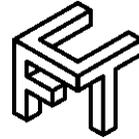




Universidad de Matanzas  
Facultad de Ciencias Técnicas



# OPTIMIZACIÓN DE TIPOLOGÍAS DE ARMADURAS PLANAS USANDO TÉCNICAS HEURÍSTICAS

Tesis Presentada como Requisito Parcial  
para la Obtención del Título de  
Máster en Ingeniería Asistida por Computadora

Autor: Ing. Alejandro Hernández Hernández

Tutor: Dr.C. Ramón Quiza Sardiñas

Matanzas, 2018

## **DECLARACIÓN DE AUTORIDAD Y NOTA LEGAL**

Yo, Alejandro Hernández Hernández, declaro que soy el único autor de la siguiente tesis, titulada *Optimización de tipologías de armaduras planas usando técnicas heurísticas* y, en virtud de tal, cedo el derecho de copia de la misma a la Universidad de Matanzas, bajo la licencia *Creative Commons* de tipo *Reconocimiento No Comercial Sin Obra Derivada*, con lo cual se permite su copia y distribución por cualquier medio siempre que mantenga el reconocimiento de sus autores, no haga uso comercial de la obra y no realice ninguna modificación de ella.

Matanzas, 20 de noviembre de 2018

## **RESUMEN**

Actualmente las técnicas heurísticas de optimización son una opción muy utilizada como herramienta para la solución de problemas complejos de búsqueda y optimización, debido a su probada capacidad de encontrar soluciones adecuadas en un tiempo computacionalmente aceptable ya que resultan ser de las pocas técnicas efectivas en problemas que antes se resolvían gracias a la intuición y experiencia de los investigadores. Empleando estas técnicas se pueden optimizar recursos materiales, tiempo de ejecución, factibilidad, costos, entre otros, lo que facilita mejoras no solo en la calidad sino también en la economía. Una de las ramas en la que más se emplean estas técnicas es la optimización estructural, en la presente investigación se realiza una comparación del desempeño de varias de estas técnicas, así como un análisis de la mejor variante en la optimización de estructuras (tipologías de armaduras planas).

**Palabras claves:** Heurísticas de optimización, optimización estructural.

## **ABSTRACT**

Nowadays multi objective evolutionary algorithms (MOEA) are worldwide used techniques as tools to solve optimization and complex search problems, because of their proved ability to find good solutions in a very short period of time, these are some of the few techniques that can find solutions to very hard problems better than many others that depend on the intuition and expertise of designers and researchers. Using this MOEA, natural resources, materials, execution time among other aspects can be optimized, and this improve not only the quality of the design but also the economy. One of the fields in which this MOEA is more applied is the structural optimization. In this thesis a comparison between some of these techniques is carried out, such as an analysis of the best ones of this MOEAs performance in the optimization of some particular structures (specially plane truss).

**Keywords:** multi objective evolutionary algorithms, structural optimization

## TABLA DE CONTENIDO

Introducción .....	1
Capítulo 1. Análisis del estado del arte de la optimización estructural y los métodos heurísticos de optimización.....	7
1.1 Generalidades.....	7
1.2 Optimización.....	7
1.2.1 Función objetivo .....	9
1.2.2 Restricción .....	11
1.2.3 Espacio de búsqueda.....	12
1.2.4 Variables .....	12
1.2.5 Óptimo local, Óptimo global y de Pareto. ....	13
1.3 Antecedentes de los algoritmos evolutivos.....	14
1.3.1 Población.....	17
1.3.2 Función de evaluación ( <i>Fitness</i> ) .....	17
1.3.3 Operadores Genéticos .....	18
1.4 Técnicas heurísticas de optimización.....	19
1.4.1 Ventajas y desventajas de las heurísticas de optimización. ....	20
1.5 Armaduras planas. Conceptualización, utilización, principales tipologías.....	21
1.5.1 Armaduras planas.....	21
1.5.2 Tipologías de armaduras planas.....	23
1.6 Software .....	28
1.7 Conclusiones Parciales.....	29

Capítulo 2 Análisis comparativo del desempeño de las diferentes técnicas heurísticas de optimización.....	30
2.1 Generalidades.....	30
2.2 Técnicas heurísticas empleadas para el estudio .....	30
2.2.1 <i>Not sorting genetic algorithms</i> II (NSGAII).....	30
2.2.2 <i>Not sorting genetic algorithms</i> III (NSGAIII) .....	31
2.2.3 <i>Multi Objective Cross Entropy</i> (MOCE+).....	32
2.2.4 <i>Simple Multi Objective Cross Entropy</i> (SMOCE).....	32
2.2.5 <i>Multi-objective evolutionary algorithm based on decomposition</i> (MOEAD).....	34
2.2.6 <i>Multi objective particle swarm optimization</i> (MOPSO).....	34
2.2.7 <i>Strength Pareto Evolutionary Algorithm</i> (SPEA-II) .....	35
2.2.8 <i>Pareto Envelope-based Selection Algorithm</i> (PESA II) .....	36
2.3 Funciones de prueba. ....	37
2.4 Parámetros que influyen en el desempeño de las heurísticas .....	44
2.5 Métricas para realizar la comparación. ....	46
2.6 Comparación. ....	47
2.7 Conclusiones parciales.....	56
Capítulo 3 aplicación de las heurísticas seleccionadas al diseño de tipologías de armaduras planas.....	57
3.1 Generalidades.....	57
3.2 Análisis del caso de optimización (configuración de armaduras planas) .....	57
3.2.1 Solución propuesta por el diseñador (Sloped). ....	60
3.2.2 Solución Optimizada con NSGA III(Sloped). ....	61

3.2.3 Solución Optimizada con SMOCE (Sloped). .....	62
3.2.4 Solución propuesta por el diseñador (Pratt).....	63
3.2.5 Solución Optimizada con NSGA III (Pratt).....	64
3.2.6 Solución Optimizada con SMOCE (Pratt).....	66
3.2.7 Solución propuesta por el diseñador (Vertical). .....	67
3.2.8 Solución Optimizada con NSGA III (Vertical). .....	68
3.2.9 Solución Optimizada con SMOCE (Vertical).....	69
3.2.10 Solución propuesta por el diseñador (English). .....	70
3.2.11 Solución Optimizada con NSGA III (English). .....	71
3.2.6 Solución Optimizada con SMOCE (English). .....	72
3.3 Conclusiones parciales.....	73
Conclusiones .....	74
Recomendaciones .....	75
Referencias Bibliográficas .....	76

## INTRODUCCIÓN

Hasta las últimas décadas del pasado siglo, muchos de los procesos de diseño fueron realizados por la experiencia e intuición del diseñador, en muchas ocasiones derivadas de la subjetividad, en vez de una aplicación intensiva de la teoría de optimización o la simulación. Recientemente esta forma de pensar ha cambiado debido a la importancia que ha tomado el campo de la optimización estructural en el diseño estructural, ya que mediante su aplicación se logra reducir costos, materiales y tiempo en los procesos de diseño realizados por los equipos de proyecto. El propósito de aplicar los conceptos de diseño óptimo a la ingeniería estructural es el de obtener una solución a un problema que cumpla con todas las limitaciones y restricciones impuestas, y que a la vez resulte ser la mejor en cuanto a uno o varios criterios de diseño previamente establecidos.

Una de las maneras de enfocar la optimización estructural es a partir de la programación genética, en la inteligencia artificial, el término programación genética se refiere a una metodología basada en los algoritmos evolutivos e inspirada en la evolución biológica para desarrollar automáticamente programas de computadoras que realicen una tarea definida por el usuario. Es una técnica de aprendizaje automático utilizada para optimizar un grupo de individuos (por ejemplo tipologías de armaduras o celosías) evaluando su capacidad de acuerdo a una función de ajuste o aptitud, permitiendo al usuario elegir la solución más idónea. Como metodología científica, la heurística es aplicable a cualquier ciencia e incluye la elaboración de medios auxiliares, principios, reglas, estrategias y programas que faciliten la búsqueda de vías de solución a problemas; o sea, para resolver tareas de cualquier tipo para las que no se cuente con un procedimiento algorítmico de solución. La importancia de minimizar el número de soluciones posibles esta enfatizada

en generar problemas razonables de optimización que se refieren generalmente en la optimización estructural a las tres categorías del diseño: el tamaño, la forma y la topología.

Esta disciplina ha ido evolucionando con el paso del tiempo hasta convertirse en un área de ingeniería llamada optimización estructural. El interés creciente en esta área en las últimas décadas es debido a la disponibilidad de computadoras baratas y poderosas, junto con los desarrollos rápidos en los métodos de análisis estructural y de optimización que juega un papel principal en muchos campos de ingeniería. Por otra parte numerosos problemas a los que se enfrentan los ingenieros civiles están descritos por la optimización de estructuras.

La optimización de estructuras (armaduras, celosías, entre otras) se ocupa de la selección de nodos y su conectividad, en otro nivel, crea búsquedas de optimización para encontrar las coordenadas óptimas de nodos existentes, finalmente, las secciones transversales de miembros de la estructura pueden ser optimizadas. En cada caso, el problema de optimización es subordinado a algunas restricciones en los desplazamientos del nodo, las cargas críticas de alabeo, entre otras. La función objetivo es en muchos casos minimizar el peso de estructura que se correlaciona con el costo global.

Elegir la mejor solución y mejorarla, depende fuertemente de la experiencia del diseñador, por lo que no es simple y no se dispone de los parámetros o la capacidad suficientes para asegurar que la última propuesta de solución que se obtenga sea la óptima, ya que es físicamente imposible evaluar el espectro completo de posibles soluciones. Entonces, resulta atractivo tener una herramienta que permita obtener

soluciones de un problema de diseño lo más cercanas posibles a la óptima, cuyo desempeño no dependa de la experiencia del diseñador, de lo que se deriva el **problema de la investigación**: Los diseños de las diferentes tipologías de armaduras planas que se realizan hoy en el país, se obtienen fundamentalmente basados en experiencias y criterios del diseñador y no de un proceso de optimización que derive en armaduras más racionales y económicas, lo que implica un gasto innecesario de recursos.

Siguiendo entonces la siguiente **hipótesis**: Si se emplean técnicas heurísticas de optimización se pueden obtener configuraciones de armaduras planas que resulten más eficientes que las propuestas por criterios de diseñador.

De esta se derivan las siguientes variables de la investigación:

- Variable independiente: Configuración de armaduras planas
- Variable dependiente: Heurística seleccionada, parámetros de diseño estructural, restricciones.

Siendo el **objetivo general**:

- Aplicar técnicas heurísticas de optimización al proceso de diseño de diferentes tipologías de armaduras planas de modo que se obtengan configuraciones más racionales que a partir de criterios del diseñador.

Teniendo como objetivos específicos:

- Analizar el estado del conocimiento sobre las técnicas heurísticas de optimización, su aplicación en el campo de la optimización estructural y las armaduras planas.
- Comparar los resultados del desempeño de varias heurísticas ante diferentes funciones de prueba.
- Analizar los resultados del diseño estructural de diferentes tipologías de armaduras teniendo en cuenta las heurísticas seleccionadas como más efectivas y el criterio del diseñador.

El cumplimiento de los objetivos trazados conlleva a que se obtenga como resultado un algoritmo para el diseño de la configuración de las armaduras planas que son usadas con mayor frecuencia, que sea racional y cumpla con la resistencia necesaria sin tener que acudir a la experiencia del diseñador, ahorrando además tiempo en la elaboración de los proyectos.

Esta investigación tendrá un **valor económico**, pues la optimización del diseño de los elementos estructurales traerá consigo un ahorro de las materias primas utilizadas en la elaboración de los mismos, además de la implicación con la conservación del medio ambiente que está implícita en el ahorro de los recursos naturales. Esta reducción de la utilización de materiales sirve además como impulso a la elaboración de estos elementos utilizados en la construcción de viviendas, teniendo entonces un carácter social.

Posee además un **valor metodológico**, pues la investigación es abordada a través de diferentes fases, etapas y pasos que permiten orientar metodológicamente la secuencia de

acciones lógicas a desarrollar y los elementos a tener en cuenta para la continuidad de la misma. Además permite la integración de los procedimientos tradicionales más utilizados por autores relacionados con la temática.

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

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

- Actualización del estado del conocimiento sobre la optimización estructural y las heurísticas de optimización.
- Descripción de las principales técnicas heurísticas utilizadas
- Comparación del desempeño de estas heurísticas a partir de funciones de pruebas.
- Aplicar las heurísticas de mejores prestaciones a la optimización de varias tipologías de armaduras planas.

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

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

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

# **CAPÍTULO 1. ANÁLISIS DEL ESTADO DEL ARTE DE LA OPTIMIZACIÓN ESTRUCTURAL Y LOS MÉTODOS HEURÍSTICOS DE OPTIMIZACIÓN**

## **1.1 Generalidades**

En este capítulo se realiza un estudio del estado del arte referente a la utilización de las principales tipologías de las armaduras planas, además de una breve reseña de los avances que ha tenido la optimización estructural en los últimos tiempos y las principales ventajas de las técnicas heurísticas de optimización sobre las tradicionales, así como la definición de los conceptos fundamentales que se deben conocer del amplio campo referente a la optimización.

## **1.2 Optimización**

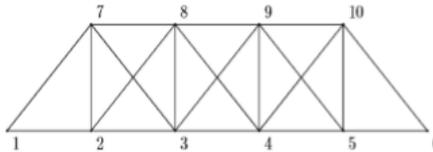
El hombre desde sus inicios se ha enfrentado constantemente a situaciones que requieren de la selección de una de varias alternativas, con el fin de dar solución a un problema o suplir alguna necesidad. Hasta hace poco tiempo, bastaba con que la alternativa escogida fuera útil para lo que se requería, sin embargo los recursos disponibles son limitados, entonces no basta con elegir una alternativa solo porque esta sirve, sino que se hace necesario elegir la “mejor” de todas las posibilidades, es decir, la requiere de menos recursos para obtener el mismo resultado.(Chen, 2013)

De este modo, surge el concepto de optimización, que se refiere a la selección del mejor elemento de un conjunto de elementos posibles, de acuerdo a un criterio de selección dado. Para esto, se deben emplear métodos de optimización que permitan conocer los mejores valores que puede asumir dicho conjunto de variables.(Gong et al., 2014).

De forma genérica, puede definirse la optimización como aquella ciencia encargada de determinar las mejores soluciones a problemas matemáticos que a menudo modelan una realidad física.(Coello and Christiansen, 2000). Una solución eficiente y precisa a estos problemas depende no sólo del tamaño del problema en términos del número de restricciones y del número de parámetros de diseño sino también de las características de la función objetivo y de las restricciones.

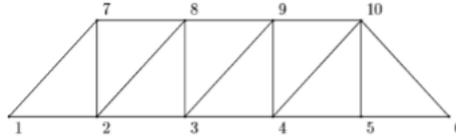
El problema de la optimización en estructuras se puede abordar desde diferentes enfoques, que se dividen fundamentalmente los siguientes:(Coello et al., 1995)

- Optimización del tamaño: Trata de determinar las medidas óptimas de los elementos que constituyen una estructura de forma predefinida. Es el enfoque más sencillo ya que el número de variables del problema es bastante limitado.



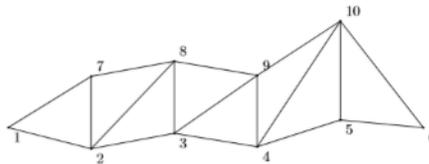
*Figura 1.1 Optimización de tamaño(Ramirez, 2014)*

- Optimización de la forma: Trata de buscar la forma óptima de una estructura de topología fija. En este enfoque, partiendo de una forma predefinida se busca optimizar esta mediante la modificación de la conectividad entre elementos o mediante la eliminación de los elementos menos esforzados.



*Figura 1.2 Optimización de forma (Ramirez, 2014)*

- Optimización de la topología: Trata de buscar la distribución óptima del material en una estructura, mediante la supresión de elementos y/o la modificación de la conectividad y/o de las coordenadas nodales. Este enfoque es el más complicado porque la cantidad de variables que aborda es elevada. En todos los trabajos publicados hasta la fecha se parte siempre de una morfología previa.



*Figura 1.3 Optimización de topología (Ramirez, 2014)*

### **1.2.1 Función objetivo**

La función objetivo, es la función que mide cualitativamente el funcionamiento del sistema en un proceso de optimización, de modo que lo que se busca es la maximización o minimización de esta, enmarcado en una gama de posibles soluciones.

Una función de evaluación heurística es una función que hace corresponder situaciones del problema con números. Es decir, da una medida conceptual de la distancia entre un estado dado y el estado objetivo. Estos valores son usados para determinar cuál operación ejecutar a continuación, típicamente seleccionando la operación que conduce a la situación con máxima o mínima evaluación. (Gharari et al., 2016)

La función debe dar un estimado útil y realista del mérito de un estado particular. La evaluación de la función en general no debe requerir un gran cálculo en su aplicación. Si la evaluación de la función es computacionalmente compleja puede ser más eficiente hacer una búsqueda a ciegas en lugar de gastar recursos (tiempo y memoria) en el cálculo de la función.

Es siempre mejor usar una función heurística con valores más altos que otras, siempre que esta no esté sobreestimada. Para un problema puede haber una colección de heurísticas admisibles  $h_1, \dots, h_m$ . Si una de ellas domina a las otras, es decir alcanza valores mayores para todos los nodos, se debe seleccionar esta.

Un problema habitual en las ejecuciones de las técnicas heurísticas surge debido a la velocidad con la que el algoritmo converge. En algunos casos la convergencia es muy rápida, lo que suele denominarse convergencia prematura, en la cual el algoritmo converge hacia óptimos locales, mientras que en otros casos el problema es justo el contrario, es decir se produce una convergencia lenta del algoritmo.

Otra técnica que se ha utilizado en el caso en que la computación de la función objetivo sea muy compleja es la denominada evaluación aproximada de la función objetivo. En algunos casos la obtención de "n" funciones objetivo aproximadas puede resultar mejor que la evaluación exacta de una única función objetivo.

Una posible solución a estos problemas pasa por efectuar transformaciones en la función objetivo. El problema de la convergencia prematura, surge a menudo cuando la selección de individuos se realiza de manera proporcional a su función objetivo. En tal caso, pueden existir individuos con una adaptación al problema muy superior al resto, que a

medida que avanza el algoritmo "dominan" a la población. Por medio de una transformación de la función objetivo, en este caso una comprensión del rango de variación de la función objetivo, se pretende que dichos "superindividuos" no lleguen a dominar a la población.

El algoritmo puede terminar por llegar a un estado para el cual no se pueden generar mejores estados. Esto puede suceder si el proceso de búsqueda ha alcanzado un máximo local, una cresta o una meseta.

Asimismo, existen diferentes tipos de funciones objetivo, de acuerdo al tipo de variable, formulación matemática y operatividad. Entre las más importantes: lineal y no lineal, unimodal y multimodal, estática y dinámica, sin ruido y con ruido, un objetivo y multiobjetivo.

### **1.2.2 Restricción**

Una restricción es un parámetro que permite ceñir, circunscribir o reducir a un límite un elemento determinado, de modo que haya validez en el diseño.(Coello et al., 2002)

Dado que estas restricciones son aplicadas en problemas de tipo físico y matemático, se pueden clasificar en dos grandes grupos:

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

En el ámbito estructural, las restricciones se refieren al comportamiento de la estructura y su estabilidad, y se imponen generalmente a los esfuerzos y deformaciones en los

elementos y por ende a sus dimensiones mínimas y máximas. Además, deben imponerse restricciones constructivas para que los elementos obtenidos tengan una secuencia y configuración lógica y factible de ser llevada a cabo en la realidad.

### **1.2.3 Espacio de búsqueda**

Se define como espacio de búsqueda al conjunto de todos los elementos que pueden ser operados por los métodos de optimización. La búsqueda generalmente se realiza con los mejores individuos de distintas áreas del espacio del problema. (MacDonald, 2005).

La representación de una solución potencial y su correspondiente interpretación brinda el espacio de búsqueda y su tamaño. Este es un punto clave para el problema: el tamaño de dicho espacio no viene determinado por el problema sino por la representación e interpretación.

### **1.2.4 Variables**

Representan las decisiones que se pueden tomar para variar el valor de la función objetivo. Se pueden clasificar como dependientes o independientes.

En el caso de la optimización estructural de armaduras planas o celosías por lo general son consideradas como variables las longitudes y secciones transversales de los elementos, es decir la posición de los nudos y la rigidez de las barras.(Pires, 2014)

### **1.2.5 Óptimo local, Óptimo global y de Pareto.**

Como definición básica, en matemáticas aplicadas y ciencia computacional, un óptimo local de un problema de optimización es una solución óptima (un máximo o un mínimo) de un conjunto vecino de soluciones candidatas. En contraposición, un óptimo global es la solución óptima de entre todas las soluciones posibles. En relación a estos dos tipos de óptimos, se pueden distinguir entre algoritmos locales y globales.

Los algoritmos de optimización local, en un caso general, sólo son capaces de detectar óptimos locales de las funciones objetivo. Se trata de algoritmos que dirigen su búsqueda evaluando la derivada de la función objetivo o alguna aproximación de ésta. (Coello et al., 2002)

Por su parte los algoritmos de optimización global permiten detectar óptimos globales, o al menos una buena aproximación de los mismos. La mayoría de los métodos heurísticos realizan una búsqueda estocástica por todo el espacio, de modo que estarían en condiciones de detectar óptimos globales de las funciones objetivo.

Óptimo de Pareto: Este término aparece cuando se afronta problemas de optimización de más de una variable y con restricciones. En estos casos no todas las soluciones son viables. Y más aún, los óptimos de Pareto hacen referencia a soluciones en las que no se puede mejorar más los valores de una variable, sin empeorar alguna de las otras condiciones. Es decir que más que alcanzar una solución única, lo que se consigue es definir una frontera de soluciones equilibradas (frente de Pareto) donde no es posible encontrar una solución que mejore en ningún sentido sin empeorar en otro. (Coello, 2012).

### **1.3 Antecedentes de los algoritmos evolutivos.**

Los Algoritmos Evolutivos incluyen técnicas relacionadas con la optimización metaheurística. Con el auge que tomaron estas teorías, y por medio de los descubrimientos de nuevas tecnologías que permitían un desarrollo computacional y matemático mucho más elevado, empezaron a investigarse teorías y algoritmos artificiales sentando sus bases en los mecanismos naturales con los trabajos de Holland. (Holland, 1975).

A partir de esta fecha, los estudios sobre algoritmos evolutivos, en cualquiera de sus facetas, se han incrementado de tal forma que ya es difícil encontrar un área de investigación en la que no hayan irrumpido con fuerza. Se han encontrado implementados en ámbitos tan diversos como finanzas, inversión, educación, transporte, redes de comunicación, planificación, logística, producción, optimización, entre otras.(Biethan and Nissen, 1995). Las primeras ideas en las que se basan técnicas heurísticas se pueden encontrar en los artículos de Holland de principios de los años 1960. En ellos se establecen los aspectos básicos para la comprensión de los principios de los sistemas adaptativos. Estos sistemas son capaces de automodificarse en respuesta a su interacción con el medio en el que están funcionando. Las teorías sobre los sistemas adaptativos están orientadas a facilitar, por un lado, la comprensión de las formas complejas de adaptación que aparecen en los sistemas naturales y, por otro, la habilidad para diseñar sistemas adaptativos robustos. A mediados de los años 1960, las ideas de Holland empezaron a plasmarse en modelos implementados en ordenador. En cada uno de estos sistemas se representaban elementos por medio del uso de genomas en los que

los mecanismos de evolución y herencia eran abstracciones de operadores genéticos como la mutación, el cruce o la inversión.

Estos estudios experimentales desembocaron en el tratamiento de problemas de búsqueda más complejos, como el reconocimiento de patrones. Asimismo, se realizaron los primeros estudios con operadores de selección de tipo elitista y las ideas básicas para el empleo de probabilidades adaptativas de mutación y cruce. También en esta época se estudió detalladamente, por primera vez, las diferentes posibilidades de reproducción y cruce. Usando dos espacios de soluciones diferentes, se experimentó con una amplia variedad de operadores de cruce.

En paralelo con estos estudios experimentales, Holland continuó trabajando en una teoría general de los sistemas adaptativos. Posteriormente evidenció todas estas ideas en su libro "*Adaptation in Natural and Artificial Systems*". Muchas de las propiedades de los algoritmos genéticos identificadas por Holland de forma teórica no pudieron ser observadas experimentalmente debido a la falta de recursos computacionales. Esto obligaba a trabajar con un número limitado de generaciones y con un número muy pequeño de individuos, habitualmente menos de 20. (Holland, 1975).

En esta misma época, se empieza a generalizar la investigación sobre las heurísticas de optimización al iniciarse trabajos en diversas universidades. Esta generalización fue muy lenta en un principio debido a cierto escepticismo creado por los malos resultados que habían obtenido las técnicas basadas en sistemas auto-organizativos y las redes de neuronas artificiales con los perceptrones que no cubrieron las expectativas creadas. Entorno a estas técnicas se creó un ambiente de pesimismo, pero a pesar de esto se

celebró en la Universidad de Michigan el congreso “*An interdisciplinary Workshop in Adaptive Systems*”, lo cual impulsó su estudio, consiguiendo que se establecieran diversos grupos de trabajo.

Dado el creciente interés en los algoritmos genéticos, se organizó en 1985 el primer “*International Congress on Genetic Algorithms*” donde se presentaron los últimos avances tanto teóricos como prácticos en el empleo de estas técnicas. El éxito de esta edición del congreso consiguió que se repitiera su celebración cada dos años. A partir de 1989, las actividades en el campo de los algoritmos genéticos habían crecido mucho, por lo que se fundó la “*International Society for Genetic Algorithms (ISGA)*”(Saka and Greem, 2012).

Fue precisamente David Goldberg, un discípulo de Holland quien alcanzó mayor éxito en la temática de optimización. Según parece tras asistir a uno de los seminarios de Holland, Goldberg que estaba interesado en encontrar el diseño óptimo de líneas para el transporte de gas, le planteó la posibilidad de emplear los Algoritmos Genéticos con este fin. Poco tiempo después logró llevar a cabo dicha tarea en su tesis doctoral. Posteriormente publicaría un libro que se convertiría un hito en el campo de la optimización y que recogía numerosas aplicaciones, muchas de ellas relacionadas con la optimización de estructuras, como el clásico problema de optimización de diez barras y seis nudos, aunque sin la restricción de desplazamientos del problema original (Goldberg, 1989).

A partir de la década de 1980 fueron aumentando la cantidad y complejidad de los problemas de optimización de estructuras que son abordados a partir del empleo de las heurísticas de optimización, un gran número de estos problemas son de armaduras planas

y espaciales.(Cheng, 2010), (Ahrari et al., 2014), (Gutierrez et al., 2012), (Hultman, 2010), (Smith et al., 2002).

### **1.3.1 Población**

Este parámetro indica el número de cromosomas que se tiene para una generación determinada. Es importante que exista una población lo suficientemente amplia, para que haya un adecuado número de reproducciones y se llegue a una solución óptima. No obstante, el tamaño de la población debe definirse de modo que no sea demasiado amplio y por lo tanto la convergencia tome mucho tiempo. Por lo general, la población inicial se genera como una combinación aleatoria de individuos a partir de un rango de variables de diseño (Chen and Zou, 2014).

### **1.3.2 Función de evaluación (*Fitness*)**

Esta función es la que vincula el algoritmo con el problema de optimización y es la que evalúa la calidad de la solución. Comienza tomando un cromosoma de entrada y retorna un número o conjunto de números que dan una medida de la calidad del cromosoma en relación a las condiciones del problema.

Usualmente se pueden distinguir los siguientes tipos de funciones para la mayoría de las técnicas heurísticas: continuas/ discontinuas, convexas/no convexas, unimodales/ multimodales, cuadráticas/no cuadráticas, dimensionalidad alta/dimensionalidad baja, determinísticas/estocásticas(Coello et al., 2002).

### 1.3.3 Operadores Genéticos

Se aplican de modo que se altere la composición de los descendientes (mutación o cruce). Análogamente a la teoría de evolución de las especies de Darwin, se distinguen varias operaciones genéticas que se deben aplicar a los individuos para evaluar una población.

- Selección: Es el operador que selecciona un grupo de individuos de la población actual para el apareamiento, dependiendo del valor de su *fitness*.(Saka and Greem, 2012)
- Cruzamiento o Reproducción: Esta es la segunda operación genética, en la cual los *strings* de los padres seleccionados son fraccionados y algunos genes son intercambiados para generar nuevos individuos. (Kona et al., 2006)
- Mutación: En la evolución, una mutación es un suceso bastante poco común. Estas contribuyen a la diversidad genética de la especie. En un algoritmo evolutivo tendrán el mismo papel, y la misma frecuencia es decir, muy baja. Una vez aplicado el cruzamiento para obtener una nueva población se aplica este tercer operador genético, la mutación, que consiste en alterar aleatoriamente los genes de algunos de los individuos de la población. Si no hay mutación, los descendientes son los mismos que había cuando se dio la reproducción. En caso de que haya mutaciones, parte del cromosoma descendiente es modificado y si la probabilidad de mutación es del 100%, significa que la totalidad del cromosoma es alterada.(Koziel and Yang, 2011)

- Elitismo: Este operador se emplea para que las operaciones de cruce y mutación no afecten un buen cromosoma indefinidamente. Lo que hace es copiar el mejor individuo de la población anterior a la siguiente, lo que permite una mejor convergencia del algoritmo. Sin embargo, deben imponerse las restricciones necesarias para que no se generen “superindividuos”.(Marler and Arora, 2004).

#### **1.4 Técnicas heurísticas de optimización.**

Al resolver una armadura con las técnicas actuales, se diseña y se halla una solución que satisfaga las condiciones establecidas por el método de prueba y error, dependiendo en gran medida de la experiencia de quien lo calcule. El resultado es una estructura factible desde el punto de vista estructural y de las normas vigentes, pero no necesariamente optimizada(Allaire et al., 2004).

Los algoritmos heurísticos por lo general tienen un tiempo de ejecución mucho menor y evidencian la posibilidad de mejorar significativamente el diseño de las estructuras, a partir de una línea de desarrollo que se impone en el mundo, pero que en Cuba aún se encuentra poco difundida y estudiada. (Vidal, 2013).

Dentro del grupo de los algoritmos heurísticos se pueden distinguir los denominados métodos metaheurísticos, que imitan fenómenos simples observados en la naturaleza y que están asociados con la inteligencia artificial. Estos algoritmos (evolutivos) tratan de adaptar el comportamiento de diferentes especies a soluciones de problemas altamente complejos mediante optimización.(Chen, 2013).

#### **1.4.1 Ventajas y desventajas de las heurísticas de optimización.**

Ventajas:

- No necesitan conocimientos específicos del problema a resolver.
- Operan de forma simultánea con varias soluciones y no de forma secuencial como las técnicas tradicionales.
- Son fáciles de ejecutar en modernas arquitecturas masivamente paralelas.
- Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras operaciones.
- Trabajan con un código del conjunto de parámetros, no con el conjunto mismo. Por trabajar a nivel de código, y no con las funciones y sus variables de control, como los otros métodos, es más difícil que la solución converja a un mínimo o máximo local.
- Buscan una población de puntos, no un único punto. Manteniendo una población de puntos muestrales bien adaptados, se reduce la probabilidad de caer en una solución falsa.
- Emplean la función objetivo, no necesitan derivadas ni otra información complementaria, tan difícil a veces de conseguir. De este modo ganan en eficiencia y en generalidad.

- Cuando se usan para problemas de optimización resultan menos afectados por los máximos o mínimos locales que las técnicas tradicionales.
- Se valen de reglas de transición estocástica, no deterministas. Los Algoritmos Genéticos se valen de operadores aleatorios para guiar la búsqueda de los mejores puntos.

Desventajas:

- Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en gran medida de los parámetros que se utilicen: tamaño de la población, número de generaciones y porcentajes de los operadores genéticos (Coello et al., 2002), (Vamplew et al., 2010, Hultman, 2010).
- Pueden converger prematuramente debido a una serie de problemas de diversa índole. (Hultman, 2010), (Coello et al., 2002)

## **1.5 Armaduras planas. Conceptualización, utilización, principales tipologías.**

Uno de los elementos estructurales más usados en la actualidad son las cerchas o armaduras, las cuales soportan cargas elevadas y cubren grandes luces, generalmente se utilizan en cubiertas de techos y puentes.

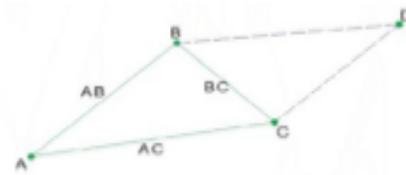
### **1.5.1 Armaduras planas**

Es una estructura reticulada simple formada por elementos rectos de sección constante, cuya longitud supera varias veces su sección transversal, se conocen como barras y se

conectan rígidamente en sus extremos denominados nodos o nudos, los esfuerzos actúan a lo largo de su eje longitudinal. Las armaduras planas o cerchas se utilizan para soportar cargas elevadas y cubrir grandes luces, pueden construirse en maderas o acero y usadas en cubiertas de techos, puentes, grúas, torres, entre otros.

Una armadura plana se idealiza como un sistema de miembros en un plano e interconectados en juntas articuladas. Las cargas pueden consistir de fuerzas concentradas en los nudos que dará como resultado fuerzas axiales de tensión y compresión en los elementos.

Con el fin de obtener la rigidez de la armadura las barras deben tener una disposición triangular, por ser geoméricamente una figura indeformable, unidas de dos en dos en sus extremos. Las uniones de las barras se llaman nudos, nudos o juntas y se resuelven generalmente con placas metálicas llamadas cartelas. Partiendo del triángulo base, formado por 3 nudos (ABC) y tres barras (AB, AC, BC) por cada nuevo nudo (D), se necesitan dos barras (BD, CD), no alineadas, para formar un nuevo triángulo, generando estructuras rígidas(Chavez, 2018).



*Figura 1.4 Triángulo base (Chavez, 2018)*

Condición de rigidez de las armaduras

La rigidez de una armadura está determinada por su capacidad de mantener la forma original luego de ser aplicadas las cargas de trabajo. La rigidez mide la estabilidad estructural de la armadura.

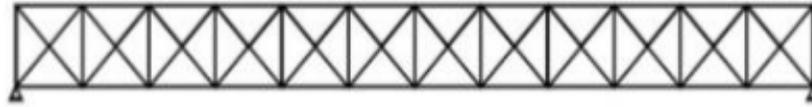
Las armaduras se equilibran externamente mediante apoyos, en muchas ocasiones isostáticos. Los extremos de cada barra son articulaciones de pasador permitiendo el giro, alrededor del nudo, el sistema de fuerzas sobre el nudo es concurrente. Cada barra de la armadura se encuentra sometida a un sistema de dos fuerzas, axiales, iguales, opuestas y colineales, que la mantienen en equilibrio.

### **1.5.2 Tipologías de armaduras planas.**

Existen diferentes tipologías pero su diferencia radica en la disposición de sus elementos, a continuación se describen algunos de los tipos de armaduras más usadas en la ingeniería.(Bellido et al., 2015).

#### **Armaduras Long**

Este tipo de armadura debe su nombre a Stephen H. Long (1784-1864), y tiene su origen hacia 1835. Los cordones superior e inferior horizontales se unen mediante montantes verticales todos ellos arriostrados por diagonales dobles, usados para aumentar la rigidez de la estructura y su capacidad de resistir cargas laterales, tales como los movimientos sísmicos y la presión de los vientos huracanados.



- Figura 1.5 Armaduras Long(Bellido et al., 2015).

### **Armaduras Howe**

La armadura *Howe* es la inversa a la *Pratt*, fue patentada en 1840 por Wiliam Howe, aunque ya había sido usada con anterioridad. Se usó mucho en el diseño de celosías de madera, está compuesta por montantes verticales entre el cordón superior e inferior. Las diagonales se unen en sus extremos donde coincide un montante con el cordón superior o inferior. Con esa disposición se lograba que los elementos verticales, que eran metálicos y más cortos estuvieran tensionados, mientras que las diagonales más largas estaban comprimidas, lo cual era económico puesto que los elementos metálicos eran más caros y con esta disposición *Howe* se minimizaba su longitud.

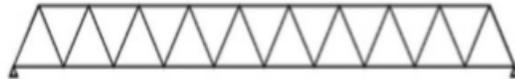


Figura 1.6 Armaduras tipo Howe(Bellido et al., 2015)

### **Armaduras Warren**

La armadura Warren, fue patentada por los ingleses James Warren y Willboughby Monzony, en 1840. El rasgo característico de este tipo de armaduras es que forman una serie de triángulos isósceles (o equiláteros), de manera que todas las diagonales tienen la misma longitud. Típicamente en una celosía de este tipo y con cargas aplicadas verticales en sus nudos superiores, las diagonales presentan alternativamente compresión y tracción.

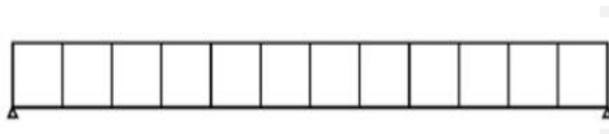
Esto, que es desfavorable desde el punto de vista resistente, presenta en cambio una ventaja constructiva. Si las cargas son variables sobre la parte superior de la celosía (como por ejemplo en una pasarela) las armaduras presentan resistencia similar para diversas configuraciones de carga.



*Figura 1.7 Armadura tipo Warren(Bellido et al., 2015)*

### **Armaduras Vierendeel**

La armadura Vierendeel, en honor al ingeniero belga A. Vierendeel, tiene como características principales: Las uniones obligatoriamente rígidas, Ausencia de diagonales inclinadas. De esta manera no aparecen formas triangulares como en la mayoría de las armaduras, sino una serie de marcos rectangulares. Se trata por tanto de una armadura empleada en edificación por el aprovechamiento de sus aperturas.



*Figura 1.8 Armaduras Vierendeel.(Bellido et al., 2015)*

### **Armaduras Pratt Plana**

Representa la adaptación de las armaduras al uso más generalizado de un nuevo material de construcción de la época: el acero. A diferencia de una armadura *Howe*, las barras está inclinadas en sentido contrario, de manera que las diagonales están sometidas a tensión, mientras que las barras verticales están comprimidas.

En esencia tiene una tipología y uso muy parecidos al Warren. Para la armadura de cuerdas paralelas, la Pratt ofrece las ventajas: menos efecto de pandeo (los miembros más largos a tracción y los cortos a compresión).

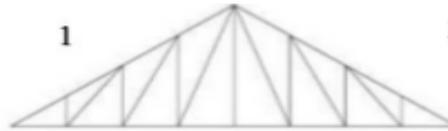


Figura 1.9 Armaduras Pratt.(Bellido et al., 2015)

### Armaduras Fink

Para techos mayores (más de 15 grados de pendiente) las armaduras *Fink* son muy usadas, las *Howe* y *Pratt* también pueden usarse pero no son tan económicas. Un hecho que la hace más económica es que la mayoría de sus miembros está en tensión, mientras que los sujetos a compresión son bastante cortos, además es importante saber que la triangulación de una armadura se proyecta tomando en cuenta el esparcimiento de los largueros, ya que usualmente es conveniente localizar los largueros sólo en los vértices de los triángulos, la triangulación principal puede subdividirse. La armadura *Fink* puede ser dividida en un gran número de triángulos y coincidir con cualquier esparcimiento de largueros.

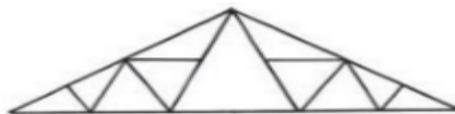
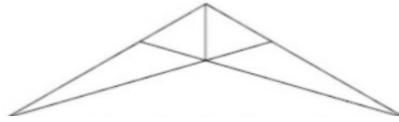


Figura 1.10 Armadura tipo Fink(Bellido et al., 2015)

### Armadura tipo tijera

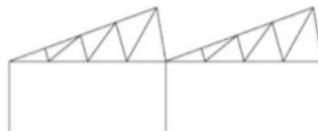
Cercha a dos aguas cuyas barras de tracción van desde la base de cada cordón hasta el punto medio del otro cordón superior. Se puede emplear para distancias pequeñas que requieran altura libre superior.



*Figura 1.11 Armadura tipo tijera.(Bellido et al., 2015)*

### **Armadura Diente de sierra**

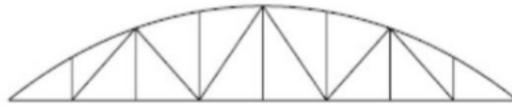
Estas armaduras pueden utilizarse cuando la separación entre columnas no es objetable y se desea una iluminación natural adecuada por medio de ventanales en construcciones anchas. Sus caras más inclinadas llevan los ventanales y están generalmente orientadas al norte para una iluminación difusa más pareja. Estructuralmente es una estructura aporticada muy eficiente y se usa mucho en fábricas de textiles.



*Figura 1.12 Armadura Diente de sierra.(Bellido et al., 2015)*

### **Armadura de cuerda y arco**

Se emplea para garajes, gimnasios y hangares pequeños. En este caso el cordón superior tiende a la configuración de un arco lo que implica una buena distribución de las tensiones en sus elementos.



*Figura 1.13 Armadura de cuerda y arco.(Bellido et al., 2015)*

## **1.6 Software**

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

Por sus características es un software utilizado ampliamente en el mundo académico y científico, dentro del cual se destaca el campo de la optimización mediante algoritmos evolutivos(Coello, 2012), (Coello et al., 2002).

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

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

Mediante SAP2000 es posible modelar complejas geometrías, definir diversos estados de carga, generar pesos propios automáticamente, asignar secciones, materiales, así como realizar cálculos estructurales de hormigón y acero basados en variadas normativas.(Chavez, 2018)

## **1.7 Conclusiones Parciales**

1. Las armaduras planas son elementos estructurales simples que trabajan a flexión, compuestas por elementos dispuestos de forma que sean capaces de resistir las cargas y transmitirlos a los apoyos actuando como una unidad, su forma más simple es la triangular, por ser esa figura geométrica la que proporciona mayor rigidez y estabilidad, trabajan en conjunto como una viga, soportan cargas de flexión, pero sus miembros individualmente soportan cargas de tracción y compresión, se emplean para cubrir grandes luces, se construyen de madera, acero u hormigón armado.

2. Las heurísticas de optimización se muestran en la actualidad como una opción muy tentativa en la optimización estructural por las ventajas que presentan sobre las técnicas tradicionales de diseño, pues han demostrado que son capaces de encontrar soluciones adecuadas en un tiempo computacionalmente aceptable. Empleando estas técnicas se pueden optimizar recursos materiales, tiempo de ejecución, factibilidad, costos, entre otros, lo que facilita mejoras no solo en la calidad sino también en la economía. Una de las ramas en la que más se emplean estas técnicas es la optimización estructural

## **CAPÍTULO 2 ANÁLISIS COMPARATIVO DEL DESEMPEÑO DE LAS DIFERENTES TÉCNICAS HEURÍSTICAS DE OPTIMIZACIÓN**

### **2.1 Generalidades.**

En este capítulo se realiza una descripción de los principios de funcionamiento de los algoritmos utilizados para comparar su desempeño, así como de las funciones de pruebas utilizadas con este propósito, además se realiza una breve descripción del proceso de obtención de los resultados del rendimiento de las heurísticas en estos problemas.

### **2.2 Técnicas heurísticas empleadas para el estudio**

#### **2.2.1 *Not sorting genetic algorithms*II (NSGAI).**

Los algoritmos genéticos son métodos metaheurísticos de optimización basados en un planteamiento probabilístico que además realizan la búsqueda por poblaciones. Su importancia radica en que es un método capaz de converger con muy alta probabilidad en un óptimo global, a diferencia de los métodos basados en gradientes que eran los que más se utilizaban hasta la aparición de estos nuevos algoritmos.

Algunos autores han reportado que NSGA (*Notsorting genetic algorithms*) tienen un rendimiento algo más bajo que MOGA (*multi-objective genetic algorithms*), además de que NSGA se acerca más a la frontera de Pareto que sus predecesores en algunos tipos de problemas.

El NSGA-II parece ser notablemente más eficiente que su predecesor NSGA, pero también presenta una cuestionable capacidad exploratoria, además el algoritmo tiende a esparcirse rápidamente y de modo apropiado en ciertas regiones con soluciones no

dominadas sin embargo parece presentar problemas en regiones con características distintas a estas dentro del espacio de búsqueda, también se ha demostrado que no es muy efectivo a medida que aumenta el número de objetivos.

### **2.2.2 *Not sorting genetic algorithms III (NSGAIII)***

Las funciones con muchos objetivos se presentan como un reto para los algoritmos evolutivos basados en la dominancia de Pareto clásica (MOEAs *Multi objective evolutionary algorithms*) incluido NSGA II, esto es debido a que la calidad de los resultados, es decir la posibilidad de los algoritmos de acercarse o predecir la frontera real de Pareto se hace más baja con el aumento de la cantidad de objetivos a analizar.

Más recientemente ha surgido NSGA III este parece tratar mejor los problemas multiobjetivos, en este caso se mantiene la diversidad entre los miembros de la población mediante el suministro y la actualización adaptativa de una serie de puntos de referencia bien distribuidos, sin embargo todavía depende de la dominancia de Pareto para acercar la población hacia el frente de Pareto, dejando así la posibilidad de la ocurrencia de la no convergencia.

Recientemente han surgido nuevas mejoras a este algoritmo precisamente tratando de suprimir esta última limitación ( $\theta$ -NSGA-III)

### **2.2.3 Multi Objective Cross Entropy (MOCE+).**

Las ventajas de los algoritmos de estimación de distribución (Estimation-of-Distribution algorithms) y los métodos específicos de entropía cruzada (*crossentropy*) son ampliamente reconocidos

(MOCE+) es una propuesta de este tipo de algoritmos, el cual está basado en un nuevo procedimiento para abordar las restricciones: (i) el uso de valores límites para seleccionar la población elitista, filtrando esta población después de cada iteración.(Beruvides et al., 2016)

### **2.2.4 Simple Multi Objective Cross Entropy (SMOCE)**

Este algoritmo *Simple Multi-Objective Cross-Entropy method* (SMOCE) está basado en su antecesor MOCE+, como en todas las técnicas de optimización multiobjetivo SMOCE se centra en determinar los mínimos de cualquier función, el algoritmo base de este método se representa a continuación(Haber et al., 2017).

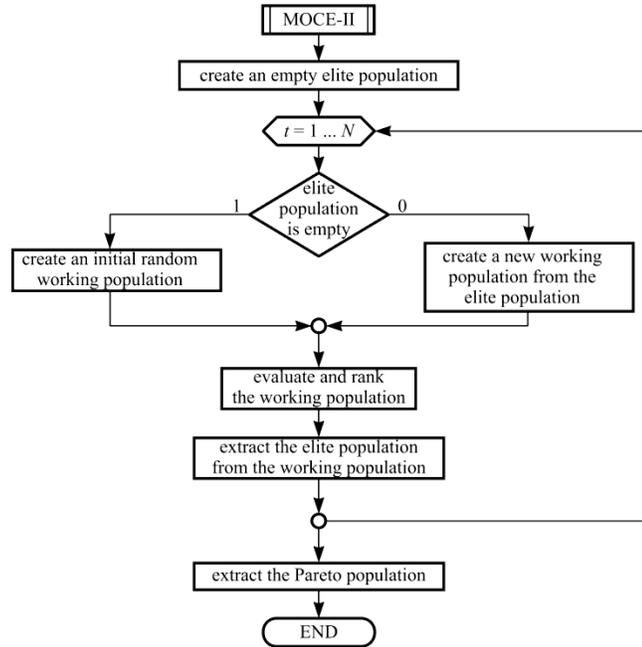


Figura 2.1 Algoritmo de SMOCE(Haber et al., 2017)

El proceso evolutivo en este algoritmo tiene lugar en un solo lazo con una única condición de parada (basado en *epoch number*, N), estas son dos de las ventajas de SMOCE sobre su antecesor MOCE+ donde eran considerados dos lazos y tres condiciones de parada, esta modificación permite la reducción de la cantidad de evaluaciones de la función objetivo y la eliminación de los parámetros de máximo número de evaluaciones y el límite de convergencia.

En este algoritmo en la primera iteración se trabaja con una población creada al azar y en las siguientes iteraciones se va creando una población a partir de la anterior y se evalúa los valores de la función objetivo después de creada la población.

En este caso las mejores soluciones (*elite solutions*) son incluidas en la próxima población lo cual introduce elitismo en el algoritmo (SMOCE), presentándose esto como otra ventaja de este método sobre MOCE+(Haber et al., 2017).

### **2.2.5 Multi-objective evolutionary algorithm based on decomposition(MOEAD)**

Normalmente las diferentes variantes de MOEA/D (*multi-objective evolutionary algorithm based on decomposition*) no presentan un mecanismo de normalización, debido a esto muestran dificultades para ajustarse correctamente al frente real de Pareto de los problemas tratados, es por ello que algunas de las variantes más recientes de MOEA/D si presentan este mecanismo de normalización (Ke et al., 2013).

Este algoritmo está basado en descomponer en varios problemas de un solo objetivo el problema real que es de hecho un problema multiobjetivo. El objetivo de cada subproblema es una agregación ponderada (lineal o no lineal) de los objetivos individuales, la relación de cada uno de los problemas con sus vecinos son definidas a partir de sus vectores de ponderación, de este modo cada subproblema es optimizado a partir de la información de sus vecinos (Ke et al., 2013).

### **2.2.6 Multi objective particle swarm optimization(MOPSO)**

*Particle swarm optimization* (PSO) es un método computacional que intenta optimizar un problema a través de iteraciones sucesivas, tratando de obtener los mejores candidatos de solución, a partir de cierta medida de calidad.(Derrac et al., 2011)

En esta técnica las posibles soluciones o partículas se van moviendo por el espacio de búsqueda a medida que sus posiciones son evaluadas, de modo que las partículas que alcanzan mejores posiciones (mejores soluciones) atraen al resto hacia esos espacios, logrando así encontrar las mejores ubicaciones posibles, lo más cercano a la frontera real de Pareto como puedan (Coello and Zalazar, 2003).

### **2.2.7 Strength Pareto Evolutionary Algorithm (SPEA-II)**

El SPEA fue introducido por Eckart Zitzler y Lothar Thiele, esta técnica fue concebida como un modo de integrar diferentes algoritmos evolutivos. SPEA usa un archivo externo conteniendo soluciones no dominadas previamente encontradas. En cada generación, los individuos no dominados son enviados para el conjunto no dominado inicial (externo). Para cada individuo en este set externo, un valor de desempeño es computado. El *fitness* de cada miembro de la población actual es computado según el desempeño de todas las soluciones no dominadas externas que lo dominan (Zitzler et al., 2001).

En este algoritmo se usa el criterio de la dominancia de Pareto para asegurar que la población está correctamente esparcida a lo largo de la frontera de Pareto. De hecho, si las soluciones no dominadas en el conjunto externo se vuelven demasiado numerosas, podría reducir la presión de selección, desacelerando así la búsqueda. Debido a esto sus autores decidieron adoptar una técnica para cortar el contenido de las soluciones no dominadas, garantizando así que se mantuvieran por debajo de cierta frontera, esta es la denominada *average link age method* (Coello et al., 2002).

SPEA2 tiene tres diferencias fundamentales con respecto a su predecesor SPEA

Incorpora una refinada técnica de *fitness* o adaptación que tomo en cuenta para cada individuo el número de individuos que lo dominan a él y el número de individuos que el domina.

Usa como técnica la llamada *a nearest neighbor density estimation* (estimación de la densidad de vecino más cercano) lo cual conduce la búsqueda de modo más eficiente. Tiene un método mejorado de truncamiento que garantiza la preservación de soluciones cercanas a la frontera. Esta técnica (SPEAII) y NSGAII son dos de las más usadas en la comparación del desempeño de nuevas heurísticas desarrolladas (Coello et al., 2002).

### **2.2.8 Pareto Envelope-based Selection Algorithm (PESA II)**

El *Pareto Envelope-based Selection Algorithm* (PESA) fue desarrollado por Corne. Este algoritmo (PESA) está compuesto por una población externa grande y una población interna considerablemente menor. Se utiliza una división tipo rejilla (muy fina) del espacio para mantener la diversidad de la selección mientras se ejecuta el MOEA, esta medida se utiliza para permitir soluciones en la población externa a través de un archivo de soluciones que evalúan vectores no dominados (Coello et al., 2002).

En PESA II se incluye una diferencia con respecto a PESA, en este caso la selección se hace por regiones y los sujetos son seleccionados no individualmente, sino por paquetes o *hiperbox*, este cambio esta motivado por el propósito de disminuir el costo computacional asociado al *ranking* de Pareto (Li et al., 2013).

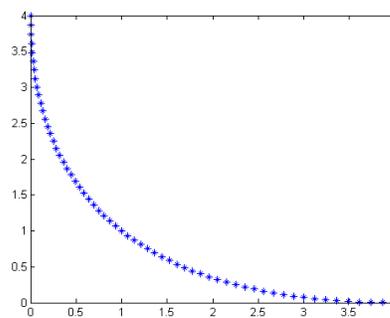
### 2.3 Funciones de prueba.

Este término se refiere a funciones preestablecidas o generadas a partir de ciertas condiciones, las mismas son muy variadas y van desde muy sencillas hasta muy complejas, su principal característica y razón de su existencia es que se conoce su frontera real de Pareto, lo que las hace muy útil a la hora de evaluar el desempeño de las diferentes heurísticas, pues se puede determinar con qué nivel de precisión son capaces de encontrar este frente de Pareto cada una de las técnicas de búsqueda (Huband et al., 2006), (Deb et al., 2001).

**MOP1** (Deb et al., 2001)

En este caso se trata de una función con frente de Pareto continuo y cóncavo, es una función relativamente sencilla.

$$\begin{aligned} f_1(x) &= x^2 \\ f_2(x) &= (x-2)^2 \end{aligned} \quad \left[ -10^5, 10^5 \right] \quad \text{ecuación 2.1}$$



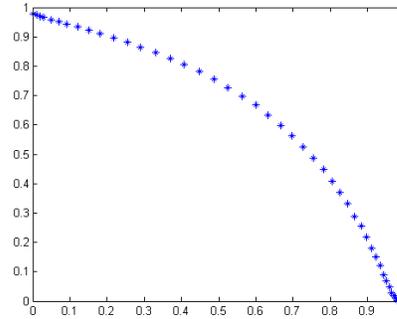
*Figura 2.1 Frontera real de Pareto de la función MOP1 (Elaborada por el autor).*

**MOP2**(Deb et al., 2001)

Se trata de una función algo más compleja que MOP1, en este caso tiene un frente de Pareto también continuo pero, a diferencia de MOP, este es convexo.

$$\begin{aligned}
 f_1(x, \dots, x_n) &= 1 - \exp\left(-\sum_{i=1}^n (x_i - 1/\sqrt{n})^2\right) \\
 f_2(x, \dots, x_n) &= 1 + \exp\left(-\sum_{i=1}^n (x_i - 1/\sqrt{n})^2\right)
 \end{aligned}
 \quad [-4, 4]$$

ecuación 2.2



*Figura 2.2 Frontera real de Pareto de la función MOP2 (Elaborada por el autor).*

**MOP3**(Deb et al., 2001)

Se trata de una función con frente de Pareto cóncavo pero, en este caso, a diferencia de MOP1 y MOP2 es discontinuo, lo cual agrega algo de complejidad.

$$\begin{aligned}
 \text{Max, } f_1(x_1, x_2) &= -1 - (A_1 - B_1)^2 - (A_2 - B_2)^2 \\
 \text{Max, } f_2(x_1, x_2) &= -(x_1 + 3)^2 - (x_2 + 1)^2 \\
 A_1 &= 0,5 \text{sen} 1 - 2 \cos 1 + \text{sen} 2 - 1,5 \cos 2 \\
 A_2 &= 1,5 \text{sen} 1 - \cos 1 + 2 \text{sen} 2 - 0,5 \cos 2 \\
 B_1 &= 0,5 \text{sen} x_1 - 2 \cos x_1 + \text{sen} x_2 - 1,5 \cos x_2 \\
 B_2 &= 1,5 \text{sen} x_1 - \cos x_1 + 2 \text{sen} x_2 - 0,5 \cos x_2
 \end{aligned}
 \quad [-\pi, \pi]$$

ecuación 2.3

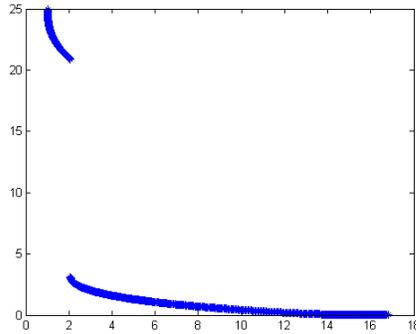


Figura 2.3 Frontera real de Pareto de la función MOP3 (Elaborada por el autor).

**MOP4**(Deb et al., 2001)

Similar al MOP3, pero en este caso presenta más de una discontinuidad.

$$f_1(x_1, x_2, x_3) = \sum_{i=1}^2 -10 \exp^{-0,2\sqrt{x_i^2 + x_{i+1}^2}} \quad [-5, 5] \quad \text{ecuación 2.4}$$

$$f_2(x_1, x_2, x_3) = \sum_{i=1}^3 |x_i|^{0,8} + 5 \text{sen}(x_i^3)$$

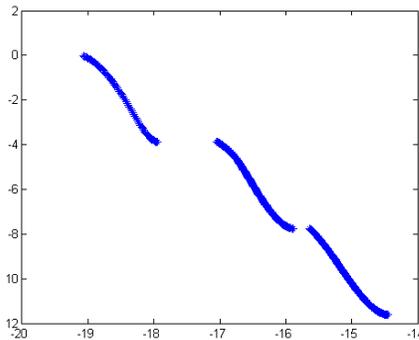


Figura 2.4 Frontera real de Pareto de la función MOP4 (Elaborada por el autor).

**MOP6**(Deb et al., 2001)

Esta función es similar a MOP3 y MOP4 pero con más discontinuidades que las dos anteriores.

$$f_1(x_1) = x_1$$

$$f_2(x_1, x_2) = (1 + 10x_2) \left( 1 - \left( \frac{x_1}{1 + 10x_2} \right)^2 - \frac{x_1}{1 + 10x_2} \text{sen}(8\pi x_1) \right)$$

[0,1] ecuación 2.5

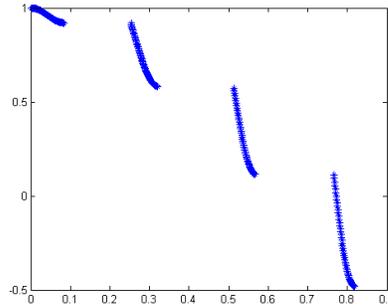


Figura 2.5 Frontera real de Pareto de la función MOP6 (Elaborada por el autor).

**ZDT1**(Deb et al., 2001)

Esta función tiene un frente de Pareto convexo y continuo. Su función y dominio se describen a continuación.

$$f_1 = y_1$$

$$g = 1 + 9 \sum_{i=1}^k \frac{z_i}{k}$$

$$h = 1 - \sqrt{f_1 / g}$$

[0,1]

ecuación 2.6

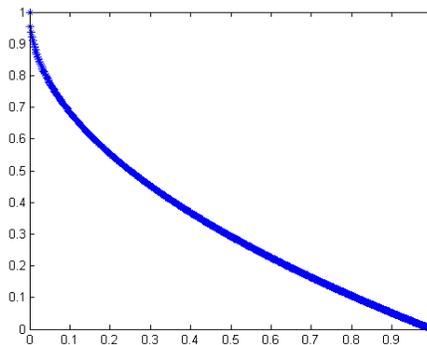


Figura 2.6 Frontera real de Pareto de la función ZDT1 (Elaborada por el autor).

### ZDT2(Deb et al., 2001)

Al igual que la función ZDT1 tiene un frente de Pareto continuo, pero en este caso cóncavo. Su función y dominio se describen a continuación.

$$\begin{aligned} f_1 &= y_1 \\ g &= 1 + 9 \sum_{i=1}^k \frac{z_i}{k} \\ h &= 1 - \left( \frac{f_1}{g} \right)^2 \end{aligned} \quad [0,1] \quad \text{ecuación 2.7}$$

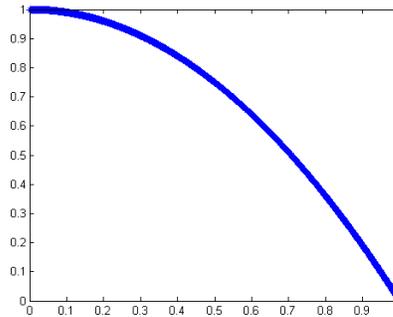


Figura 2.7 Frontera real de Pareto de la función ZDT2 (Elaborada por el autor).

### ZDT3(Deb et al., 2001)

A diferencia de ZDT1 y ZDT2 tiene un frente de Pareto discontinua, convexo al igual que ZDT1 y diferente a ZDT2. Su función y dominio se describen a continuación

$$\begin{aligned} f_1 &= y_1 \\ g &= 1 + 9 \sum_{i=1}^k \frac{z_i}{k} \\ h &= 1 - \sqrt{\frac{f_1}{g} - \left( \frac{f_1}{g} \right)^2} \text{sen}(10\pi f_1) \end{aligned} \quad [0,1] \quad \text{ecuación 2.8}$$

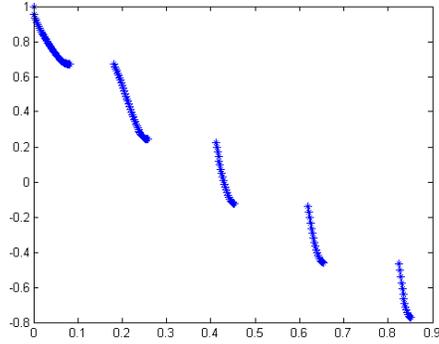


Figura 2.8 Frontera real de Pareto de la función ZDT3 (Elaborada por el autor).

**WFG2**(Deb et al., 2001)

$$h_{m-1:M-1} = \text{convex}$$

$$h_M = \text{disc}_M (\alpha = \beta = 1, A = 5)$$

$$t_{i=1:k}^2 = y_i$$

$$t_{i=k+1:k+t/2}^2 = r - \text{nonsep} \left( \left\{ y_{k+2(i-k)-1}, y_{k+2(i-k)} \right\}, 2 \right)$$

ecuación 2.9

$$t_{i=1:M-1}^3 = r - \text{sum} \left( \left\{ y_{(i-1)k/(M-1)+1}, \dots, y_{ik/(M-1)} \right\}, \{1, \dots, 1\} \right)$$

$$t_M^3 = r - \text{sum} \left( \left\{ y_{k+1}, \dots, y_{k+i/2} \right\}, \{1, \dots, 1\} \right)$$

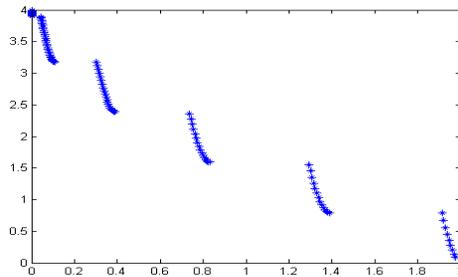


Figura 2.9 Frontera real de Pareto de la función WFG2 (Elaborada por el autor).

**WFG3**(Deb et al., 2001)

$$h_{m=1:M} = \text{linear}_m (\text{degenerate})$$

$$t^{1:3} = \text{idemWFG2}$$

ecuación 2.10

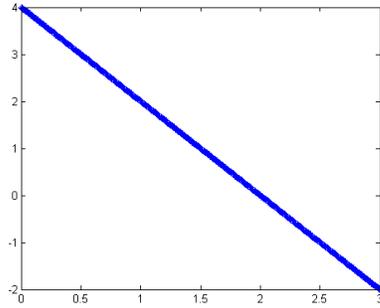


Figura 2.10 Frontera real de Pareto de la función WFG2 (Elaborada por el autor).

**WFG4**(Deb et al., 2001)

$$h_{m-1:M} = \text{concave}_m$$

$$t_{i=1:n}^1 = s - \text{multi}(y_i, 30, 10, 0.35)$$

$$t_{i=1:M-1}^2 = r - \text{sum}\left(\{y_{(i-1)k/(M-1)+1}, \dots, y_{ik/(M-1)}\}, \{1, \dots, 1\}\right)$$

ecuación 2.11

$$t_M^2 = r - \text{sum}\left(\{y_{k+1}, \dots, y_n\}, \{1, \dots, 1\}\right)$$

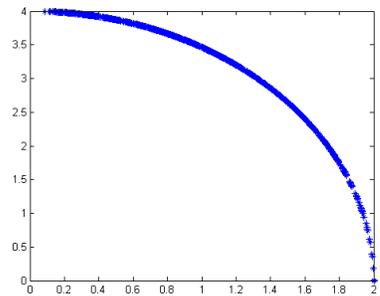


Figura 2.11 Frontera real de Pareto de la función WFG4 (Elaborada por el autor).

**WFG5**(Deb et al., 2001)

$$h_{m-1:M} = \text{concave}_m$$

$$t_{i=1:n}^1 = s.\text{decept}(y_i, 0.35, 0.001, 0.05)$$

$$t^2 = \text{idemWFG4}$$

ecuación 2.12

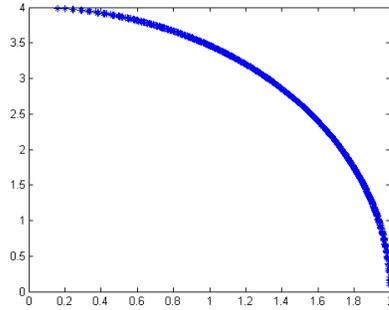


Figura 2.12 Frontera real de Pareto de la función WFG5 (Elaborada por el autor).

**WFG6**(Deb et al., 2001)

$$h_{m-1:M} = \text{concave}_m$$

$$t^1 = \text{idemWFG1}$$

$$t_{i=1:M-1}^2 = r - \text{nonsep}\left(\{y_{(i-1)k/(M-1)+1}, \dots, y_{ik/(M-1)}\}, k / (M - 1)\right)$$

ecuación 2.13

$$t_M^2 = r - \text{nonsep}\left(\{y_{k+1}, \dots, y_n\}, l\right)$$

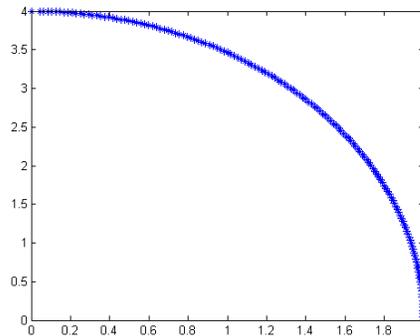


Figura 2.13 Frontera real de Pareto de la función WFG6 (Elaborada por el autor).

## 2.4 Parámetros que influyen en el desempeño de las heurísticas

Todas las simulaciones realizadas se hicieron en una computadora personal con Intel Core i3-2120 CPU (3.30 GHz) and 4.0 GB RAM. Durante el tiempo que duraba el experimento con cada algoritmo solo se dedicaba la computadora a esta tarea, no

realizándose ninguna otra que pudiera afectar el tiempo de ejecución, uno de los parámetros a medir en el estudio.

En todos los casos el objetivo es obtener para cada problema de prueba, usando cada uno de los algoritmos antes mencionados el desempeño de estos últimos en cuanto a la calidad de los resultados (aproximación a la frontera real de Pareto) y tiempo de ejecución, para ello se efectuaron varias iteraciones, variando solamente los parámetros cantidad de generaciones (G o Z) y tamaño de la población (N) hasta obtener los mejores resultados posibles (soluciones no dominadas)(Chen and Zou, 2014).

A continuación, se muestra un ejemplo con una función de prueba y uno de los algoritmos seleccionados, en el resto de los casos el proceso fue idéntico. Se nota como al variar los parámetros cantidad de generaciones y tamaño de la población varían el hiperárea *ratio* y el tiempo de ejecución.

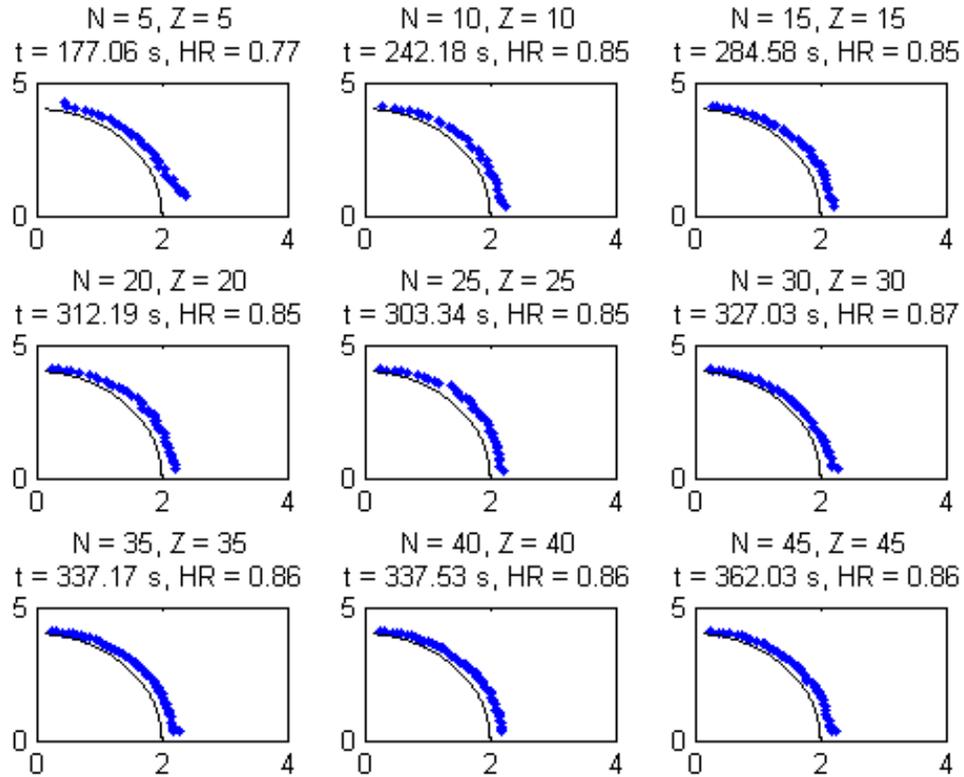


Figura 2.14 Proceso iterativo de un algoritmo para obtener la mejor frontera de Pareto posible HR (hiper área ratio),  $t$  (tiempo de ejecución) (Elaborado por el autor)

## 2.5 Métricas para realizar la comparación.

En este caso se usan como parámetros para medir el desempeño de los algoritmos ante las funciones de prueba el tiempo de ejecución o de cómputo y la relación de hiperárea, este último representa la calidad de los resultados pues es una buena medida de que tan cercano se encuentra la frontera de Pareto alcanzada por el algoritmo y la frontera real de Pareto.

## 2.6 Comparación.

*Tabla 2.1 Soluciones no dominadas obtenidas con cada heurística ante diferentes funciones de prueba. N (tamaño de la población, G (cantidad de generaciones), HR (hiper-área ratio), T (tiempo de ejecución en segundos)*

Problema	NSGA II				NSGA III				MOCE+				SMOCE			
	N	G	HR	T	N	G	HR	T	N	G	HR	T	N	G	HR	T
MOP1	250	300	1	23.7	30	30	0.91	14	15	10	1	5.46	2505	525	0.99	5.2
MOP2	300	300	1	42.8	60	60	0.99	82.2	15	10	1	7.58	10	1000	1	6.3
MOP3	250	300	1	7	60	60	0.98	82.6	15	10	1	3.75	2505	525	1	3.4
MOP4	250	250	1	66.3	60	60	0.98	82.4	15	10	0.99	21.6	2505	525	0.99	20.3
MOP6	300	300	1	57.1	60	60	0.98	81.3	15	10	1	13.2	2505	525	1	11.5
ZDT1	300	300	0.79	45.3	40	40	0.91	24.2	15	10	1	88.5	2505	525	0.97	80.2
ZDT2	300	300	0.56	131	40	40	0.9	25.4	15	10	0.94	88.6	2505	525	0.62	81.6
ZDT3	300	300	0.53	126	40	40	0.91	25.6	15	10	0.85	69.4	2505	525	0.82	62.1
WFG2	688	172	0.86	60	50	50	0.79	53	68	17	1	60	2505	525	1	75
WFG3	768	192	0.73	73	60	60	0.73	85.4	68	17	0.72	80	2505	525	1	65
WFG4	1008	252	1	63	40	40	0.79	26.1	68	17	0.77	65	2505	525	1	62
WFG5	1008	252	0.99	61	60	60	0.73	83.3	76	19	0.77	71	2505	525	1	62
WFG6	736	184	0.81	77	60	60	0.81	82.4	76	19	0.71	65	2505	525	0.82	77

*Tabla 2.2 Soluciones no dominadas obtenidas con cada heurística ante diferentes funciones de prueba. N (tamaño de la población, G (cantidad de generaciones), HR (hiper-área ratio), T (tiempo de ejecución en segundos)*

Problema	MOEAD				MOPSO				SPEA II				PESA II			
	N	G	HR	T	N	G	HR	T	N	G	HR	T	N	G	HR	T
MOP1	80	80	0.99	149	180	180	0.99	15.3	90	90	1	39.6	100	100	0.75	95.4
MOP2	60	60	0.93	78.7	180	180	0.97	14.7	90	90	0.93	39.8	100	100	0.76	97.8
MOP3	60	60	0.97	73.9	180	180	1	42.2	90	90	1	39.8	90	90	0.99	91.5
MOP4	60	60	0.7	50.9	180	180	0.95	11.1	90	90	0.94	39	120	120	0.87	92.5
MOP6	60	60	0.92	70.1	180	180	1	21.9	90	90	0.99	39.7	120	120	0.88	92.4
ZDT1	70	70	0.89	101	180	180	1	41.7	60	60	0.97	72	90	90	0.97	74.4
ZDT2	80	80	0.65	120	80	80	0.83	14.6	60	60	0.83	70.9	90	90	0.98	75.5
ZDT3	70	70	0.35	90.2	300	300	0.89	90.7	60	60	0.68	71.6	90	90	0.76	74.5
WFG2	264	66	0.81	63	532	133	0.77	71	224	56	0.57	344	344	86	0.82	63
WFG3	256	64	0.74	73	308	77	0.75	65	248	62	0.52	74	320	80	0.73	74
WFG4	256	64	0.69	66	340	85	0.77	80	248	62	0.7	67	336	84	0.81	65
WFG5	256	64	0.77	65	280	70	0.78	80	248	62	0.6	67	320	80	0.75	67
WFG6	264	66	0.72	66	308	77	0.69	70	248	62	0.52	77	320	80	0.75	66

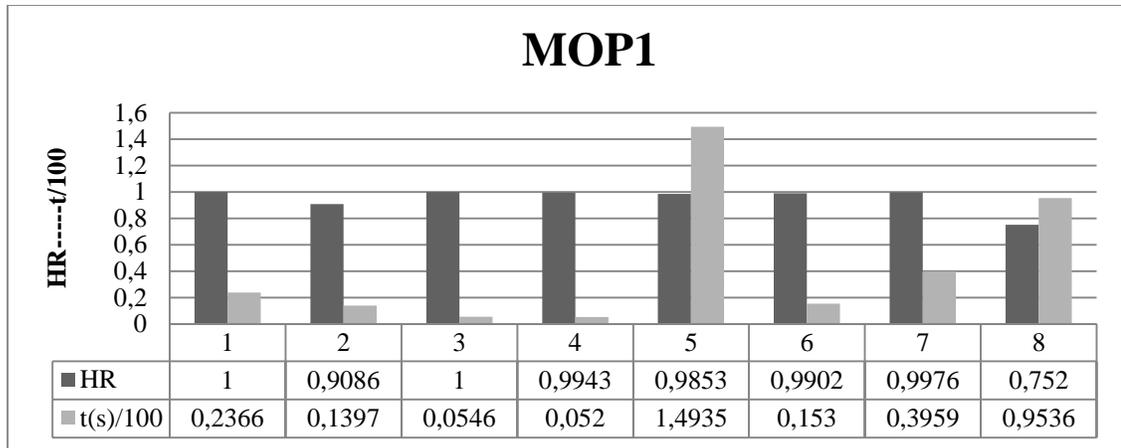


Figura 2.15 Desempeño de las diferentes heurísticas en la solución del problema MOP1  
 NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II  
 (7), PESA II (8)

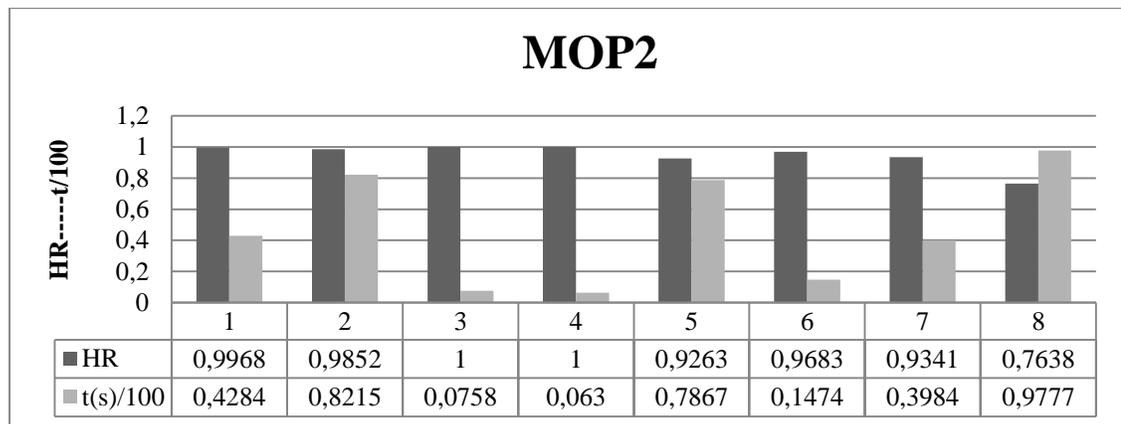
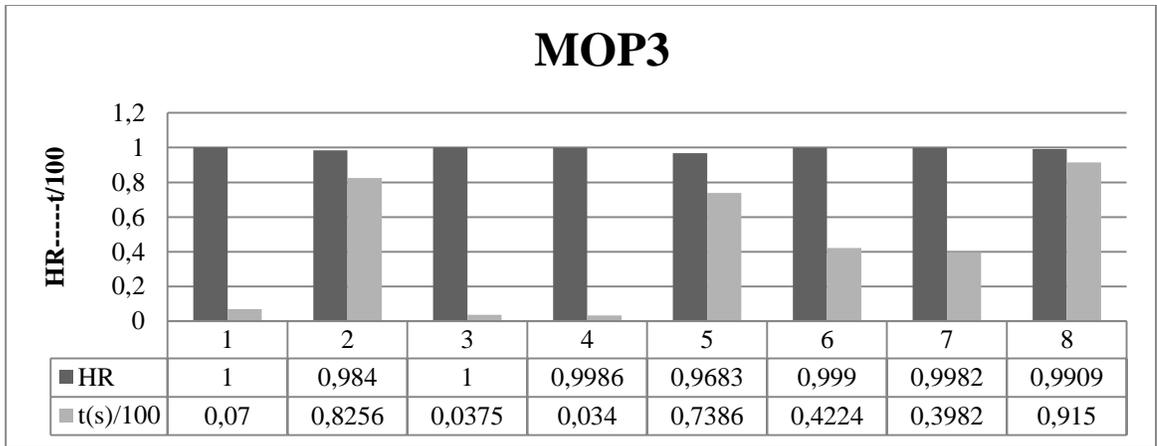
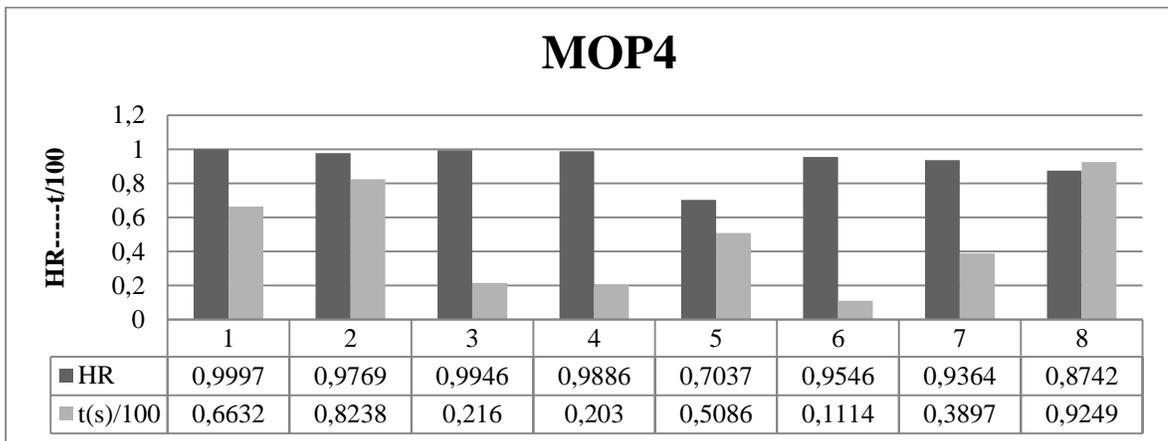


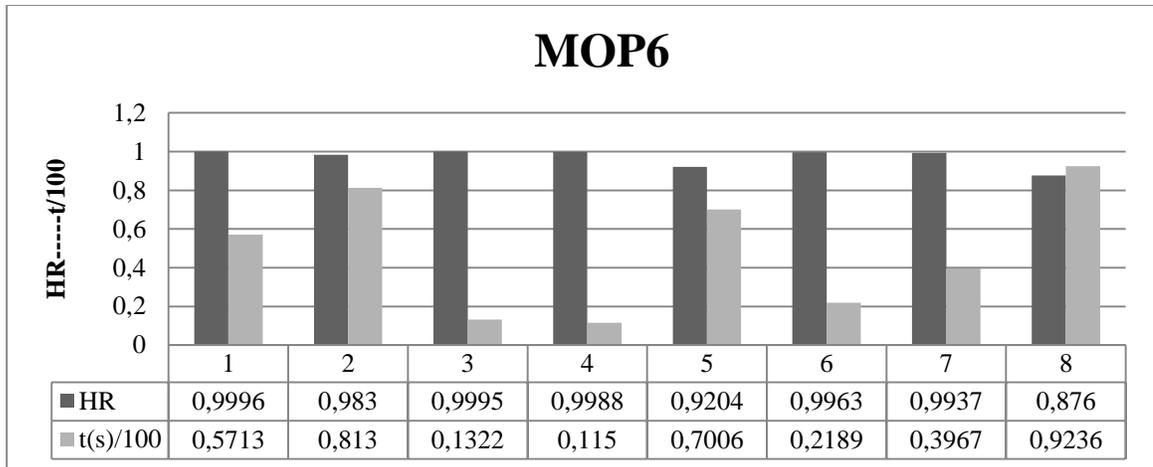
Figura 2.16 Desempeño de las diferentes heurísticas en la solución del problema MOP2  
 NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II  
 (7), PESA II (8)



*Figura 2.17 Desempeño de las diferentes heurísticas en la solución del problema MOP3*  
*NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II*  
*(7), PESA II (8)*

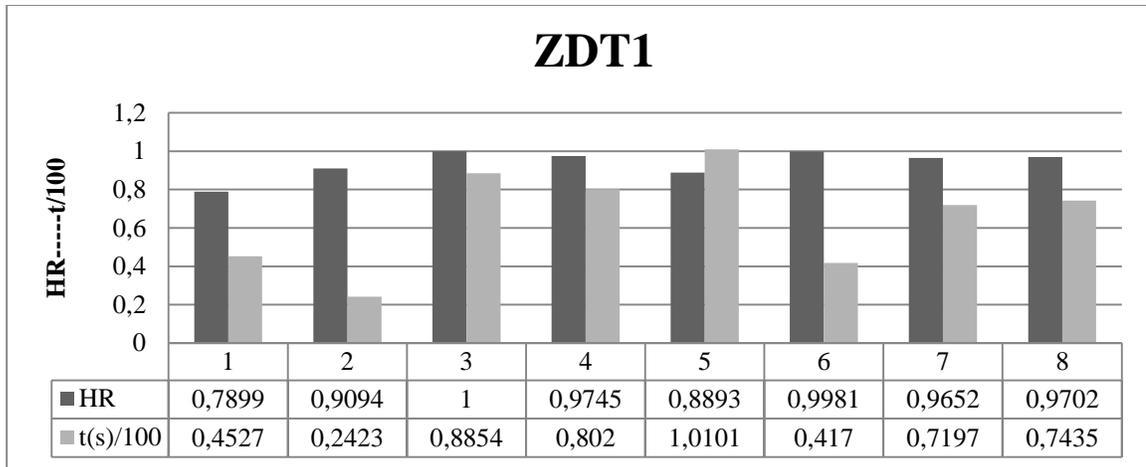


*Figura 2.18 Desempeño de las diferentes heurísticas en la solución del problema MOP4*  
*NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II*  
*(7), PESA II (8)*

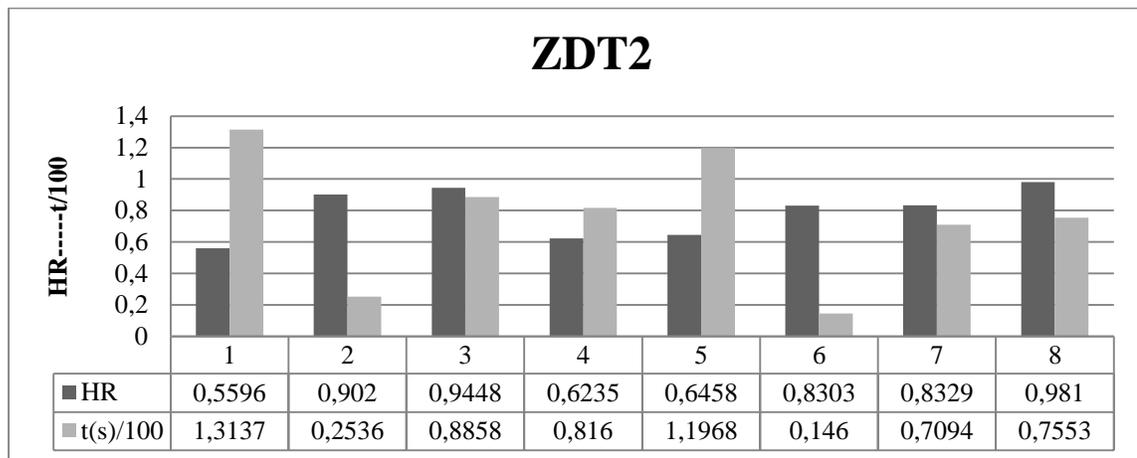


*Figura 2.19 Desempeño de las diferentes heurísticas en la solución del problema MOP6 NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II (7), PESA II (8)*

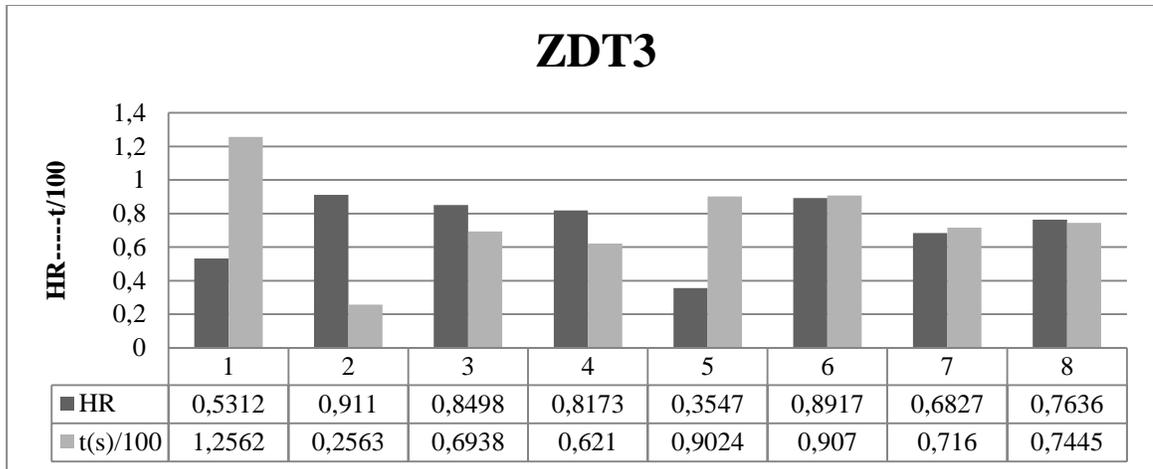
En las funciones MOP en general la mayoría de las heurísticas tienen buenos desempeños solo PESA II tiene un desempeño marcadamente más bajo que el resto y en algunos casos MOEAD también tiene un desempeño algo menor que el resto, en cuanto al tiempo de ejecución (relacionado al costo computacional) son MOCE+ y su sucesor SMOCE las que tienen mejores resultados, pues además de tener un muy buen desempeño (HR) lo logran empleando muy poco tiempo.



*Figura 2.20 Desempeño de las diferentes heurísticas en la solución del problema ZDT1*  
*NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II*  
*(7), PESA II (8)*



*Figura 2.21 Desempeño de las diferentes heurísticas en la solución del problema ZDT2*  
*NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II*  
*(7), PESA II (8)*



*Figura 2.22 Desempeño de las diferentes heurísticas en la solución del problema ZDT3  
NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II  
(7), PESA II (8)*

En este tipo de funciones (ZDT) en general el desempeño de todas las heurísticas disminuye y aumenta el tiempo de ejecución, en este caso no se muestra una marcada tendencia de que alguna heurística sea mejor que otra sino que son ligeramente superiores NSGA III, MOCE+ y SMOCE sobre el resto.

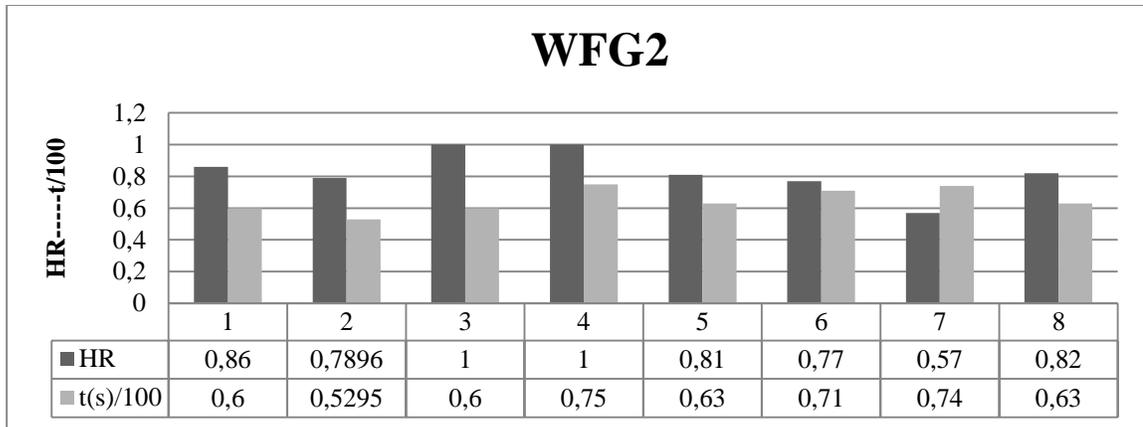


Figura 2.23 Desempeño de las diferentes heurísticas en la solución del problema WFG2  
 NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II  
 (7), PESA II (8)

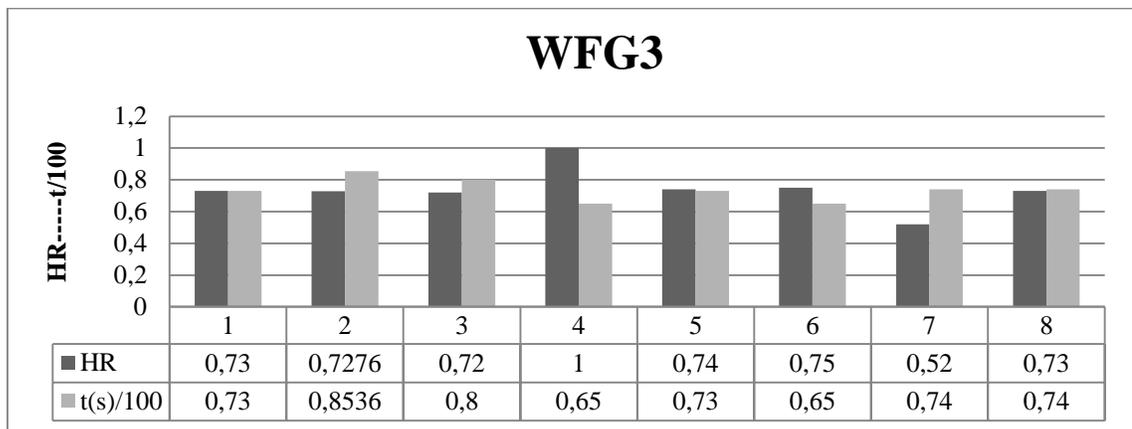


Figura 2.24 Desempeño de las diferentes heurísticas en la solución del problema WFG3  
 NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II  
 (7), PESA II (8)

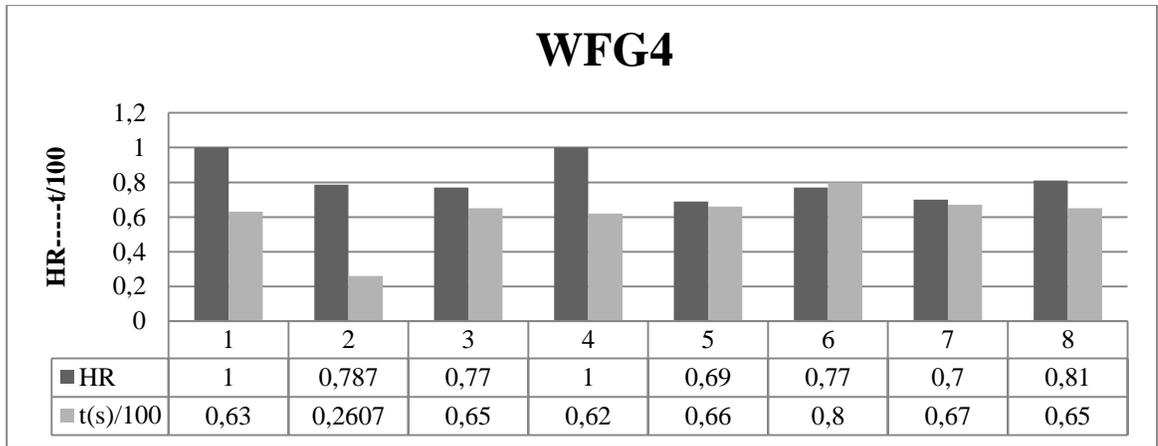


Figura 2.25 Desempeño de las diferentes heurísticas en la solución del problema WFG4  
 NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II  
 (7), PESA II (8)

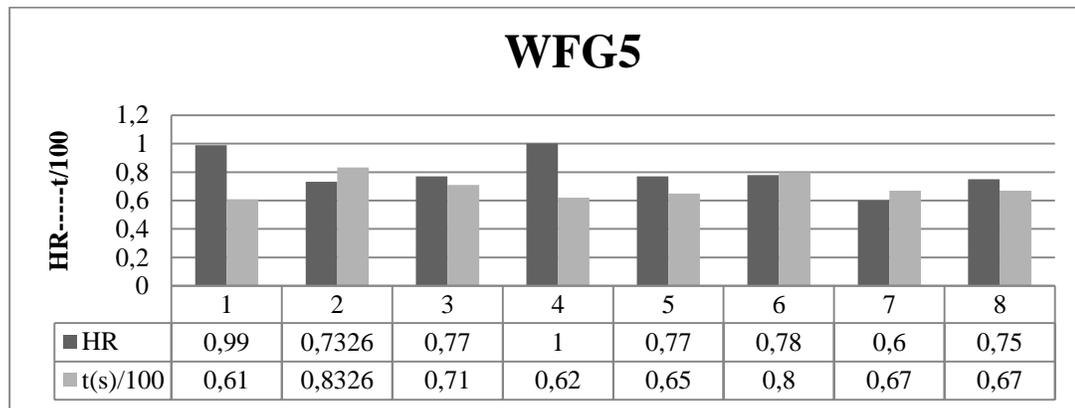
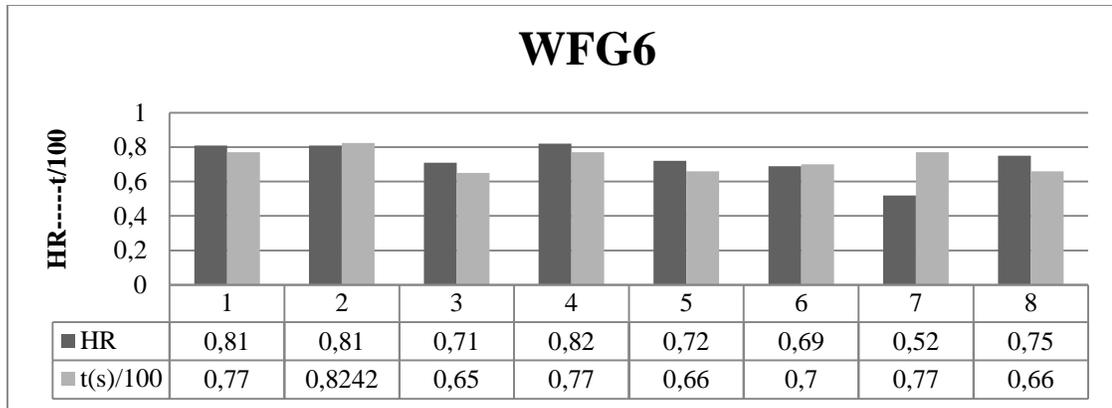


Figura 2.26 Desempeño de las diferentes heurísticas en la solución del problema WFG5  
 NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II  
 (7), PESA II (8)



*Figura 2.27 Desempeño de las diferentes heurísticas en la solución del problema WFG6  
NSGA II (1), NSGA III (2), MOCE+ (3), SMOCE (4), MOEAD (5), MOPSO (6), SPEA II  
(7), PESA II (8)*

En el caso de las funciones WFG se nota que SMOCE es la técnica con mejor desempeño pues alcanza los mayores valores de HR empleando valores de tiempo de ejecución similares al resto.

## 2.7 Conclusiones parciales

1. Los algoritmos evolutivos analizados son variados y aplicados en múltiples campos, a partir de las funciones de prueba descritas y a través de los parámetros HR y tiempo de ejecución se demuestra que a medida que se complejizan los problemas disminuyen los rendimientos de las técnicas analizadas, siendo en todos los casos SMOCE, su antecesor MOCE+ y NSGA III las técnicas que mejor desempeño tienen en estos problemas.

## **CAPÍTULO 3 APLICACIÓN DE LAS HEURÍSTICAS SELECCIONADAS AL DISEÑO DE TIPOLOGÍAS DE ARMADURAS PLANAS.**

### **3.1 Generalidades**

El objetivo fundamental de este capítulo es analizar los resultados de los desempeños de los algoritmos que mejores resultados obtuvieron con las funciones de prueba en la optimización de las diferentes tipologías de armaduras planas propuestas.

### **3.2 Análisis del caso de optimización (configuración de armaduras planas)**

Las armaduras planas, fundamentalmente las tipologías *Pratt*, *English*, *Verticaly Slope* son de las estructuras más utilizadas en la actualidad por su ya probada eficacia, tanto en puentes, cubiertas o torres de comunicaciones o electricidad.

En todos los casos la función objetivo resulta las tensiones en las barras de las armaduras, debido a que disminuyendo lo más posible estos valores se obtendría secciones transversales más pequeñas lo que influye directamente en el peso final de la estructura y más importante aun en el costo relacionado con el gasto de materiales para su ejecución.

En el caso de la tipología *Pratt* las variables que intervienen en la función son las alturas en los extremos (longitud total del primer montante), la altura de cada montante, la longitud de todos los elementos del cordón inferior, lo cual fija por supuesto las longitudes de los elementos del cordón superior.

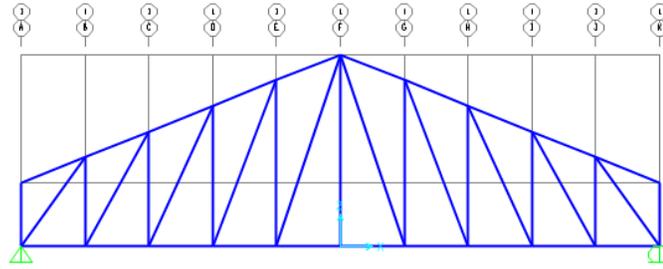


Figura 3.1 Configuración de la tipología Pratt (elaborada por el autor en SAP 2000)

En el caso de la tipología *English* las variables son exactamente las mismas, solo varía la configuración (disposición) de las diagonales.

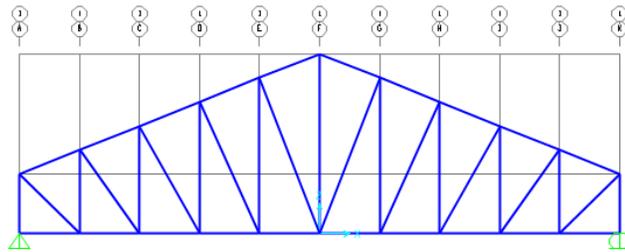


Figura 3.2 Configuración de la tipología English (elaborada por el autor en SAP 2000)

En el caso de la tipología *Sloped* disminuye la cantidad de variables con respecto a las tipologías *English* y *Pratt*, pues en este caso la altura total de la armadura no varía en cada nodo del cordón superior, sino que se mantiene constante.

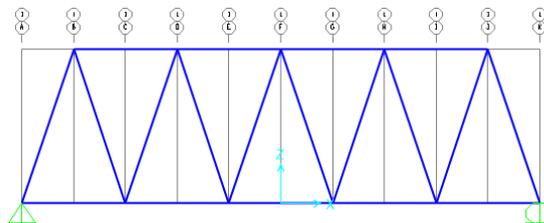
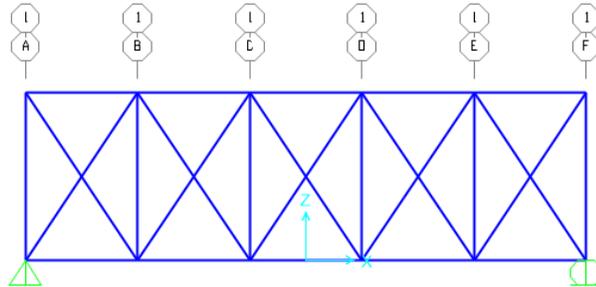


Figura 3.3 Configuración de la tipología Sloped (elaborada por el autor en SAP 2000)

La tipología *Vertical* no es muy diferente a la anterior, el número de variables es el mismo solo aumenta la complejidad para obtener la función objetivo.



*Figura 3.4 Configuración de la tipología Vertical (elaborada por el autor en SAP 2000)*

Obtener una configuración racional, resistente y económica ha sido una tarea difícil a la que se enfrentan los ingenieros pues muchas son las variables a tener en cuenta, siendo la experiencia y criterio del diseñador una parte importante del proceso, sin embargo, en muchas ocasiones se dejan de analizar soluciones que pueden ser sino optimas al menos mejor que la propuesta definitiva, pues nunca se tuvo en consideración.

Sin embargo, con el uso de los algoritmos propuestos estas soluciones no escaparían a la vista de los que diseñan y se pueden ahorrar considerables recursos solo con el uso de estas técnicas, por esto es que se propone una programación en MatLab que permita a los diseñadores hacer uso de ella para ajustarla a las condiciones de cada problema y así obtener las mejores soluciones posibles. A continuación, se propone un problema tipo para validar el funcionamiento de los algoritmos propuestos, los valores de longitudes y cargas pueden ser cambiados según las necesidades del diseñador para este estudio se mantendrán los valores en cada caso pues el objetivo es comparar los resultados obtenidos por criterio de diseñador y usando los algoritmos propuestos.

En este se requiere salvar una longitud  $L$  (en este caso de 50 m), con apoyos extremos (simplemente apoyados) y sin apoyos intermedios; usando la menor variedad de perfiles posibles para una mejor facilidad de armado. Para ello se utiliza una carga distribuida  $q$  (de 25 kN/m en este caso, puede también usarse cargas puntuales en los nodos que se desee).

### 3.2.1 Solución propuesta por el diseñador (Sloped).

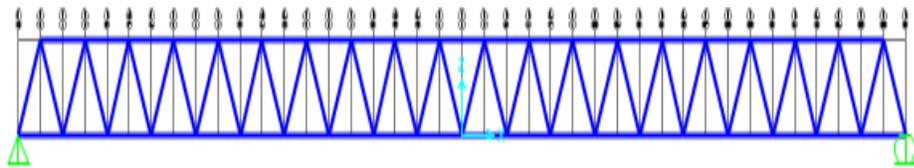


Figura 3.4 Armadura Sloped. Fuente: Elaborado por el autor en el software SAP2000.

Sección recomendada por el software: S20x96

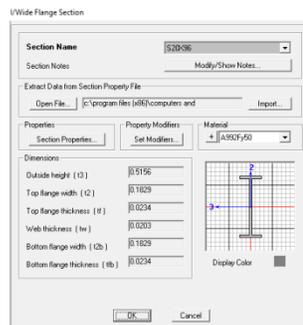


Figura 3.5 Sección S20x96. Fuente: Software SAP2000

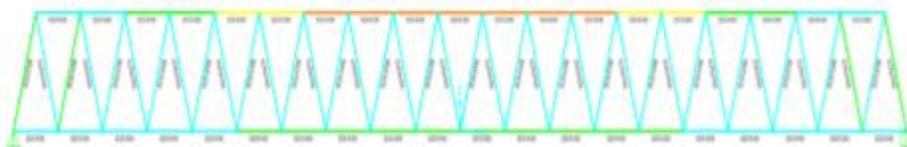


Figura 3.6 Chequeo de la estructura con la sección S20x96. Fuente: Software SAP2000.

Como se aprecia en la Figura 3.6 no existe ningún elemento sobretensionado por lo que el diseñador puede concluir que esta solución es aceptable.

### 3.2.2 Solución Optimizada con NSGA III(Sloped).

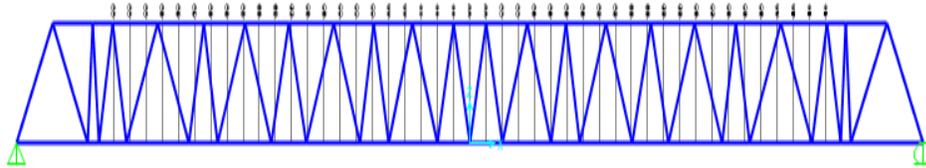


Figura 3.7 Armadura Sloped. Fuente: Elaborado por el autor en el software SAP2000.

Sección recomendada por el software: S20x86



Figura 3.8 Sección S20x86. Fuente: Software SAP2000.

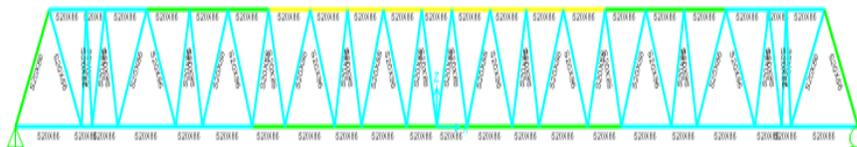


Figura 3.9 Chequeo de la estructura con la sección S20x86. Fuente: Software SAP2000.

Como se aprecia en la Figura 3.9 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

Al comparar ambas soluciones se refleja claramente que en la propuesta optimizada con NSGA III se utiliza una menor sección (S20x86), con un área de 0.0163 m<sup>2</sup>, lo que representa una disminución del 10.5% con respecto a la sección propuesta por el diseñador (S20x96) con un área de 0.0182 m<sup>2</sup>.

### 3.2.3 Solución Optimizada con SMOCE (Sloped).

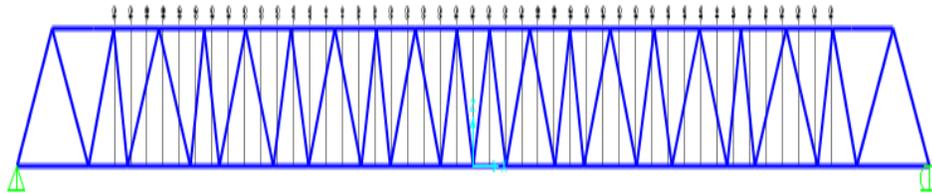


Figura 3.10 Armadura Sloped. Fuente: Elaborado por el autor en el software SAP2000.

Sección recomendada por el software: S20x75

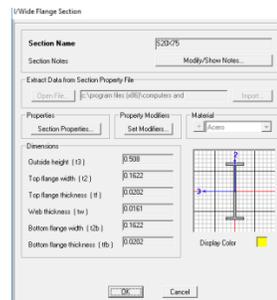


Figura 3.5 Sección S20x86. Fuente: Software SAP2000.

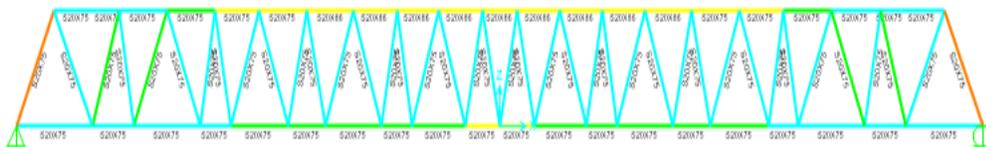


Figura 3.11 Chequeo de la estructura con la sección S20x86. Fuente: Software SAP2000.

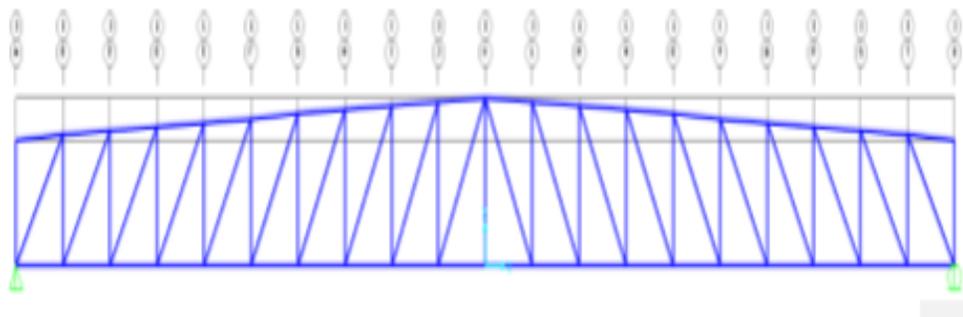
Como se aprecia en la Figura 3.11 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

Al comparar las soluciones se refleja claramente que en la propuesta optimizada con SMOCE se utiliza una menor sección (S20x75), con un área de 0.0142 m<sup>2</sup>, que las usadas cuando se realizó el análisis con NSGA III (S20x86) con un área de 0.0163 m<sup>2</sup> y mucho menor que cuando se hizo por experiencia (S20x96) con un área de 0.0182 m<sup>2</sup>.

*Tabla 3.1 Comparación de los resultados del desempeño de las técnicas empleadas en la tipología Sloped*

Técnica	Sección	Área (m <sup>2</sup> )	Disminución % respecto a la por experiencia	Disminución % respecto a la optimizada con NSGA III
Experiencia	S20x96	0.0182	-	-
NSGA III	S20x86	0.0163	10.43956044	-
SMOCE	S20x75	0.0142	21.97802198	12.88343558

### 3.2.4 Solución propuesta por el diseñador (Pratt).



*Figura 3.12 Armadura Pratt. Fuente: Elaborado por el autor en el software SAP2000.*

Sección recomendada por el software: S24x106.



Figura 3.13 Sección S24x106. Fuente: Software SAP2000

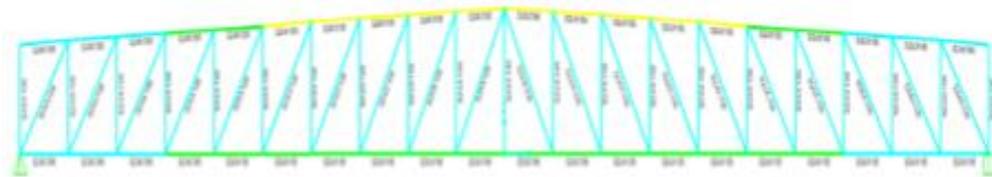


Figura 3.14 Chequeo de la estructura con la sección S24x106. Fuente: Software SAP2000

Como se aprecia en la Figura 3.14 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología. Conclusión a la que puede arribar el diseñador al usar este modelo estructural.

### 3.2.5 Solución Optimizada con NSGA III (Pratt).

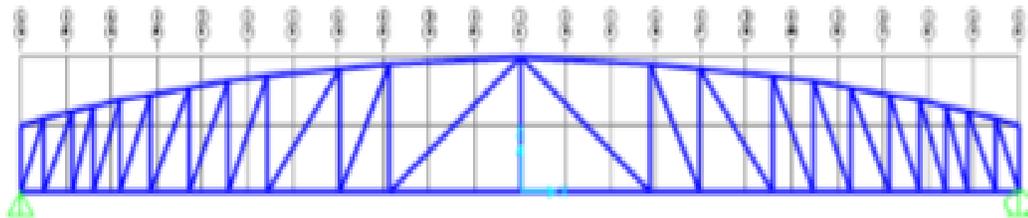


Figura 3.15 Armadura Pratt Optimizada. Fuente: Elaborado por la autora en el software SAP2000.



*Figura 3.16 Sección S20x86. Fuente: Software SAP2000*



*Figura 3.17 Chequeo de la estructura con la sección S20x86. Fuente: Software SAP2000*

Como se aprecia en la Figura 3.17 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

Al igual que en la Sloped si se comparan ambas soluciones se puede apreciar que en la propuesta optimizada con NSGA III se utiliza una menor sección (S20x86), con un área de  $0.0163 \text{ m}^2$ , mientras que en la propuesta por experiencia del diseñador se usa una sección mayor (S24x106), con un área de  $0.0201 \text{ m}^2$ .

### 3.2.6 Solución Optimizada con SMOCE (Pratt).

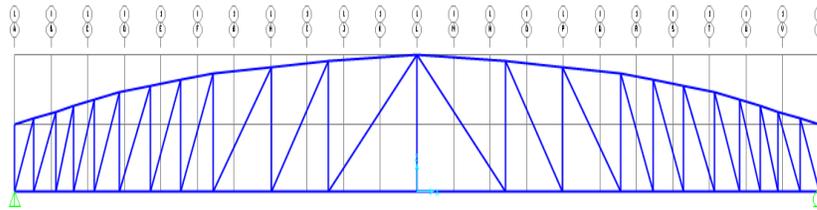


Figura 3.18 Armadura Pratt Optimizada. Fuente: Elaborado por la autora en el software SAP2000.

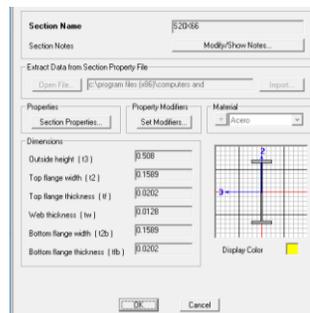


Figura 3.11 Sección S20x66. Fuente: Software SAP2000

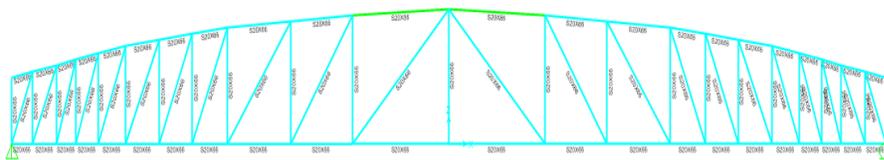


Figura 3.19 Chequeo de la estructura con la sección S20x66. Fuente: Software SAP2000

Como se aprecia en la Figura 3.19 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

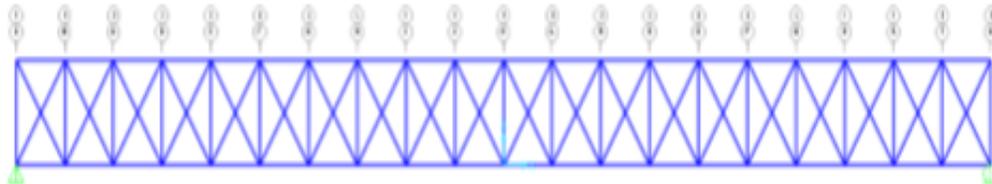
En este caso también se logra una mayor disminución en el área de los perfiles usando SMOCE, pues por experiencia se usa una sección (S24x106), con un área de  $0.0201 \text{ m}^2$ ,

con NSGA III una sección (S20x86), con un área de 0.0163 m<sup>2</sup> y con SMOCE una sección S20x66 con área de sección transversal de 0.0125 m<sup>2</sup>.

*Tabla 3.2 Comparación de los resultados del desempeño de las técnicas empleadas en la tipología Pratt*

Técnica	Sección	Área (m <sup>2</sup> )	Disminución % respecto a la por experiencia	Disminución % respecto a la optimizada con NSGA III
Experiencia	S24x106	0.0201	-	-
NSGA III	S20x86	0.0163	18.90547264	-
SMOCE	S20x66	0.0125	37.81094527	23.31288344

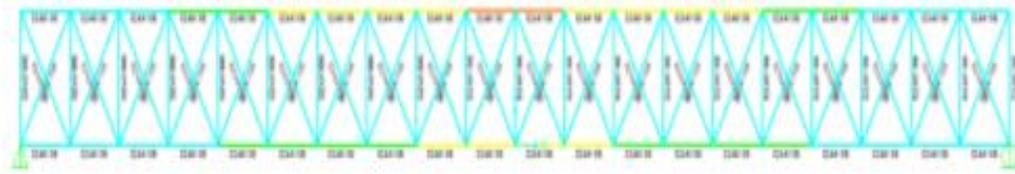
### 3.2.7 Solución propuesta por el diseñador (Vertical).



*Figura 3.20 Armadura Vertical. Fuente: Elaborado por la autora en el software SAP2000.*



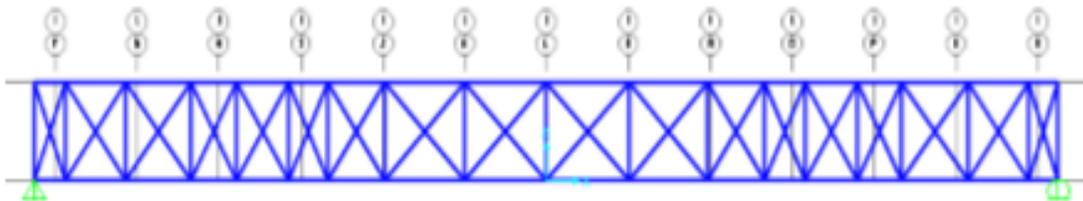
*Figura 3.21 Sección S24x106. Fuente: Software SAP2000*



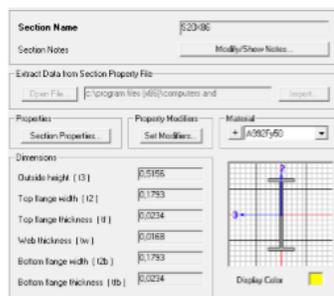
*Figura 3.22 Chequeo de la estructura con la sección S24x106. Fuente: Software SAP2000*

Como se aprecia en la Figura 3.22 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

### 3.2.8 Solución Optimizada con NSGA III (Vertical).



*Figura 3.23 Armadura Vertical Optimizada. Fuente: Elaborado por el autor en el software SAP2000.*



*Figura 3.24 Sección S20x86. Fuente: Software SAP2000.*

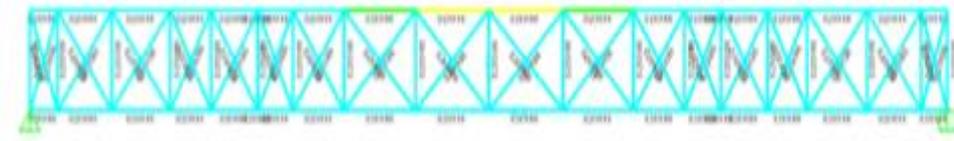


Figura 3.25 Chequeo de la estructura con la sección S20x86. Fuente: Software SAP2000.

Como se aprecia en la Figura 3.25 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

Se comparan ambas soluciones y se aprecia que en la propuesta optimizada con NSGA III se utiliza una menor sección (S20x86), con un área de  $0.0163 \text{ m}^2$  que con la propuesta del diseñador (S24x106), con un área de  $0.0201 \text{ m}^2$ .

### 3.2.9 Solución Optimizada con SMOCE (Vertical).

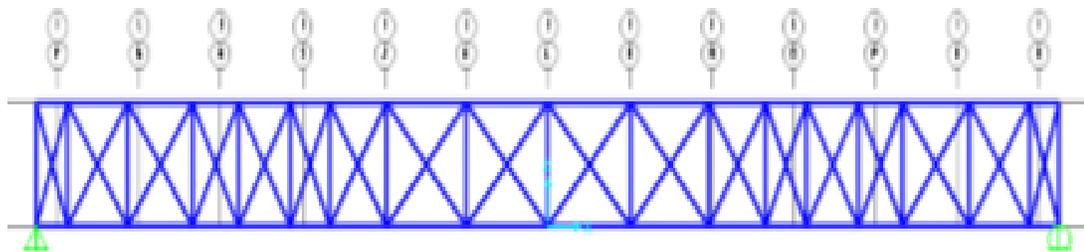


Figura 3.26 Armadura Vertical Optimizada. Fuente: Elaborado por el autor en el software SAP2000.



Figura 3.27 Sección S20x86. Fuente: Software SAP2000.

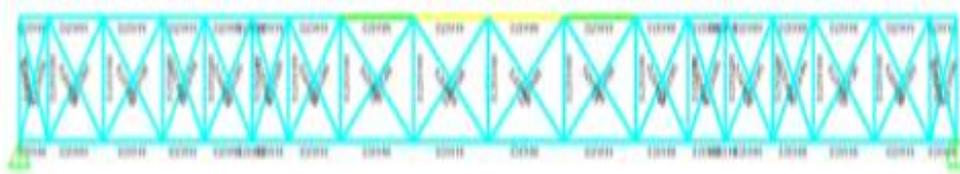


Figura 3.28 Chequeo de la estructura con la sección S20x86. Fuente: Software SAP2000.

Como se aprecia en la Figura 3.18 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

En este caso se logra la misma disminución del área de la sección transversal de la sección con ambas técnicas, SMOCE y NSGA III con respecto a la propuesta del diseñador.

### 3.2.10 Solución propuesta por el diseñador (English).

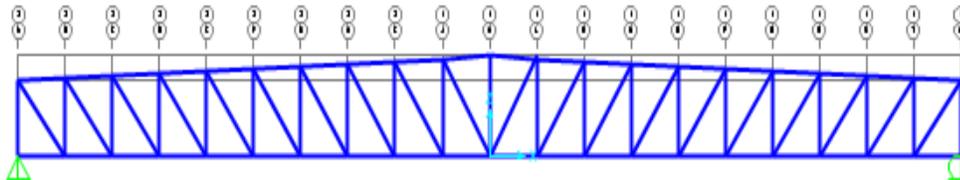


Figura 3.12 Armadura English. Fuente: Elaborado por el autor en el software SAP2000.

Sección recomendada por el software: S24x106.



Figura 3.13 Sección S24x106. Fuente: Software SAP2000

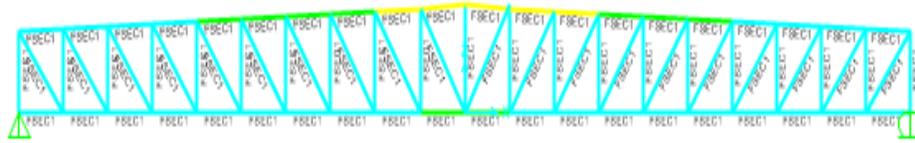


Figura 3.14 Chequeo de la estructura con la sección S24x106. Fuente: Software SAP2000

Como se aprecia en la Figura 3.9 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

### 3.2.11 Solución Optimizada con NSGA III (English).

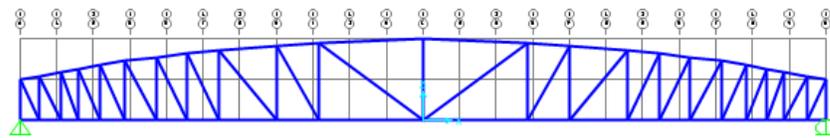


Figura 3.15 Armadura English Optimizada. Fuente: Elaborado por el autor en el software SAP2000.

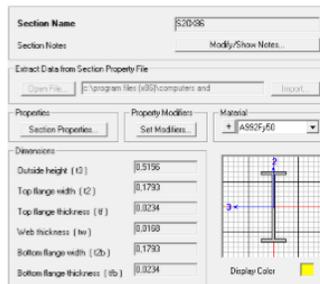


Figura 3.16 Sección S20x86. Fuente: Software SAP2000

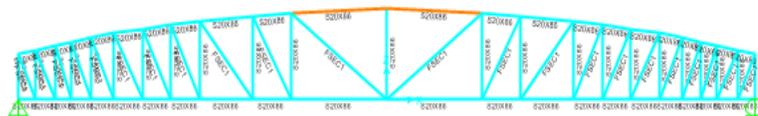
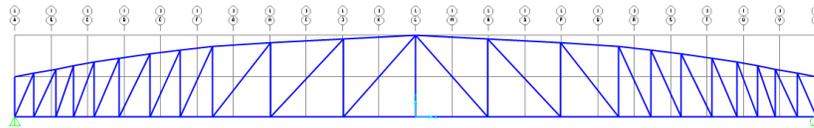


Figura 3.17 Chequeo de la estructura con la sección S20x86. Fuente: Software SAP2000

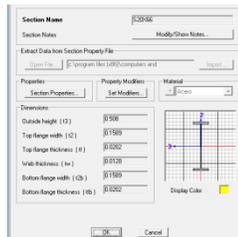
Como se aprecia en la Figura 3.17 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

En este caso los resultados son exactamente iguales al obtenido en la tipología *Pratt*

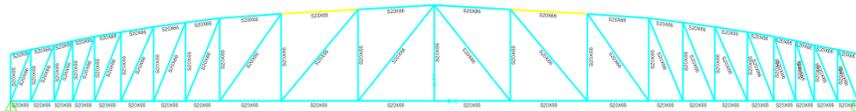
### 3.2.6 Solución Optimizada con SMOCE (English).



*Figura 3.18 Armadura English Optimizada. Fuente: Elaborado por el autor en el software SAP2000.*



*Figura 3.11 Sección S20x66. Fuente: Software SAP2000*



*Figura 3.19 Chequeo de la estructura con la sección S20x66. Fuente: Software SAP2000*

Como se aprecia en la Figura 3.19 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

En este caso los resultados de las secciones transversales obtenidos por todas las técnicas son iguales a las obtenidas en la tipología *Pratt*.

*Tabla 3.2 Comparación de los resultados del desempeño de las técnicas empleadas en la tipología English*

Técnica	Sección	Área (m <sup>2</sup> )	Disminución % respecto a la por experiencia	Disminución % respecto a la optimizada con NSGA III
Experiencia	S24x106	0.0201	-	-
NSGA III	S20x86	0.0163	18.90547264	-
SMOCE	S20x66	0.0125	37.81094527	23.31288344

### **3.3 Conclusiones parciales**

1. Al analizar los resultados obtenidos en cada caso de estudio con la aplicación de las técnicas seleccionadas NSGAIII y SMOCE con las programaciones establecidas y los obtenidos por la experiencia del diseñador, se puede apreciar cuan eficiente resultan ser estas técnicas, sobre todo SMOCE pues se reducen las secciones en todos los casos en los que no se realizó un previo dimensionamiento, demostrando que a través de la optimización estructural empleando MOEAs, se obtienen estructuras más ligeras, económicas y a la vez resistentes

## CONCLUSIONES

Como resultado final del trabajo desarrollado, se ha podido arribar a las siguientes conclusiones:

- 1.-Las heurísticas de optimización se han convertido en la actualidad en unas de las técnicas más difundidas y utilizadas en diversos campos, siendo uno de ellos la optimización estructural, especialmente las estructuras reticuladas planas.
2. Los resultados de las diferentes heurísticas seleccionadas varían según la complejidad del problema, en sentido general se destacan tres de ellas como las de mejor rendimiento ante estos problemas (SMOCE, MOCE+ y NSGA III).
3. Es evidente la mejora en la racionalidad del diseño cuando se emplean algoritmos de búsqueda heurística a problemas de armaduras planas (*English, Pratt, Sloped y Vertical*) pues se logran secciones transversales más pequeñas que redundan finalmente en estructuras más ligeras y económicas, a la vez que son resistentes a las cargas impuestas.

## **RECOMENDACIONES**

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

1. Extender el estudio a otras tipologías de armaduras planas y a armaduras tridimensionales.

## REFERENCIAS BIBLIOGRÁFICAS

- AHRARI, A., ATAI, A. & DEB, K. 2014. Simultaneous Topology, Shape and Size Optimization of Truss Structures by Fully Stressed Design Based on Evolution Strategy.
- ALLAIRE, G., JOUVE, F. & TOAEDE, A. 2004. Structural optimization using sensitivity analysis and a level-set method. . *Computational Physics*.
- BELLIDO, M., AYRA, D. & AGUADO, R. 2015. Estructura de puente de armadura Warren. Análisis aplicativo. Diversificación Productiva y del Fortalecimiento de la Educación.
- BERUVIDES, G., QUIZA, R. & HABER, R. 2016. Multi-objective optimization based on an improved cross-entropy method. A case of study of a micro-scale manufacturing process. *Information Sciences*.
- BIETHAN, J. & NISSEN, V. 1995. Evolutionary algorithms in management applications.
- COELLO, RUDNICK & CHRISTIANSEN 1995. Using genetic algorithms for optimal design of trusses.
- COELLO, C. 2012. Constraint-Handling techniques used with evolutionary algorithms. Mexico.
- COELLO, C. & CHRISTIANSEN, C. 2000. Multiobjective optimization of trusses using genetic algorithms.
- COELLO, C., LAMONT, G. & VANVELDHUIZEN, D. 2002. *Evolutionary Algorithms for Solving Multi-Objective Problems*.
- COELLO, C. & ZALAZAR, M. 2003. MOPSO: A proposal for multiple objective particle swarm optimization.
- CHAVEZ, A. 2018. *Optimización estructural de armaduras planas empleando técnicas heurísticas (Algoritmos Genéticos)*. Diploma, Universidad de Matanzas.
- CHEN, X. 2013. Learnig with sparsity: Structures, Optimization and Applications. *School of Computer Science Carnegie Mellon University*.
- CHEN, Y. & ZOU, X. 2014. Runtime analysis of a multi-objective evolutionary algorithm for obtaining finite approximations of Pareto fronts.
- CHENG, J. 2010. Optimum design of steel truss arch bridges using a hybrid genetic algorithm. *Constructional Steel Research*.

- DEB, K., THIELE, L., LAUMANN, M. & ZITZLER, E. 2001. Scalable Test Problems for Evolutionary Multi-Objective Optimization. *IEEE*.
- DERRAC, J., GARCIA, S., MOLINA, D. & HERRERA, F. 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*.
- GHARARI, R., POURSALEHI, N., ABBASI, M. & AGHAIE, M. 2016. Implementation of strength Pareto evolutionary algorithm II in the multiobjective burnable poison placement optimization of KWU pressurized water reactor. *Nuclear Engineering and Technology*.
- GOLDBERG, D. 1989. Genetic Algorithms in Search, Optimization and Machine Learning.
- GONG, D., WANG, G., SUN, X. & HAN, Y. 2014. A set-based genetic algorithm for solving the many-objective optimization problem. *Springer*.
- GUTIERREZ, N., PENICHE, R., HERRERA, G., ALVARADO, R. & CARRION, F. 2012. A Non-Symmetrical Solution Applying a Genetic Algorithm with Natural Crossover for the Structural Optimization of Truss Structures. *Ingeniería Investigación y Tecnología, XIII*.
- HABER, R., BERUVIDES, G., QUIZA, R. & HERNANDEZ, A. 2017. A simple multi-objective optimization based on the cross-entropy method. *IEEE*.
- HOLLAND, J. 1975. Adaptation in Natural and Artificial Systems. *The University of Michigan Press*.
- HUBAND, S., HINGSTON, P., BARONE, L. & WHILE, L. 2006. A Review of Multi-objective Test Problems and a Scalable Test Problem Toolkit *IEEE*, 10.
- HULTMAN, M. 2010. *Weight optimization of steel trusses by a genetic algorithm*. Master, Lund University.
- KE, L., ZHANG, Q. & BATTITI, R. 2013. MOEA/D-ACO: A Multiobjective Evolutionary Algorithm Using Decomposition and Ant Colony *IEEE*.
- KONA, A., COIT, D. & SMITH, A. 2006. Multi-objective optimization using genetic algorithms: A tutorial.
- KOZIEL, S. & YANG, X. 2011. Computational Optimization, Methods and Algorithms, Typeset & Cover Design.
- LI, M., YANG, S., LIU, X. & WANG, K. 2013. IPESA-II: Improved Pareto Envelope-Based Selection Algorithm II. *In: PURSHOUSE, R., FLEMING, P., FONSECA,*

- C., GRECO, S. & SHAW, J. (eds.) *International Conference on Evolutionary Multi-Criterion Optimization*. UK.
- MACDONALD, R. 2005. *Genetic Algorithms with Steel Structures a Literatura Review*. Master, Brigham Young University.
- MARLER, T. & ARORA, J. 2004. Survey of multi-objective optimization methods for engineering. *Struct Multidisc Optim*.
- PIRES, J. 2014. *Structural Optimization using the Finite Element Method*. Universidad Catolica Portuguesa.
- RAMIREZ, S. 2014. *Metodología basada en Algoritmos Genéticos y Programación en Paralelo para el Diseño Óptimo de Armaduras de Acero*. Maestría en ingeniería civil.
- SAKA, M. & GREEM, Z. 2012. Mathematical and metaheuristic application in design optimization of steel frame structures: An extensive review.
- SMITH, J., HODGINS, J., OPPENHEIM, I. & WITKIN, A. 2002. Creating Models of Truss Structures with Optimization.
- VAMPLEW, P., DAZELEY, R., BERRY, A., ISSABEKOV, R. & DEKKER, E. 2010. Empirical evaluation methods for multiobjective reinforcement learning algorithms.
- VIDAL, A. 2013. *Algoritmos Heurísticos en optimización*. Maaster, Universidad de Santiago de Compostela.
- ZITZLER, E., LAUMANN, M. & THIELE, L. 2001. SPEA2: Improving the strength pareto evolutionary algorithm. *ETHzurich*.