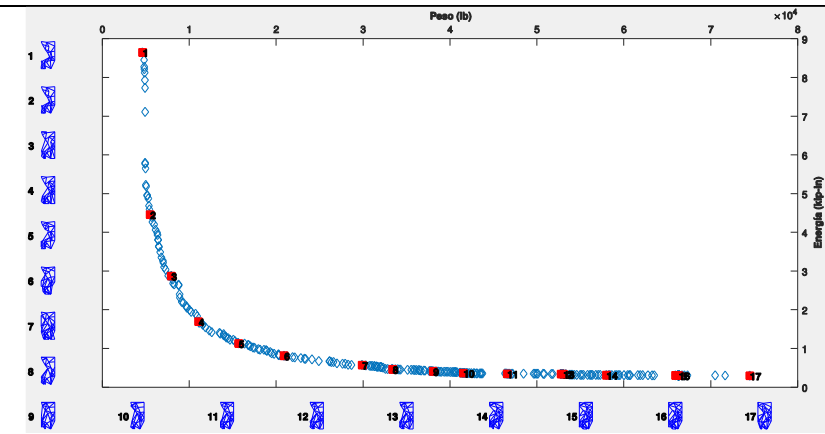
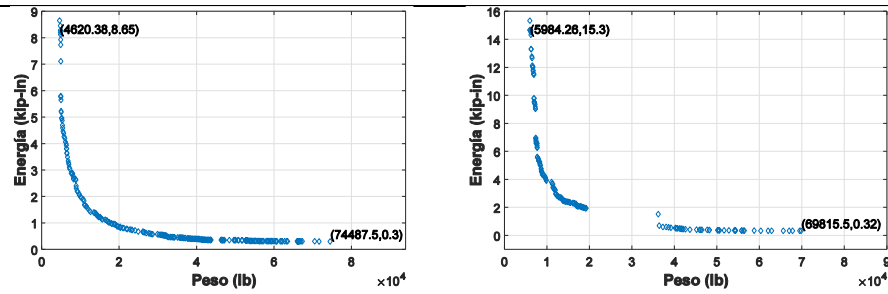
6.4.3.1. Estructura base de 101 elementos (derivada del caso de carga 1)

Tabla 31.

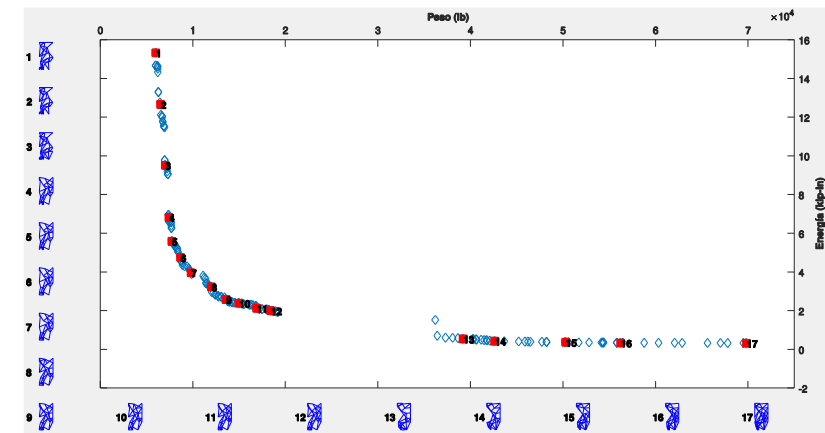
Desempeño del algoritmo AOTMOH en la optimización de la estructura base 101 elementos



(e) Frente de Pareto envolvente

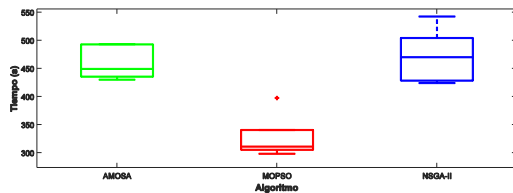


(a) Soluciones de la estructura base de 101 elementos con restricciones tipo A

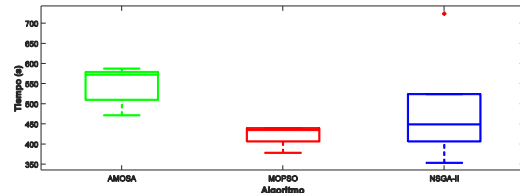


(b) Soluciones de la estructura base de 101 elementos con restricciones tipo B

Figura 86. Soluciones decodificadas del frente de Pareto envolvente de la estructura base de 101 elementos

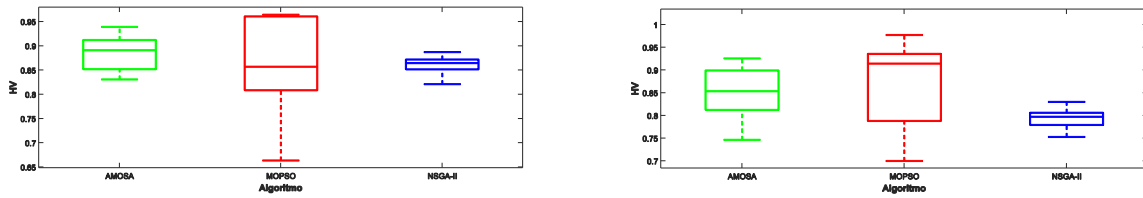


(a) Con restricciones tipo A



(b) Con restricciones tipo B

Figura 87. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 101 elementos



(a) Con restricciones tipo A

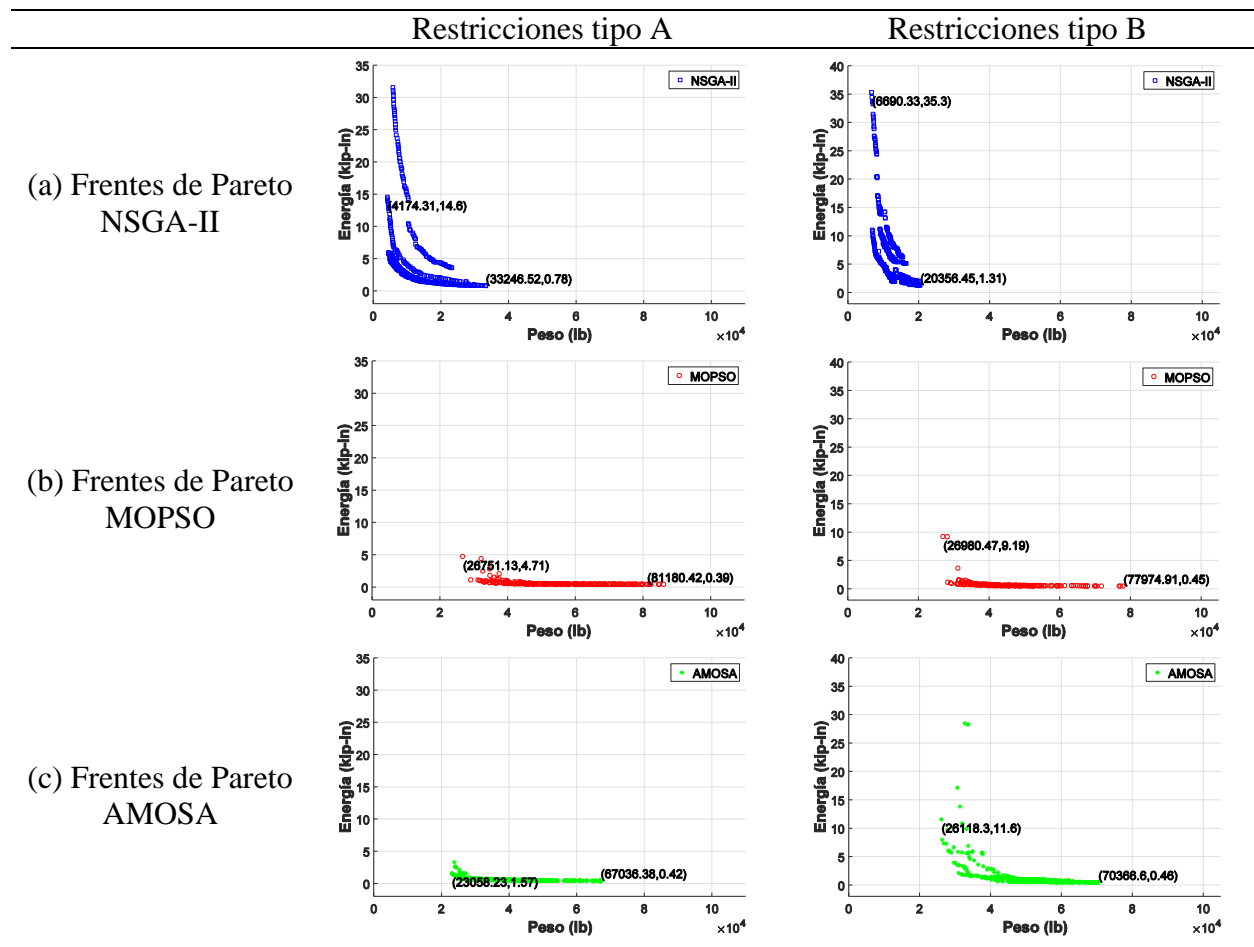
(b) Con restricciones tipo B

Figura 88. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 101 elementos

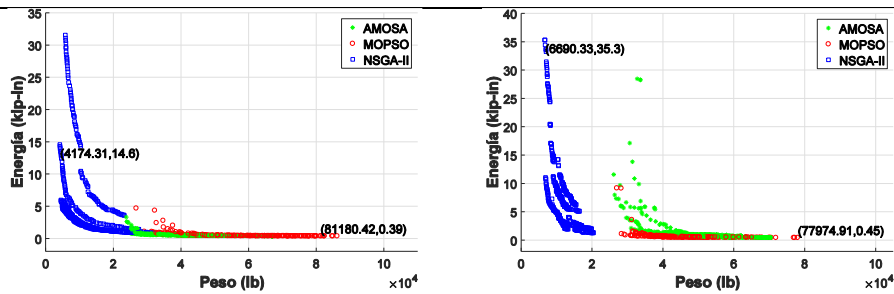
6.4.3.2. Estructura base de 85 elementos (derivada del caso de carga 2)

Tabla 32.

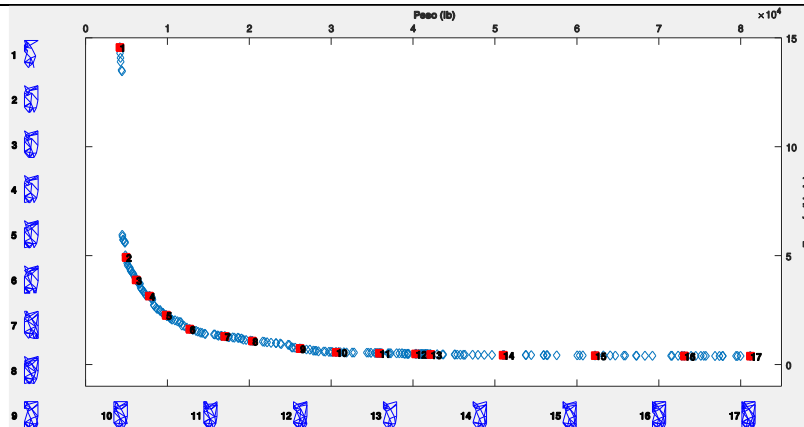
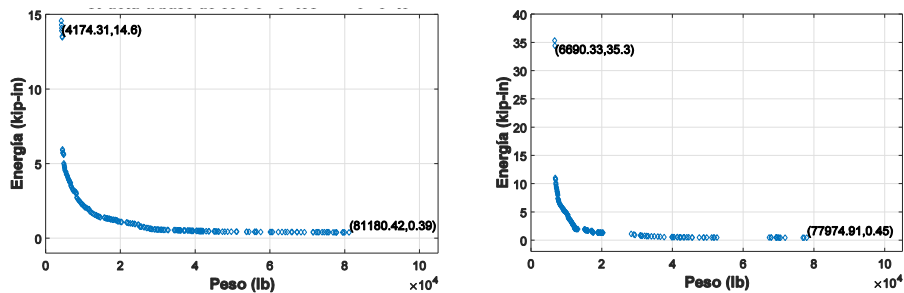
Desempeño del algoritmo AOTMOH en la optimización de la estructura base 85 elementos



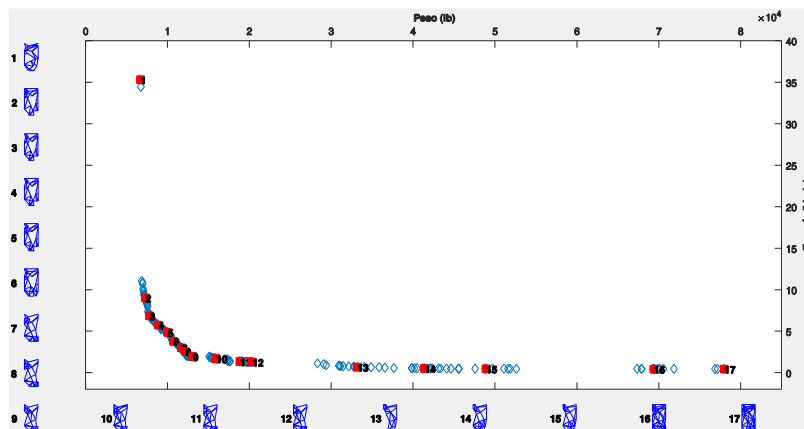
(d) Frentes de Pareto  
NSGA-II, MOPSO y  
AMOSa



(e) Frente de Pareto  
envolvente



(a) Soluciones de la estructura base de 85 elementos con restricciones tipo A



(b) Soluciones de la estructura base de 85 elementos con restricciones tipo B

Figura 89. Soluciones decodificadas del frente de Pareto envolvente de la estructura base de 85 elementos

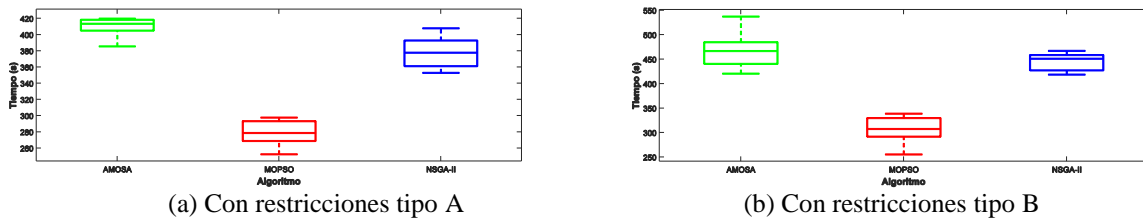


Figura 90. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 85 elementos

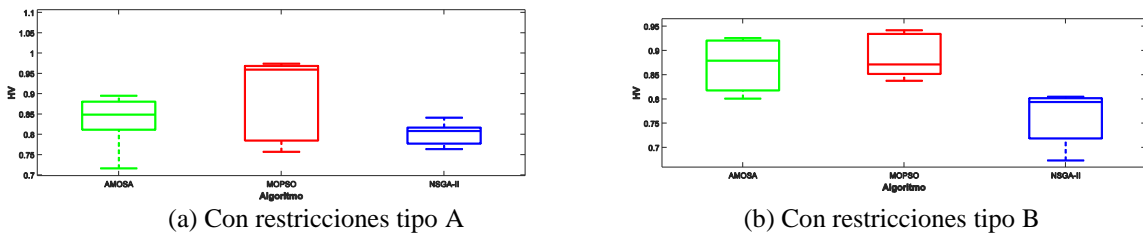
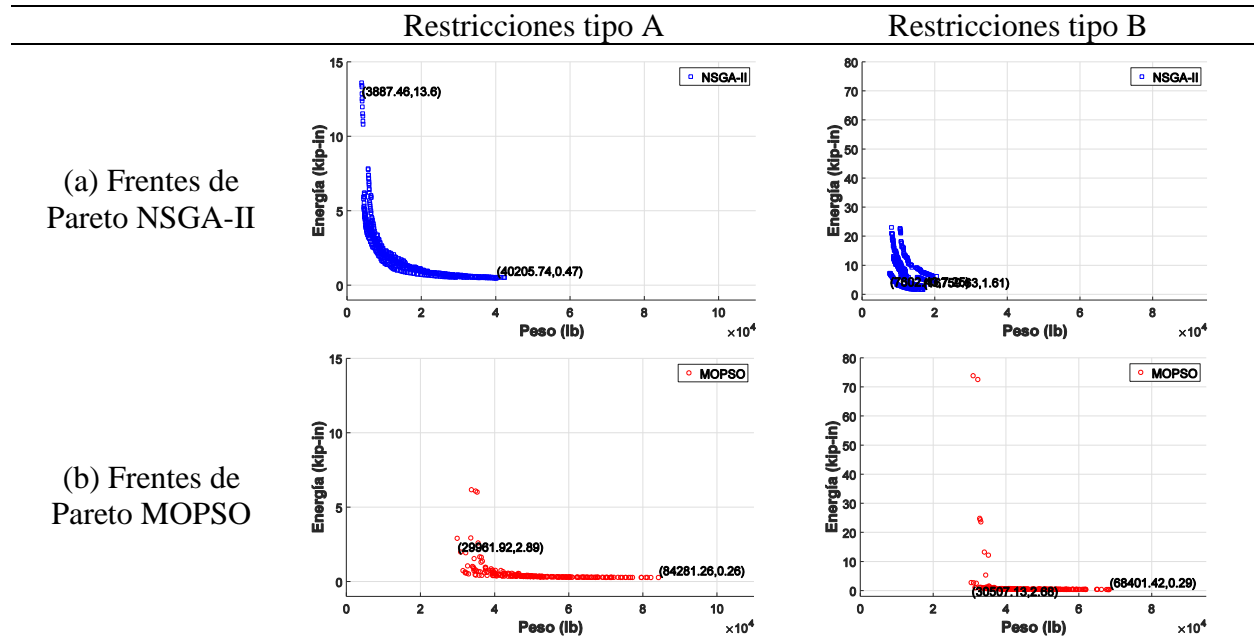


Figura 91. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 85 elementos

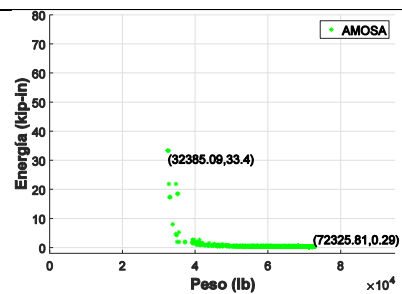
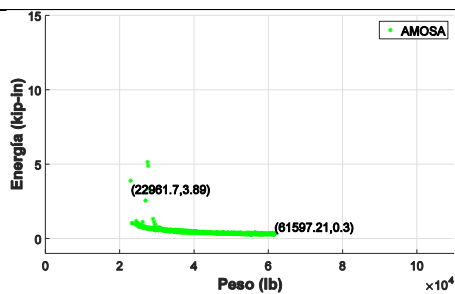
6.4.3.3. Estructura base de 88 elementos (derivada del caso de carga 3)

Tabla 33.

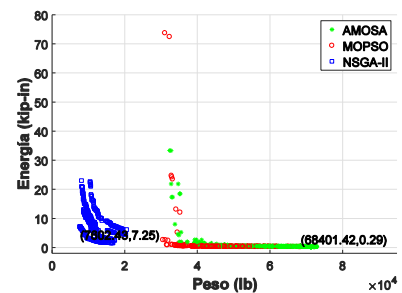
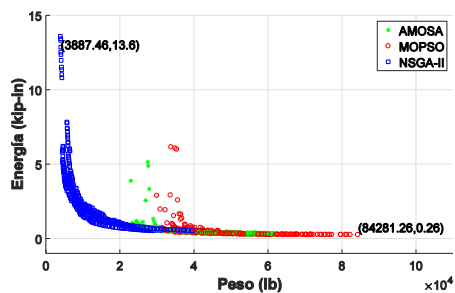
Desempeño del algoritmo AOTMOH en la optimización de la estructura base 88 elementos



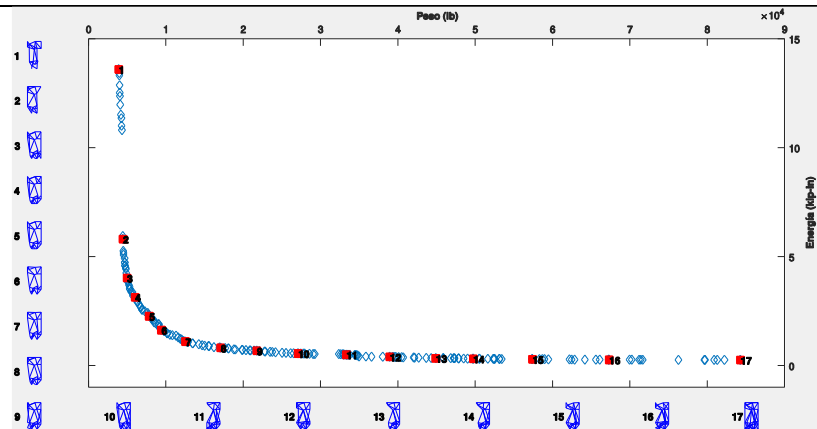
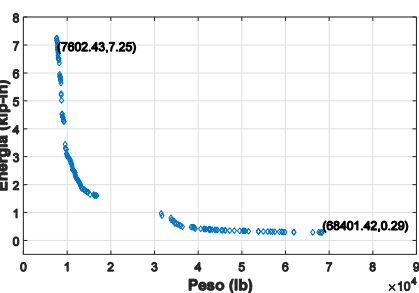
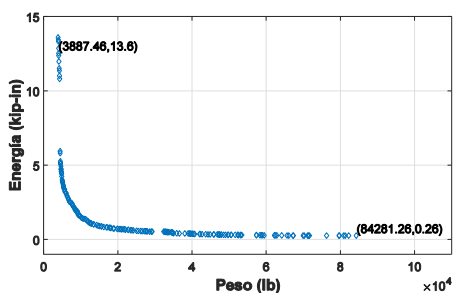
(c) Frentes de Pareto AMOSA



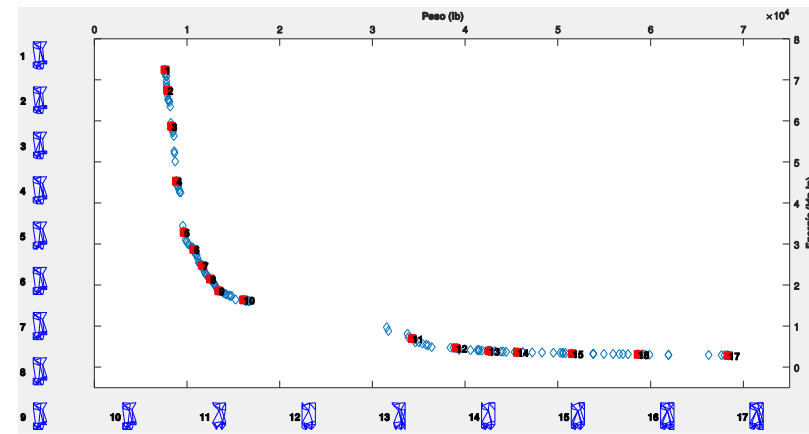
(d) Frentes de Pareto NSGA-II, MOPSO y AMOSA



(e) Frente de Pareto envolvente

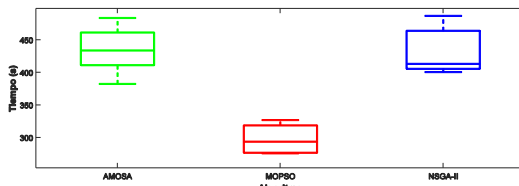


(a) Soluciones de la estructura base de 88 elementos con restricciones tipo A

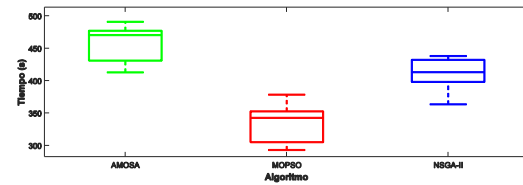


(b) Soluciones de la estructura base de 88 elementos con restricciones tipo B

Figura 92. Soluciones decodificadas del frente de Pareto envolvente de la estructura base de 88 elementos

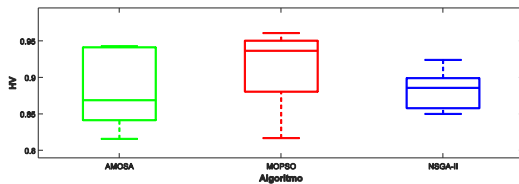


(a) Con restricciones tipo A

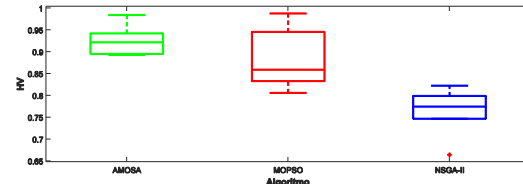


(b) Con restricciones tipo B

Figura 93. Diagramas de cajas para tiempo computacional de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 88 elementos



(a) Con restricciones tipo A



(b) Con restricciones tipo B

Figura 94. Diagramas de cajas para HV de los algoritmos AMOSA, MOPSO y NSGA-II en la optimización topológica multiobjetivo de la estructura base de 88 elementos

**6.4.4. Análisis de resultados del proceso de optimización topológica multiobjetivo de la torre plana con el método TEM.** En la Sección 6.3 se aplicó el algoritmo AOTMOH en el proceso de diseño de una torre de transmisión de energía eléctrica plana, considerando un espacio de diseño continuo y empleando el método TEM para generar la estructura base. Para la Tabla 31, Tabla 32 y Tabla 33, correspondientes a los resultados de desempeño de los algoritmos que hacen parte del

AOTMOH (NSGA-II, MOPSO y AMOSA) en las 3 estructuras base analizadas (101, 85 y 88 elementos), puede observarse que para los dos problemas resueltos en cada estructura (con restricciones tipo A y restricciones tipo B), el algoritmo NSGA-II es el que presenta un mejor desempeño en la región del frente de Pareto correspondiente al peso mínimo, ya que los algoritmos MOPSO y AMOSA no pudieron obtener resultados en esta región. Para la región intermedia y región de mínima energía de deformación, los algoritmos que presentaron mejores resultados fueron el MOPSO y AMOSA, ya que en estas regiones fue el algoritmo NSGA-II quien no logró cubrirlas. De esta manera, al combinar los mejores resultados de los 3 algoritmos fue posible obtener un frente de Pareto envolvente que abarcara la mayor región posible de soluciones óptimas a lo largo de los dos objetivos.

Cuando se analiza el tiempo computacional empleado por los algoritmos para resolver una ejecución del problema de optimización topológica multiobjetivo de las 3 estructuras base (101, 85 y 88 elementos), de acuerdo a la Figura 87, Figura 90 y Figura 93, el más rápido sigue siendo el MOPSO con un tiempo que oscila en el rango de 270-450 s, seguido por el NSGA-II con un rango de 360-550 s y el AMOSA con un tiempo entre 400 y 600 s. Con la ejecución secuencial de los tres algoritmos metaheurísticos, considerando que en cada ejecución el tiempo empleado es de 600 s y se ejecutan 5 veces el costo total del algoritmo AOTMOH para generar el frente de Pareto envolvente es de 9000 s (2.5 h). Esto demuestra que pueden obtenerse respuestas óptimas en un tiempo razonable para estructuras de celosía que pueden tener una cantidad considerable de elementos.

Al analizar la convergencia y diversidad calculando la métrica HV, para los frentes de Pareto envolvente de las estructuras base generadas a partir de los 3 casos de carga para las restricciones tipo A y B (ver parte (e) de la Tabla 31, Tabla 32 y Tabla 33), se obtuvieron los siguientes valores:

- Estructura base de 101 elementos: Restricciones tipo A (Tabla 31 (e) izquierda) HV=0.9416 y restricciones tipo B (Tabla 31 (e) derecha) HV=0.9168.
- Estructura base de 85 elementos: Restricciones tipo A (Tabla 32 (e) izquierda) HV=0.9631 y restricciones tipo B (Tabla 32 (e) derecha) HV=0.9785.
- Estructura base de 88 elementos: Restricciones tipo A (Tabla 33 (e) izquierda) HV=0.9677 y restricciones tipo B (Tabla 33 (e) derecha) HV=0.8877.

El cálculo de HV usando el frente de Pareto envolvente, permite aprovechar los mejores valores obtenidos por cada algoritmo metaheurístico (NSGA-II, MOPSO y AMOSA) en diferentes regiones del espacio objetivo, mejorando los resultados individuales de HV obtenidos en cada ejecución (ver Figura 88, Figura 91 y Figura 94).

Con respecto a los valores extremos (solución de mínimo peso y solución de mínima energía de deformación) de los dos frentes de Pareto envolventes generados para cada estructura base en la parte (f) de la Tabla 31, Tabla 32 y Tabla 33 respectivamente, puede observarse que con el uso de las restricciones tipo A, se obtuvieron estructuras más livianas en la región de peso mínimo y estructuras con menos energía en la zona de energía de deformación mínima, en comparación con las restricciones tipo B. En este caso, el tipo de restricciones consideradas influye de manera significativa en el valor de la respuesta óptima.

Finalmente, para las soluciones óptimas decodificadas del frente de Pareto envolvente en la Figura 86, Figura 89 y Figura 92, puede observarse que estas torres presentan geometrías con formas diversas entre sí (es decir diferentes topologías o patrones de conectividad), las cuales son todas óptimas desde el punto de vista técnico, ya que cumplen con los dos objetivos planteados (minimizar el peso y la energía de deformación), las restricciones de diseño consideradas en el proceso de optimización (ej. esfuerzo axial, esfuerzo de pandeo, límites de las áreas de las

secciones transversales) y todas son estrictamente estables desde el punto de vista del análisis estructural.

Cuando el diseñador requiere una estructura que sea liviana, entonces se seleccionará de la región del frente de Pareto con peso mínimo. Por otro lado, si se requiere una estructura rígida que presente pocas deformaciones internas, entonces la solución adecuada se encuentra en la región del frente de Pareto correspondiente a la energía de deformación mínima. Si los dos objetivos tienen igual importancia y se quiere una estructura en la que exista un equilibrio entre el peso y la energía de deformación, entonces la solución seleccionada se encuentra en la región intermedia del frente de Pareto.

De igual manera, aunque las geométricas generadas con el algoritmo AOTMOH usando el método TEM pueden parecer un poco extrañas, en comparación con las formas convencionales de las torres de energía actuales, su uso puede ser de gran utilidad en el diseño conceptual, dando lugar a nuevas geometrías estéticamente innovadoras y atractivas, que además estructuralmente pueden tener un mejor desempeño, ya que para su generación se usó un método que usa principios básicos de la mecánica de sólidos (el método de las trayectorias de esfuerzo), para colocar los elementos clave de las estructuras en los lugares donde se transmiten las mayores cargas de tensión y de compresión hacia los apoyos, lo cual constituye el objetivo básico de la optimización topológica, que es encontrar patrones óptimos de conectividad para soportar y transmitir cargas. Además, el uso de las trayectorias de esfuerzo puede tener una aplicación potencial en el área de la manufactura aditiva (el proceso de agregar materiales, usualmente capa por capa para crear objetos a partir de modelos de computo CAD 3D), ya que pueden generarse estructuras óptimas (estructuras aporticadas y de celosía) estéticamente atractivas que pueden construirse usando

impresoras 3D y que pueden ser de gran utilidad en diversas áreas de la ingeniería civil, mecánica aeronáutica e industrial.

## 7. Conclusiones

En este trabajo se desarrolló un proceso de optimización topológica multiobjetivo (denominado algoritmo AOTMOH) que minimiza el peso y la energía de deformación (de forma simultánea) de estructuras de celosía (armaduras o cerchas) planas. Como métodos de optimización se emplearon tres de los algoritmos metaheurísticos multiobjetivo más populares en términos de citaciones (AMOS, MOPSO y NSGA-II). La estrategia implementada para generar la estructura base incluye el método clásico NCN (niveles de conectividad de nodos) y el método TEM (trayectorias de esfuerzo y macroelementos).

El uso de los algoritmos metaheurísticos en los problemas de optimización multiobjetivo es adecuado, ya que son métodos eficientes (para este tipo de problemas) que permiten calcular de forma aproximada la solución óptima (en este caso el frente de Pareto, de acuerdo a las definiciones presentadas en la Sección 3.1), usando modelos matemáticos inspirados en la naturaleza.

El problema de optimización topológica de estructuras de celosía (también conocida como optimización topológica discreta) es un problema complejo que busca obtener los mejores patrones de conectividad para transmitir cargas de manera segura a los apoyos. La calidad de las soluciones depende en gran medida de las características geométricas de la estructura base, ya que esta constituye el universo estructural (todas las posibles estructuras solución) a partir de la cual los algoritmos de optimización eliminan elementos (barras) hasta obtener geometrías (topologías) óptimas.

Los algoritmos de optimización que hacen parte del AOTMOH (NSGA-II, MOPSO y AMOSA) fueron validados de forma cualitativa (usando la representación gráfica del frente de Pareto) y cuantitativa (usando las métricas GD,  $\Delta$ , HV), evaluando su convergencia y diversidad en una serie de problemas multiobjetivo de referencia que incluyeron funciones matemáticas sin restricciones, funciones matemáticas con restricciones y estructuras de celosía (armadura o cerchas) planas. Los resultados indicaron que el uso de los tres algoritmos es adecuado, ya que su desempeño en los problemas con las mismas características (naturaleza de las funciones objetivo, variables y restricciones) que el problema de optimización topológica multiobjetivo abordado en este trabajo fue satisfactorio. De esta manera, usando el algoritmo AOTMOH se implementó la estrategia de generación de un único frente de Pareto (conocido como Pareto envolvente), que combina los mejores resultados de optimización de los tres algoritmos (NSGA-II, MOPSO y AMOSA). Lo anterior resulta conveniente, ya que puede explorarse una región más amplia del frente de Pareto a lo largo de los dos objetivos, aprovechando las fortalezas de los tres algoritmos.

Las soluciones representadas en el frente de Pareto envolvente, para cada uno de los problemas de optimización topológica multiobjetivo de estructuras de celosía abordados en este trabajo, son todas óptimas desde el punto de vista matemático para las dos funciones objetivo (minimizar peso y energía de deformación), cumplen con todas las restricciones de diseño y son estables. Cuando se requiere de estructuras que sean económicas y livianas el diseñador busca la solución más adecuada en la región del frente de Pareto con el menor valor del peso. Cuando las estructuras deben ser rígidas, entonces la solución óptima se selecciona de la región del frente de Pareto con los valores mínimos de energía de deformación. Por otro lado, si los dos objetivos son prioritarios, y se requiere de una estructura que sea liviana, pero que al mismo tiempo presente la menor

deformación posible, entonces esta debe ser seleccionada de la región intermedia del frente de Pareto.

Las soluciones óptimas generadas con el algoritmo AOTMOH para las estructuras base generadas con el método TEM (ver problemas de la Secciones 6.3 y 6.4.3), presentan geometrías cuya forma puede parecer extraña en comparación con las estructuras tradicionales. Sin embargo, su uso como herramienta de apoyo en las labores de diseño (ya sea en la fase conceptual, preliminar o definitiva) puede ser de gran utilidad e interés para los diseñadores estructurales, ya que pueden generarse estructuras innovadoras desde el punto de vista estético (nuevas formas y patrones de conectividad) y de desempeño estructural (teniendo en cuenta que la estrategia usa el concepto de trayectorias de esfuerzo para generar los elementos de la estructura base en la dirección definida por los esfuerzos principales).

Los algoritmos de optimización metaheurísticos (NSGA-II, MOPSO y AMOSA) pueden implementarse usando los equipos de cómputo convencionales disponibles, logrando resultados satisfactorios en tiempos razonables (En este trabajo se utilizó un computador portátil con sistema operativo Windows 10, Procesador Intel ® Core™ i5-3230, CPU 2.60GHz y memoria RAM de 8.0 GB). Para las estructuras base con el mayor número de elementos (101, 85 y 88), correspondientes a los problemas de la Sección 6.4.3, el número de diseños realizado por cada algoritmo (NSGA-II, MOPSO, AMOSA) en cada ejecución para cada una de ellas fue de 50000, en un tiempo que estuvo en el rango de los 350 a 600 segundos (6 a 10 minutos). El algoritmo más rápido fue el MOPSO (270 a 450 s), seguido por NSGA-II (360 a 550 s) y AMOSA (400-600 s). Como cada algoritmo se ejecutó 5 veces y tomando como referencia que en promedio el tiempo empleado por cada algoritmo es de 600 segundos por ejecución (el mayor tiempo de los 3 algoritmos metaheurísticos), para la construcción del frente de Pareto envolvente con las mejores

soluciones óptimas no dominadas del proceso de optimización, se combinaron las mejores soluciones del total de 750000 diseños realizados por los tres algoritmos metaheurísticos (NSGA-II, MOPSO y AMOSA) en un tiempo de 9000 s (2.5 horas). Lo anterior indica que, para la implementación de las estrategias de optimización, en este problema, no es necesario el uso de computadores de alto desempeño.

Teniendo en cuenta la gran cantidad de pruebas e iteraciones que realizan los algoritmos de optimización para obtener la mejor solución posible, pueden convertirse en herramientas de apoyo de gran utilidad en las labores de diseño estructural, obteniendo ahorros significativos de tiempo (teniendo en cuenta que el proceso iterativo de diseño es desarrollado por un algoritmo de optimización). La tarea adicional del diseñador consiste en tomar la decisión de seleccionar cual sería solución más conveniente del frente de Pareto envolvente, de acuerdo a sus preferencias o información adicional previa relacionada con la prioridad de los objetivos, factores económicos, detalles constructivos, entre otros.

Con el trabajo realizado se dio cumplimiento al objetivo general definido en la Sección 1.1. El primer objetivo específico puede verificarse mediante los resultados de la Sección 5.1. El segundo objetivo específico con los resultados de las Secciones 4.3 y 5.2. El cumplimiento del tercer y cuarto objetivo se verifica con la solución de los problemas del Capítulo 6.

Los principales aportes de este trabajo son: (a) El desarrollo de una estrategia de optimización topológica multiobjetivo que permite la automatización del proceso de diseño de estructuras de celosía planas. (b) La idea del frente de Pareto envolvente (haciendo analogía a la envolvente de las combinaciones de carga en el diseño estructural) con las mejores soluciones obtenidas por los tres algoritmos metaheurísticos (NSGA-II, MOPSO y AMOSA) logrando una mayor convergencia y diversidad en las soluciones del frente de Pareto. (c) La estrategia para generar la estructura base

usando el método TEM (trayectorias de esfuerzo y macroelementos), ya que permite ubicar los nodos y los elementos iniciales de esta estructura en lugares clave del dominio de diseño. Con el uso del concepto de las trayectorias de esfuerzo (Beyer, 2015; Gao, Liu, et al., 2017) se ubican los elementos clave de la estructura base en las direcciones definidas por los esfuerzos principales ( $\sigma_1$  y  $\sigma_2$ ), las cuales proporcionan las rutas óptimas de transmisión de cargas. Con el uso del método de macroelementos (X. Zhang et al., 2016) se generan elementos adicionales para garantizar que la estructura base inicial sea estable.

La formulación matemática del proceso de optimización definido en el algoritmo AOTMOH puede modificarse para incluir otro tipo de funciones objetivo (ej. minimizar la energía potencial, el esfuerzo máximo, entre otras) y restricciones de diseño (ej. frecuencias naturales y volumen máximo). lo cual es un proceso relativamente sencillo.

Aunque el proceso de optimización topológica multiobjetivo desarrollado en este trabajo fue aplicado en estructuras de celosía planas, es posible extender su aplicación a estructuras tridimensionales, ajustando algunas de sus funciones internas. Además, para tener un uso más directo en la industria puede emplearse la optimización de tamaño (para una geometría conocida calcular el valor óptimo de las secciones transversales).

El uso de las trayectorias de esfuerzo puede ser extendido en el área de la manufactura aditiva, ya que pueden obtenerse estructuras con un buen desempeño estructural que pueden construirse usando impresoras 3D y ser de gran utilidad en diferentes áreas de la industria.

Finalmente, aunque la mayor parte de las aplicaciones de los algoritmos metaheurísticos en optimización estructural tratan con problemas a pequeña escala (requiriendo de futuros estudios enfocados en aplicaciones de gran escala) y existen muchos algoritmos nuevos (muchos algoritmos hacen más difícil entender los mecanismos de su funcionamiento en general, haciendo que sea

necesario tener un enfoque unificado para analizar y clasificar los algoritmos, para entenderlos de una manera más profunda) (Gandomi et al., 2013), su aplicación como herramienta de apoyo en los problemas de diseño estructural, permite la automatización de este proceso, mediante la solución de un problema de optimización matemático, proporcionando una amplia cantidad de soluciones óptimas aceptables, con tiempos y recursos computacionales razonables.

### Referencias bibliográficas

- Altair Engineering Inc. (2019). Altair OptiStruct (Version 2019) [Computer software]. Troy, Michigan: Altair Engineering, Inc.
- Alzate Acevedo, C. (2016). *Diseño Conceptual de torres de transmisión de energía utilizando el método de optimización topológica considerando análisis tridimensional y cargas producidas por cables y viento (Tesis de maestría)*. Universidad Nacional de Colombia.
- American Institute of Steel Construction (AISC). (2016). *Specification for Structural Steel Buildings (ANSI/AISC 360-16)*. Chicago, USA.
- American Institute of Steel Construction (AISC). (2017). AISC Shapes Database v15.0. Retrieved May 22, 2019, from <https://www.aisc.org/globalassets/aisc/manual/v15.0-shapes-database/aisc-shapes-database-v15.0.xlsx>
- Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., & Sigmund, O. (2011). Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1), 1–16. <https://doi.org/10.1007/s00158-010-0594-7>
- ANSYS Inc. (2018). ANSYS Workbench (Version 2019 R1) [Computer software]. Canonsburg, Pennsylvania: ANSYS, Inc.
- Arora, J. S. (2017a). Introduction to Design Optimization. In *Introduction to Optimum Design* (4th ed., pp. 3–18). Elsevier. <https://doi.org/10.1016/B978-0-12-800806-5.00001-9>

- Arora, J. S. (2017b). More on Numerical Methods for Unconstrained Optimum Design. In *Introduction to Optimum Design* (4th ed., pp. 455–509). Elsevier. <https://doi.org/10.1016/B978-0-12-800806-5.00011-1>
- Arora, J. S. (2017c). Optimum Design Problem Formulation. In *Introduction to Optimum Design* (4th ed., pp. 19–70). Elsevier. <https://doi.org/10.1016/B978-0-12-800806-5.00002-0>
- Asociación Colombiana de Ingeniería Sísmica (AIS). (2010). *Reglamento Colombiano de Construcción Sismo Resistente (NSR-10)*. Bogotá, D.C.
- Assimi, H., & Jamali, A. (2018). A hybrid algorithm coupling genetic programming and Nelder–Mead for topology and size optimization of trusses with static and dynamic constraints. *Expert Systems with Applications*, 95, 127–141. <https://doi.org/10.1016/j.eswa.2017.11.035>
- Assimi, H., Jamali, A., & Nariman-zadeh, N. (2017). Sizing and topology optimization of truss structures using genetic programming. *Swarm and Evolutionary Computation*, 37, 90–103. <https://doi.org/10.1016/j.swevo.2017.05.009>
- Bandyopadhyay, S., Saha, S., Maulik, U., & Deb, K. (2008). A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, 12(3), 269–283. <https://doi.org/10.1109/TEVC.2007.900837>
- Barr, M. (2013). Bi-Objective Optimization Based on Compromise Method for Horizontal Fragmentation in Relational Data Warehouses. *International Journal of Machine Learning and Computing*, 3(3), 250–254. <https://doi.org/10.7763/IJMLC.2013.V3.313>
- Bekdaş, G., Nigdeli, S. M., & Yang, X. S. (2015). Sizing optimization of truss structures using flower pollination algorithm. *Applied Soft Computing Journal*, 37, 322–331. <https://doi.org/10.1016/j.asoc.2015.08.037>
- Bendsøe, M. P., & Sigmund, O. (2004). *Topology Optimization: Theory, Methods, and*

- Applications* (2nd ed.). Berlin, Heidelberg: Springer Berlin Heidelberg.  
<https://doi.org/10.1007/978-3-662-05086-6>
- Beyer, F. R. (2015). *Principal Stress Trajectories in Numerical Solid Mechanics: An Algorithm for the Visualisation of Structural Element's Stress States in Two and Three Dimensions (PhD thesis)*. Technische Universität Dresden. Retrieved from <https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-178497>
- Branke, J., Deb, K., Miettinen, K., & Słowiński, R. (Eds.). (2008). *Multiobjective Optimization* (1st ed., Vol. 5252). Berlin, Heidelberg: Springer Berlin Heidelberg.  
<https://doi.org/10.1007/978-3-540-88908-3>
- Cao, H., Qian, X., & Zhou, Y. (2018). Large-scale structural optimization using metaheuristic algorithms with elitism and a filter strategy. *Structural and Multidisciplinary Optimization*, 57(2), 799–814. <https://doi.org/10.1007/s00158-017-1784-3>
- Cazacu, R., & Grama, L. (2014). Steel Truss Optimization Using Genetic Algorithms and FEA. *Procedia Technology*, 12, 339–346. <https://doi.org/10.1016/j.protcy.2013.12.496>
- Chen, X., & Liu, Y. (2015). *Finite Element Modeling and Simulation with ANSYS Workbench* (2nd ed.). Boca Raton: CRC Press.
- Cheng, M. Y., Prayogo, D., Wu, Y. W., & Lukito, M. M. (2016). A Hybrid Harmony Search algorithm for discrete sizing optimization of truss structure. *Automation in Construction*, 69, 21–33. <https://doi.org/10.1016/j.autcon.2016.05.023>
- Christensen, P., & Klarbring, A. (2009). *An Introduction to Structural Optimization* (1st ed., Vol. 153). Dordrecht: Springer Netherlands. <https://doi.org/10.1007/978-1-4020-8666-3>
- Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3), 256–

279. <https://doi.org/10.1109/TEVC.2004.826067>

Coello Coello, C., Lamont, G. B., & van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems* (2nd ed.). Boston, MA: Springer US. <https://doi.org/10.1007/978-0-387-36797-2>

Coello Coello, C.A., & Lechuga, M. S. (2002). MOPSO: a proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)* (Vol. 2, pp. 1051–1056). IEEE. <https://doi.org/10.1109/CEC.2002.1004388>

Coello Coello, Carlos A. (2019). EMOO Web Page. Retrieved September 13, 2019, from <http://delta.cs.cinvestav.mx/~ccoello/EMOO/testfuncs/>

Colciencias. (2013). *Plan Estratégico de Ciencia, Tecnología e Innovación en Energía y Minería 2013-2022*. Bogotá, D.C.

Computers and Structures Inc. (CSI). (2018). SAP2000 (Version 20.2.0) [Computer software]. Walnut Creek, California: Computers and Structures, Inc.

Couceiro, I., París, J., Martínez, S., Colominas, I., Navarrina, F., & Casteleiro, M. (2016). Structural optimization of lattice steel transmission towers. *Engineering Structures*, 117, 274–286. <https://doi.org/10.1016/j.engstruct.2016.03.005>

D'Amico, B., & Pomponi, F. (2018). Sustainability Tool to Optimise Material Quantities of Steel in the Construction Industry. *Procedia CIRP*, 69(May), 184–188. <https://doi.org/10.1016/j.procir.2017.10.006>

Dassault Systèmes (3DS). (2018a). DS Simulia Abaqus CAE (Version 2018) [Computer software]. Vélizy-Villacoublay: Dassault Systèmes.

Dassault Systèmes (3DS). (2018b). DS Simulia Tosca (Version 2018) [Computer software].

Vélizy-Villacoublay: Dassault Systèmes.

- de Souza, R. R., Fadel Miguel, L. F., Lopez, R. H., Miguel, L. F. F., & Torii, A. J. (2016). A procedure for the size, shape and topology optimization of transmission line tower structures. *Engineering Structures*, *111*, 162–184. <https://doi.org/10.1016/j.engstruct.2015.12.005>
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. <https://doi.org/10.1109/4235.996017>
- Deb, Kalyanmoy. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, *186*(2–4), 311–338. [https://doi.org/10.1016/S0045-7825\(99\)00389-8](https://doi.org/10.1016/S0045-7825(99)00389-8)
- Deb, Kalyanmoy. (2001). *Multi-Objective Optimization using Evolutionary Algorithms* (1st ed.). Chichester: John Wiley & Sons Ltd.
- Deb, Kalyanmoy, & Gulati, S. (2001). Design of truss-structures for minimum weight using genetic algorithms. *Finite Elements in Analysis and Design*, *37*(5), 447–465. [https://doi.org/10.1016/S0168-874X\(00\)00057-3](https://doi.org/10.1016/S0168-874X(00)00057-3)
- Deb, Kalyanmoy, & Sundar, J. (2006). Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation - GECCO '06* (Vol. 2, p. 635). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1143997.1144112>
- Degertekin, S. O., Lamberti, L., & Ugur, I. B. (2017). Sizing, layout and topology design optimization of truss structures using the Jaya algorithm. *Applied Soft Computing Journal*. <https://doi.org/10.1016/j.asoc.2017.10.001>
- Descamps, B., & Coelho, R. F. (2013). Graph Theory in Evolutionary Truss Design Optimization.

- In *Metaheuristic Applications in Structures and Infrastructures* (pp. 241–268). Elsevier.  
<https://doi.org/10.1016/B978-0-12-398364-0.00010-3>
- Descamps, B., & Filomeno Coelho, R. (2014). The nominal force method for truss geometry and topology optimization incorporating stability considerations. *International Journal of Solids and Structures*, *51*(13), 2390–2399. <https://doi.org/10.1016/j.ijsolstr.2014.03.003>
- Dhiman, G., & Kumar, V. (2018). Multi-objective spotted hyena optimizer: A Multi-objective optimization algorithm for engineering problems. *Knowledge-Based Systems*, *150*, 175–197. <https://doi.org/10.1016/j.knosys.2018.03.011>
- Du, K.-L., & Swamy, M. N. S. (2016). *Search and Optimization by Metaheuristics*. Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-41192-7>
- Faramarzi, A., & Afshar, M. H. (2012). Application of cellular automata to size and topology optimization of truss structures. *Scientia Iranica*, *19*(3), 373–380. <https://doi.org/10.1016/j.scient.2012.04.009>
- Felkner, J., Chatzi, E., & Kotnik, T. (2015). Interactive truss design using Particle Swarm Optimization and NURBS curves. *Journal of Building Engineering*, *4*, 60–74. <https://doi.org/10.1016/j.jobbe.2015.08.004>
- Frans, R., & Arfiadi, Y. (2014). Sizing, shape, and topology optimizations of roof trusses using hybrid genetic algorithms. *Procedia Engineering*, *95*(Scescm), 185–195. <https://doi.org/10.1016/j.proeng.2014.12.178>
- Gan, B. S., Hara, T., Han, A., Alisjahbana, S. W., & As'Ad, S. (2017). Evolutionary ACO Algorithms for Truss Optimization Problems. *Procedia Engineering*, *171*, 1100–1107. <https://doi.org/10.1016/j.proeng.2017.01.467>
- Gandomi, A. H., Yang, X.-S., Talatahari, S., & Alavi, A. H. (2013). Metaheuristic Algorithms in

- Modeling and Optimization. In *Metaheuristic Applications in Structures and Infrastructures* (pp. 1–24). Elsevier. <https://doi.org/10.1016/B978-0-12-398364-0.00001-2>
- Gao, G., Li, Y., Pan, H., Chen, L., & Liu, Z. (2017). An effective members-adding method for truss topology optimization based on principal stress trajectories. *Engineering Computations*, 34(6), 2088–2104. <https://doi.org/10.1108/EC-05-2016-0166>
- Gao, G., Liu, Z., Li, Y., & Qiao, Y. (2017). A new method to generate the ground structure in truss topology optimization. *Engineering Optimization*, 49(2), 235–251. <https://doi.org/10.1080/0305215X.2016.1169050>
- Geem, Z. W. (Ed.). (2009). *Harmony Search Algorithms for Structural Design Optimization* (1st ed., Vol. 239). Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-03450-3>
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* (1st ed.). Boston, MA: Addison-Wesley Longman Publishing Co., Inc.
- Gonçalves, M. S., Lopez, R. H., & Miguel, L. F. F. (2015). Search group algorithm: A new metaheuristic method for the optimization of truss structures. *Computers and Structures*, 153, 165–184. <https://doi.org/10.1016/j.compstruc.2015.03.003>
- Goodno, B. J., & Gere, J. M. (2018). *Mechanics of Materials* (9th ed.). Boston: Cengage Learning.
- Hare, W., Nutini, J., & Tesfamariam, S. (2013). A survey of non-gradient optimization methods in structural engineering. *Advances in Engineering Software*, 59, 19–28. <https://doi.org/10.1016/j.advengsoft.2013.03.001>
- Hasançebi, O., & Azad, S. K. (2015). Adaptive dimensional search: A new metaheuristic algorithm for discrete truss sizing optimization. *Computers and Structures*, 154, 1–16. <https://doi.org/10.1016/j.compstruc.2015.03.014>

- Hibbeler, R.C. (2012). *Análisis Estructural* (8th ed.). México: Pearson Educación de México, S.A de C.V.
- Hibbeler, Russell Charles. (2011). *Mechanics of materials* (8th ed.). Upper Saddle River: Pearson Prentice Hall.
- Hosseini, S. S., Hamidi, S. A., Mansuri, M., & Ghoddosian, A. (2015). Multi Objective Particle Swarm Optimization (MOPSO) for Size and Shape Optimization of 2D Truss Structures. *Periodica Polytechnica Civil Engineering*, 59(1), 9–14. <https://doi.org/10.3311/PPci.7341>
- Kanarachos, S., Griffin, J., & Fitzpatrick, M. E. (2017). Efficient truss optimization using the contrast-based fruit fly optimization algorithm. *Computers and Structures*, 182, 137–148. <https://doi.org/10.1016/j.compstruc.2016.11.005>
- Kar, A. K. (2016). Bio inspired computing - A review of algorithms and scope of applications. *Expert Systems with Applications*, 59, 20–32. <https://doi.org/10.1016/j.eswa.2016.04.018>
- Kassimali, A. (2012). *Matrix Analysis of Structures* (2nd ed.). Stamford: Cengage Learning.
- Kaveh, A., & Khayatazad, M. (2013). Ray optimization for size and shape optimization of truss structures. *Computers and Structures*, 117, 82–94. <https://doi.org/10.1016/j.compstruc.2012.12.010>
- Kaveh, A., & Mahdavi, V. R. (2015). A hybrid CBO-PSO algorithm for optimal design of truss structures with dynamic constraints. *Applied Soft Computing Journal*, 34, 260–273. <https://doi.org/10.1016/j.asoc.2015.05.010>
- Kaveh, A., & Mahdavi, V. R. (2018). Multi-objective colliding bodies optimization algorithm for design of trusses. *Journal of Computational Design and Engineering*, 1–11. <https://doi.org/10.1016/j.jcde.2018.04.001>
- Kaveh, A., & Talatahari, S. (2009). A particle swarm ant colony optimization for truss structures

- with discrete variables. *Journal of Constructional Steel Research*, 65(8–9), 1558–1568.  
<https://doi.org/10.1016/j.jcsr.2009.04.021>
- Kaveh, Ali, & Ilchi Ghazaan, M. (2018). *Meta-heuristic Algorithms for Optimal Design of Real-Size Structures* (1st ed.). Cham: Springer International Publishing.  
<https://doi.org/10.1007/978-3-319-78780-0>
- Kelly, D. W., & Tosh, M. W. (2000). Interpreting load paths and stress trajectories in elasticity. *Engineering Computations*, 17(2), 117–135. <https://doi.org/10.1108/02644400010313084>
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference On*, 4, 1942–1948 vol.4.  
<https://doi.org/10.1109/ICNN.1995.488968>
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671>
- Knowles, J. D., & Corne, D. W. (2000). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2), 149–172.  
<https://doi.org/10.1162/106365600568167>
- Kuo, H. C., Chiu, J. T., & Lin, C. H. (2012). Intelligent Garbage Can Decision-Making Model evolution algorithm for optimization of structural topology of plane trusses. *Applied Soft Computing Journal*, 12(9), 2719–2727. <https://doi.org/10.1016/j.asoc.2012.03.011>
- Lieu, Q. X., Do, D. T. T., & Lee, J. (2018). An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. *Computers & Structures*. <https://doi.org/10.1016/j.compstruc.2017.06.016>
- Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). *Geographic Information Science and Systems* (4th ed.). Hoboken: John Wiley & Sons, Inc.

- Luh, G. C., & Lin, C. Y. (2011). Optimal design of truss-structures using particle swarm optimization. *Computers and Structures*, 89(23–24), 2221–2232. <https://doi.org/10.1016/j.compstruc.2011.08.013>
- Mahesh, K., Nallagownden, P., & Elamvazuthi, I. (2016). Advanced Pareto Front Non-Dominated Sorting Multi-Objective Particle Swarm Optimization for Optimal Placement and Sizing of Distributed Generation. *Energies*, 9(12), 982. <https://doi.org/10.3390/en9120982>
- McCormac, J. C., & Csernak, S. F. (2012). *Diseño de estructuras de acero* (5th ed.). Mexico: Alfaomega Grupo Editor, S.A de C.V.
- MIDAS Information Technology Co. Ltd. (MIDAS IT). (2018). midas NFX (Version 2018 R1) [Computer software]. Seongnam: MIDAS Information Technology Co., Ltd.
- Miguel, L. F. F., Lopez, R. H., & Miguel, L. F. F. (2013). Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm. *Advances in Engineering Software*, 56, 23–37. <https://doi.org/10.1016/j.advengsoft.2012.11.006>
- Millán Páramo, C. A. (2013). *Optimización topológica de elementos estructurales: un enfoque estocástico (Tesis de maestría)*. Universidad Industrial de Santander.
- Mitropoulou, C. C., Fourkiotis, Y., Lagaros, N. D., & Karlaftis, M. G. (2013). Evolution Strategies-Based Metaheuristics in Structural Design Optimization. In *Metaheuristic Applications in Structures and Infrastructures* (pp. 79–102). Elsevier. <https://doi.org/10.1016/B978-0-12-398364-0.00004-8>
- Mokarram, V., & Banan, M. R. (2018). A new PSO-based algorithm for multi-objective optimization with continuous and discrete design variables. *Structural and Multidisciplinary Optimization*, 57(2), 509–533. <https://doi.org/10.1007/s00158-017-1764-7>
- Moreno-De-Luca, L., & Begambre Carrillo, O. J. (2013). Multi-Objective Heuristic Computation

- Applied to Architectural and Structural Design: A Review. *International Journal of Architectural Computing*, 11(4), 363–392. <https://doi.org/10.1260/1478-0771.11.4.363>
- Moreno De Luca, L. (2013). *Computación evolutiva aplicada al diseño de estructuras reticulares (Tesis de maestría)*. Universidad Industrial de Santander.
- Mortazavi, A., & Toğan, V. (2016). Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer. *Structural and Multidisciplinary Optimization*, 54(4), 715–736. <https://doi.org/10.1007/s00158-016-1449-7>
- MSC Software Corporation. (2018). MSC Nastran (Version 2018) [Computer software]. Newport Beach, California: MSC Software Corporation.
- Multiscale Multiphysics Design Optimization Laboratory (M2DO). (2018). OpenLSTO (Version 1.0) [Computer software]. San Diego, California: M2DO Multiscale Multiphysics Design Optimization Laboratory.
- Navidi, W. (2015). *Statistics for Engineers and Scientists* (4th ed.). New York: McGraw-Hill Education.
- Ngatchou, P., Zarei, A., & El-Sharkawi, A. (2005). Pareto Multi Objective Optimization. In *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems* (pp. 84–91). IEEE. <https://doi.org/10.1109/ISAP.2005.1599245>
- Noilublao, N., & Bureerat, S. (2011). Simultaneous topology, shape and sizing optimisation of a three-dimensional slender truss tower using multiobjective evolutionary algorithms. *Computers and Structures*, 89(23–24), 2531–2538. <https://doi.org/10.1016/j.compstruc.2011.08.010>
- Oliver Olivella, X., & Agelet de Saracibar, C. (2017). *Continuum Mechanics for Engineers: Theory and Problems* (2nd ed.). Barcelona. <https://doi.org/10.13140/RG.2.2.25821.20961>

- Pandey, A., Datta, R., & Bhattacharya, B. (2017). Topology optimization of compliant structures and mechanisms using constructive solid geometry for 2-d and 3-d applications. *Soft Computing*, 21(5), 1157–1179. <https://doi.org/10.1007/s00500-015-1845-8>
- Pereira, O. J. B. A., & de Almeida, J. P. B. M. (1994). Automatic drawing of stress trajectories in plane systems. *Computers & Structures*, 53(2), 473–476. [https://doi.org/10.1016/0045-7949\(94\)90219-4](https://doi.org/10.1016/0045-7949(94)90219-4)
- Querin, O. M., Victoria, M., Alonso, C., Ansola, R., & Martí, P. (2017). Introduction. In *Topology Design Methods for Structural Optimization* (pp. 1–13). Elsevier. <https://doi.org/10.1016/B978-0-08-100916-1.00001-5>
- Rahami, H., Kaveh, A., & Gholipour, Y. (2008). Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. *Engineering Structures*, 30(9), 2360–2369. <https://doi.org/10.1016/j.engstruct.2008.01.012>
- Richardson, J. N., Adriaenssens, S., Bouillard, P., & Coelho, R. F. (2012). Multiobjective topology optimization of truss structures with kinematic stability repair. *Structural and Multidisciplinary Optimization*, 46(4), 513–532. <https://doi.org/10.1007/s00158-012-0777-5>
- Röber, M., & Todtermuschke, M. (2014). Multiobjective topology optimization of truss structures for assembly devices using intelligent swarm techniques. *Procedia CIRP*, 23(C), 241–245. <https://doi.org/10.1016/j.procir.2014.10.083>
- Robert McNeel & Associates. (2019a). Grasshopper (Version 1.0.0) [Computer software]. Seattle, Washington: Robert McNeel & Associates.
- Robert McNeel & Associates. (2019b). Rhinoceros (Version 6 SR13) [Computer software]. Seattle, Washington: Robert McNeel & Associates.
- Rozvany, G. I. N. (Ed.). (1997). *Topology Optimization in Structural Mechanics* (1st ed.). Vienna:

- Springer Vienna. <https://doi.org/10.1007/978-3-7091-2566-3>
- Sadd, M. H. (2014). Stress and Equilibrium. In *Elasticity* (3rd ed., pp. 55–80). Kidlington: Elsevier. <https://doi.org/10.1016/B978-0-12-408136-9.00003-9>
- Sahab, M. G., Toropov, V. V., & Gandomi, A. H. (2013). A Review on Traditional and Modern Structural Optimization. In *Metaheuristic Applications in Structures and Infrastructures* (pp. 25–47). Elsevier. <https://doi.org/10.1016/B978-0-12-398364-0.00002-4>
- Saka, M. P., Hasańcebi, O., & Geem, Z. W. (2016). Metaheuristics in structural optimization and discussions on harmony search algorithm. *Swarm and Evolutionary Computation*, 28, 88–97. <https://doi.org/10.1016/j.swevo.2016.01.005>
- Savsani, V. J., Tejani, G. G., Patel, V. K., & Savsani, P. (2017). Modified meta-heuristics using random mutation for truss topology optimization with static and dynamic constraints. *Journal of Computational Design and Engineering*, 4(2), 106–130. <https://doi.org/10.1016/j.jcde.2016.10.002>
- Su, R., Wang, X., Gui, L., & Fan, Z. (2011). Multi-objective topology and sizing optimization of truss structures based on adaptive multi-island search strategy. *Structural and Multidisciplinary Optimization*, 43(2), 275–286. <https://doi.org/10.1007/s00158-010-0544-4>
- Talatahari, S., & Kaveh, A. (2013). Optimum Design of Skeletal Structures via Big Bang–Big Crunch Algorithm. In *Metaheuristic Applications in Structures and Infrastructures* (pp. 173–205). Elsevier. <https://doi.org/10.1016/B978-0-12-398364-0.00008-5>
- Talbi, E.-G. (2009). *Metaheuristics: From design to implementation* (1st ed.). Hoboken, NJ, USA: John Wiley & Sons, Inc. <https://doi.org/10.1002/9780470496916>
- Technical University of Denmark (DTU). (2014). TopOpt3D (Version 1) [Computer software]. Copenhagen: DTU Mechanical Engineering and DTU Compute.

- Tejani, G. G., Kumar, S., & Gandomi, A. H. (2019). Multi-objective heat transfer search algorithm for truss optimization. *Engineering with Computers*. <https://doi.org/10.1007/s00366-019-00846-6>
- Tejani, G. G., Pholdee, N., Bureerat, S., & Prayogo, D. (2018). Multiobjective adaptive symbiotic organisms search for truss optimization problems. *Knowledge-Based Systems, 161*, 398–414. <https://doi.org/10.1016/j.knosys.2018.08.005>
- Tejani, G. G., Savsani, V. J., Bureerat, S., Patel, V. K., & Savsani, P. (2019). Topology optimization of truss subjected to static and dynamic constraints by integrating simulated annealing into passing vehicle search algorithms. *Engineering with Computers, 35*(2), 499–517. <https://doi.org/10.1007/s00366-018-0612-8>
- Tejani, G. G., Savsani, V. J., Patel, V. K., & Savsani, P. V. (2018). Size, shape, and topology optimization of planar and space trusses using mutation-based improved metaheuristics. *Journal of Computational Design and Engineering, 5*(2), 198–214. <https://doi.org/10.1016/j.jcde.2017.10.001>
- Tejani, G., Savsani, V., & Bureerat, S. (2018). *Truss Topology Optimization: A review* (1st ed.). Scholar's Press. Retrieved from [https://www.researchgate.net/publication/326377131\\_Truss\\_Topology\\_Optimization\\_A\\_Review](https://www.researchgate.net/publication/326377131_Truss_Topology_Optimization_A_Review)
- The Mathworks Inc. (2017). MATLAB R2017a (Version 9.2.0) [Computer software]. Natick, Massachusetts: The Mathworks, Inc.
- Tomšič, P., & Duhovnik, J. (2014). Simultaneous topology and size optimization of 2D and 3D trusses using evolutionary structural optimization with regard to commonly used topologies. *Advances in Mechanical Engineering, 2014*. <https://doi.org/10.1155/2014/864807>

- Tort, C., Şahin, S., & Hasańcebi, O. (2017). Optimum design of steel lattice transmission line towers using simulated annealing and PLS-TOWER. *Computers and Structures*, 179, 75–94. <https://doi.org/10.1016/j.compstruc.2016.10.017>
- Unidad de Planeación Minero Energética (UPME). (2018). *Plan de Expansi3n de Referencia Generaci3n – Transmisi3n 2017-2031*. Bogot3, D.C. Retrieved from [https://www1.upme.gov.co/Energia\\_electrica/Plan\\_GT\\_2017\\_2031\\_PREL.pdf](https://www1.upme.gov.co/Energia_electrica/Plan_GT_2017_2031_PREL.pdf)
- Yang, X.-S. (2010). *Engineering Optimization* (1st ed.). Hoboken, NJ, USA: John Wiley & Sons, Inc. <https://doi.org/10.1002/9780470640425>
- Yang, X.-S. (2013). Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers*, 29(2), 175–184. <https://doi.org/10.1007/s00366-012-0254-1>
- Zadeh, P. M., & Shirazi, M. A. S. (2013). Multidisciplinary Design and Optimization Methods. In *Metaheuristic Applications in Structures and Infrastructures* (pp. 103–127). Elsevier. <https://doi.org/10.1016/B978-0-12-398364-0.00005-X>
- Zavala, G., Nebro, A. J., Luna, F., & Coello Coello, C. A. (2016). Structural design using multi-objective metaheuristics. Comparative study and application to a real-world problem. *Structural and Multidisciplinary Optimization*, 53(3), 545–566. <https://doi.org/10.1007/s00158-015-1291-3>
- Zavala, G. R., Nebro, A. J., Luna, F., & Coello Coello, C. A. (2014). A survey of multi-objective metaheuristics applied to structural optimization. *Structural and Multidisciplinary Optimization*, 49(4), 537–558. <https://doi.org/10.1007/s00158-013-0996-4>
- Zhang, W., Zhu, J., & Gao, T. (2016). *Topology Optimization in Engineering Structure Design* (1st ed.). London: ISTE Press Ltd.
- Zhang, X., Maheshwari, S., Ramos, A. S., & Paulino, G. H. (2016). Macroelement and Macropatch

Approaches to Structural Topology Optimization Using the Ground Structure Method.  
*Journal of Structural Engineering*, 142(11), 04016090.  
[https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0001524](https://doi.org/10.1061/(ASCE)ST.1943-541X.0001524)

Zhang, X. S., Maheshwari, S., Ramos, A. S., & Paulino, G. H. (2018). Closure to “Macroelement and Macropatch Approaches to Structural Topology Optimization Using the Ground Structure Method” by Xiaojia Zhang, Sushant Maheshwari, Adeildo S. Ramos Jr., and Glaucio H. Paulino. *Journal of Structural Engineering*, 144(10), 07018009.  
[https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0002179](https://doi.org/10.1061/(ASCE)ST.1943-541X.0002179)

## Apéndices

### Apéndice A. Algoritmos de optimización metaheurísticos

#### A.1. Algoritmo NSGA-II (Non-dominated Sorting Genetic Algorithm II)

El procedimiento esquemático del algoritmo NSGA-II se muestra en la Figura A.1 :

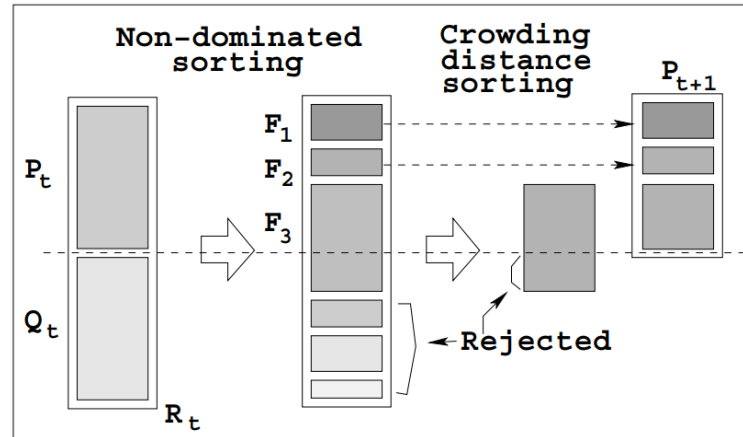


Figura A.1. Procedimiento esquemático del algoritmo NSGA-II

Fuente: Branke et al. (2008)

La explicación del esquema mostrado en la Figura A.1:

Para una generación  $t$ , la población de descendencia  $Q_t$  es creada a partir de la población de los padres  $P_t$ , usando los operadores genéticos tradicionales: selección, cruzamiento y mutación. Después estas dos poblaciones se combinan para formar una población  $R_t$  de tamaño  $2N$  (como  $R_t$  incluye la población  $P_t$  y  $Q_t$ , el elitismo está asegurado, ya que los padres tienen la posibilidad de competir con los hijos por un lugar en la siguiente generación). Después, se clasifica la población  $R_t$  usando *fast nondominated sorting*, para identificar los diferentes frentes no dominados ( $F_i, i = 1, 2, 3, \dots, etc$ ), de acuerdo rango o clase de no dominación. A cada una de las soluciones se le asigna un valor de calidad (fitness) igual al rango de no dominación al cual pertenece. De esta manera se asume un problema de minimización de fitness (K. Deb et al., 2002).

Después, se crea la nueva población de la siguiente generación  $P_{t+1}$  de tamaño  $N$ , a partir de la población combinada  $R_t$ : Las soluciones que pertenecen al mejor frente no dominado  $\mathcal{F}_1$ , son las mejores soluciones en la población combinada, por lo cual deben tener un mayor énfasis que las demás soluciones de los otros frentes. Si el tamaño de  $\mathcal{F}_1$  es menor de  $N$ , todas las soluciones de  $\mathcal{F}_1$  se incluyen en  $P_{t+1}$ . Los miembros restantes de  $P_{t+1}$  se eligen de los frentes siguientes de acuerdo al orden de clasificación. De esta manera, las soluciones de  $\mathcal{F}_2$  son las siguientes, seguidas por  $\mathcal{F}_3$ , y así sucesivamente. Este proceso se sigue hasta que ningún otro frente pueda ser acomodado. El último frente no dominado, más allá del cual ningún otro puede ser acomodado se denomina  $\mathcal{F}_l$ . Generalmente el número de soluciones totales que se incluyen desde  $\mathcal{F}_1$  hasta  $\mathcal{F}_l$  es mayor que el tamaño de la población  $N$ . Para elegir exactamente  $N$  miembros en la población  $P_{t+1}$ , las soluciones del último frente  $\mathcal{F}_l$  se clasifican usando el valor de *crowding distance*. Las soluciones de  $\mathcal{F}_l$  se organizan en orden descendente de magnitud de *crowding distance*, y se eligen las mejores soluciones necesarias hasta llenar los espacios disponibles (K. Deb et al., 2002).

A continuación, en la Tabla A.1 se muestra el pseudocódigo del algoritmo NSGA-II:

Tabla A.1. Pseudocódigo del algoritmo NSGA-II

NSGA-II	
$t = 1$	% Iniciar contador de soluciones
Inicializar $P_t$	% Generar de forma aleatoria y evaluar una población de padres, de tamaño $N$
$\mathcal{F} = non\_dominated\_sorting(P_t)$	% Identificar los diferentes frentes no dominados $\mathcal{F}_i$ , $i = 1,2,3, \dots, etc.$
$crowding\_distance\_assignment(\mathcal{F})$	% Calcular <i>crowding distance</i> para cada frente $\mathcal{F}_i$
$Q_t = operadores\_geneticos(P_t, \mathcal{F})$	% Crear población de descendencia, aplicando operadores genéticos a $P_t$
$R_t = P_t \cup Q_t$	% Combinar $P_t$ con $Q_t$ para formar $R_t$ de tamaño $2N$
$\mathcal{F} = non\_dominated\_sorting(R_t)$	% Identificar los diferentes frentes no dominados $\mathcal{F}_i$ , $i = 1,2,3, \dots, etc.$
$crowding\_distance\_assignment(\mathcal{F})$	% Calcular <i>crowding distance</i> para cada frente $\mathcal{F}_i$
$P_t = reduccion(R_t, \mathcal{F})$	% Nueva población de padres de tamaño $N$ , a partir de la reducción de $R_t$
<b>FOR</b> $t = 1$ hasta $gen$	% $gen$ es el número de generaciones o iteraciones
$Q_t = operadores\_geneticos(P_t, \mathcal{F})$	% Crear población de descendencia, aplicando operadores genéticos a $P_t$
$R_t = P_t \cup Q_t$	% Combinar $P_t$ con $Q_t$ para formar $R_t$ de tamaño $2N$
$\mathcal{F} = non\_dominated\_sorting(R_t)$	% identificar los diferentes frentes no dominados $\mathcal{F}_i$ , $i = 1,2,3, \dots, etc.$
$crowding\_distance\_assignment(\mathcal{F})$	% Calcular <i>crowding distance</i> para cada frente $\mathcal{F}_i$
$P_t = reduccion(R_t, \mathcal{F})$	% Nueva población de padres de tamaño $N$ , a partir de la reducción de $R_t$
<b>END</b>	

Nota: Adaptado de K. Deb et al. (2002)

Las soluciones óptimas de Pareto, corresponden al primer frente no dominado de  $R_t$ , cuando se ha cumplido el número máximo de generaciones.

**A.1.1. Clasificación no dominada de las soluciones.** El algoritmo NSGA-II usa el enfoque *fast nondominated sorting* para la clasificación de una población  $P$  en diferentes frentes no dominados, de acuerdo al nivel o rango de dominación de cada una de sus soluciones (ver Figura A.2).

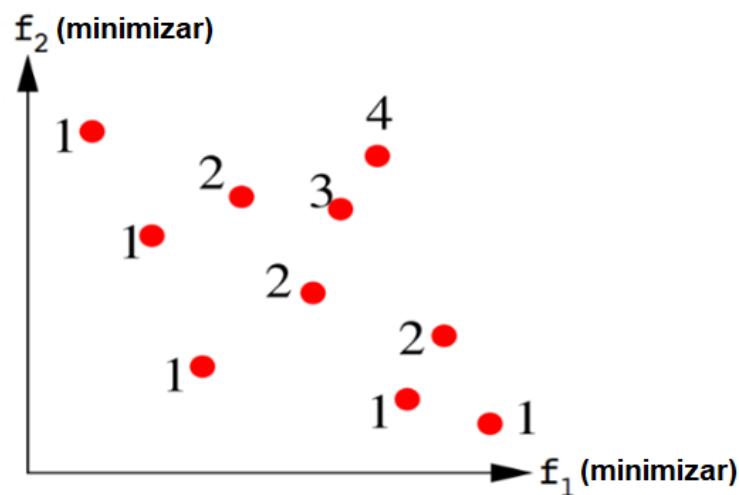


Figura A.2. Clasificación no dominada (nondominated sorting) de los puntos solución de una población  
Fuente: Branke et al. (2008)

El procedimiento inicia calculando para cada una de las soluciones de  $P$ , dos entidades: (a) contador de dominación  $n_p$  el cual indica cual número de soluciones que dominan la solución  $p$ . (b)  $S_p$ , el cual indica el conjunto de soluciones que  $p$  domina (K. Deb et al., 2002).

Todas las soluciones del primer frente no dominado (rango o nivel 1) tendrán el valor de  $n_p = 0$ . Ahora, para cada solución  $p$  con  $n_p = 0$  se visita cada miembro  $q$ , presente en su conjunto  $S_p$  y se reduce el valor  $n_p$  en uno. Haciendo esto, si para algún miembro,  $n_p$  se convierte en cero, este se coloca en una lista separada  $Q$ . Estas soluciones pertenecen al segundo frente no dominado (rango o nivel 2). El procedimiento anterior se continúa para la miembro de  $Q$  y el tercer frente no

dominado es identificado. Este proceso se continua hasta que todos los frentes son identificados.

El Pseudocódigo se presenta a continuación en la Tabla A.2:

Tabla A.2. *Pseudocódigo del enfoque fast nondominated sorting*

fast_nondominated_sorting (P)	
$\mathcal{F}_1 = \emptyset$	% Definir variable para guardar los miembros del primer frente
<b>FOR</b> cada $p \in P$	
$S_p = \emptyset$	% Definir variable que guarda conjunto de soluciones dominadas por $p$
$n_p = 0$	% Definir el contador de dominación de $p$
<b>FOR</b> cada $q \in P$	
<b>IF</b> ( $p < q$ )	% Si $p$ domina $q$
$S_p = S_p \cup \{q\}$	% Añadir $q$ al conjunto de soluciones dominadas por $p$
<b>ELSEIF</b> ( $q < p$ )	
$n_p = n_p + 1$	% Incrementar el contador de dominación de $p$
<b>END</b>	
<b>IF</b> $n_p = 0$	
$prank = 1$	% Asignar rango o clase no dominación 1 a la solución $p$
$\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$	% Añadir $p$ al primer frente no dominado
<b>END</b>	
<b>END</b>	
$i = 1$	% Iniciar contador de frentes
<b>WHILE</b> $\mathcal{F}_i \neq \emptyset$	
$Q = \emptyset$	% Definir variable para guardar los miembros del frente $i + 1$
<b>FOR</b> cada $p \in \mathcal{F}_i$	
<b>FOR</b> cada $q \in S_p$	
$n_q = n_q - 1$	% Disminuir el contador de dominación de $q$ en uno
<b>IF</b> $n_q = 0$	
$qrank = i + 1$	% Asignar rango o clase no dominación $i + 1$ a la solución $p$
$Q = Q \cup \{q\}$	% Añadir $q$ al a la variable que guarda miembros del frente $i + 1$
<b>END</b>	
<b>END</b>	
<b>END</b>	
$i = i + 1$	% Aumentar contador de frentes
$\mathcal{F}_i = Q$	% Asignar los miembros respectivos al frente $i$
<b>END</b>	

Nota: Adaptado de K. Deb et al. (2002)

**A.1.1.1. Preservación de la diversidad.** El algoritmo NSGA-II usa el enfoque de comparación crowded-comparision approach para mantener la diversidad entre los miembros de la población. Para describir este enfoque, primero se definen dos conceptos: métrica de estimación de densidad y el operador de comparación crowded-comparision operator.

**Estimación de densidad.** Para obtener un estimado de la densidad de soluciones que rodean una solución particular  $i$  en la población  $P$ , se calcula la distancia promedio de dos puntos (uno a cada lado del punto de interés, es decir punto  $i - 1$  y punto  $i + 1$ ) a lo largo de cada uno de los

objetivos. Esta cantidad  $i_{distance}$  sirve como un estimado del perímetro del cuboide formado usando los puntos vecinos de  $i$  como vértices. Este valor es lo que se denomina crowding distance. En la Figura A.3 el cuboide que representa el valor de crowding distance de la solución  $i$  en su frente (marcada en círculo rojo) se representa en un recuadro de líneas punteadas.

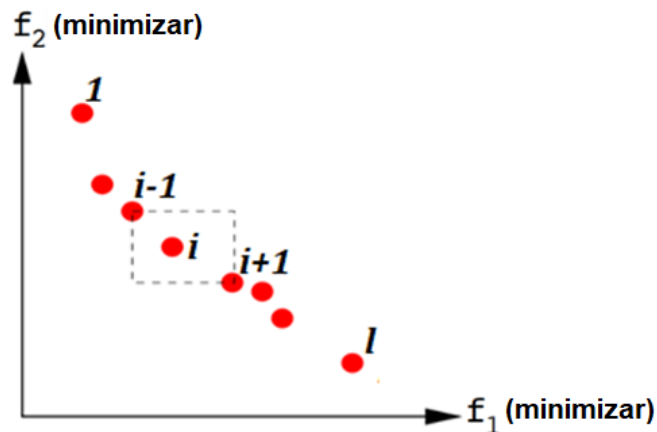


Figura A.3. Crowding distance para un punto  $i$  que hace parte de un frente de Pareto (color rojo)  
Fuente: Branke et al., (2008).

Para el cálculo del valor de *crowding distance*, es necesario ordenar la población para cada función objetivo en orden de magnitud ascendente. Por lo tanto, para cada función objetivo, a las soluciones de borde (soluciones con los valores menores y mayores de las funciones objetivo (para la Figura A.3 son las soluciones 1 y  $l$ ) se les asigna un valor de distancia igual a infinito, para asegurar su preferencia sobre las soluciones intermedias. Para las soluciones intermedias se les asigna un valor de distancia igual a la diferencia absoluta normalizada de los valores de la función objetivo de dos soluciones adyacentes. Este cálculo se realiza para cada una de las funciones objetivo. El valor de *crowding distance* es la suma de las distancias individuales que corresponden a cada una de las funciones objetivo. Cada objetivo se normaliza antes de calcular el valor de *crowding distance* (K. Deb et al., 2002). El pseudocódigo para la asignación del valor *crowding distance* en cada una de las soluciones un frente no dominado arbitrario  $\mathcal{F}$  se presenta en la Tabla A.3.

Tabla A.3. *Pseudocódigo crowding distance*

crowding_distance_assignment ( $\mathcal{F}$ )	
$l = \ \mathcal{F}\ $	% Número de soluciones en el frente $\mathcal{F}$
<b>FOR</b> cada $i \in \mathcal{F}$	
$d_i = 0$	% Inicializar <i>crowding distance</i> para cada solución $i$
<b>END</b>	
<b>FOR</b> cada objetivo $m$	% Para cada objetivo $m = 1, 2, 3, \dots, M$
$J = \text{sort}(\mathcal{F}, m)$	% Índice de las soluciones del frente $\mathcal{F}$ en orden ascendente para objetivo $m$
$d_{J(1)} = d_{J(l)} = \infty$	% <i>crowding distance</i> de las soluciones extremas (1 y $l$ ) para el objetivo $m$
<b>FOR</b> $i = 2$ hasta $l - 1$	
$d_{J(i)} = d_{J(i)} + \frac{f_m^{J(i+1)} - f_m^{J(i-1)}}{f_m^{\max} - f_m^{\min}}$	% <i>crowding distance</i> de las soluciones intermedias
<b>END</b>	
<b>END</b>	

Nota: Adaptado de K. Deb et al. (2002)

Donde  $f_m^{J(i+1)}$  se refiere al valor de la función objetivo  $m$  para solución con  $J(i+1)$  de  $\mathcal{F}$ .  $f_m^{\max}$  y  $f_m^{\min}$  corresponden al valor máximo y mínimo de la función objetivo  $m$  respectivamente. Cuando se ha asignado el valor de *crowding distance* a todas las soluciones del frente  $\mathcal{F}$ , es posible comparar dos soluciones. Una solución con un valor menor de esta distancia, indica que está rodeada por un mayor de soluciones (K. Deb et al., 2002).

**Crowded tournament selection operator.** El operador de comparación crowded comparison operator ( $\prec_n$ ) compara dos soluciones y retorna el ganador del torneo (comparación). Además, guía el proceso de selección en varias etapas del algoritmo NSGA-II hacia un conjunto de soluciones óptimas de Pareto uniformemente distribuidas. El operador asume que cada individuo  $i$  de la población tiene dos atributos (K. Deb et al., 2002):

- Rango o clase de no dominación ( $i_{rank}$  o  $r_i$ )
- Crowding distance ( $i_{distance}$  o  $d_i$ )

Usando *crowded tournament selection operator* una solución  $i$  gana un torneo a una solución  $j$  si alguna de las siguientes condiciones es verdadera:

- Si la solución  $i$  tiene un mejor rango que  $j$ , es decir  $r_i < r_j$

- Si tienen el mismo rango, pero la solución  $i$  tiene un mejor valor de *crowding distance* que la solución  $j$ , es decir  $r_i = r_j$  y  $d_i > d_j$ .

La primera condición asegura que la solución seleccionada se encuentre en un mejor frente no dominado. La segunda condición resuelve el empate entre dos soluciones que se encuentran en el mismo frente no dominado, decidiendo con el valor de *crowding distance*. En este caso, se prefiere la solución que se encuentra en la región menos poblada del frente, es decir la solución con mayor valor de *crowding distance* (K. Deb et al., 2002).

## A.2. Algoritmo MOPSO (Multiobjective Particle Swarm Optimization)

El pseudocódigo del algoritmo MOPSO se muestra en la Tabla A.4:

Tabla A.4. Pseudocódigo *del algoritmo MOPSO*

MOPSO	
<b>FOR</b> $i = 1$ to $MAX$	
Initialize $POP(i)$	% Iniciar la posición de la población de partículas de tamaño $MAX$
$VEL(i) = 0$	% Iniciar la velocidad de las partículas de la población
Evaluate $POP(i)$	% Evaluar cada partícula
$P_{best}(i) = POP(i)$	% Inicializar memoria con la mejor posición de cada partícula
<b>END</b>	
Initialize $REP$	% Guardar partículas no dominadas en el repositorio usando hipercubos
$ITE = 1$	
<b>WHILE</b> $ITE \leq MAXITE$	
<b>FOR</b> $i = 1$ to $MAX$	
% Actualizar velocidad de las partículas	
$VEL(i) = \omega * VEL(i) + [c_1 * r_1 * (P_{best}(i) - POP(i))] + [c_2 * r_2 * (REP(h) - POP(i))]$	
$POP(i) = POP(i) + VEL(i)$	% Calcular las nuevas posiciones de las partículas
$LB(i) \leq POP(i) \leq UB(i)$	% Mantener las partículas dentro del espacio de búsqueda
Mutate $POP(i)$	% Aplicar operador de mutación
Evaluate $POP(i)$	% Evaluar cada partícula
<b>IF</b> $POP(i) < P_{best}(i)$	
$P_{best}(i) = POP(i)$	% Actualizar mejor posición de la partícula
<b>END</b>	
<b>END</b>	
Update $REP$	% Actualizar el contenido del repositorio
$ITE = ITE + 1$	% Incrementar ciclo
<b>END</b>	

Nota: Adaptado de C. A. Coello Coello & Lechuga (2002)

Cuando se inicializa el repositorio  $REP$  se generan los hipercubos del espacio de búsqueda explorado hasta el momento. Las partículas no dominadas guardadas en el repositorio se localizan usando estos hipercubos como un sistema coordinado, donde las coordenadas de cada partícula están definidas de acuerdo a los valores de la función objetivo.

En la ecuación de actualización de la velocidad,  $\omega$  es el valor de la inercia (toma un valor de 0.4),  $c_1$  es el parámetro cognitivo (valores entre 0 y 4, usualmente se toma 2),  $c_2$  es el parámetro cognitivo (valores entre 0 y 4, usualmente se toma 2).  $r_1$  y  $r_2$  son dos números aleatorios en el rango  $[0,1]$ .  $P_{best}(i)$  es la mejor posición que la partícula  $i$  ha tenido.  $REP(h)$  es un valor que se toma del repositorio, donde el índice  $h$  se selecciona de la siguiente manera: Los hipercubos que contienen más de una partícula se les asigna un valor de fitness igual a dividir cualquier número  $x > 1$  por el número de partículas que contiene. Esto con el fin de disminuir el valor de fitness de los hipercubos que contienen más partículas. De esta forma se aplica un mecanismo de diversidad a las soluciones no dominadas guardadas en el repositorio. Posteriormente se aplica selección por medio del método de la ruleta usando el valor de fitness para seleccionar el hipercubo del cual se va a tomar la correspondiente partícula. Una vez se ha seleccionado el hipercubo, se selecciona de forma aleatoria una partícula dentro de este hipercubo.  $POP(i)$ . Es el valor de la posición actual de la partícula  $i$

Cuando se actualizar el contenido del repositorio  $REP$ , también se actualiza la representación geográfica de las partículas dentro de los hipercubos. Esta actualización consiste en insertar todas soluciones actuales no dominadas de las partículas en el repositorio. Cualquier solución dominada en el repositorio se elimina en el proceso. Como el tamaño del repositorio es limitado, cuando se complete su capacidad máxima se aplica un segundo criterio para guardar una partícula: las partículas que se encuentren en regiones menos pobladas del espacio objetivo tienen una mayor prioridad sobre las partículas que se encuentran en regiones más pobladas.

Las soluciones óptimas de Pareto corresponden a las soluciones guardadas en el repositorio  $REP$  cuando se ha cumplido el número máximo de iteraciones  $MAXITE$ .

**A.2.1. Descripción del repositorio.** La principal función del repositorio externo (o archivo) **REP** es mantener un registro histórico de las soluciones no dominadas encontradas durante el proceso de búsqueda del algoritmo MOPSO. Este repositorio consiste de dos partes: El controlador del archivo y la grilla adaptativa (Coello et al., 2004).

**A.2.1.1. Controlador del archivo.** Su función es decidir si cierta solución debe ser añadida al archivo o no. Las soluciones no dominadas encontradas en cada iteración de la población de partículas se comparan (una por una) con respecto a las soluciones que se encuentran guardadas en el repositorio **REP**, el cual al iniciar el proceso de búsqueda se encuentra vacío. Si este archivo externo (repositorio) se encuentra vacío, entonces la nueva solución es aceptada (ver Figura A.4 caso 1). Si la nueva solución es dominada por alguna solución del archivo externo, entonces la nueva solución se descarta (ver Figura A.4 caso 2). De otra manera si ninguna de las soluciones del archivo externo domina la nueva solución, entonces la nueva solución se añade al archivo externo (ver Figura A.4 caso 3). Si en el archivo hay soluciones dominadas por la nueva solución, estas soluciones se eliminan del repositorio y se añade la nueva solución (ver Figura A.4 caso 4). Finalmente, si la población de soluciones del archivo externo ha alcanzado su capacidad máxima, entonces se activa la grilla adaptativa (Coello et al., 2004).

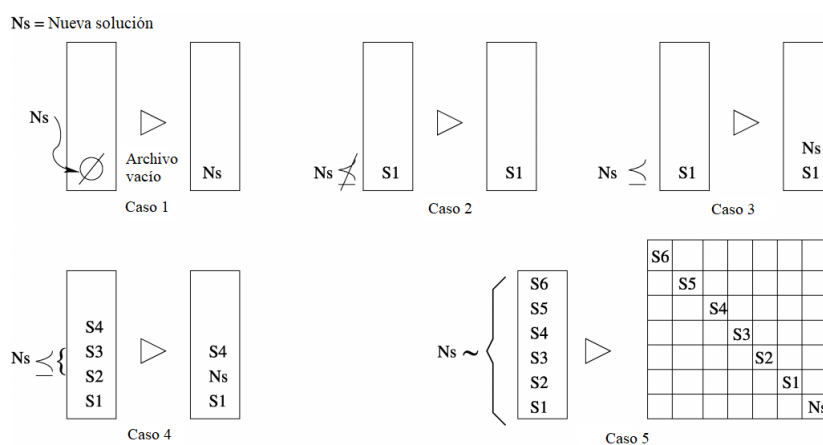


Figura A.4. Posibles casos para el control del archivo.

Fuente: Adaptado de Coello et al. (2004)

**A.2.1.2. Grilla adaptativa.** Para producir frentes de Pareto uniformemente distribuidos, el algoritmo MOPSO usa una variación de la grilla adaptativa (adaptive grid) usada por el algoritmo PAES (ver Figura A.5).

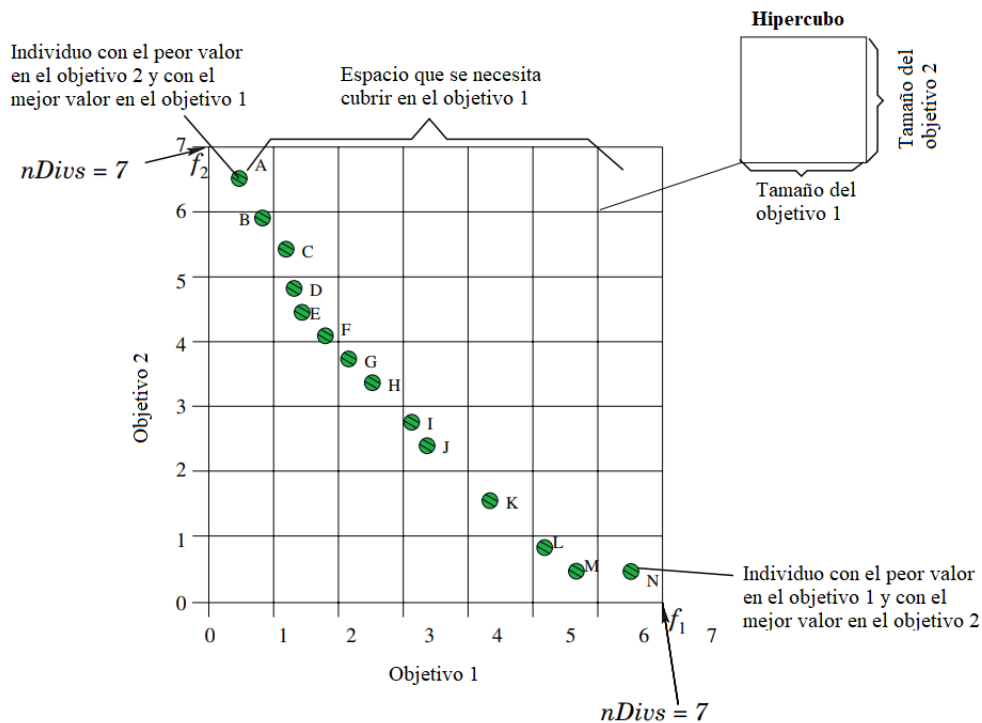


Figura A.5. Representación gráfica de la grilla adaptativa  
Fuente: C. Coello Coello, Lamont, & van Veldhuizen (2007)

La grilla adaptativa es un espacio formado por hipercubos, los cuales tienen tantas componentes como funciones objetivo existen. Cada hipercubo se puede interpretar como una región geográfica que contiene un número de soluciones. La grilla adaptativa se usa para distribuir de manera uniforme la mayor cantidad posible de hipercubos. Para lograrlo, es necesario proporcionar cierta información (número de subdivisiones de la grilla  $nDivs$ ), la cual depende del problema que se está resolviendo (Coello et al., 2004).

En la grilla adaptativa, el espacio de las funciones objetivo se divide en hipercubos, en los cuales se insertan las nuevas soluciones (ver Figura A.6). Si los nuevos individuos insertados en el archivo se encuentran por fuera de los límites actuales de la grilla, esta debe ser recalculada y

cada individuo que se encuentra en la grilla debe ser reubicado nuevamente de acuerdo a los nuevos límites (ver Figura A.7) (Coello et al., 2004).

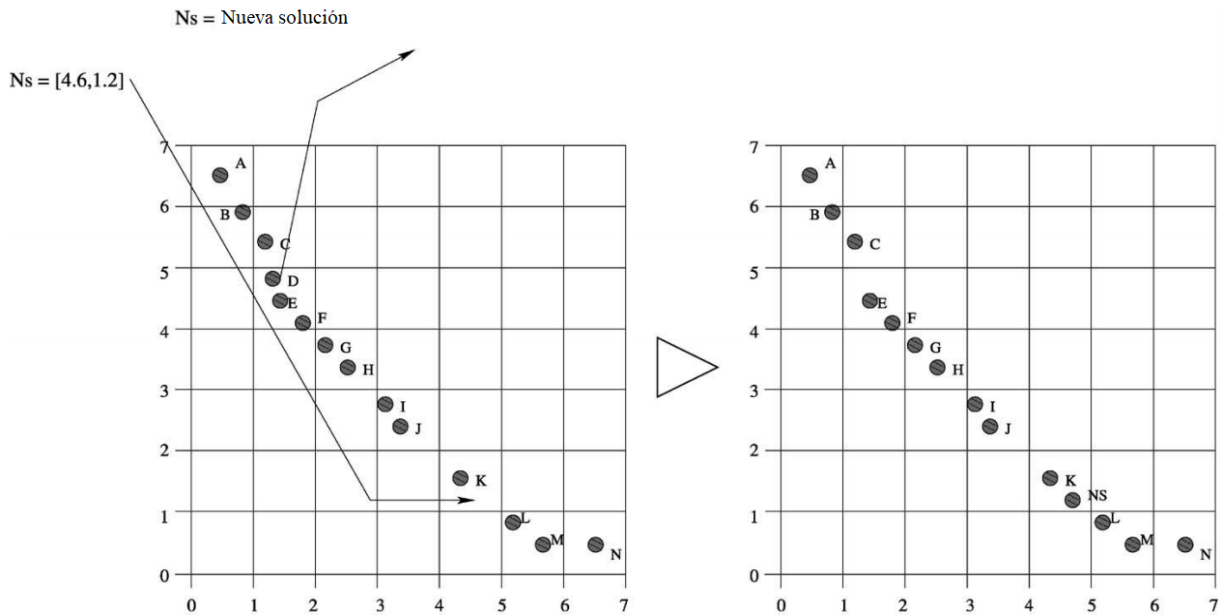


Figura A.6. Representación gráfica de la inserción de una nueva solución en la grilla adaptativa, cuando la nueva solución se encuentra dentro de los límites actuales de la grilla  
Fuente: Coello et al. (2004)

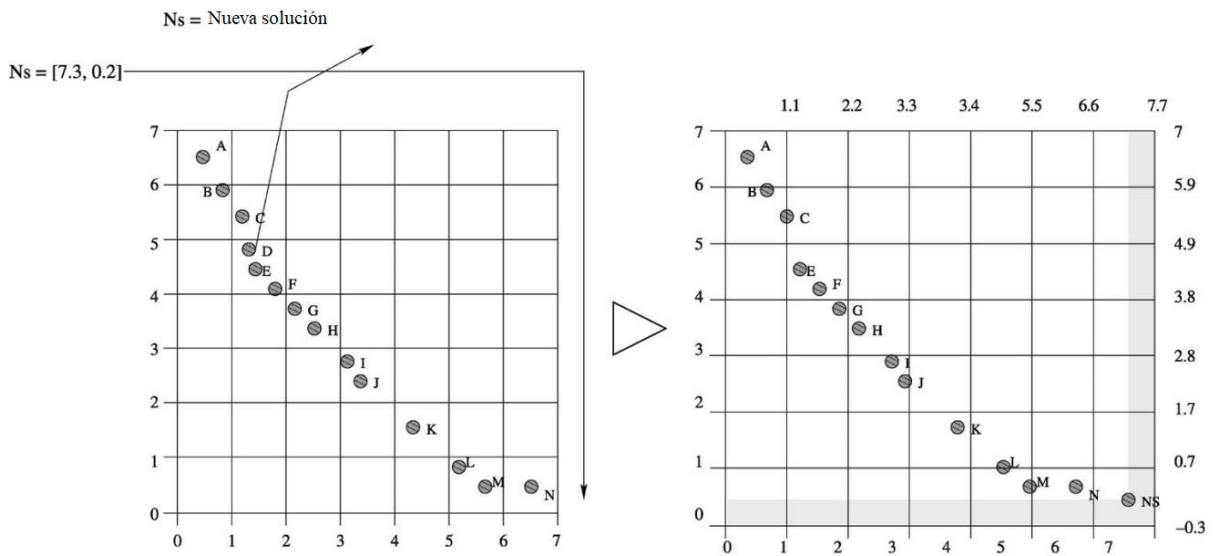


Figura A.7. Representación gráfica de la inserción de una nueva solución en la grilla adaptativa, cuando la nueva solución se encuentra por fuera de los límites actuales de la grilla  
Fuente: Coello et al. (2004)

**A.2.2. Uso de un operador de mutación.** El algoritmo PSO es conocido por su alta velocidad de convergencia. Sin embargo, esta propiedad puede ser perjudicial en el contexto de optimización multiobjetivo, ya que el algoritmo podría converger a un frente de Pareto local de soluciones no dominadas (el equivalente a un óptimo local en optimización global de un solo objetivo). Para solucionar este problema, Coello et al. (2004) desarrollaron un operador de mutación con el cual se intenta explorar el espacio de búsqueda con todas las partículas, al inicio de las iteraciones. Después, a medida que aumenta el número de iteraciones se disminuye rápidamente el número de partículas afectadas por el operador de mutación (ver Figura A.8).

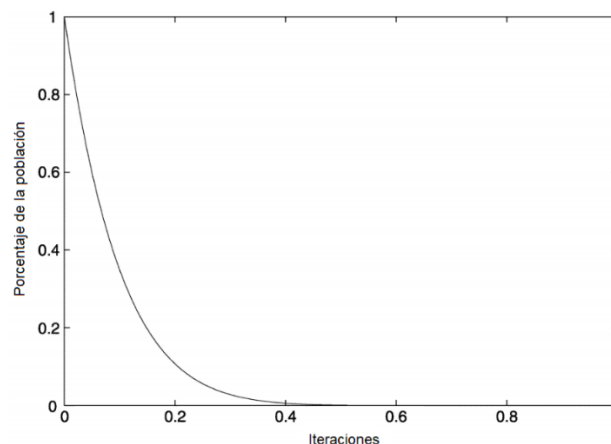


Figura A.8. Comportamiento del operador de mutación.  
Fuente: Coello et al. (2004)

En el eje horizontal, se muestra el número de iteraciones ejecutadas por el algoritmo MOPSO, expresado como un porcentaje. En el eje vertical se muestra el porcentaje de la población que es afectada por el operador de mutación

El operador de mutación se aplica no solo a las partículas de la población, sino también al rango de las variables de decisión del problema de optimización. Lo que hace es cubrir el rango de cada variable de decisión al inicio de la búsqueda, y posteriormente se disminuye este rango usando una función no lineal. En la Figura A.8 se puede observar que al comienzo de la búsqueda todas las partículas de la población están afectadas por el operador de mutación (al igual que el rango

completo de las variables de decisión). Lo que se intenta es producir un comportamiento del algoritmo altamente “explorativo”. A medida que el número de iteraciones avanza, el efecto del operador de mutación disminuye. El pseudocódigo del operador de mutación se muestra a continuación en la Tabla A.5:

Tabla A.5. *Operador de mutación del algoritmo MOPSO*

---

<b>Mutation-Operator</b> ( <i>particle, dims, current_gen, tot_ite, mut_rate</i> )
% <i>particle</i> : partícula a ser mutada
% <i>dims</i> : número de dimensiones o variables de decisión de la partícula a ser mutada
% <i>current_ite</i> : iteración actual
% <i>tot_ite</i> : número total de iteraciones
% <i>mut_rate</i> : tasa de mutación
% <i>rand</i> : número aleatorio entre [0,1]
% <i>which_dim</i> : variable de decisión a ser mutada
% UB: vector con el límite superior de las variables de decisión
% LB: vector con el límite inferior de las variables de decisión
% <i>mutrange</i> : rango de mutación de la variable a ser mutada
% ub: nuevo límite superior de la variable a ser mutada
% lb: nuevo límite inferior de la variable a ser mutada
<b>IF</b> $rand < \left(1 - \frac{current\_ite}{tot\_ite}\right)^{\frac{5}{mut\_rate}}$
<i>which_dim</i> = <i>random</i> (1, <i>dims</i> ) % Seleccionar aleatoriamente la variable a ser mutada
<i>mut_range</i> = (UB( <i>which_dim</i> ) - LB( <i>which_dim</i> )) * $\left(1 - \frac{current\_ite}{tot\_ite}\right)^{\frac{5}{mut\_rate}}$
ub = <i>particle</i> ( <i>which_dim</i> ) + <i>mut_range</i>
lb = <i>particle</i> ( <i>which_dim</i> ) - <i>mut_range</i>
<b>IF</b> lb < LB( <i>which_dim</i> )
lb = LB( <i>which_dim</i> )
<b>END</b>
<b>IF</b> ub > UB( <i>which_dim</i> )
ub = UB( <i>which_dim</i> )
<b>END</b>
<i>particle</i> ( <i>which_dim</i> ) = lb + <i>rand</i> * (ub - lb) % Calcular el valor de la variable mutada usando los nuevos límites
<b>END</b>

---

### A.3. Algoritmo AMOSA (Archived Multiobjective Simulated Annealing)

El pseudocódigo del algoritmo AMOSA se presenta en la Tabla A.6:

Tabla A.6. Pseudocódigo *del algoritmo AMOSA*

Algoritmo AMOSA
<pre> Definir: <math>T_{max}</math>, <math>T_{min}</math>, <math>HL</math>, <math>SL</math>, <math>iter</math>, <math>\alpha</math>, <math>temp = T_{max}</math> Inicializar el <i>Archivo</i> <math>current\_pt = random(Archivo)</math> % Seleccionar una solución aleatoria del <i>Archivo</i> <b>WHILE</b> (<math>temp &gt; T_{min}</math>)   <b>FOR</b> <math>i = 1</math> hasta <math>iter</math>     <math>new\_pt = perturbar(current\_pt)</math>     % Chequear el estado de dominación entre <math>new\_pt</math> y <math>current\_pt</math>     <b>IF</b> (<math>current\_pt</math> domina <math>new\_pt</math>) % Caso 1       <math>\Delta dom_{avg} = \frac{(\sum_{i=1}^k \Delta dom_{i,new\_pt}) + \Delta dom_{current\_pt,new\_pt}}{k+1}</math> % <math>k</math>: número de soluciones del <i>Archivo</i> que dominan <math>new\_pt</math>, <math>k \geq 0</math>       <math>prob = \frac{1}{1 + \exp(\Delta dom_{avg} * temp)}</math>       Definir <math>new\_pt</math> como <math>current\_pt</math> con una probabilidad = <math>prob</math>     <b>ELSEIF</b> (<math>current\_pt</math> y <math>new\_pt</math> son no dominadas entre sí) % Caso 2       % Chequear el estado de <math>new\_pt</math> con las soluciones del <i>Archivo</i>       <b>IF</b> (<math>new\_pt</math> es dominada por <math>k</math> soluciones del <i>Archivo</i>, <math>k \geq 1</math>) % Caso 2(a)         <math>\Delta dom_{avg} = \frac{(\sum_{i=1}^k \Delta dom_{i,new\_pt})}{k}</math>         <math>prob = \frac{1}{1 + \exp(\Delta dom_{avg} * temp)}</math>         Definir <math>new\_pt</math> como <math>current\_pt</math> con una probabilidad = <math>prob</math>       <b>ELSEIF</b> (<math>new\_pt</math> es no dominada con todas las soluciones del <i>Archivo</i>) % Caso 2(b)         Definir <math>new\_pt</math> como <math>current\_pt</math> y añadir <math>new\_pt</math> al <i>Archivo</i>         <b>IF</b> (<math>tamaño\ del\ Archivo &gt; SL</math>)           Aplicar <i>clustering</i> al <i>Archivo</i> y reducir su tamaño a <math>HL</math>         <b>END</b>       <b>ELSEIF</b> (<math>new\_pt</math> domina <math>k</math> soluciones del <i>Archivo</i>, <math>k \geq 1</math>) % Caso 2(c)         Definir <math>new\_pt</math> como <math>current\_pt</math> y añadir <math>new\_pt</math> al <i>Archivo</i>         Remover todas las soluciones <math>k</math> dominadas del <i>Archivo</i>       <b>END</b>     <b>ELSEIF</b> (<math>new\_pt</math> dominada <math>current\_pt</math>) % Caso 3       % Chequear el estado de <math>new\_pt</math> con las soluciones del <i>Archivo</i>       <b>IF</b> (<math>new\_pt</math> es dominada por <math>k</math> soluciones del <i>archivo</i>, <math>k \geq 1</math>) % Caso 3(a)         <math>\Delta dom_{min} = \text{valor mínimo de la diferencia de las cantidades de dominación entre } new\_pt \text{ y cada } k</math>         <math>prob = \frac{1}{1 + \exp(-\Delta dom_{min})}</math>         Definir el punto de <i>Archivo</i> para el cual corresponde el valor de <math>\Delta dom_{min}</math> como <math>current\_pt</math> con una probabilidad = <math>prob</math>, de lo contrario definir <math>new\_pt</math> como <math>current\_pt</math>       <b>ELSEIF</b> (<math>new\_pt</math> es no dominada con todas las soluciones del <i>Archivo</i>) % Caso 3(b)         Definir <math>new\_pt</math> como <math>current\_pt</math> y añadir <math>new\_pt</math> al <i>Archivo</i>         Remover <math>current\_pt</math> del <i>Archivo</i>, si se encuentra ahí       <b>IF</b> (<math>tamaño\ del\ Archivo &gt; SL</math>)         Aplicar <i>clustering</i> al <i>Archivo</i> y reducir su tamaño a <math>HL</math>       <b>END</b>     <b>ELSEIF</b> (<math>new\_pt</math> domina <math>k</math> soluciones del <i>Archivo</i>, <math>k \geq 1</math>) % Caso 3(c)       Definir <math>new\_pt</math> como <math>current\_pt</math> y añadir <math>new\_pt</math> al <i>archivo</i>       Remover todas las soluciones <math>k</math> dominadas del <i>Archivo</i>     <b>END</b>   <b>END</b>   <math>temp = \alpha * temp</math> <b>END</b> <b>IF</b> (<math>tamaño\ del\ Archivo &gt; SL</math>)   Aplicar <i>clustering</i> al <i>archivo</i> y reducir su tamaño a <math>HL</math> <b>END</b> </pre>

Nota: Adaptado de Bandyopadhyay et al. (2008)

Las soluciones óptimas de Pareto corresponden a las soluciones guardadas en el *Archivo* cuando se ha alcanzado la temperatura  $T_{min}$ .

Los parámetros generales del algoritmo AMOSA son:

- *HL*: Máximo tamaño del Archivo al final del proceso de optimización. Su tamaño es igual al número de soluciones óptimas no dominadas requeridas por el usuario.
- *SL*: Máximo tamaño del Archivo hasta el cual puede ser llenado, antes de aplicar una técnica de *clustering* para reducir su tamaño a *HL*.
- $T_{max}$ : Temperatura máxima (temperatura inicial)
- $T_{min}$ : Temperatura mínima (temperatura final)
- *temp*: Temperatura actual
- *iter*: Número de iteraciones en cada temperatura
- $\alpha$ : tasa de enfriamiento (parámetro del Simulated Annealing SA)

La explicación de cada uno de los pasos del algoritmo es la siguiente (Bandyopadhyay et al., 2008):

**A.3.1. Inicialización del Archivo.** El algoritmo comienza con la inicialización ( $\gamma \times SL$ ) soluciones, donde ( $\gamma > 1$ ). Cada una de estas soluciones se refina usando una técnica de *Hill Climbing*, aceptando una nueva solución solo si esta domina la solución anterior. Este proceso se realiza por algunas iteraciones. Después de esto, las soluciones no dominadas obtenidas de este proceso se añaden al **Archivo**, hasta un límite máximo de **HL**. Si el número de soluciones excede el valor de **HL**, se usa una técnica de *clustering* (*Single Linkage Algorithm*) para reducir el tamaño hasta **HL**. Lo anterior significa que el **Archivo**, inicialmente tiene un número máximo de soluciones no dominadas igual a **HL**. En la fase de inicialización es posible obtener un **Archivo** de tamaño uno, es decir una sola solución no dominada, caso en el cual se generan otras nuevas soluciones dominadas por la única solución existente en el **Archivo**.

**A.3.2. Técnica de *clustering*.** Como técnica de *clustering* el algoritmo AMOSA usa el método *Single Linkage Algorithm*. Esta técnica permite agrupar las soluciones del archivo en clústeres. Se permite que el tamaño del Archivo pueda incrementarse hasta **SL** ( $> \mathbf{HL}$ ) para que no sea necesario el uso constante de la función *clustering* y favorecer la formación de soluciones no dominadas más diversas (una mejor distribución uniforme en el frente de Pareto). Si el valor de **SL** es superado, las soluciones se agrupan en **HL** clusters. Las **HL** soluciones se obtienen seleccionando la solución más representativa de cada clúster.

**A.3.3. Cantidad de dominación.** El algoritmo AMOSA usa el concepto de cantidad de dominación para calcular la probabilidad de aceptación de una nueva solución **new\_pt** en el **Archivo**. Dadas dos soluciones **a** y **b** la cantidad de dominación se define con la Ecuación (A.1):

$$\Delta dom_{a,b} = \prod_{i=1}^M \frac{|f_i(a) - f_i(b)|}{R_i}, \quad f_i(a) \neq f_i(b) \quad (\text{A.1})$$

Donde  $M$  es el número de objetivos y  $R_i$  es el rango del objetivo  $i$ . El rango de un objetivo puede ser desconocido a priori. En este caso, las soluciones presentes en el *Archivo*, la nueva solución "new\_pt" y la solución inicial o actual "current\_pt" se usan para calcular el rango de objetivo. En la Figura A.9 se muestra de forma gráfica el concepto de  $\Delta dom_{a,b}$  para dos funciones objetivo  $f_1$  y  $f_2$ .

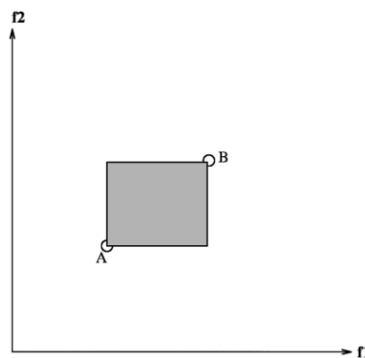


Figura A.9. Cantidad total de dominación (área del rectángulo sombreado) entre dos soluciones A y B  
Fuente: (Bandyopadhyay et al., 2008)

**A.3.4. Proceso principal del algoritmo AMOSA.** Inicialmente, una de las soluciones del *Archivo* se asigna (de forma aleatoria) como solución actual o inicial "*current\_pt*", a una temperatura  $temp = T_{max}$ . La solución "*current\_pt*" se perturba (un pequeño cambio de localización en espacio de las variables de decisión) para generar una nueva solución denominada "*new\_pt*". Se chequea el estado de dominación de "*new\_pt*" con respecto a "*current\_pt*" y las soluciones presentes en el *Archivo*. Con base en el estado de dominación entre "*current\_pt*" y "*new\_pt*" pueden presentarse tres casos:

**A.3.4.1. Caso 1.** "*current\_pt*" domina "*new\_pt*" y  $k$  puntos ( $k \geq 0$ ) del *Archivo* dominan "*new\_pt*" (ver Figura A.10)

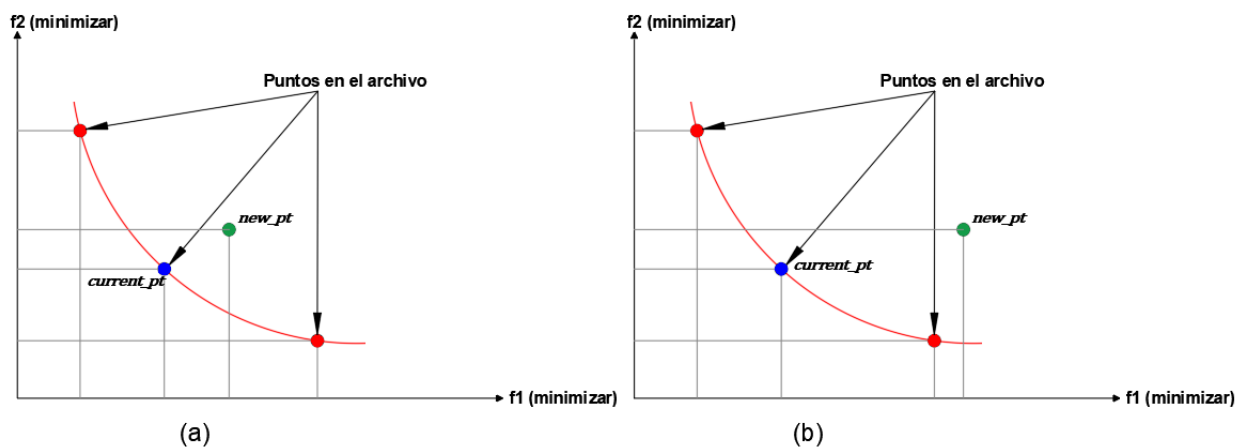


Figura A.10. Estado de dominación caso 1  
Fuente: Adaptado de Bandyopadhyay et al. (2008))

En este caso "*new\_pt*" es seleccionado como "*current\_pt*" con un valor de probabilidad definido en la Ecuación (A.2):

$$prob = \frac{1}{1 + \exp(\Delta dom_{avg} * temp)} \quad (A.2)$$

Donde  $\Delta dom_{avg}$  es el promedio de la cantidad de dominación de los puntos que dominan "*new\_pt*", es decir los  $k$  puntos del *Archivo* y "*current\_pt*" y se calcula con la Ecuación (A.3).

A medida que  $k$  se incrementa el valor de  $\Delta dom_{avg}$  aumenta, disminuyendo la probabilidad de selección.

$$\Delta dom_{avg} = \frac{(\sum_{i=1}^k \Delta dom_{i,new\_pt}) + \Delta dom_{current\_pt,new\_pt}}{k + 1} \quad (A.3)$$

Cuando  $k = 0$ , "current\_pt" se encuentra en el archivo. Si  $k \geq 1$  "current\_pt" puede o no estar en el *Archivo*.

**A.3.4.2. Caso 2.** "current\_pt" y "new\_pt" son no dominadas entre sí. Dependiendo del estado de dominación de "new\_pt" con las soluciones del *Archivo*, se presentan tres situaciones:

- Caso 2(a): new\_pt es dominado por  $k$  puntos ( $k \geq 1$ ) del *Archivo* (ver Figura A.11).

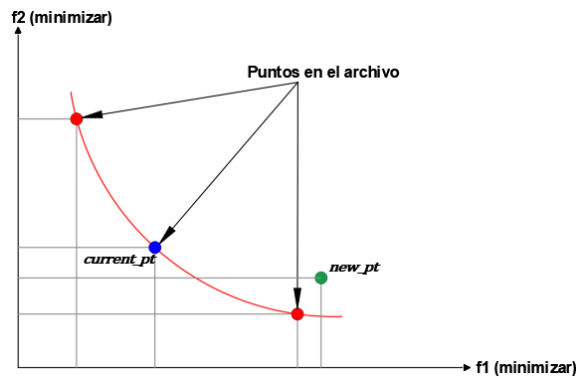


Figura A.11. Estado de dominación caso 2(a)  
Fuente: Adaptado de Bandyopadhyay et al. (2008)

En este caso "new\_pt" es seleccionado como "current\_pt" con un valor de probabilidad calculada con la Ecuación (A.4):

$$prob = \frac{1}{1 + \exp(\Delta dom_{avg} * temp)} \quad (A.4)$$

Donde  $\Delta dom_{avg}$  es el promedio de la cantidad de dominación de los puntos que dominan "new\_pt", es decir los  $k$  puntos del *Archivo* y se calcula con la Ecuación (A.5).

$$\Delta dom_{avg} = \frac{\sum_{i=1}^k \Delta dom_{i,new\_pt}}{k} \quad (A.5)$$

"current\_pt" puede o no estar en el *Archivo*.

- Caso 2b: *new\_pt* es no dominada con respecto a todos los puntos del *Archivo* (ver Figura A.12).

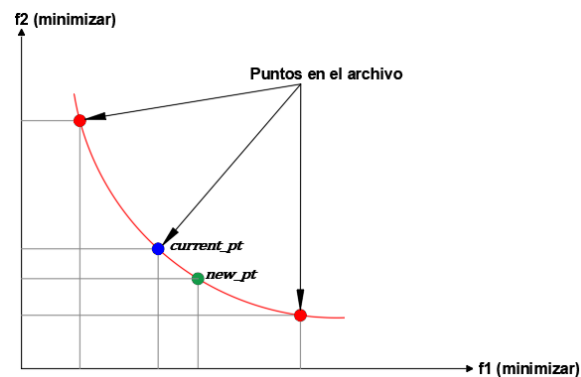


Figura A.12. Estado de dominación caso 2(b)  
Fuente: Adaptado de Bandyopadhyay et al. (2008)

En esta situación, "*new\_pt*" es seleccionado como "*current\_pt*" y se añade al *Archivo*. Si el tamaño del *Archivo* es mayor a *SL*, se aplica *clustering* para reducir el tamaño a *HL*.

- Caso 2(c): "*new\_pt*" domina  $k$  ( $k \geq 1$ ) puntos del archivo (ver Figura A.13)

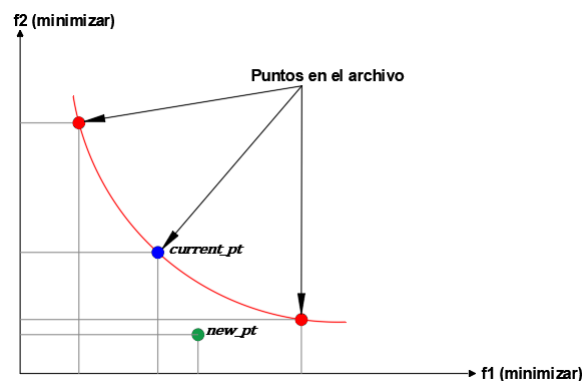


Figura A.13. Estado de dominación caso 2(c)  
Fuente: Adaptado de Bandyopadhyay et al. (2008)

En esta situación, "*new\_pt*" es seleccionado como "*current\_pt*" y se añade al *Archivo*. Todos los puntos  $k$  dominados se eliminan del *Archivo*. "*current\_pt*" puede o no estar en el *Archivo*.

**A.3.4.3. Caso 3.** "*new\_pt*" domina "*current\_pt*". Dependiendo del estado de dominación de "*new\_pt*" con las soluciones del *Archivo*, se presentan tres situaciones:

- Caso 3(a): "*new\_pt*" es dominado por  $k$  puntos ( $k \geq 1$ ) del *Archivo* (ver Figura A.14)

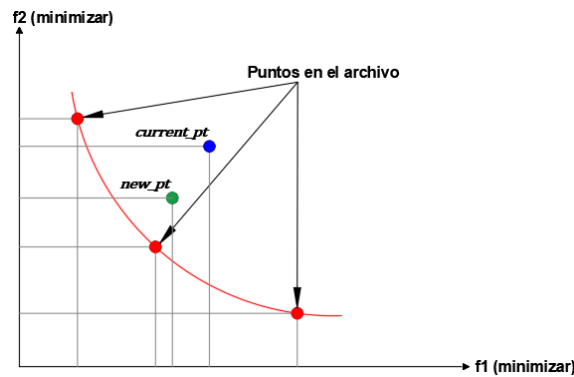


Figura A.14. Estado de dominación caso 3(a)  
Fuente: Adaptado de Bandyopadhyay et al. (2008)

Esta situación se genera sólo si "current\_pt" no es un miembro del *Archivo*. Se calcula el mínimo de la diferencia de la cantidad de dominación  $\Delta dom_{min}$  entre "new\_pt" y los  $k$  puntos. El punto del archivo que corresponde a la mínima diferencia, se selecciona como "current\_pt" con un valor de probabilidad dado por la Ecuación (A.6):

$$prob = \frac{1}{1 + \exp(-\Delta dom_{min})} \quad (A.6)$$

De otra manera, "new\_pt" es seleccionado como "current\_pt".

- Caso 3(b): "new\_pt" es no dominada con respecto a todos los puntos del *Archivo*, excepto "current\_pt" si este pertenece al *Archivo* (ver Figura A.15).

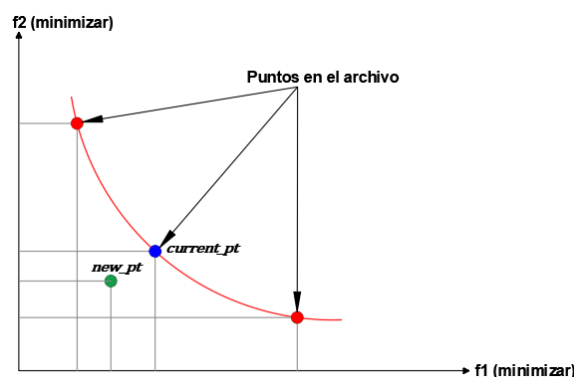


Figura A.15. Estado de dominación caso 3(b)  
Fuente: Adaptado de Bandyopadhyay et al. (2008)

En esta situación, "new\_pt" es seleccionado como "current\_pt" y se añade al *Archivo*. Si "current\_pt" se encuentra en el *Archivo* es removida. De otra manera si el tamaño del *Archivo*

es mayor a  $SL$ , se aplica *clustering* para reducir el tamaño a  $HL$ . "*current\_pt*" puede o no estar en el *Archivo*.

- Caso 3(c). "*new\_pt*" domina  $k$  ( $k \geq 1$ ) puntos del *Archivo* (ver Figura A.16)

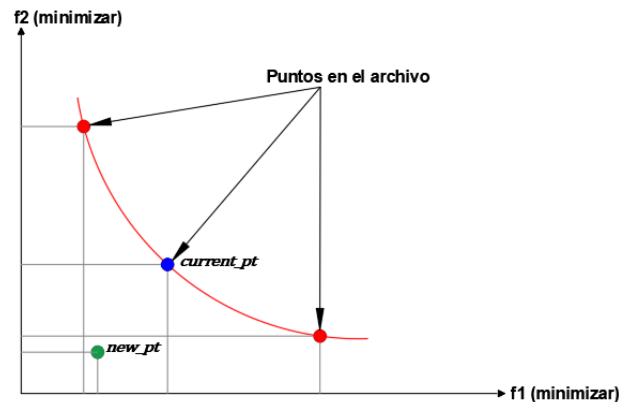


Figura A.16. Estado de dominación caso 3(c)  
Fuente: Adaptado de Bandyopadhyay et al. (2008)

En esta situación "*new\_pt*" es seleccionado como "*current\_pt*" y se añade al *Archivo*. Todos los puntos  $k$  dominados se eliminan del *Archivo*. "*current\_pt*" puede o no estar en el *Archivo*.

El proceso descrito anteriormente se repite "*iter*" veces para cada temperatura "*temp*". La temperatura es reducida usando  $\alpha * temp$ , (donde  $\alpha$  es la tasa de enfriamiento), hasta alcanzar la temperatura mínima  $T_{min}$ .

## Apéndice B. Trayectorias de esfuerzo

### B.1. Definición del concepto de trayectorias de esfuerzo

Las trayectorias de esfuerzo son las envolventes del campo vectorial determinadas por los esfuerzos principales. Las trayectorias de esfuerzo son tangentes a las direcciones de los esfuerzos principales en cada punto del dominio de diseño. Para el caso de esfuerzo plano ( $\sigma_x, \sigma_y, \tau_{xy}$ ), los esfuerzos principales  $\sigma_1$  y  $\sigma_2$ , dan lugar a dos familias de trayectorias de esfuerzo que forman una red ortogonal compuesta por líneas libres de esfuerzo cortante (Kelly & Tosh, 2000; Oliver Olivella & Agelet de Saracibar, 2017; Pereira & de Almeida, 1994). Estas trayectorias son de gran utilidad para entender las rutas de carga (como las cargas externas se mueven a través de la estructura a los puntos de soporte), proporcionan información relacionada con la forma en la cual el flujo de los esfuerzos principales ocurre en el plano de análisis y están relacionadas con la optimización estructural (Oliver Olivella & Agelet de Saracibar, 2017; Sadd, 2014). Han sido ampliamente usadas en el análisis de experimentos fotoelásticos y son adecuadas para describir el comportamiento de una estructura (Pereira & de Almeida, 1994).

Las dos familias de trayectorias de esfuerzo son (Oliver Olivella & Agelet de Saracibar, 2017):

- Trayectorias de esfuerzo  $\sigma_1$ : Tangentes a la dirección del mayor esfuerzo principal
- Trayectorias de esfuerzo  $\sigma_2$ : Tangentes a la dirección del menor esfuerzo principal

Como ejemplo, en la Figura B.1 y Figura B.2 se muestra la distribución de las trayectorias de esfuerzo para una viga cantiléver con una carga puntual en el extremo libre, para una viga simplemente apoyada con una carga uniformemente distribuida (Goodno & Gere, 2018) y para una viga en forma de muro (Beyer, 2015) respectivamente.

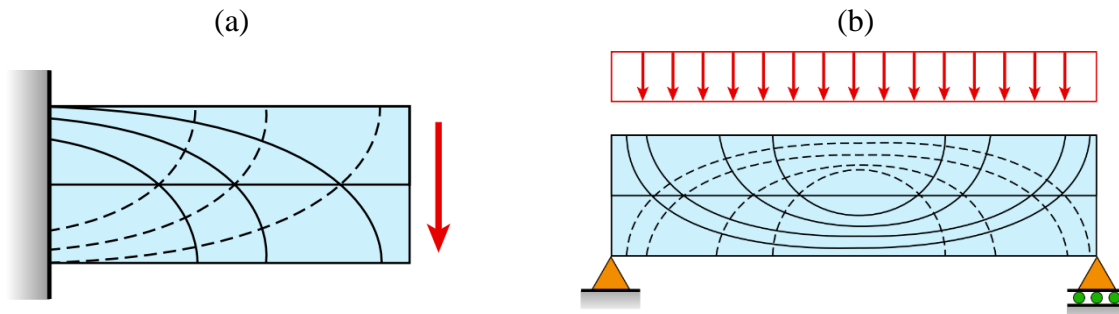


Figura B.1. Trayectorias de esfuerzo para (a) viga cantiléver y (b) viga simplemente apoyada. Trayectorias de esfuerzo  $\sigma_1$  (líneas continuas) y trayectorias de esfuerzo  $\sigma_2$  (líneas punteadas)

Fuente: Goodno & Gere (2018)

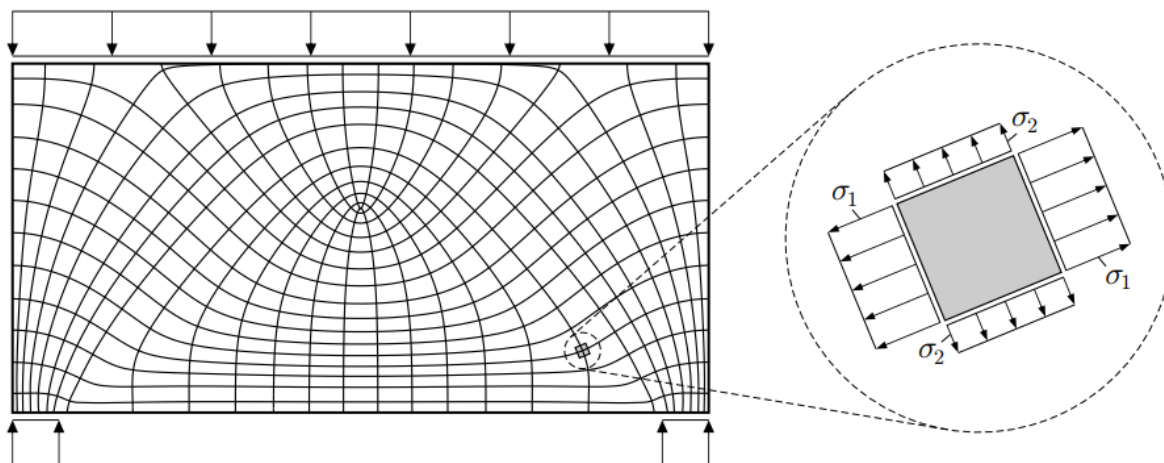


Figura B.2. Trayectorias de esfuerzo para una viga en forma de muro (izquierda). Elemento infinitesimal orientado de acuerdo a la dirección de los esfuerzo principales  $\sigma_1$  y  $\sigma_2$

Fuente: Beyer (2015)

Para el cálculo las trayectorias de esfuerzo se tiene en cuenta las siguientes definiciones (Oliver Olivella & Agelet de Saracibar, 2017):

Un *punto singular* se caracteriza por el estado de esfuerzo definido en la Ecuación (B.1),

$$\sigma_x = \sigma_y \quad \gamma \quad \tau_{xy} = 0 \quad (\text{B.1})$$

Un *punto neutral* es un punto singular caracterizado por el estado de esfuerzo de la Ecuación (B.2),

$$\sigma_x = \sigma_y = \tau_{xy} = 0 \quad (\text{B.2})$$

Todas las direcciones en un punto singular son direcciones de esfuerzos principales. Por consiguiente, las trayectorias de esfuerzo tienden a perder su regularidad en los puntos singulares y pueden generarse cambios bruscos en su dirección.

Si se considera la ecuación general de una trayectoria de esfuerzo como  $y = f(x)$ , y el valor del ángulo formado por la dirección del esfuerzo principal  $\sigma_1$  con respecto al eje  $x$  como  $\varphi_p$  (ver Figura B.3) (Oliver Olivella & Agelet de Saracibar, 2017):

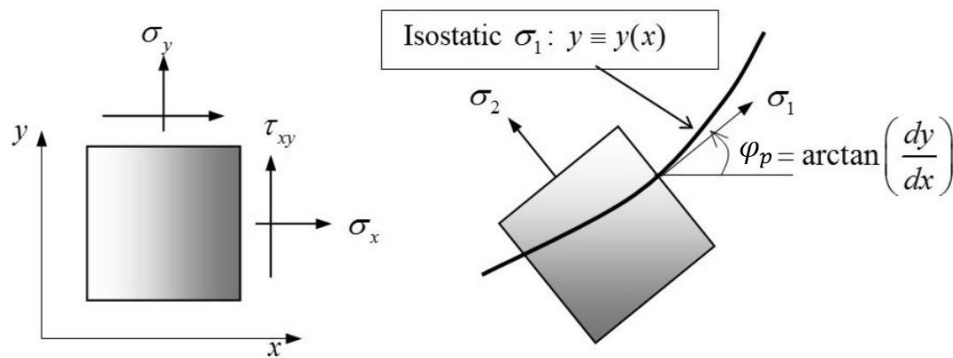


Figura B.3. Determinación de la ecuación diferencial de una trayectoria de esfuerzo  
Fuente: Adaptado de Oliver Olivella & Agelet de Saracibar (2017)

La orientación del ángulo principal  $\varphi_p$ , se puede calcular usando la Ecuación (B.3),

$$\tan 2\varphi_p = \frac{2\tau_{xy}}{\sigma_x - \sigma_y} \quad (\text{B.3})$$

Ahora, para la trayectoria definida como  $y = y(x)$ , su pendiente está dada por la Ecuación (B.4),

$$\tan \varphi_p = \frac{dy}{dx} \quad (\text{B.4})$$

Usando la identidad trigonométrica de la Ecuación (B.5),

$$\tan 2\varphi_p = \frac{2 \tan \varphi_p}{1 - \tan^2 \varphi_p} \quad (\text{B.5})$$

Igualando las Ecuaciones (B.3) y (B.5),

$$\frac{2\tau_{xy}}{\sigma_x - \sigma_y} = \frac{2 \tan \varphi_p}{1 - \tan^2 \varphi_p} \quad (\text{B.6})$$

Reemplazando la Ecuación (B.4) en la Ecuación (B.6),

$$\frac{2\tau_{xy}}{\sigma_x - \sigma_y} = \frac{2 \left( \frac{dy}{dx} \right)}{1 - \left( \frac{dy}{dx} \right)^2} \quad (\text{B.7})$$

Simplificando la Ecuación (B.7) se obtiene la ecuación diferencial de la trayectoria de esfuerzo, para el plano  $xy$ :

$$\frac{dy}{dx} = -\frac{\sigma_x - \sigma_y}{2\tau_{xy}} \pm \sqrt{1 + \left( \frac{\sigma_x - \sigma_y}{2\tau_{xy}} \right)^2} \quad (\text{B.8})$$

La Ecuación diferencial (B.8) puede ser integrada para generar las trayectorias de esfuerzo  $y(x)$ . Para algunos casos especiales la ecuación puede solucionarse de forma analítica, sin embargo, la mayoría de las distribuciones de esfuerzo presentan formas complicadas que requieren del uso de integración numérica u otros métodos numéricos para obtener una aproximación de la solución (Sadd, 2014).

## B.2. Cálculo numérico de las trayectorias de esfuerzo

**B.2.1. Cálculo de los esfuerzos principales.** Para obtener las direcciones principales de esfuerzo 1 y 2 ( $\varphi_{p1}$  y  $\varphi_{p2}$ ) (Figura B.4 derecha), a partir de las componentes de esfuerzo  $\sigma_x$ ,  $\sigma_y$  y  $\tau_{xy}$  (Figura B.4 izquierda), en un punto, se usan las ecuaciones de transformación de esfuerzo (Russell Charles Hibbeler, 2011).

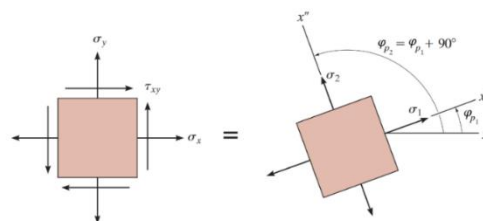


Figura B.4. Esfuerzos principales en el plano para un punto específico  
Fuente: Hibbeler (2011)

A partir de la ecuación de transformación de esfuerzo cortante,

$$\tau_{x'y'} = -\frac{\sigma_x - \sigma_y}{2} \operatorname{sen}2\varphi + \tau_{xy} \operatorname{cos}2\varphi \quad (\text{B.9})$$

Los esfuerzos principales ocurren cuando el valor del esfuerzo cortante  $\tau_{x'y'}$  de la Ecuación (B.9) es cero. Despejando  $\varphi$ , se obtiene la dirección del esfuerzo principal  $\varphi_p$ , dada por la Ecuación (B.10).

$$\begin{aligned} \tan 2\varphi_p &= \frac{2\tau_{xy}}{\sigma_x - \sigma_y} \\ \varphi_p &= \frac{1}{2} \tan^{-1} \left( \frac{2\tau_{xy}}{\sigma_x - \sigma_y} \right) \end{aligned} \quad (\text{B.10})$$

El ángulo principal 1,  $\varphi_{p_1}$ , se calcula de acuerdo a la Ecuación (B.11):

$$\varphi_{p_1} = \begin{cases} 45^\circ & \text{Si } \sigma_x - \sigma_y = 0 \text{ y } 2\tau_{xy} > 0 \\ -45^\circ & \text{Si } \sigma_x - \sigma_y = 0 \text{ y } 2\tau_{xy} < 0 \\ \frac{1}{2} \tan^{-1} \left( \frac{2\tau_{xy}}{\sigma_x - \sigma_y} \right) & \text{Si } \sigma_x - \sigma_y > 0 \\ \frac{1}{2} \tan^{-1} \left( \frac{2\tau_{xy}}{\sigma_x - \sigma_y} \right) + 90^\circ & \text{Si } \sigma_x - \sigma_y < 0 \end{cases} \quad (\text{B.11})$$

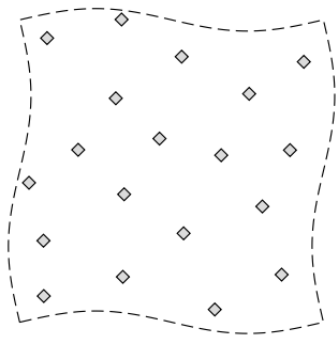
El ángulo principal 2,  $\varphi_{p_2}$ , es perpendicular a  $\varphi_{p_1}$ , es decir

$$\varphi_{p_2} = \varphi_{p_1} + 90^\circ \quad (\text{B.12})$$

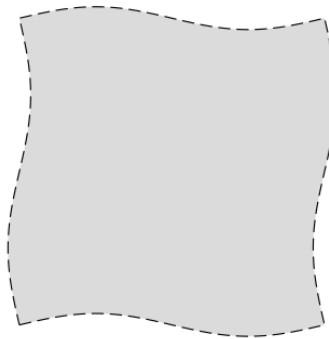
A continuación, se describe en detalle un método numérico (Beyer, 2015) que permite calcular de forma aproximada las trayectorias de esfuerzo a partir de las direcciones principales calculadas con las Ecuaciones (B.11) y (B.12).

**B.2.2. Descripción general del algoritmo de Beyer.** El estado de esfuerzo para un dominio de diseño arbitrario plano y continuo, puede determinarse por medio del método de elementos finitos (ej. con la ayuda de un software como ANSYS® o ABAQUS®). La información del estado de esfuerzos calculada de esta manera, está disponible en puntos discretos distribuidos en todo el

dominio de diseño (Figura B.5(a)). Como las trayectorias de esfuerzo pasan entre los puntos discretos, se hace necesario el uso de un método de interpolación que permita conocer el estado de esfuerzo en un punto arbitrario del espacio continuo de diseño (Figura B.5(b)), a partir de la información inicial del estado de esfuerzo en los puntos discretos (Figura B.5(a)).



(a) Información del estado de esfuerzo como resultado del cálculo numérico en puntos discretos



(b) Conclusión del estado de esfuerzo en todo el dominio de diseño por medio de una aproximación

Figura B.5. Determinación del estado de esfuerzo de un área con base en la interpolación de los puntos conocidos

Fuente: Beyer (2015)

Cuando se determina una trayectoria de esfuerzo, se requiere que ésta presente cambios suaves de curvatura a lo largo de los puntos que conforman dicha trayectoria en una dirección principal en particular (1 o 2), lo cual significa que no deben existir cambios repentinos en su dirección (ejemplo:  $\pm 90^\circ$ ) (Beyer, 2015).

El cálculo de una trayectoria requiere de un punto de partida, a partir del cual se inicia la ruta de la trayectoria. Este punto se define como  $P_0$  y se encuentra en una localización arbitraria del dominio de diseño. Para este punto se pueden identificar dos direcciones principales de esfuerzo, 1 y 2 (Figura B.6(a)), calculadas con las Ecuaciones (B.11) y (B.12). Pueden seguirse ambas direcciones de esfuerzo, lo que resulta en el registro de dos trayectorias. Sin embargo, se debe tener en cuenta el sentido en el cual debe registrarse la trayectoria dentro de la dirección principal seleccionada, ya que para cada dirección principal existen dos opciones: *Para la dirección principal 1 existen dos sentidos de movimiento  $1'$  y  $1''$ . De igual manera para 2 existen  $2'$  y  $2''$ .*

Para explicar la estrategia se selecciona  $1'$ , como sentido de movimiento de la dirección principal 1 (Beyer, 2015).

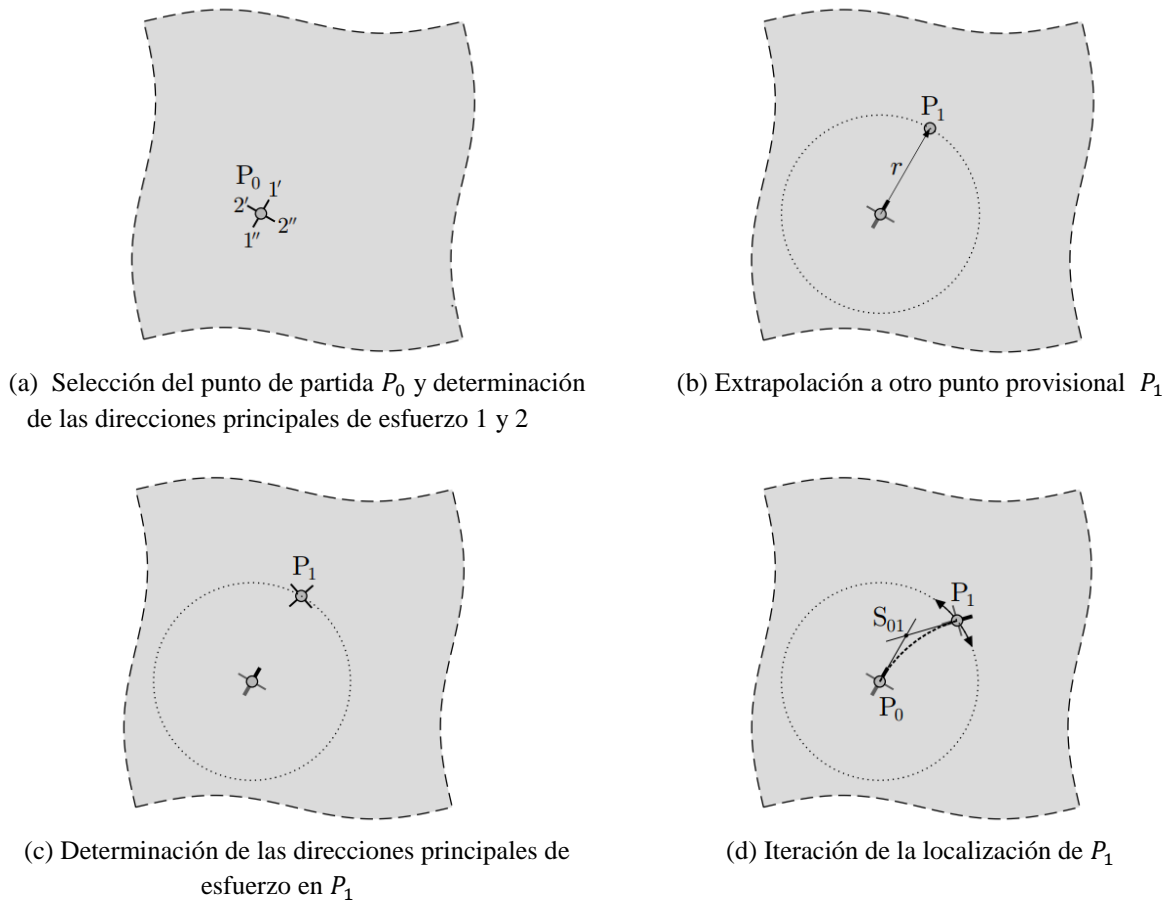


Figura B.6. Representación básica del procedimiento de determinación de las trayectorias de esfuerzo  
Fuente: Beyer (2015)

Para Beyer (2015), el objetivo del algoritmo numérico desarrollado es que las soluciones de las trayectorias de esfuerzo obtenidas, sean los más independientes posible del sistema de coordenadas. Mientras que en los métodos de integración numérica como *Runge-Kutta* o *método de Euler*, el tamaño del paso está relacionado con un sistema de coordenadas fijo, en este algoritmo se introduce un paso “real” “ $r$ ” que indica la distancia real entre los nodos solución de una trayectoria de esfuerzo y no la proyección con respecto a un eje de referencia.

Comenzando desde el punto de partida  $P_0$  (Figura B.6(a)), un punto lejano provisional  $P_1$ , es extrapolado en la dirección inicial seleccionada (Ejemplo, dirección  $1'$ , Figura B.6(b)) con un paso  $r$ . En este punto se determinan sus direcciones principales (Figura B.6(c)), y con base en la posición inicial  $P_0$ , su posición actual  $P_1$  es iterada en los bodes del círculo, de tal manera que los puntos  $P_0$  y  $P_1$  puedan unirse mediante un segmento de arco circular, tal que las tangentes de este arco circular en los puntos  $P_0$  y  $P_1$  correspondan a la respectiva dirección principal de esfuerzo (Figura B.6(d)). El objetivo es alcanzado cuando las distancias  $\overline{P_0S_{01}}$  y  $\overline{S_{01}P_1}$  tienen la misma longitud. Estas distancias corresponden a una continuación rectilínea de la dirección del esfuerzo principal iniciando desde  $P_0$  o  $P_1$  hasta la intersección en  $S_{01}$ .

Finalmente cuando se encuentra la posición final de  $P_1$ , este punto se toma como el nuevo punto de partida  $P_0$ , a partir del cual se dará el siguiente paso. El procedimiento se repite hasta cumplir con un criterio de finalización (ejemplo, cuando se cruza un borde del espacio de diseño). Cuando se cruza el borde, se ha calculado media trayectoria de la dirección 1, en el sentido  $1'$ . La segunda parte se encuentra comenzando la iteración desde el punto original  $P_0$ , pero ahora siguiendo el sentido  $1''$ . Cuando se han calculado las dos mitades, se unen formando la trayectoria total calculada en la dirección principal 1. Para la dirección principal 2 se realiza el mismo procedimiento ejecutado para 1. De esta manera se han encontrado dos líneas de trayectoria de esfuerzo que son ortogonales, una en la dirección del esfuerzo principal 1 y la otra en dirección del esfuerzo principal 2.

### B.2.3. Descripción detallada del algoritmo de Beyer

**B.2.3.1. Interpolación del estado de esfuerzo.** A partir de la información numérica del estado de esfuerzo en los puntos discretos del dominio de diseño (Figura B.5(a)), se requiere interpolar los valores del estado de esfuerzo en un punto arbitrario del dominio de diseño, en función de los valores discretos conocidos, cuando se están calculando las trayectorias de esfuerzo. Para este proceso se pueden usar técnicas de interpolación espacial comúnmente aplicadas en áreas como la Geodesia y la Fotogrametría, para la creación de modelos digitales de terrenos y superficies. Un ejemplo de estas técnicas es el método de interpolación de suma de superficies elementales (definidas por la función central  $k$ ) (Beyer, 2015). También pueden usarse otros métodos populares de interpolación espacial como el método IDW (Inverse Distance Weight) (Longley, Goodchild, Maguire, & Rhind, 2015). A continuación, se explica el método IDW, el cual fue usado en este trabajo para interpolar el estado de esfuerzos:

Método de interpolación IDW (Inverse Weighted Method): Es un método exacto de interpolación espacial, y es el método que se usa con más frecuencia en el análisis de Información Geográfica. En este método se estiman valores desconocidos como promedios ponderados (weighted averages) sobre las mediciones conocidas en los puntos cercanos, dando un mayor peso a los puntos cercanos. Los pasos generales del método son (Longley et al., 2015):

- a) Denotar el punto de interés que se quiere interpolar como  $\mathbf{x}$ , y los puntos donde las medidas fueron tomadas como  $\mathbf{x}_i$  (donde  $i$  va desde 1 hasta  $n$ , si existen  $n$  datos conocidos).
- b) Denotar el valor de interpolación desconocido como  $z(\mathbf{x})$  y las mediciones conocidas como  $z_i$ . Dar a cada uno de los puntos un peso  $w_i$ , el cual será evaluado con base en la distancia  $d_i$  de  $\mathbf{x}_i$  a  $\mathbf{x}$ . La Figura B.7 explica esta notación con un diagrama.

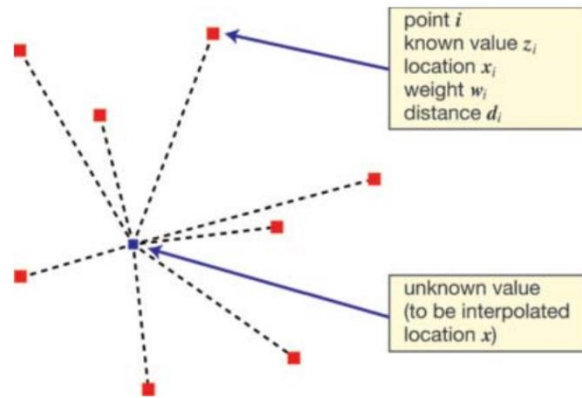


Figura B.7. Notación usada en las ecuaciones que definen la interpolación espacial  
Fuente: Longley et al.(2015)

c) Calcular el promedio ponderado (weighted average) en  $\mathbf{x}$  con la Ecuación (B.13):

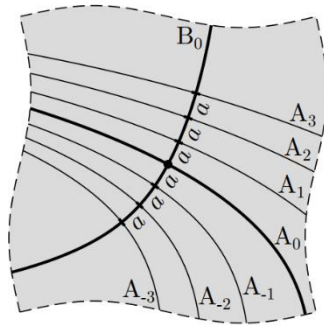
$$z(\mathbf{x}) = \frac{\sum_i w_i z_i}{\sum_i w_i} \quad (\text{B.13})$$

El valor interpolado es un promedio sobre los valores observados, ponderado por los pesos  $w$ 's. Hay varias formas para definir los pesos  $w$ . La opción que más se usa es calcularlos como los cuadrados inversos de las distancias (Longley et al., 2015):

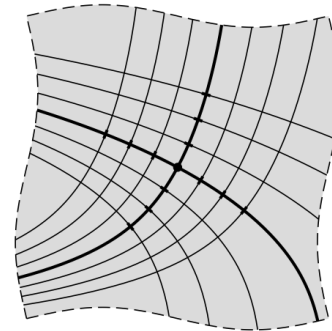
$$w_i = \frac{1}{d_i^2} \quad (\text{B.14})$$

De esta manera el método IDW proporciona una forma sencilla de conocer los valores de un campo continuo en puntos donde no existen mediciones (Longley et al., 2015) .

**B.2.3.2. Selección de los puntos de partida y distancia de control.** Las trayectorias de esfuerzo, teóricamente existen de una forma infinitamente densa, por lo cual, la representación de todas las trayectorias no sería de utilidad, ya que no se revelaría ninguna información. Por esta razón la dirección de una trayectoria solo puede ser reconocida si la trayectoria puede percibirse como una línea. En este sentido solo unas pocas trayectorias son representadas en el espacio de diseño, y su selección debe hacerse de alguna manera. La determinación de los puntos de partida de las trayectorias se hace de forma automática a partir de un único punto de partida inicial (Figura B.8).



(a) Punto de partida inicial (●) y determinación de puntos de partida lejanos a lo largo de la trayectoria del punto de partida inicial



(b) Completar los puntos de partida lejanos en la otra dirección

Figura B.8. Determinación de los puntos de partida de las trayectorias usando distancias constantes  
Fuente: Beyer (2015)

La Figura B.8(a) muestra la sección un componente plano. En el medio, se selecciona un punto de partida. Con base en este punto se determinan las dos trayectorias de esfuerzo marcadas en negrita,  $A_0$  y  $B_0$ . Se especifica una distancia de separación constante denotada con la letra  $a$  (Figura B.8(a)). Esta distancia corresponde a la longitud de arco de un segmento de  $B_0$ , limitado por las trayectorias  $A_i$  y  $A_{i+1}$ . La Figura B.8(b) muestra el resultado de la implementación de este proceso en ambas direcciones principales.

En este procedimiento, la selección del punto de partida inicial es decisivo para los resultados finales de espaciamiento de las trayectorias. La Figura B.9 muestra nuevamente las trayectorias encontradas en la Figura B.8 con una distancia constante  $a$ . En las áreas señaladas de la Figura B.9 (color magenta y cian), puede observarse que la separación inicial constante a lo largo de las trayectorias no se mantiene constante entre sí. De esta manera, la densidad o el número de trayectorias depende de la elección del punto de partida inicial.

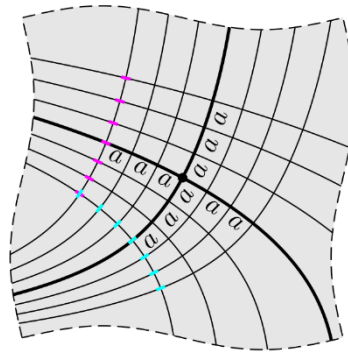


Figura B.9. Cambios en las relaciones de distancia entre trayectorias de esfuerzo

Fuente: Beyer (2015)

**B.2.3.3. Procedimiento de iteración.** Para la iteración del punto  $P_1(x_1, y_1)$  a partir del punto  $P_0(x_0, y_0)$ , se realiza una extrapolación lineal de  $P_0$  con un tamaño paso igual a  $r$  en la dirección principal seleccionada  $\varphi_0$  (de las cuatro opciones disponibles 1', 1'', 2' y 2'', de acuerdo a la Sección B.2.2) de  $P_0$  (ver Figura B.10(a)):

$$\begin{aligned} x_1 &= x_0 + r \cos \varphi_0 \\ y_1 &= y_0 + r \sin \varphi_0 \end{aligned} \quad (\text{B.15})$$

**Determinación de la dirección de seguimiento de ruta más relevante:** En el punto  $P_1$ , la Ecuación (B.10) calcula la dirección del ángulo principal  $\varphi_p$ . En este punto existen dos direcciones principales (1 y 2), y cada una de ellas tiene dos posibles direcciones (1', 1'', 2' y 2'', ver Figura B.6(a)). Estas cuatro posibles rutas del punto  $P_1$  (ver Figura B.6(b)), se calculan inicialmente de acuerdo a la Ecuación (B.16):

$$\varphi_1^I = \varphi_p, \quad \varphi_1^{II} = \varphi_p + 90^\circ, \quad \varphi_1^{III} = \varphi_p + 180^\circ, \quad \varphi_1^{IV} = \varphi_p + 270^\circ \quad (\text{B.16})$$

Teniendo en cuenta la naturaleza continua de la ruta de las trayectorias, de las cuatro opciones posibles de cambio de dirección en el punto  $P_1$ , se selecciona la que tenga una menor variación con respecto a  $\varphi_0$ . Para verificar cuál de las rutas  $\varphi_1 \in \{\varphi_1^I, \varphi_1^{II}, \varphi_1^{III}, \varphi_1^{IV}\}$  cumple con esta condición, se aplica la definición de producto escalar de la Ecuación (B.17):

$$\cos \alpha = \frac{u \cdot v}{\|u\| \|v\|} \quad (\text{B.17})$$

En esta Ecuación (B.17) el ángulo  $\alpha$  corresponde al menor ángulo positivo entre dos vectores  $u$  y  $v$ . Para el caso de las trayectorias, en vez de usar los ángulos  $\varphi_0$ , y  $\varphi_1 \in \{\varphi_1^I, \varphi_1^{II}, \varphi_1^{III}, \varphi_1^{IV}\}$  y poder aplicar el producto escalar, se definen sus correspondientes vectores unitarios (ver Figura B.10 (c)) :

$$\Phi_0 = \begin{bmatrix} \cos \varphi_0 \\ \sin \varphi_0 \end{bmatrix}, \quad \Phi_1(\varphi_1) = \begin{bmatrix} \cos \varphi_1 \\ \sin \varphi_1 \end{bmatrix} \tag{B.18}$$

Si en la Ecuación (B.17) en vez de los vectores  $u$  y  $v$ , se usan los vectores  $\Phi_0$  y  $\Phi_1(\varphi_1)$ , de la Ecuación (B.18), el ángulo formado por estos dos vectores es:

$$\alpha(\varphi_1) = \cos^{-1}(\Phi_0 \cdot \Phi_1(\varphi_1)) \tag{B.19}$$

Se evalúa la Ecuación (B.19) para todos los  $\varphi_1 \in \{\varphi_1^I, \varphi_1^{II}, \varphi_1^{III}, \varphi_1^{IV}\}$  (ver Figura B.10(d)), y el que de un menor resultado de  $\alpha$ , se toma como la dirección definitiva  $\varphi_1$  en el punto  $P_1$ .

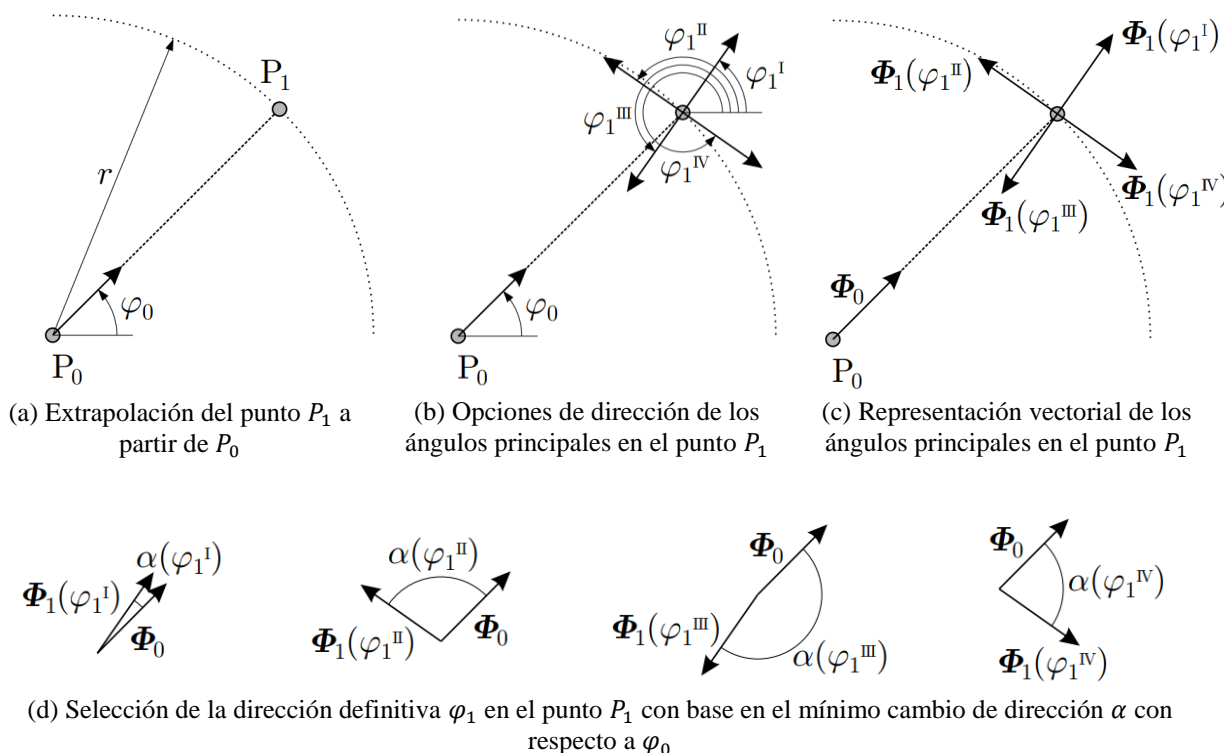


Figura B.10. Determinación de la dirección de seguimiento de la ruta más relevante  
Fuente: (Beyer, 2015)

**Determinación del rango de iteración:** La posición del punto  $P_1$  es iterada a lo largo de un arco circular (con radio  $r$ , que corresponde al tamaño del paso). A partir de la posición inicial extrapolada de  $P_1$ , se determina si la rotación del punto  $P_1$  (dependiente al ángulo  $\varphi_1$ ), se debe realizar en sentido positivo o negativo, de acuerdo con la Figura B.11(a). Para calcular el sentido de rotación de  $P_1$ , se define una variable de cambio  $\kappa_{rot} \in \{-1, 1\}$ , la cual indica el sentido de rotación (1 para rotación anti horaria y -1 para rotación horaria).

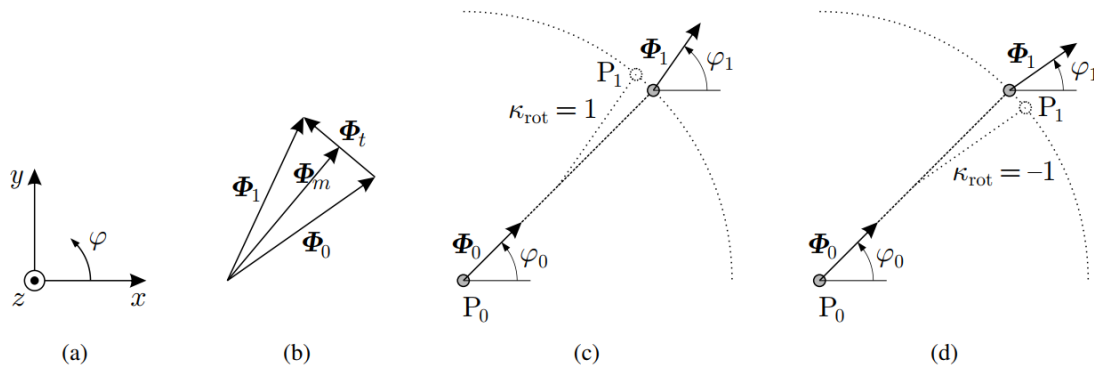


Figura B.11. Determinación del rango de iteración de  $P_1$ , con base en el sentido de la rotación definido por  $\kappa_{rot}$  del cambio en la dirección de  $\varphi_1$   
 Fuente: Beyer (2015)

Una forma para determinar el valor de  $\kappa_{rot}$ , es usar los vectores  $\Phi_0$  y  $\Phi_1$ , además de la definición de producto vectorial (producto cruz). Es necesario convertir los vectores  $\Phi_0$  y  $\Phi_1$  a vectores tridimensionales, añadiendo una tercera componente, de acuerdo a la Ecuación (B.20):

$$\Phi_0 = \begin{bmatrix} \cos \varphi_0 \\ \sin \varphi_0 \\ 0 \end{bmatrix}, \quad \Phi_1(\varphi_1) = \begin{bmatrix} \cos \varphi_1 \\ \sin \varphi_1 \\ 0 \end{bmatrix} \quad (\text{B.20})$$

A partir de estos vectores tridimensionales  $\Phi_1$  y  $\Phi_2$ , se pueden definir dos nuevos vectores  $\Phi_m$  y  $\Phi_t$  ortogonales entre sí (ver Figura B.11(b)):

$$\Phi_m = \frac{1}{2}(\Phi_1 + \Phi_0), \quad \Phi_t = \Phi_1 - \Phi_0 \quad (\text{B.21})$$

Los vectores  $\Phi_m$  y  $\Phi_t$ , son vectores que siguen en el plano  $xy$ . El producto cruz de estos dos vectores da como resultado un vector perpendicular al plano  $xy$ , con una componente en el eje  $z$  de  $z_{01}$  :

$$\Phi \times \Phi_t = \frac{1}{2} \begin{bmatrix} \cos \varphi_1 + \cos \varphi_0 \\ \sin \varphi_1 + \sin \varphi_0 \\ 0 \end{bmatrix} \times \begin{bmatrix} \cos \varphi_1 - \cos \varphi_0 \\ \sin \varphi_1 - \sin \varphi_0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ z_{01} \end{bmatrix} \quad (\text{B.22})$$

A partir de esta información puede calcularse el valor de la variable  $\kappa_{rot}$  (el cual indica el sentido de rotación de  $P_1$ ):

$$z_{01} = (\cos \varphi_1 + \cos \varphi_0)(\sin \varphi_1 - \sin \varphi_0) - (\sin \varphi_1 + \sin \varphi_0)(\cos \varphi_1 - \cos \varphi_0) \quad (\text{B.23})$$

El factor  $1/2$  de la Ecuación (B.22), no influye en el cálculo de la evaluación del signo de  $z_{01}$ , razón por la cual no se tiene en cuenta en la Ecuación (B.23). Para las iteraciones de rotación mostradas en las Figura B.10(c) y Figura B.10(d), el valor de  $\kappa_{rot}$  fue calculado de acuerdo a la Ecuación (B.23). De esta manera la iteración de rotación puede dar inicio de manera dirigida con un sentido adecuado. Si el valor de  $\kappa_{rot} = 0$ , esto indica que  $\varphi_1 = \varphi_0$ , por lo cual no es necesario realizar iteraciones de rotación y la posición inicial extrapolada de  $P_1$ , puede tomarse directamente como la posición final de  $P_1$  y ser solución del presente paso.

**Iteraciones de rotación del punto  $P_1$ :** Para la descripción de la posición del punto  $P_1$  en el arco circular, se introduce la el ángulo  $\varphi_{01}$  (ver Figura B.12(a)). Durante la iteración, la posición del punto  $P_1$  es controlada modificando el valor de este ángulo  $\varphi_{01}$ , de forma incremental (ya sea aumentando o disminuyendo el valor). El cálculo de las coordenadas  $(x_1, y_1)$  del punto  $P_1$  se continua de la misma manera a partir de la Ecuación (B.15), pero en vez de  $\varphi_0$ , se usa el valor de  $\varphi_{01}$ . El valor del ángulo  $\varphi_1$ , se sigue calculando de acuerdo a las Ecuaciones (B.16), (B.17), (B.18), (B.19) y la Figura B.10(d).

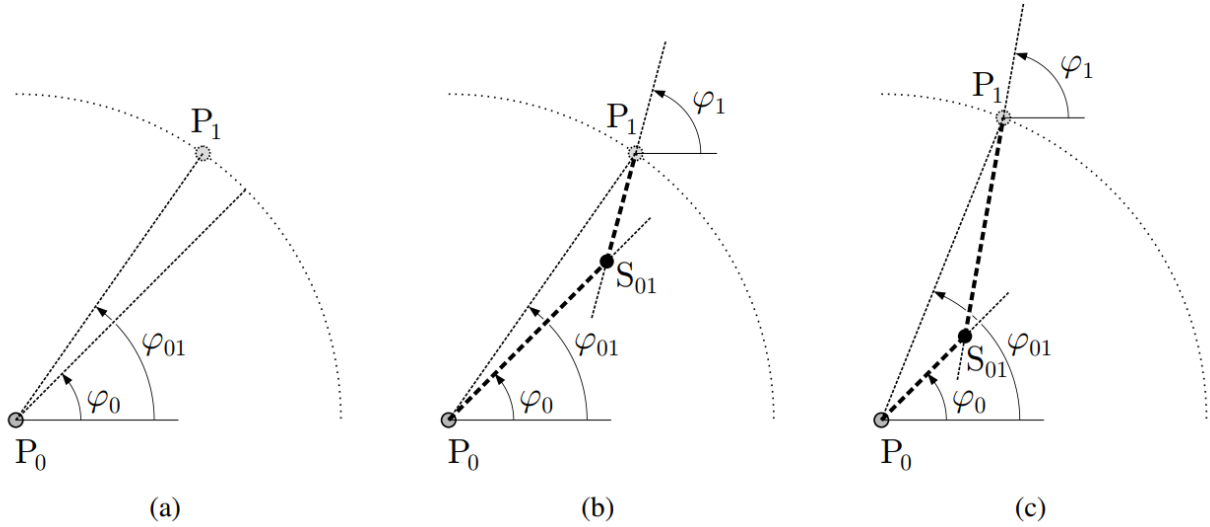


Figura B.12. Representación del principio de iteración con el ángulo de control  $\varphi_{01}$   
Fuente: Beyer (2015)

De acuerdo a lo descrito en la Sección B.2.2 (ver Figura B.6(d)), el propósito de la iteración del punto  $P_1$ , es que las distancias  $\overline{P_0S_{01}}$  y  $\overline{S_{01}P_1}$  sean iguales. El punto  $S_{01}$  es la intersección de la línea  $g_0$  (que pasa por los puntos  $P_0$  y  $S_{01}$ ), y de la línea  $g_1$  (que pasa por los puntos  $P_1$  y  $S_{01}$ ), las cuales están definidas en su forma vectorial como:

$$g_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + l_0 \begin{bmatrix} \cos \varphi_0 \\ \sin \varphi_0 \end{bmatrix}, \quad g_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} - l_1 \begin{bmatrix} \cos \varphi_1 \\ \sin \varphi_1 \end{bmatrix} \quad (\text{B.24})$$

El punto de intersección de las líneas  $g_0$  y  $g_1$  se calcula cuando  $g_0 = g_1$ , lo cual permite calcular los parámetros  $l_0$  y  $l_1$ :

$$l_0 = \frac{(y_1 - y_0) \cos \varphi_1 - (x_1 - x_0) \sin \varphi_1}{\cos \varphi_1 \sin \varphi_0 - \sin \varphi_1 \cos \varphi_0}$$

$$l_1 = \frac{-(y_1 - y_0) \cos \varphi_0 + (x_1 - x_0) \sin \varphi_0}{\cos \varphi_1 \sin \varphi_0 - \sin \varphi_1 \cos \varphi_0} \quad (\text{B.25})$$

Donde  $l_0$  en términos de cantidad, corresponde a la longitud de la distancia  $\overline{P_0S_{01}}$ , y  $l_1$  corresponde a la longitud de la distancia  $\overline{S_{01}P_1}$ . El signo negativo en la ecuación de la recta  $g_1$ , se agrega para que el valor de  $l_1$  sea positivo. El objetivo de la iteración es que  $l_0 = l_1$  o que  $l_1 - l_0 = 0$ . Como esta condición no puede cumplirse numéricamente por ningún medio, se acepta una

solución aproximada con la menor desviación posible. La desviación se calcula en función del tamaño del paso  $r$ , con el fin de evitar los efectos de escala. De esta manera, el criterio para terminar la iteración de rotación de  $P_1$  está dado por la Ecuación (B.26):

$$\frac{|l_1 - l_0|}{r} < \Delta_{tol} \tag{B.26}$$

Donde  $\Delta_{tol}$ , es un límite de tolerancia el cual, a menos que se especifique lo contrario, se selecciona con un valor de 0.0001.

La Figura B.13 muestra posibles soluciones de ubicación del punto  $P_1$  que pueden generarse durante la iteración de rotación. De igual manera, se esboza el valor del ángulo  $\varphi_1$  en seis puntos igualmente espaciados a lo largo del arco circular (con valor de radio igual al tamaño de paso  $r$ ).

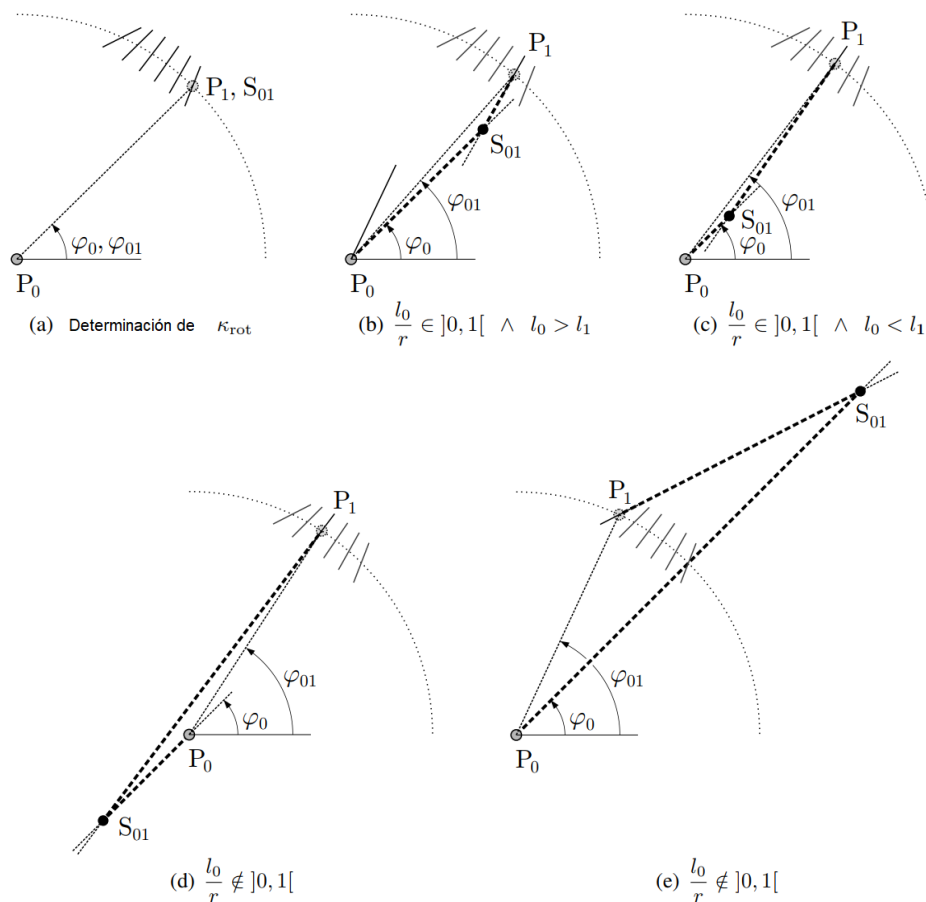


Figura B.13. Rango de control del ángulo  $\varphi_{01}$  y regla de los posibles casos presentes durante la iteración de  $P_1$   
Fuente: Beyer (2015)

La Figura B.13(a) muestra el primer paso antes de comenzar la iteración. En esta posición se determina el valor inicial de  $\kappa_{rot}$ . En el ejemplo presentado, el valor inicial de  $\kappa_{rot} = 1$ , es decir la iteración comienza rotando en sentido anti horario de las manecillas del reloj. En la Figura B.13(b) y Figura B.13(c) se muestran dos iteraciones de rotación de  $P_1$ , en las cuales el punto de intersección  $S_{01}$  se encuentra entre el los puntos  $P_0$  y  $P_1$ . En el primer caso (Figura B.13(b))  $l_0 > l_1$ , y en el segundo caso (Figura B.13(c))  $l_0 < l_1$ . Lo anterior indica que la localización del punto  $P_1$  donde  $l_0 = l_1$ , se encuentra entre las dos iteraciones realizadas. Si una iteración resulta en el caso presentado en la Figura B.13(b), esto significa que la siguiente iteración debe realizarse en sentido positivo de  $\kappa_{rot}$  (paso hacia adelante), es decir una rotación anti horaria. En el caso presentado en la Figura B.13(c), la siguiente iteración debe ser contraria al sentido positivo de  $\kappa_{rot}$  (paso hacia atrás), es decir una rotación horaria. Esta distinción se hace mediante la definición de una variable  $\kappa_{it}$ , a la cual se le asigna un valor de 1 para un paso hacia adelante y -1 para un paso hacia atrás. El valor inicial de  $\kappa_{it}$  es 1, es decir se inicia con un valor de paso hacia adelante. El ángulo de control  $\varphi_{01}$ , para una iteración de rotación se calcula con base en el valor previo y un incremento del ángulo  $\Delta\varphi_{01}$ , usando la Ecuación (B.27):

$$\varphi_{01} = \varphi_{01} + \kappa_{rot}\kappa_{it}\Delta\varphi_{01} \quad (\text{B.27})$$

Antes de la primera iteración de rotación se asigna a  $\varphi_{01}$  el valor de  $\varphi_0$ . Como incremento inicial del ángulo se establece el valor  $\Delta\varphi_{01} = \alpha/3$ , donde  $\alpha$  es el valor calculado de acuerdo a la Ecuación (B.19) con la dirección  $\varphi_1$  del punto  $P_1$  inicialmente extrapolado (ver Figura B.11). Si el valor de la variable  $\kappa_{it}$  cambia para la siguiente iteración, el valor del incremento del ángulo  $\Delta\varphi_{01}$  se reduce a la mitad.

Si el punto de intersección  $S_{01}$ , no se encuentra entre el los puntos  $P_0$  y  $P_1$ , como se observa en la Figura B.13(d) y Figura B.13(e), el incremento del ángulo se considera demasiado grande y su

valor se reduce a la mitad ( $\Delta\varphi_{01} = 0.5\Delta\varphi_{01}$ ), y se inicia nuevamente el proceso de iteración desde la ubicación original del punto  $P_1$ , es decir ( $\varphi_{01} = \varphi_0$  y  $\kappa_{it} = 1$ ). Teóricamente, pueden presentarse problemas para determinar el punto de intersección  $S_{01}$ , por ejemplo si las líneas  $g_0$  y  $g_1$  son paralelas (ver Figura B.14(a)). Aunque desde el punto de vista numérico es poco probable que este caso suceda en la práctica, debe verificarse antes de calcular el punto de intersección con la Ecuación (B.29). Como este caso representa una situación intermedia de la Figura B.13(d) y Figura B.13(e), es posible proceder de la misma manera que se hace para estos casos, es decir reiniciar la iteración de rotación del punto  $P_1$ , ( $\varphi_{01} = \varphi_0$  y  $\kappa_{it} = 1$ ) y reducir el incremento del ángulo ( $\Delta\varphi_{01} = 0.5\Delta\varphi_{01}$ ).

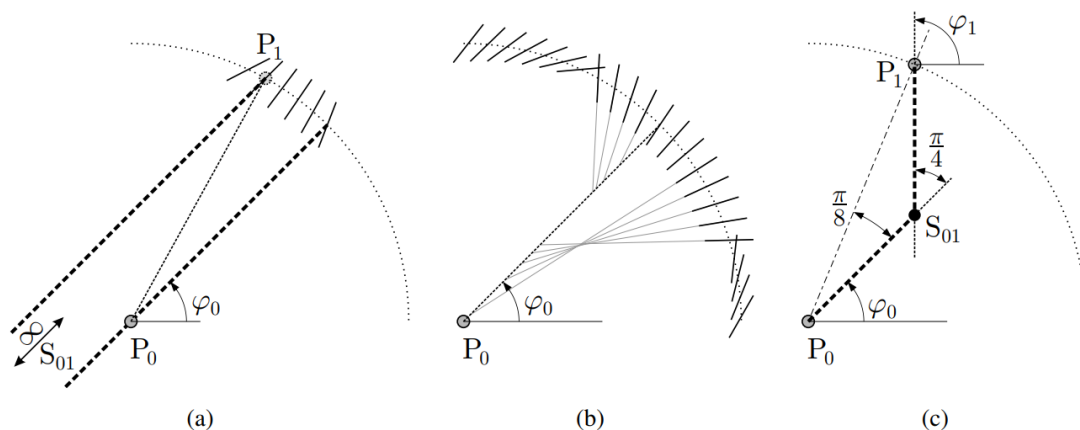


Figura B.14. Casos especiales y límites durante la iteración  
Fuente: Beyer (2015)

La Figura B.14(b), muestra una situación en la cual no se pudo encontrar una solución significativa, es decir no se puede cumplir la Ecuación (B.26). Las envolventes de la orientación de las tangentes direccionales, se derivan del cálculo de  $\varphi_1$ , de acuerdo a las Ecuaciones (B.16), (B.17), (B.18), (B.19) y la Figura B.10(d). La razón por la cual no fue posible encontrar una solución adecuada es que el tamaño del paso  $r$ , es demasiado grande, por lo que una aproximación como un segmento de arco circular no es posible para este caso. La iteración puede resultar en resultados sin sentido o nunca terminaría. Por esta razón tiene sentido limitar el rango de iteración.

La Figura B.14(c) muestra un límite para este valor. En el límite ilustrado la dirección principal de esfuerzo en el punto  $P_1$  difiere de la dirección principal en  $P_0$  en  $\pi/4$ . Entonces su desviación de la ortogonal en la dirección principal en  $P_0$  sería muy grande. La asignación de un valor mínimo de  $\alpha$  en la Ecuación (B.19) no es confiable en esta área, por lo tanto, el rango de iteración debe limitarse al menos a  $|\varphi_{01} - \varphi_0| < \pi/8$  con respecto a  $\varphi_{01}$ . Sin embargo, por razones de precisión, el rango debe ser elegido mucho más pequeño que este límite. A continuación, se complementa esta parte.

**Control del tamaño del paso:** La aproximación de las trayectorias de esfuerzo por medio de segmentos de arco circulares (ejemplo con una curvatura constante) es en general mejor, cuanto menor sea el cambio de dirección dentro de un paso. Por lo tanto, las secciones (dominio de diseño) de gran curvatura deben tratarse con pequeños incrementos. Esto es apropiado para resolver las áreas de mayor curvatura, de forma más detallada. Adicionalmente, en el análisis final una trayectoria debe poder representarse por medio de una línea poligonal simple. Para este propósito, el cambio en la dirección del paso de una trayectoria está limitado a un ángulo  $\alpha_{lim}$ . El cambio existente en la dirección  $\alpha$ , es similar al ángulo que forma el segmento de arco circular (ver Figura B.15(a)). Este corresponde al ángulo  $\alpha(\varphi_1)$ , calculado con la Ecuación (B.19) si se usa el ángulo  $\varphi_1$  del punto iterado  $P_1$ .

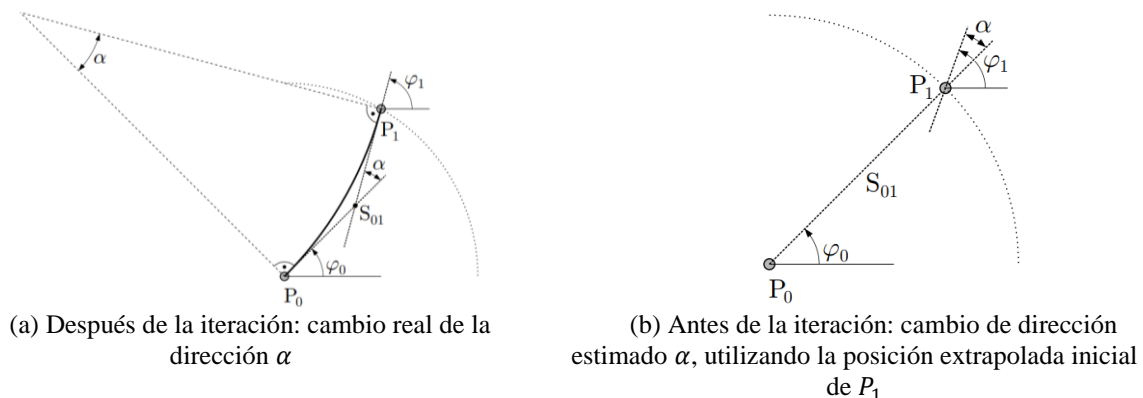


Figura B.15. Cambio de dirección estimado y determinado para un paso de una trayectoria  
Fuente: Beyer (2015)

Después de la iteración del punto  $P_1$ , de acuerdo a la sección anterior, se puede determinar el cambio de dirección  $\alpha$ , el cual puede ser menor, igual o mayor al límite establecido  $\alpha_{lim}$ . Con base en este resultado se decide si la iteración debe repetirse con una disminución del tamaño del paso  $r$ , ejemplo si no se cumple el criterio  $\alpha < \alpha_{lim}$ . De lo contrario, se puede continuar con el siguiente paso de la trayectoria. En cualquier caso, sin importar la decisión tomada, tiene sentido definir un nuevo tamaño de paso  $r = r_{new}$ , para el cual no es necesaria ninguna repetición (adicional) de la iteración. Suponiendo que una aproximación lineal entre el tamaño del paso  $r$  y el cambio de dirección  $\alpha$ :

$$\frac{r_{new}}{r} \approx \frac{\alpha_{lim}}{\alpha} \quad (B.28)$$

Con incrementos lo suficientemente pequeños y un valor de  $\alpha_{lim}$  lo suficientemente pequeño, esta suposición es bastante útil. En consecuencia, se puede proporcionar un valor guía para un nuevo incremento del paso  $r$ :

$$r_{new} \approx \frac{\alpha_{lim}}{\alpha} r \quad (B.29)$$

Como la Ecuación (B.29) es el resultado de una sola estimación, el tamaño real del nuevo paso se establece con un valor ligeramente menor para compensar las sobreestimaciones leves de  $r_{new}$ , para que no sea necesario repetir la iteración en el presente paso de la trayectoria o en el paso siguiente:

$$r = r_{new} = c_{red} \frac{\alpha_{lim}}{\alpha} \quad (B.30)$$

La selección del factor  $c_{red} = 0.9$  en la Ecuación (B.30) ha resultado ser adecuado. cuando el criterio  $\alpha < \alpha_{lim}$  se cumple, el tamaño del paso puede incrementarse. En este caso el tamaño del paso está limitado a un tamaño máximo de paso  $r_{max}$ :

$$r = r_{new} = \min\left(c_{red} \frac{\alpha_{lim}}{\alpha}, r_{max}\right) \quad (B.31)$$

Si  $\alpha = 0$ ,  $r = r_{new} = r_{max}$ , valor que corresponde al límite de la Ecuación (B.31) cuando el valor de  $\alpha \rightarrow 0$ .

Las medidas descritas hasta ahora para el ajuste del tamaño de paso  $r$ , una vez se ha completado la iteración son suficientes para cumplir el criterio  $\alpha < \alpha_{lim}$  en cada paso de la trayectoria. Sin embargo, en áreas donde se presenta una alta reducción del tamaño de paso para cumplir con este criterio, se requiere de múltiples iteraciones, lo que afecta el tiempo de cálculo computacional. Aunque este problema no ocurriría si el valor del parámetro  $c_{red}$  es mucho menor, se usarían incrementos mucho menores de los necesarios para el tamaño del paso. Esto incrementaría el número total de pasos de la trayectoria, lo que a su vez significa más iteraciones.

En lugar de ello, se usa la dirección al inicio de la iteración (ver Figura B.13(a)), para estimar el cambio de dirección esperado. Para pequeños  $\alpha_{lim}$  se tiene que:

$$\alpha r \ll r \quad (B.32)$$

Lo que significa que la distancia desde la posición inicial del punto  $P_1$  antes de la iteración, y el punto  $P_1$  iterado (después de la iteración), es mucho menor que sus distancias al punto  $P_0$ . Con base en esto, se asume que la dirección  $\varphi_1$  para determinar  $\alpha(\varphi_1)$  (la cual realmente es la dirección en el punto  $P_1$  iterado), puede estimarse como  $\varphi_1$ , en la posición inicial de  $P_1$  antes de comenzar la iteración. El cambio de dirección estimado de esta manera se denomina  $\tilde{\alpha}$ . Antes de dar inicio a la iteración, se verifica si el cambio de dirección estimado cumple el criterio  $\tilde{\alpha} < \alpha_{lim}$ . Si no se cumple, se asume que el cambio real de dirección  $\alpha$ , también será demasiado grande y esta iteración no dará un resultado que cumpla con el criterio  $\alpha < \alpha_{lim}$ . Por lo tanto, el tamaño del paso se reduce de acuerdo usando la Ecuación (B.33):

$$r = c_{red} r \quad (B.33)$$

Además de reducir el paso, el valor de  $\tilde{\alpha}$  es reevaluado hasta que se cumpla el criterio  $\tilde{\alpha} < \alpha_{lim}$ . Sólo entonces se da inicio a la iteración del punto  $P_1$ . De esta manera, se evitan virtualmente múltiples iteraciones. La Figura B.16 muestra una trayectoria determinada con el método de control de paso descrito. Los segmentos se muestran con un ancho de línea mayor para que los tamaños del paso sean reconocibles. Para su determinación se seleccionaron los siguientes parámetros:  $\alpha_{lim} = \frac{\pi}{180} \text{ rad}$ ,  $c_{red} = 0.9$  y  $r_{max} = \frac{l_{char}}{30}$ .

$l_{char}$  es la menor dimensión del dominio de diseño (para la Figura B.16, la distancia vertical). En este ejemplo se seleccionó  $r_{max} = \frac{l_{char}}{30}$ , con el fin de reconocer los segmentos individuales de la Figura B.16, sin embargo se recomienda usar  $r_{max} = \frac{l_{char}}{100}$ , para obtener una mejor resolución de la trayectoria calculada.

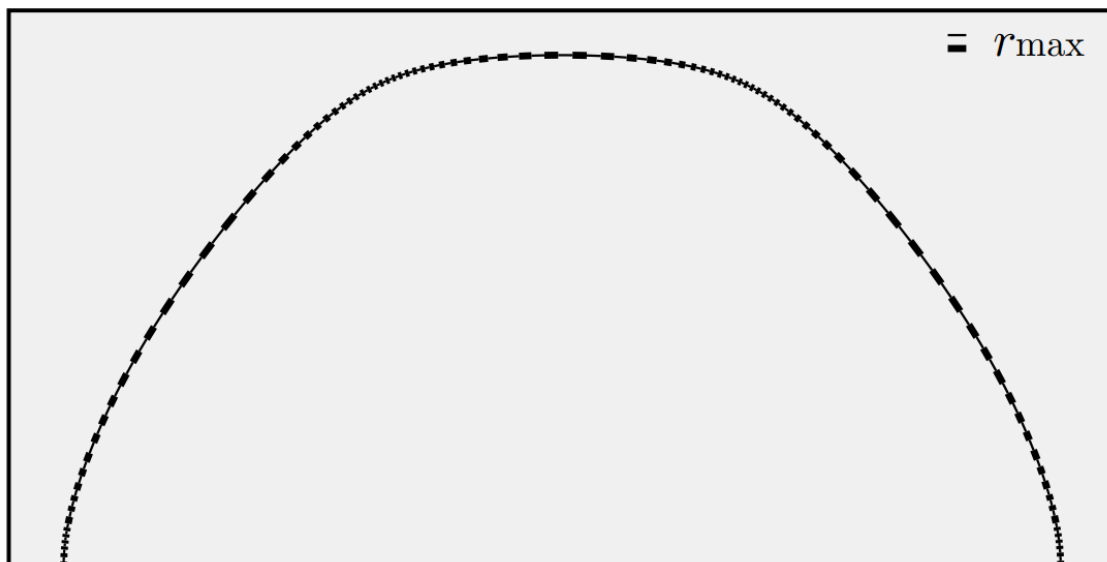


Figura B.16. Ejemplo de aplicación del método de control de paso. Los pasos de la trayectoria se muestran con un ancho de línea mayor. Límite del tamaño de paso  $r_{max}$  a escala  
Fuente: (Beyer, 2015).

**Criterio de parada:** Si se tiene que el valor de  $r = 0$ , como resultado de la reducción del tamaño de paso, este sería un criterio de terminación de la trayectoria, ya que una continuación con un

tamaño de paso mayor no proporciona resultados confiables y la continuación con  $r = 0$  no es posible.

El criterio de mayor importancia para dar por terminada una trayectoria, es cuando se cruza un borde del espacio de diseño. Comenzando para el punto de inicio de una trayectoria el cual debe estar dentro del dominio de diseño, se verifica en cada uno de los pasos siguientes de la trayectoria si el punto  $P_1$  se encuentra dentro del dominio de diseño. Si no es así, el proceso de determinación de la trayectoria debe ser terminado. El paso en el cual el punto  $P_1$  se encuentra por primera vez por fuera de los límites del dominio de diseño puede ser descartado o mejor, se mezcla con el borde, calculando el punto de intersección, para que la trayectoria termine exactamente en el borde. Para determinar si un punto se encuentra por dentro o por fuera del dominio de diseño se usa el siguiente método:

Usando un punto  $A$ , el cual se selecciona por fuera del dominio de diseño, se construye una línea recta entre los puntos  $A$  y  $P_1$ . Si el número de cortes (intersecciones) con los bordes del dominio de diseño es impar,  $P_1$  se encuentra dentro del dominio de diseño, caso contrario se encuentra por fuera. En la Figura B.17(a) y Figura B.17(b) muestran esquemáticamente estos casos.

A partir de la información de las coordenadas  $xy$  de cada uno de los bordes que componen el dominio de diseño, es posible conocer el valor de las coordenadas extremas ( $x_{min}$ ,  $x_{max}$ ,  $y_{min}$ ,  $y_{max}$ ). Las coordenadas  $(x_A, y_A)$  del punto  $A$ , se calculan como:

$$\begin{bmatrix} x_A \\ y_A \end{bmatrix} = \begin{bmatrix} x_{max} + c_{Ax}(x_{max} - x_{min}) \\ y_{max} + c_{Ay}(y_{max} - y_{min}) \end{bmatrix} \quad (\text{B.34})$$

Donde valores positivos para  $c_{Ax}$  y  $c_{Ay}$ , aseguran que el punto  $A$  se encuentra por fuera del dominio de diseño. Se selecciona de forma arbitraria los valores de  $c_{Ax} = c_{Ay} = \frac{1}{2}$ .

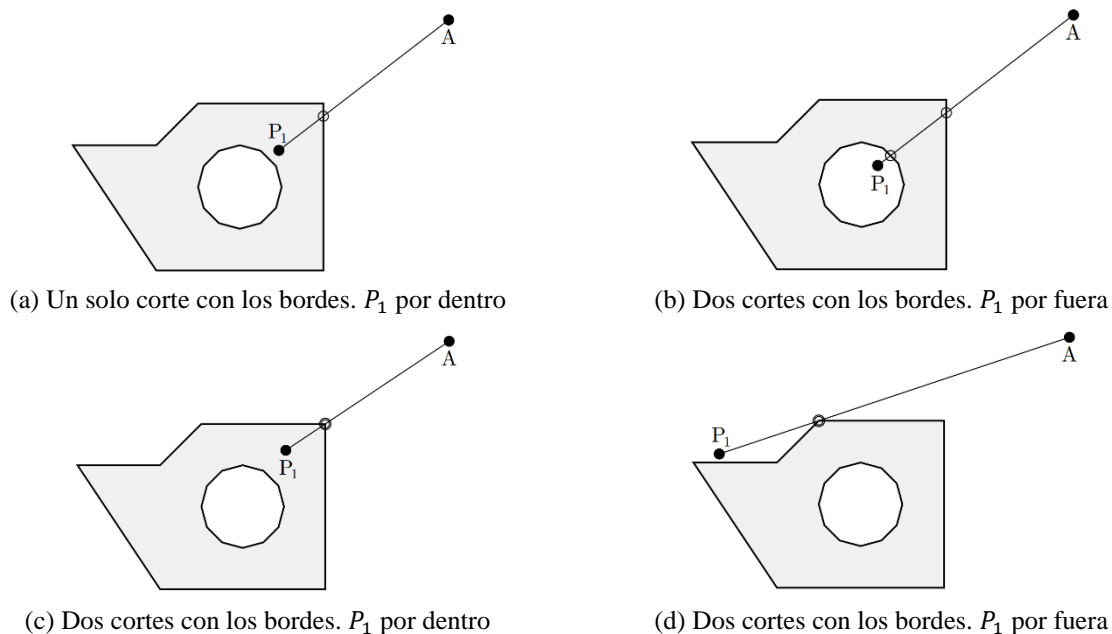


Figura B.17. Determinación de la posición de un punto  $P_1$ , por dentro o por fuera del dominio de diseño  
Fuente: Beyer (2015)

Para saber el número de bordes que corta la línea  $AP_1$ , se examinan todos los bordes del espacio de diseño, segmento por segmento. Para un segmento en particular se denota el punto inicial como  $R$  y el punto final como  $R'$ , La ecuaciones de las rectas  $g_{AP_1}$  (conformada por los puntos  $A$  y  $P_1$ ) y  $g_{RR'}$  (conformada por los puntos  $R$  y  $R'$ ), se definen de forma vectorial como:

$$\begin{aligned} g_{RR'} &= \begin{bmatrix} x_R \\ y_R \end{bmatrix} + t_{RR'} \begin{bmatrix} x_{R'} - x_R \\ y_{R'} - y_R \end{bmatrix} \\ g_{AP_1} &= \begin{bmatrix} x_A \\ y_A \end{bmatrix} + t_{AP_1} \begin{bmatrix} x_{P_1} - x_A \\ y_{P_1} - y_A \end{bmatrix} \end{aligned} \quad (\text{B.35})$$

El punto de intersección de las dos líneas se calcula cuando  $g_{RR'} = g_{AP_1}$ . Al realizar las operaciones correspondiente se obtiene que los parámetros  $t_{RR'}$  y  $t_{AP_1}$ , pueden calcularse como:

$$\begin{aligned} t_{RR'} &= \frac{(y_A - y_R)(x_{P_1} - x_A) - (x_A - x_R)(y_{P_1} - y_A)}{(y_{R'} - y_R)(x_{P_1} - x_A) - (x_{R'} - x_R)(y_{P_1} - y_A)} \\ t_{AP_1} &= \frac{(y_A - y_R)(x_{R'} - x_R) - (x_A - x_R)(y_{R'} - y_R)}{(y_{R'} - y_R)(x_{P_1} - x_A) - (x_{R'} - x_R)(y_{P_1} - y_A)} \end{aligned} \quad (\text{B.36})$$

Se cuenta como decisivos los cortes que se encuentran en las rutas  $\overline{AP_1}$  y  $\overline{RR'}$ . En consecuencia, se debe aplicar el criterio definido por la Ecuación (B.37):

$$0 < t_{AP_1} \leq 1 \quad (\text{B.37})$$

Donde la posición del punto  $P_1$ , dentro del borde  $\overline{AP_1}$  se considera dentro. Si se cumple esta condición, se distinguen tres casos:

- a) Caso 1: Si  $0 < t_{RR'} < 1$ , la intersección se encuentra dentro del borde  $\overline{RR'}$ , el contador de número de borde se aumenta en uno.
- b) Caso 2: Si  $t_{RR'} = 0 \vee t_{RR'} = 1$ , la intersección se encuentra en la esquina de un borde e intersecta dos bordes en un mismo punto. Como se muestra en la Figura B.17(c) y Figura B.17(d), en estos casos no puede decidirse si el punto se encuentra por dentro o por fuera del dominio de diseño. Por lo tanto, el punto  $A$  se mueve, por ejemplo cambiando una de sus coordenadas:

$$y_A = y_A + \frac{c_{Ay}}{100} (y_{max} - y_{min}) \quad (\text{B.38})$$

Como todos los bordes se verifican con respecto al mismo punto  $A$ , el número de bordes verificado hasta este paso debe descartarse y la verificación debe iniciar nuevamente con el nuevo punto  $A$ , para todos los bordes.

- c) Caso 3: Si adicionalmente a la condición del Caso 2,  $t_{AP_1} = 1$ , entonces el punto  $P_1$ , se encuentra ubicado en uno de los puntos originales del borde, es decir se encuentra por dentro del dominio de diseño y no es necesario verificar los demás bordes. La existencia de este caso es muy rara.

Si no se cumple la condición de la Ecuación (B.37), ni tampoco los Casos 1 y 2, se continúa examinando los demás bordes, pero no se aumenta el contador del número de cortes con los bordes.

## Apéndice C. Fundamentos del análisis y diseño estructural de celosías

### C.1. Análisis estructural de celosías planas

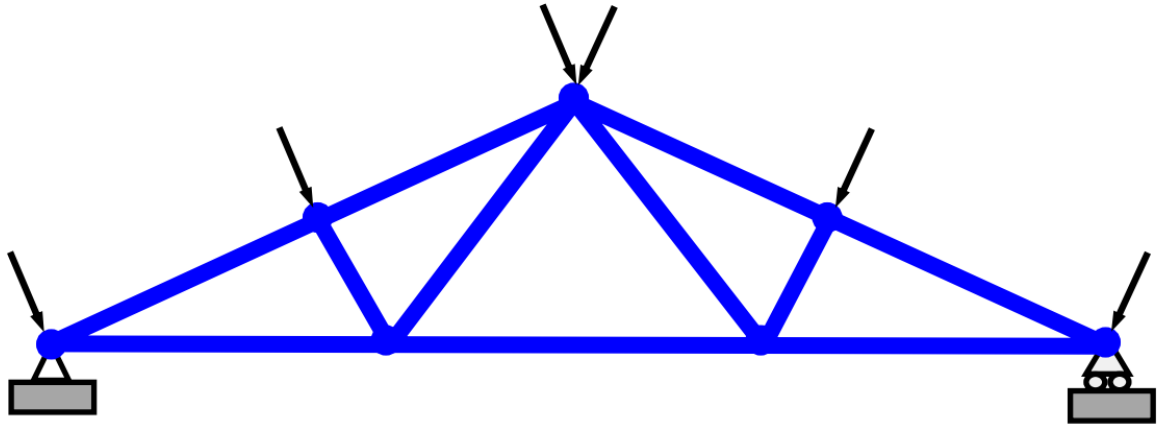
Las estructuras de celosía (armaduras o cerchas) se usan con frecuencia en el diseño de edificaciones, puentes y torres. Son estructuras con formas generalmente triangulares compuestas por barras esbeltas cuyas uniones están conectadas por medio de pernos, remaches o soldadura. Este tipo de estructuras permite cubrir grandes espacios y distancias y ofrece soluciones livianas y económicas en muchos problemas de ingeniería. Cuando todos los elementos, nodos y cargas se encuentran sobre un solo plano, se denomina celosía plana (de lo contrario, si tiene elementos y nodos que se extienden en el espacio 3D, se denomina celosía espacial). Para el análisis estructural de celosías generalmente se asume que (Chen & Liu, 2015; R.C. Hibbeler, 2012):

- Los elementos tienen un área de sección transversal uniforme y están conectados por medio de uniones libres de fricción.
- Las cargas se aplican solamente en los nodos y no a lo largo de la longitud de los elementos.

Con base en lo anterior, se considera que los elementos soportan solo cargas axiales de tensión/compresión y puede desprejarse la resistencia a la flexión. Además, el problema de análisis estructural puede tratarse como un problema lineal elástico, con base en las siguientes consideraciones (Chen & Liu, 2015; R.C. Hibbeler, 2012):

- Pequeñas deformaciones (el patrón de carga no cambia debido a la forma deformada de la estructura).
- Material elástico (no se considera plasticidad o fractura).
- Cargas estáticas (la carga se aplica a la estructura de una manera lenta y constante).

Como una celosía es el ensamblaje de barras axiales, cada una de ellas se trata como un elemento discreto tipo barra (ver Figura C.1) (Chen & Liu, 2015).



*Figura C.1.* Modelo discreto de una armadura plana de un techo  
Fuente: Chen & Liu (2015)

Un elemento barra permite describir la deformación y esfuerzo en elementos esbeltos de una estructura con un área de sección transversal constante y que se encuentran con cargas de tensión y compresión a lo largo de su eje. El comportamiento estructural del elemento puede establecerse usando el método de la rigidez directa (Chen & Liu, 2015).

**C.1.1. Método la rigidez directa.** El método de la rigidez directa, puede usarse para el análisis de estructuras estáticamente determinadas e indeterminadas. Con método de la rigidez se obtienen los desplazamientos y las fuerzas de forma directa. Su aplicación requiere subdividir la estructura en elementos finitos discretos (elementos barra) e identificar sus puntos extremos como nodos. Se determinan las propiedades de fuerza-desplazamiento en cada elemento y después se relacionan entre si usando las ecuaciones de equilibrio de fuerzas escritas en los nodos. Luego estas relaciones, para toda la estructura, se agrupan en lo que se denomina matriz de rigidez de la estructura  $\mathbf{k}$ . Una vez establecido esto, se pueden determinar los desplazamientos desconocidos de los nodos para cualquier carga dada sobre la estructura. Al conocer estos desplazamientos pueden calcularse las fuerzas externas e internas en la estructura utilizando las relaciones de fuerza-desplazamiento para cada elemento (R.C. Hibbeler, 2012).

**C.1.1.1. Formulación del elemento barra 1D.** Un elemento barra con dos nodos extremos se presenta a continuación:

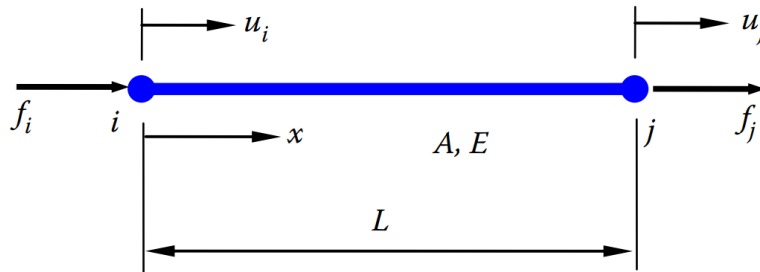


Figura C.2. Notación para un elemento barra  
Fuente: Chen & Liu (2015)

Asumiendo que el desplazamiento  $u$  varía linealmente a lo largo del eje de la barra, en términos de los dos desplazamientos nodales  $u_i$  y  $u_j$ , se puede escribir,

$$u(x) = \left(1 - \frac{x}{L}\right) u_i + \frac{x}{L} u_j \quad (\text{C.1})$$

Se tiene que

$$\varepsilon = \frac{u_j - u_i}{L} = \frac{\Delta}{L} \quad (\text{C.2})$$

$$\sigma = E\varepsilon = \frac{E\Delta}{L} \quad (\text{C.3})$$

También se tiene que,

$$\sigma = \frac{F}{A} \quad (\text{C.4})$$

Combinando las Ecuaciones (C.3) y (C.4) se tiene,

$$F = \frac{EA}{L} \Delta = k\Delta \quad (\text{C.5})$$

Donde  $k = EA/L$  es la matriz de rigidez de la barra. Como la barra se comporta como un resorte se puede concluir que la matriz de rigidez del elemento es:

$$\mathbf{k} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (\text{C.6})$$

La ecuación de equilibrio del elemento es:

$$\mathbf{ku} = \mathbf{f}$$

$$\frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_i \\ u_j \end{Bmatrix} = \begin{Bmatrix} f_i \\ f_j \end{Bmatrix} \quad (\text{C.7})$$

Para un elemento barra 1D se tiene un grado de libertad en cada uno de los nodos a lo largo del eje axial. El grado de libertad GDL es el número de componentes del vector de desplazamiento en cada uno de los nodos.

El significado físico de los coeficientes de  $\mathbf{k}$  es el siguiente: La columna  $j$ -ésima de  $\mathbf{k}$  ( $j = 1$  o  $2$ ) representa las fuerzas aplicadas a la barra, para mantener la forma deformada debido al desplazamiento unitario en el nodo  $j$  y cero desplazamiento en el otro nodo.

**C.1.1.2. Elemento barra en 2D.** Para analizar estructuras de celosía (armaduras) planas es necesario extender la formulación del elemento barra 1D a 2D.

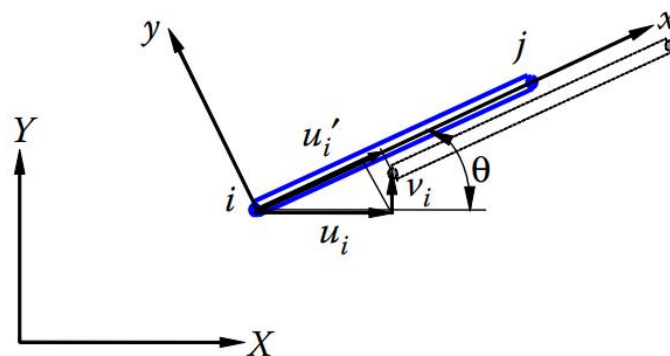


Figura C.3. Coordenadas locales y globales para una barra en el espacio 2D  
Fuente: Chen & Liu (2015)

**Transformación de coordenadas:** Los vectores de desplazamiento en el sistema de coordenadas local y global, se relacionan de la siguiente manera:

$$\begin{aligned} u_i' &= u_i \cos \theta + v_i \sin \theta = [l \quad m] \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \\ v_i' &= -u_i \sin \theta + v_i \cos \theta = [-m \quad l] \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \end{aligned} \quad (\text{C.8})$$

Donde

$$l = \cos \theta \text{ y } m = \sin \theta$$

En forma matricial,

$$\begin{aligned} \begin{Bmatrix} u_i' \\ v_i' \end{Bmatrix} &= \begin{bmatrix} l & m \\ -m & l \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \\ \mathbf{u}_i' &= \tilde{\mathbf{T}} \mathbf{u}_i \end{aligned} \quad (\text{C.9})$$

Donde la matriz de transformación  $\tilde{\mathbf{T}}$  es,

$$\tilde{\mathbf{T}} = \begin{bmatrix} l & m \\ -m & l \end{bmatrix} \quad (\text{C.10})$$

$\tilde{\mathbf{T}}$  es una matriz ortogonal, es decir:  $\tilde{\mathbf{T}}^{-1} = \tilde{\mathbf{T}}^T$

Para los dos nodos de la barra, los desplazamientos locales en función de los desplazamientos globales,

$$\begin{Bmatrix} u_i' \\ v_i' \\ u_j' \\ v_j' \end{Bmatrix} = \begin{bmatrix} l & m & 0 & 0 \\ -m & l & 0 & 0 \\ 0 & 0 & l & m \\ 0 & 0 & -m & l \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix} \quad (\text{C.11})$$

$$\mathbf{u}' = \mathbf{T} \mathbf{u} \text{ con } \mathbf{T} = \begin{bmatrix} \tilde{\mathbf{T}} & 0 \\ 0 & \tilde{\mathbf{T}} \end{bmatrix} \quad (\text{C.12})$$

Las fuerzas nodales se transforman de la misma manera,

$$\mathbf{f}' = \mathbf{T} \mathbf{f} \quad (\text{C.13})$$

En el sistema de coordenadas locales,

$$\frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_i' \\ u_j' \end{Bmatrix} = \begin{Bmatrix} f_i' \\ f_j' \end{Bmatrix} \quad (\text{C.14})$$

En su forma aumentada,

$$\frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u_i' \\ v_i' \\ u_j' \\ v_j' \end{Bmatrix} = \begin{Bmatrix} f_i' \\ 0 \\ f_j' \\ 0 \end{Bmatrix} \quad (\text{C.15})$$

$$\mathbf{k}'\mathbf{u}' = \mathbf{f}'$$

Usando las transformaciones de las Ecuaciones (C.12) y (C.13) en la Ecuación (C.15),

$$\mathbf{k}'\mathbf{T}\mathbf{u} = \mathbf{T}\mathbf{f} \quad (\text{C.16})$$

Multiplicando ambos lados por  $\mathbf{T}^T$  y aplicando la identidad de la matriz de transformación  $\mathbf{T}^T\mathbf{T} = \mathbf{I}$  se obtiene,

$$\mathbf{T}^T\mathbf{k}'\mathbf{T}\mathbf{u} = \mathbf{f} \quad (\text{C.17})$$

De esta manera la matriz de rigidez del elemento en coordenadas globales es,

$$\mathbf{k} = \mathbf{T}^T\mathbf{k}'\mathbf{T} \quad (\text{C.18})$$

La cual es una matriz simétrica de 4x4, que en su forma explícita es,

$$\mathbf{k} = \frac{EA}{L} \begin{bmatrix} u_i & v_i & u_j & v_j \\ l^2 & lm & -l^2 & -lm \\ lm & m^2 & -lm & -m^2 \\ -l^2 & -lm & l^2 & lm \\ -lm & -m^2 & lm & m^2 \end{bmatrix} \quad (\text{C.19})$$

El cálculo de los cosenos directores  $l$  y  $m$ :

$$l = \cos \theta = \frac{X_j - X_i}{L} \quad m = \sin \theta = \frac{Y_j - Y_i}{L} \quad (\text{C.20})$$

**Identificación de los elementos y nodos:** Inicialmente se debe identificar los elementos o miembros de la estructura de celosía (armadura) y sus nodos. Se identifica los nodos cercano y lejano de cada elemento, para indicar la dirección del elemento desde el nodo cercano hacia el nodo lejano (Kassimali, 2012).

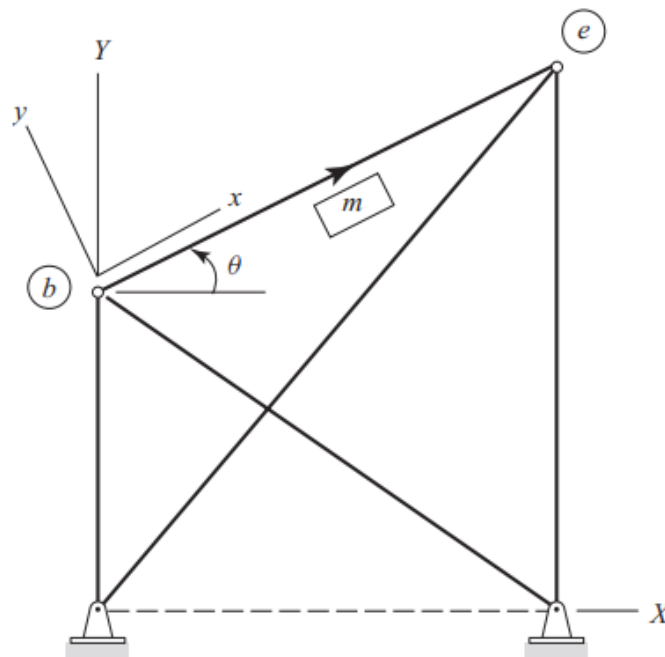


Figura C.4. Representación de un elemento  $m$  (nodo cercano  $b$ , nodo lejano  $e$ ) de una armadura con ejes locales  $xy$  rotados un ángulo  $\theta$  anti horario, respecto a los ejes globales  $XY$   
Fuente: (Kassimali, 2012)

## C.2. Requisitos de diseño de los elementos estructurales

**C.2.1. Elementos a tensión.** En la Especificación para Construcciones de Acero ANSI/AISC 360-16 (AISC, 2016), Capítulo D, y en el Reglamento Colombiano de Construcción Sismo Resistente NSR-10 (AIS, 2010), Sección F.2.4, se estipula que la resistencia nominal de un elemento a tensión  $P_n$ , será la más pequeña de los valores obtenidos sustituyendo las dos expresiones siguientes (teniendo en cuenta el método de diseño LRFD):

- a) Para el estado límite de fluencia en la sección bruta (con la idea de prevenir alargamiento excesivo del elemento),

$$P_n = F_y A_g \quad (\text{AISC D2-1, NSR-10 F.2.4.2-1})$$

$$\phi_t P_n = \phi_t F_y A_g \quad (\text{Resistencia de diseño por tensión, } \phi_t = 0.9)$$
(C.21)

- b) Para fractura por tensión en la sección neta en la que se encuentren agujeros de tornillos y remaches,

$$P_n = F_u A_e \quad (\text{AISC D2-2, NSR-10 F.2.4.2-2})$$

$$\phi_t P_n = \phi_t F_u A_e \quad (\text{Resistencia de diseño a la fractura por tensión, } \phi_t = 0.75)$$
(C.22)

Donde  $A_e$  es el área neta efectiva del elemento en  $in^2$  ( $mm^2$ ),  $A_g$  es el área bruta del elemento en  $in^2$  ( $mm^2$ ),  $F_y$  es el esfuerzo mínimo de fluencia del acero en  $ksi$  ( $MPa$ ) y  $F_u$  es el esfuerzo mínimo de tensión en  $ksi$  ( $MPa$ ).

El área neta efectiva  $A_e$  se calcula usando la fórmula:

$$A_e = A_n U \quad (\text{AISC D3 - 1, NSR - 10 F. 2.4.3 - 1})$$
(C.23)

Donde el área neta  $A_n$  es el valor del área bruta  $A_g$  de la sección menos el área de los agujeros y  $U$  es el factor de corte diferido determinado de acuerdo a la Tabla D3.1 de la especificación ANSI/AISC 360-16 (AISC, 2016) o la Tabla F.2.4.3-1 de la norma NSR-10 (AIS, 2010).

La relación de esbeltez de un elemento es el cociente de la longitud no soportada  $L$  y su radio de giro mínimo  $r$ . El propósito de esta limitación para elementos a tensión es garantizar que posea suficiente rigidez para prevenir deflexiones o vibraciones indeseables. Aunque los elementos a tensión no están sujetos a pandeo bajo cargas normales, pueden ocurrir inversión de esfuerzos en estos elementos durante el transporte, montaje y también debido a las cargas sísmicas y de viento. Para elementos a tensión, excepto varillas, la especificación ANSI/AISC 360-16 (AISC, 2016), Sección D.1 y la norma NSR-10 (AIS, 2010), Sección F.2.4.1 sugieren que la relación máxima de esbeltez para elementos a tensión, preferiblemente no debe exceder el valor de 300.

$$\frac{L}{r} \leq 300$$
(C.24)

**C.2.2. Elementos a compresión.** Las secciones a compresión pueden clasificarse como elementos no esbeltos y elementos esbeltos. Un elemento no esbelto es aquel en el cual la relación de ancho a espesor de sus elementos a compresión no excede el valor límite  $\lambda_r$  de la Tabla B4.1a de la especificación ANSI/AISC 360-16 (AISC, 2016) o tabla F.2.2.4-1a de la norma NSR-10

(AIS, 2010). Si la relación ancho a espesor excede  $\lambda_r$ , la sección se define como una sección de elemento esbelto. Para el caso de un perfil con lados de ángulos simples (como los usados en este proyecto de investigación para las armaduras), la razón de ancho espesor  $b/t$  debe cumplir (AISC, 2016; AIS, 2010):

$$\frac{b}{t} \leq 0.45 \sqrt{\frac{E}{F_y}} \quad (\text{C.25})$$

Los elementos a compresión pueden clasificarse como largos cortos o intermedio (McCormac & Csernak, 2012):

a) En los elementos largos la fórmula de Euler predice muy bien la resistencia en las que el esfuerzo axial de pandeo permanece por debajo del límite proporcional. Este tipo de elementos se pandea elásticamente.

b) En los elementos cortos el esfuerzo de falla será igual al esfuerzo de fluencia y no ocurrirá el pandeo.

c) En los elementos intermedios algunas fibras alcanzar el esfuerzo de fluencia y otras no. Los miembros fallarán tanto por fluencia como por pandeo y su comportamiento se denomina inelástico. La mayoría de las columnas caen en este rango. Para que la fórmula de Euler sea aplicable a este tipo de elementos, se modifica de acuerdo al concepto de módulo reducido o el concepto de módulo de tangente para tener en cuenta la presencia de esfuerzos residuales.

En la especificación ANSI/AISC 360-16 (AISC, 2016), capítulo E y en la norma NSR-10 (AIS, 2010), Sección F.2.5 se proporciona la ecuación de Euler para columnas largas con pandeo elástico y una ecuación parabólica empírica para columnas cortas e intermedias. Con estas ecuaciones se determina un esfuerzo de pandeo a flexión  $F_{cr}$ , para un elemento a compresión. Una vez calculado este esfuerzo para un elemento en particular, se multiplica por el área de la sección transversal para

obtener su resistencia nominal  $P_n$ . La resistencia de diseño LRFD de un elemento a compresión puede determinarse de la siguiente manera:

$$P_n = F_{cr}A_g \quad (\text{AISC E3-1, NSR-10 F.2.5.3-1})$$

$$\phi_c P_n = \phi_c F_{cr}A_g \quad (\text{Resistencia de diseño por compresión, } \phi_c = 0.9)$$
(C.26)

Las siguientes expresiones muestran cómo se determina el esfuerzo de pandeo por flexión  $F_{cr}$ , para elementos no esbeltos:

a) 
$$\text{Si } \frac{KL}{r} \leq 4.71 \sqrt{\frac{E}{F_y}} \quad \left( \text{o } \frac{F_y}{F_e} \leq 2.25 \right)$$

(C.27)

$$F_{cr} = \left[ 0.658 \left( \frac{F_y}{F_e} \right) \right] F_y \quad (\text{AISC E3-2, NSR-10 F.2.5.3-2})$$

b) 
$$\text{Si } \frac{KL}{r} > 4.71 \sqrt{\frac{E}{F_y}} \quad \left( \text{o } \frac{F_y}{F_e} > 2.25 \right)$$

(C.28)

$$F_{cr} = 0.877 F_e \quad (\text{AISC E3-3, NSR-10 F.2.5.3-3})$$

En estas expresiones  $F_e$  es el esfuerzo de pandeo crítico elástico (esfuerzo de Euler), calculado con la longitud efectiva del elemento  $KL$ .

$$F_e = \frac{\pi^2 E}{\left( \frac{KL}{r} \right)^2} \quad (\text{AISC E3 - 4, NSR - 10 F. 2.5.3 - 4})$$
(C.29)

Para elementos diseñados a compresión, la relación de esbeltez efectiva preferiblemente no debe exceder el valor de 200, de acuerdo a la Sección E.2 de la ANSI/AISC 360-16 (AISC, 2016) o la sección F.2.5.2 de la NSR-10 (AIS, 2010):

$$\frac{KL}{r} \leq 200$$
(C.30)

Este valor se basa en un criterio de ingeniería, en la economía práctica y en el hecho de tener un cuidado especial para conservar la integridad de un elemento esbelto durante la fabricación, el flete y el montaje.

En las ecuaciones anteriores  $A_g$  es el área bruta del elemento en  $in^2$  ( $mm^2$ ),  $E$  es el módulo de elasticidad del acero igual a 29000  $ksi$  (200000  $MPa$ ),  $F_y$  es el esfuerzo mínimo de fluencia en  $ksi$  ( $MPa$ ),  $K$  es el factor de longitud efectiva del elemento (para el caso de una armadura su valor es 1),  $L$  es la longitud efectiva del elemento sin arriostamiento y  $r$  es el radio de giro de la sección.

### C.2.3. Perfiles estructurales

A continuación, en la Tabla C.1 se presenta el catálogo de perfiles estructurales empleado en los problemas de optimización abordados en este trabajo:

Tabla C.1. *Tabla de perfiles estructurales tipo L*

#	Perfil	A ( $in^2$ )	b/t	$r_x, r_y$ (in)	#	Perfil	A ( $in^2$ )	b/t	$r_x, r_y$ (in)
1	L2X2X1/8	0.491	16.0	0.620	31	L6X6X3/8	4.38	16.0	1.87
2	L2X2X3/16	0.722	10.6	0.612	32	L4X4X5/8	4.61	6.40	1.20
3	L2-1/2X2-1/2X3/16	0.901	13.3	0.771	33	L5X5X1/2	4.79	10.0	1.53
4	L2X2X1/4	0.944	8.00	0.605	34	L6X6X7/16	5.08	13.7	1.86
5	L3X3X3/16	1.09	16.0	0.933	35	L4X4X3/4	5.44	5.33	1.18
6	L2X2X5/16	1.16	6.39	0.598	36	L6X6X1/2	5.77	12.0	1.86
7	L2-1/2X2-1/2X1/4	1.19	10.0	0.764	37	L5X5X5/8	5.90	8.00	1.52
8	L2X2X3/8	1.37	5.33	0.591	38	L6X6X9/16	6.45	10.7	1.85
9	L3X3X1/4	1.44	12.0	0.926	39	L5X5X3/4	6.98	6.67	1.50
10	L2-1/2X2-1/2X5/16	1.46	7.99	0.756	40	L6X6X5/8	7.13	9.60	1.84
11	L3-1/2X3-1/2X1/4	1.70	14.0	1.09	41	L8X8X1/2	7.84	16.0	2.49
12	L2-1/2X2-1/2X3/8	1.73	6.67	0.749	42	L5X5X7/8	8.00	5.71	1.49
13	L3X3X5/16	1.78	9.58	0.918	43	L6X6X3/4	8.46	8.00	1.82
14	L4X4X1/4	1.93	16.0	1.25	44	L8X8X9/16	8.77	14.2	2.49
15	L3-1/2X3-1/2X5/16	2.10	11.2	1.08	45	L8X8X5/8	9.69	12.8	2.48
16	L3X3X3/8	2.11	8.00	0.910	46	L6X6X7/8	9.75	6.86	1.81
17	L2-1/2X2-1/2X1/2	2.26	5.00	0.735	47	L6X6X1	11.0	6.00	1.79
18	L4X4X5/16	2.40	12.8	1.24	48	L8X8X3/4	11.5	10.7	2.46
19	L3X3X7/16	2.43	6.85	0.903	49	L8X8X7/8	13.3	9.14	2.45
20	L3-1/2X3-1/2X3/8	2.50	9.33	1.07	50	L10X10X3/4	14.5	13.3	3.10
21	L3X3X1/2	2.76	6.00	0.895	51	L8X8X1	15.1	8.00	2.43
22	L4X4X3/8	2.86	10.7	1.23	52	L10X10X7/8	16.8	11.4	3.07
23	L3-1/2X3-1/2X7/16	2.89	7.99	1.06	53	L10X10X1	19.0	10.0	3.05
24	L5X5X5/16	3.07	16.0	1.56	54	L10X10X1-1/8	21.3	8.85	3.03
25	L3-1/2X3-1/2X1/2	3.25	7.00	1.05	55	L12X12X1	23.0	12.0	3.70
26	L4X4X7/16	3.30	9.13	1.22	56	L10X10X1-1/4	23.4	8.00	3.02
27	L5X5X3/8	3.65	13.3	1.55	57	L10X10X1-3/8	25.6	7.25	3.00
28	L6X6X5/16	3.67	19.2	1.88	58	L12X12X1-1/8	25.8	10.6	3.68
29	L4X4X1/2	3.75	8.00	1.21	59	L12X12X1-1/4	28.4	9.60	3.66
30	L5X5X7/16	4.22	11.4	1.54	60	L12X12X1-3/8	31.1	8.70	3.64

Nota: Adaptado de AISC (2017)

**Apéndice D. Descripción del contenido de la carpeta de anexos**

En la carpeta de anexos (pueden visualizarlos en la Base de Datos de la Biblioteca UIS) se encuentran guardados todos los archivos de programación del proceso de optimización topológica multiobjetivo desarrollado en este trabajo (Algoritmo AOTMOH). La información se encuentra ordenada de la siguiente manera:

1. En la carpeta “01. Estrategia\_de Generación\_Estructura\_Base”, se encuentran los programas que permiten generar la estructura base usando el método NCN y el método TEM.
2. En la carpeta “02. Algoritmos\_Metaheurísticos\_AMOSA\_MOPSO\_NSGAII”, se encuentran los algoritmos metaheurísticos de optimización multiobjetivo de estructuras de celosía programados en MATLAB®. Se encuentran ordenados por carpetas de la siguiente manera:
  - ✓ **Optimización de tamaño**: Los algoritmos se encuentran programados para ser usados en el problema de optimización de tamaño. La carpeta está organizada internamente en dos subcarpetas: subcarpeta 1 si se usa la estructura base generada con el método NCN o con el método TEM (que es un archivo llamado “GroundStruct.mat” que tiene la información de coordenadas de los nodos y la conectividad de los elementos) y subcarpeta 2 si se carga la información de coordenadas y conectividad con un archivo de texto. A su vez estas subcarpetas están organizadas en dos opciones dependiendo si se aplican las restricciones de diseño de los problemas de referencia (benchmark), o se aplican las restricciones de diseño de las normas ANSI/AISC 360-16 y NSR-10 Título F.
  - ✓ **Optimización topológica**: Los algoritmos se encuentran programados para ser usados en el problema de optimización topológica. La carpeta está organizada internamente de la misma manera que la carpeta de optimización de tamaño.

3. En la carpeta “03. Interfaz\_Grasshopper\_NSgaiI” se encuentra la interfaz del proceso de optimización topológica multiobjetivo para el algoritmo NSGA-II, integrado con el software Rhinoceros®-Grasshopper®, para representar gráficamente a medida que avanzan las iteraciones, la evolución geométrica de la solución del frente de Pareto correspondiente a mínimo peso.
4. En la carpeta “04. Interfaz\_Grasshoppercode” se encuentra el código de programación de la interfaz gráfica en el software Rhinoceros®-Grasshopper®, el cual puede adaptarse a todos los algoritmos de optimización topológica de la carpeta “02. Algoritmos\_Metaheurísticos\_AMOSA\_MOPSO\_NSgaiI”, tomando como guía la carpeta “03. Interfaz\_Grasshopper\_NSgaiI”.
5. En la carpeta “05. Algoritmo\_Análisis\_Estructural” se encuentra el código de análisis estructural de celosías planas implementado en este trabajo, el cual fue integrado en cada uno de los algoritmos de optimización multiobjetivo.