

UNIVERSIDAD DE COSTA RICA
SISTEMA DE ESTUDIOS DE POSGRADO
TÍTULO DEL TFIA

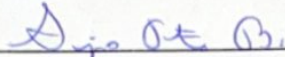
DESARROLLO DE UN MÉTODO NUMÉRICO PARA LA
OPTIMIZACIÓN MULTI-OBJETIVO DEL DISEÑO DE
RESORTES HELICOIDALES A COMPRESIÓN

Trabajo final de investigación aplicada sometido a la consideración de la Comisión del Programa de Estudios de Posgrado en Ingeniería Mecánica para optar al grado y título de Maestría Profesional en Ingeniería Mecánica.

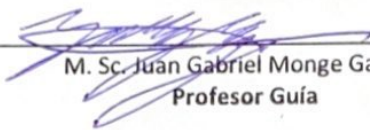
ERICK VAN PATTEN RIVERA

Ciudad Universitaria Rodrigo Facio
2022

"Este trabajo final de investigación aplicada fue aceptado por la Comisión del Programa de Estudios de Posgrado en Ingeniería Mecánica de la Universidad de Costa Rica, como requisito parcial para optar al grado y título de Maestría Profesional en Ingeniería Mecánica."



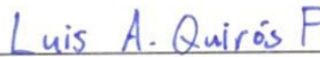
M. Sc. Sergio Ferreto Brenes
Representante de la Decana
Sistema de Estudios de Posgrado



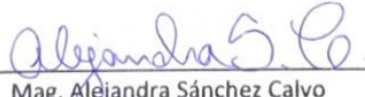
M. Sc. Juan Gabriel Monge Gapper
Profesor Guía



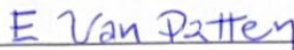
M. Sc. André Licks Miranda
Lector



M.Sc. Luis Quirós Fonseca
Lector



Mag. Alejandra Sánchez Calvo
Representante del Director del Programa de Posgrado en Ingeniería Mecánica



Erick Van Patten Rivera
Sustentante

Tabla de contenido

Hoja de aprobación	ii
Resumen	v
Lista de cuadros.....	vi
Lista de figuras	vii
1. Introducción	1
1.1. Descripción general	1
1.2. Justificación	2
1.3. Objetivos.....	2
1.3.1. Objetivo general.....	2
1.3.2. Objetivos específicos	2
1.4. Metodología general	3
1.5. Alcance y limitaciones	4
1.6. Aportes y productos materiales	4
2. Marco teórico	6
2.1. Válvulas de control.....	6
2.2. Resortes helicoidales.....	7
2.2.1. Parámetros de diseño.....	7
2.2.2. Esfuerzo máximo y pandeo	8
2.3. Fatiga no armónica	9
2.4. Diseño por robustez.....	12
2.5. Algoritmos para optimización	14
2.5.1. Optimización multiobjetivo	15
2.5.2. Algoritmos evolutivos	15
2.5.3. Algoritmo genético ordenado no dominado (NSGA II)	17
2.5.4. Selección, recombinación y mutación.....	19
3. Metodología	24
3.1. Sinopsis metodológica	24
3.2. Modelo dinámico.....	25
3.3. Optimización del resorte	30
3.3.1. Funciones objetivo	30
3.4. Restricciones	33
3.5. Casos de carga	35
3.6. Implementación del algoritmo	38
3.7. Subrutinas del programa.....	39
3.7.1. Creación de la población inicial	39
3.7.2. Validación de la población.....	40
3.7.3. Evaluación de las funciones objetivo	41
3.7.4. Ordenamiento no dominante y distanciamiento mínimo.....	42

3.7.5. Selección, apareamiento y mutación.....	44
3.8. Parámetros del algoritmo.....	45
3.9. Comentarios finales.....	46
4. Resultados y discusión	47
4.1. Estructura del código.....	47
4.2. Casos de estudio.....	49
4.2.1. Carga de onda de presión.....	49
4.2.2. Carga armónica simétrica.....	51
4.2.3. Carga aleatoria	53
4.3. Comentarios generales.....	55
5. Conclusión	57
5.1. Conclusiones y aportes.....	57
5.2. Recomendaciones y trabajo futuro.....	60
Referencias	65
Anexos	66
Anexo A.1. Validación del algoritmo	67
Anexo A.2. Guía para correr el programa con funciones objetivo	76
Anexo A.3. Artículo técnico	77
Anexo A.4. Código del algoritmo.....	82

Resumen

En esta investigación se define la implementación de un método numérico que utiliza un algoritmo genético para la optimización multi-objetivo de resortes helicoidales con el objetivo de minimizar la masa mientras se maximiza la resistencia a la fatiga y la robustez de su diseño. El método numérico y los casos de estudio en los que se prueba se enfocan principalmente a los resortes utilizados en las válvulas de control de flujo, sin embargo la metodología se puede cambiar a ser aplicable a otras aplicaciones en resorte similares. Para la implementación del algoritmo se determinan los requerimientos de diseño de los resortes y las restricciones a las que estarán sometidos, después de esto se programa el algoritmo y se prueba con funciones teóricas para verificar su efectividad.

Una vez probado el algoritmo se realiza un modelo dinámico de masa amortiguador resorte con múltiples grados de libertad y se traslada esto a un conjunto de formulaciones matemáticas con las cuales se ejecuta la optimización. Finalmente se presentan y analizan tres casos de estudios con diferentes patrones de fuerza: un patrón sinusoidal, una onda de presión y una fuerza aleatoria. Para los diferentes casos de estudio se obtienen las frontera pareto-óptimas en las cuales se encuentran las combinaciones de variables de diseño para el diámetro del resorte, diámetro del alambre y número de espiras activas que tendrán una mejor relación de masa a resistencia a la fatiga y robustez del diseño.

Abstract

The scope of this work is the implementation a numeric method that utilizes a genetic algorithm to do a multi-objective optimization of helicoidally springs with the intent to minimize the mass, while maximizing the fatigue resistance and the design robustness. The numeric method and the study cases that are analyzed are focused towards the optimization of springs used in flow control valves; however, the used methodology can be applied to other similar applications. In order to implement the algorithm the design requirements and restrictions were defined, then a dynamic model of the spring in the control valve was created and converted into the mathematic equations that will be optimized through the algorithm.

Three study cases are presented and analyzed to assess the performances of the algorithm. One is that of a sinusoidal force applied to the spring; another is a random load that varies in time; and that of a load emulating a pressure surge. In each of the study cases, the pareto-fronts are presented along with the combination of variables for spring diameter, wire diameter and number of active coils for each of the individuals. This pareto-front will also define the combinations of design variables that provide the best ratio of mass to fatigue resistance and robust design according to the proposed mathematical model.

Cítese este trabajo como:

Van Patten-Rivera, Erick (2022) Desarrollo de un método numérico para la optimización multi-objetivo del diseño de resortes helicoidales a compresión. Trabajo Final de Graduación para optar por el título de Maestría Profesional en Ingeniería Mecánica. San José: Universidad de Costa Rica.

Lista de cuadros

2.1. Identificación de variables frecuentes en diseño de resorte (Fuente: el autor).	8
3.1. Valores de las variables utilizadas para los casos de carga. (Fuente: el autor).	38
3.2. Valores de las variables utilizadas para correr el algoritmo. (Fuente: el autor.)	45
4.1. Ejemplos de resultados para el caso de estudio con fuerza de presión (Fuente: el autor).....	51
4.2. Ejemplos de resultados para el caso de estudio con fuerza sinusoidal armónica (Fuente: el autor).....	51
4.3. Ejemplos de resultados para el caso de estudio con fuerzas aleatorias (Fuente: el autor).	54

Lista de figuras

2.1. Figuras representativas de la válvula	7
2.2. Imagen representativa de las variables del resorte. L es la longitud del resorte, Del diámetro del resorte, del diámetro del alambre y n_a la cantidad de espiras activas. (Fuente: el autor.)	7
2.3. Imagen representativa de la torsión en un resorte. (Fuente: [7].)	8
2.4. Flujograma del conteo de ciclos <i>rain flow</i> . (Fuente: el autor.)	10
2.5. Imagen representativa del conteo de ciclos <i>rain flow</i> . (Fuente: [21].)	11
2.6. Gráfico que ejemplifica un diseño robusto. La imagen muestra la diferencia entre un punto de optimización convencional y una optimización por robustez de diseño. (Fuente: [16].)	13
2.7. Imagen representativa de una frontera pareto-óptima dominante. (Fuente: [33].)	15
2.8. Diagrama básico de un algoritmo evolutivo. (Fuente: el autor)	16
2.9. Diagrama simplificado de un algoritmo evolutivo. (Fuente: [38].)	17
2.10. Diagrama representativo de la distancia de hacinamiento o apilamiento. (Fuente: [41].)	18
2.11. Diagrama representativo del algoritmo NSGA-II. (Fuente: [43].)	18
2.12. Diagrama representativo de la distancia de la selección por ruleta. (Fuente: [44].)	20
2.13. Imagen representativo de la selección por torneo. (Fuente: [45].)	20
2.14. Ejemplo de un problema de optimización utilizando selección de ruleta (a) y torneo (b). (Fuente: [44].)	21
2.15. Imagen representativa de la recombinación. (Fuente: el autor.)	22
2.16. Imagen representativa de la recombinación. (Fuente: [48].)	22
2.17. Imagen representativa de la distribución de la mutación polinomial. (Fuente: [50].)	23
3.1. Flujo metodológico. (Fuente: el autor.)	24
3.2. Imagen representativa de la configuración de la válvula. (Fuente: el autor.) .	25
3.3. Imagen representativa del sistema de n grados de libertad (Fuente: el autor.)	27
3.4. Diagrama representativo del flujo del programa de cálculo de fatiga. (Fuente: el autor.)	27
3.5. Diagrama de cuerpo libre del primer elemento. (Fuente: el autor.)	28
3.6. Diagrama de cuerpo libre de un elemento i en el medio. (Fuente: el autor.)	28
3.7. Diagrama de cuerpo libre del último elemento. (Fuente: el autor.)	29
3.8. Ondas de presión a utilizar para los casos de carga. (Fuente: [29].)	35
3.9. Patrón de fuerza sinusoidal. (Fuente: el autor.)	36
3.10. Gráfica de fuerza aleatoria utilizada para el caso de estudio. (Fuente: el autor.)	37
3.11. Patrón de fuerza simulando una onda de presión. (Fuente: el autor.)	37

3.12. Detalle de patrón de fuerza simulando una onda de presión. (Fuente: el autor).....	38
3.13. Diagrama de flujo del proceso total. (Fuente: el autor).....	39
3.14. Flujograma del programa de validación de la población. (Fuente: el autor.)	40
3.15. Diagrama de la subrutina de evaluación de funciones objetivo. (Fuente: el autor.)	41
3.16. Diagrama de flujo para calcular los nuevos cromosomas (Fuente: el autor).	45
4.1. Estructura general del código creado (Fuente: el autor).....	48
4.2. Resultado de frontera pareto-óptima para fuerza onda de presión (Fuente: el autor).....	49
4.3. Distribución de las variables para solución de la onda de presión (Fuente: el autor).....	50
4.4. Resultado de frontera pareto-óptima para fuerza sinusoidal (Fuente: el autor).....	52
4.5. Distribución de las variables para solución de la fuerza sinusoidal. (Fuente: el autor).....	52
4.6. Resultado de frontera pareto-óptima para fuerza aleatoria. (Fuente: el autor).....	53
4.7. Distribución de las variables para solución la fuerza aleatoria. (Fuente: el autor).....	54
5.1. Gráfica de la frontera pareto-óptima del problema de Kursawe teórica. (Fuente: [62].).....	67
5.2. Resultados obtenidos con el algoritmo para el problema de optimización Kursawe. (Fuente: el autor.).....	68
5.3. Gráfica de la frontera pareto-óptima del problema Poloni de dos objetivos teórica. (Fuente: [63].).....	68
5.4. Gráfica de la frontera pareto-óptima del problema Poloni de dos objetivos obtenido con el algoritmo creado. (Fuente: el autor).....	69
5.5. Gráfica de la frontera pareto-óptima del problema de la función Schaffer No.1 teórica. (Fuente: [64].).....	70
5.6. Gráfica de la frontera pareto-óptima del problema de la función Schaffer No.2 teórica. (Fuente: [64].).....	70
5.7. Gráfica de la frontera pareto-óptima del problema de la función Schaffer No.1 obtenida. (Fuente: el autor).....	71
5.8. Gráfica de la frontera pareto-óptima del problema de la función Schaffer No.2 obtenida. (Fuente: el autor).....	71
5.9. Gráfica de la frontera pareto-óptima del problema de Viennet teórica. (Fuente: [65].).....	72
5.10. Gráfica de la frontera pareto-óptima del problema de Viennet práctica. (Fuente: el autor).....	72
5.11. Gráfica de la frontera pareto-óptima del problema Constr-Ex teórica. (Fuente: [39].).....	73
5.12. Gráfica de la frontera pareto-óptima del problema Constr-Ex obtenida con el algoritmo. (Fuente: el autor).....	73
5.13. Gráfica de la frontera pareto-óptima del problema Osyczka y Kundu teórica. (Fuente: [66].).....	74
5.14. Gráfica de la frontera pareto-óptima del problema Osyczka y Kundu obtenida con el algoritmo (Fuente: el autor).....	75



UNIVERSIDAD DE
COSTA RICA

SEP Sistema de
Estudios de Posgrado

Autorización para digitalización y comunicación pública de Trabajos Finales de Graduación del Sistema de Estudios de Posgrado en el Repositorio Institucional de la Universidad de Costa Rica.

Yo, Erick Van Patten Rivera, con cédula de identidad 115680590, en mi condición de autor del TFG titulado Desarrollo de un método numérico para la optimización multi-objetivo del diseño de resortes helicoidales a compresión

Autorizo a la Universidad de Costa Rica para digitalizar y hacer divulgación pública de forma gratuita de dicho TFG a través del Repositorio Institucional u otro medio electrónico, para ser puesto a disposición del público según lo que establezca el Sistema de Estudios de Posgrado. SI NO *

*En caso de la negativa favor indicar el tiempo de restricción: _____ año (s).

Este Trabajo Final de Graduación será publicado en formato PDF, o en el formato que en el momento se establezca, de tal forma que el acceso al mismo sea libre, con el fin de permitir la consulta e impresión, pero no su modificación.

Manifiesto que mi Trabajo Final de Graduación fue debidamente subido al sistema digital Kerwá y su contenido corresponde al documento original que sirvió para la obtención de mi título, y que su información no infringe ni violenta ningún derecho a terceros. El TFG además cuenta con el visto bueno de mi Director (a) de Tesis o Tutor (a) y cumplió con lo establecido en la revisión del Formato por parte del Sistema de Estudios de Posgrado.

E. Van Patten

FIRMA ESTUDIANTE

Nota: El presente documento constituye una declaración jurada, cuyos alcances aseguran a la Universidad, que su contenido sea tomado como cierto. Su importancia radica en que permite abreviar procedimientos administrativos, y al mismo tiempo genera una responsabilidad legal para que quien declare contrario a la verdad de lo que manifiesta, puede como consecuencia, enfrentar un proceso penal por delito de perjurio, tipificado en el artículo 318 de nuestro Código Penal. Lo anterior implica que el estudiante se vea forzado a realizar su mayor esfuerzo para que no sólo incluya información veraz en la Licencia de Publicación, sino que también realice diligentemente la gestión de subir el documento correcto en la plataforma digital Kerwá.

Capítulo 1

Introducción

1.1. Descripción general

La industria de las válvulas de control de presión y flujo tuvo un valor de mercado de venta de alrededor de ocho miles de millones de dólares en el 2019. Además, se estima que alcance un valor de mercado de once miles de millones de dólares para el 2025 de manera global [1]. Una gran parte de el mercado de válvulas de control va a dirigida hacia la industria de gas y petróleo [2]. En esta industria se requiere de estas válvulas hidráulicas en las líneas para el control de la presión y el flujo [3].

El correcto funcionamiento de las válvulas es esencial para que las líneas que llevan hidrocarburos funcionen adecuadamente y operen de manera segura [4]. Algunos de los usos comunes de estas válvulas son en líneas de llenado de tanques de combustible, llenado de camiones de distribución de gasolina y, de manera generalizada, en el trasiego de hidrocarburos de un punto a otro como válvulas de alivio de presión [5]. El funcionamiento de estas válvulas es altamente dependiente de la correcta operación del resorte que utilizan. Es por esto que el diseño de los resortes juega un papel fundamental en el funcionamiento y confiabilidad de la válvulas [6].

Los resortes utilizados en estas válvulas de control comúnmente son diseñados mediante un conjunto de buenas prácticas de manufactura que se han determinado debido a la complejidad y cantidad de variables que utilizan [7]. Una forma de lidiar con el proceso complejo de diseño contemplando factores como resistencia contra fatiga y costo de manufactura es la aplicación de algoritmos de optimización. De esta forma, Zhang [8] aplicó un proceso de optimización multi-objetivo en partes de automóviles contemplando la masa del resorte y la confiabilidad del mismo Kim [9] implementó un algoritmo de optimización aplicado sobre un modelo de resorte dinámico en una válvula de control de una tubería con el fin de minimizar la masa y el tiempo de respuesta. Zhou [10] desarrolló un algoritmo de optimización en resortes de diafragma de un embrague de automóvil con el fin de maximizar la resistencia a la fatiga y fuerzas externas del diseño. Sin embargo, basado en la investigación bibliográfica y el estado de la cuestión realizado para este trabajo, no se ha encontrado en la literatura la aplicación de algoritmos de optimización multi-objetivo que contemple la minimización de masa, resistencia contra fatiga así como robustez en el diseño de resortes de compresión en válvulas de control. Por lo tanto, en este trabajo pretende desarrollar un método numérico que utilice un algoritmo genético multi-objetivo para estimar el conjunto de soluciones óptimas que pueden tener las variables de diseño de un resorte contemplando las funciones objetivo.

1.2. Justificación

El diseño de resortes a compresión para la industria es uno de los problemas de diseño más complejos de optimizar [11]. Esto por contar con un amplio ámbito de soluciones posibles, de las cuales es dificultoso escoger la solución viable que reduce los costos al máximo y por ser un problema de optimización que requiere lidiar con variables discretas y continuas simultáneamente [12]. El tener esta combinación de tipos de variables, en las cuales algunas no son doblemente derivables, descarta métodos de optimización más eficientes como los algoritmos de gradiente y de optimización combinatorial [13]. Los resortes son vitales en una amplia gama de productos de producción como válvulas de control, sistemas de pistones, caudalímetros, entre otros. Lo cual implica que lograr una reducción de costo por unidad tiene grandes impactos en la disminución de gastos de producción [14].

El control del diseño sobre la variabilidad de manufactura de los resortes es también un factor de importancia [15]. Al tomar en cuenta el diseño robusto ayuda a controlar la estabilidad de los resultados de manufactura al mover el punto de diseño sin incurrir en gastos adicionales en la tolerancia de los materiales [16]. Esto a su vez tiene un ahorro económico significativo al tener resultados de manufactura más constantes sin la necesidad de invertir adicionalmente en el control de calidad, que suelen ser costos [17].

Otro punto de interés para el diseño de los resortes es utilizar la combinación de variables que necesita la menor cantidad de masa. La menor cantidad de masa implica un menor costo de fabricación del material. Sin embargo, se tomará en cuenta la resistencia a la falla por fatiga con la finalidad de asegurar el correcto funcionamiento del componente y no comprometer la operación de la válvula [18].

Además de esto, otro de los factores a tomar en cuenta al optimizar es que en el diseño de resortes se tiene un ámbito de posibles magnitudes para cada variable de diseño y que al fijar algunos de ellos para optimizar una única variable puede comprometer la solución que no es de mayor beneficio para todos los frentes. Es ahí donde la optimización multi-objetivo aporta un gran valor al optimizar para varios de los factores. Por esta razón, que el enfoque propuesto contribuye a generar un alto valor para la industria al optimizar diseños. Además de esto proporciona una gama de opciones óptimas entre las cuales se puede escoger la solución deseada.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar un método numérico para la optimización multi-objetivo del diseño de resortes helicoidales a compresión con la finalidad de minimizar la masa y maximizar la resistencia a la fatiga y la robustez del diseño.

1.3.2. Objetivos específicos

- Identificar los criterios de diseño, formulaciones matemáticas, restricciones y funciones objetivo para el planteamiento del problema de optimización.
- Implementar el algoritmo NSGA II para la optimización de los resortes helicoidales
- Evaluar el rendimiento del algoritmo para la verificación de su efectividad.

1.4. Metodología general

Durante esta sección se pretende describir los pasos metodológicos a completar. Estos serán detallados a profundidad en el capítulo específico de la metodología. Sin embargo, a manera introductoria se mencionarán las técnicas de investigación y ejecución con la finalidad de enmarcar la metodología bajo el cual se desarrolló esta investigación.

Investigación teórica.

Para apoyar los conocimientos en esta área del diseño, se efectuará una recopilación bibliográfica de artículos científicos y libros de diseño con énfasis en optimización multi-objetivo y diseño de resortes, estos darán el sustento teórico al trabajo de diseño. Para estos efectos se utilizarán los recursos en línea del sistema bibliotecario de la Universidad de Costa Rica así como otras fuentes externas.

Definición de parámetros y criterios de diseño

Una vez realizada la investigación teórica se procederá a detallar, definir y justificar los criterios y parámetros de diseño bajo los cuales estará sometido el resorte. En este segmento se definirán, además, las restricciones que afectan al algoritmo y los ámbitos posibles para todas las variables.

Proposición del algoritmo a utilizar

Una vez definidos los parámetros de diseño a utilizar, se procederá a proponer la versión del algoritmo NSGA II a utilizar, los criterios de paro, cantidad de iteraciones y todos los otros parámetros del mismo. En esta fase además, se hace un diagrama de flujo que facilite el entendimiento del algoritmo de manera gráfica.

Implementación del algoritmo.

En esta etapa del trabajo se implementará el algoritmo utilizando lenguaje de MatLab. Esto se realizará siguiendo la formulación propuesta en el paso metodológico anterior, y convirtiéndolo en el código requerido para implementarlo. Esta etapa conlleva, además, todo lo necesario para poder ejecutar el algoritmo propuesto.

Ejecución del algoritmo en diferentes casos de diseño.

Una vez implementado el algoritmo, se pretende correr bajo diferentes casos específicos con la finalidad de poder compararlo con lo que hay disponible en la literatura como punto de referencia. Cabe destacar que, debido a que las condiciones de operación y objetivos no son los mismos no se pueden comparar directamente. Sin embargo, es un buen punto de referencia para empezar a validar el funcionamiento del algoritmo. Además se pretende analizar la robustez de los resultados, comparando el cambio en los resultados al cambiar los parámetros de diseño para corroborar su confiabilidad.

Evaluación del método numérico utilizado.

Finalmente, una vez implementado el algoritmo se pretende realizar una comparación y evaluación del método utilizado. Esto se realizará haciendo varias corridas del algoritmo para verificar su convergencia y que sus resultados sean homogéneos. Además se hará

una comparación de las condiciones en la frontera pareto-óptima para las corridas entre los casos de estudio propuestos, con la finalidad poder valorar los resultados obtenidos a partir del algoritmo.

1.5. Alcance y limitaciones

Esta investigación pretende implementar un algoritmo de optimización para resortes helicoidales usando lenguaje Matlab. Sin embargo, no pretende desarrollar el programa como un ejecutable independiente, ni se provee interfase de usuario; cualquier persona que desee hacer uso del programa podrá hacerlo usando directamente el código fuente. El proyecto se enfocará principalmente en el desarrollo del código y en la verificación de los resultados obtenidos a partir del mismo.

Asimismo, la investigación no pretende correr pruebas con prototipos o con probetas físicas para verificar resultados. Como entregables finales, se presenta el código del programa, junto con la definición del problema y un análisis de los resultados obtenidos. Sin embargo, el método numérico desarrollado no garantiza encontrar un óptimo absoluto, sino encontrar el óptimo relativo señalado por la forma de las funciones objetivo. Por otro lado, en esta investigación no se pretende optimizar todos los accesorios que componen la válvula. Asimismo, las cargas que experimenta el resorte es considerado como dato de entrada en el proceso de optimización.

1.6. Aportes y productos materiales

En esta sección se presentan los productos y aportes intelectuales que surgen a partir de esta investigación. Esta lista no pretende ser exhaustiva, sino que describe las principales contribuciones creadas.

Método de cálculo para optimización de resortes

Uno de los aportes de este trabajo será la descripción y definición del método de cálculo creado para la optimización de resortes. Este método incluye el análisis dinámico del sistema con la finalidad de determinar las fuerzas críticas contra fatiga, la robustez del diseño y la minimización de la masa. Además, verifica el cumplimiento de las restricciones según se detalla en el capítulo de metodología.

Código fuente original con la implementación del algoritmo.

En la sección de anexos se presenta el código y subprogramas utilizados para implementar el algoritmo. El anexo contiene detallado el código fuente original con todos los subprogramas junto con una explicación en los comentarios del mismo. Tener el código fuente implica que el algoritmo puede analizarse con mucho mayor detalle y recrear la metodología utilizada a lo largo de este trabajo. Asimismo, se presenta una guía de los pasos para utilizar el programa y poder recrear los casos de estudio.

Manuscrito de artículo presentado en la jornada de investigación.

Se presenta el manuscrito del resumen extendido presentado en la tercera Jornada de Investigación de la Universidad de Costa Rica del año 2021. El manuscrito representa una ponencia de investigación realizada que contiene un resumen de la metodología así como una representación de los resultados obtenidos.

Memoria de aplicación a casos particulares y su consecuente análisis.

En el transcurso del documento de la investigación se presenta el detalle de los casos de estudio particulares que se analizaron. Además en capítulo 4 del documento se muestran los resultados obtenidos al utilizar el método numérico en estos casos de estudio, así como un análisis de los mismos.

Durante este capítulo se lograron definir las bases bajo las cuales se desarrolló esta investigación, además de los objetivos que se cumplieron a lo largo de la misma. También hace énfasis en identificar el alcance que tendrá el tema de investigación y las limitaciones que regirán el desarrollo de las siguientes partes del documento de investigación.

Capítulo 2

Marco teórico

El propósito de este capítulo es establecer y sintetizar los fundamentos teóricos y antecedentes técnicos que sustentaran esta investigación. Se describe tanto el proceso de diseño de resortes como la teoría necesaria para poder comprender con claridad las decisiones tomadas en la implementación del algoritmo.

2.1. Válvulas de control

El resorte en el cuál se enfocará esta investigación pertenece a una válvula de control como la que se muestra en la figura 2.1a. Estas válvulas de control funcionan mediante un pistón controlado por un resorte que puede desplazarse a lo largo de un cilindro. La compresión y resistencia de este resorte se puede lograr mediante un tornillo que poseen. La posición en la que se coloca este tornillo se escoge según el punto de operación en el cuál se desee que la válvula opere [5]. Es por esta razón que el resorte de la válvula es uno de los elementos críticos para una operación correcta y segura de la válvula.

La ilustración mostrada en la figura 2.1b muestra de una manera más clara y simplificada el funcionamiento y estructura de la válvula. La válvula funciona como una caída de presión en la línea por la que pasa el fluido, por lo que en la figura 2.1b el valor de P_1 es mayor al valor de P_2 . Además la precompresión está demarcada en la figura por F_1 , que se logra a través del ajuste de un tornillo en la parte superior [19]. Esta precompresión es determinada por el usuario según las presiones de trabajo y se tomará en cuenta al momento de realizar el análisis dinámico de la válvula.

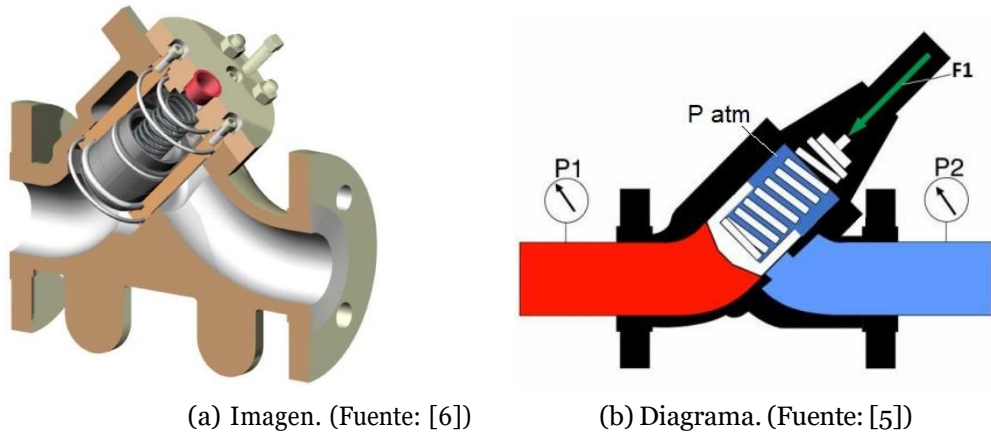


Figura 2.1: Figuras representativas de la válvula

2.2. Resortes helicoidales a compresión

2.2.1. Parámetros de diseño

Durante este segmento y a lo largo del trabajo se utilizarán frecuentemente variables relacionadas con el diseño de resorte. El cuadro 2.1 define las variables que se usarán y sus unidades con la finalidad de evitar de definir las múltiples veces a lo largo del trabajo. La figura 2.2 presenta una imagen en la cual se resaltan las variables geométricas del resorte que están definidas en el cuadro.

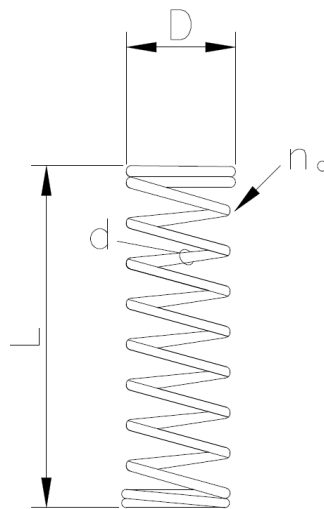


Figura 2.2: Imagen representativa de las variables del resorte. L es la longitud del resorte, D el diámetro del resorte, d el diámetro del alambre y n_a la cantidad de espiras activas. (Fuente: el autor.)

Cuadro 2.1: Identificación de variables frecuentes en diseño de resorte (Fuente: el autor).

Variable	Significado (unidades)
d	Diámetro del alambre (mm)
D	Diámetro exterior del resorte (cm)
E	módulo de Young del material (Pa)
F_{max}	Máxima fuerza del resorte antes de comprimirse totalmente (N)
L_{libre}	Longitud del resorte libre (cm)
G	Módulo de cizalladura o a cortante del material (Pa)
W	Factor de corrección de Wahl
k	constante del resorte (N/m)
n_a	Cantidad de espiras activas del resorte
n_t	Cantidad de espiras totales del resorte
ρ	Densidad del material (kg/m^3)
θ	Ángulo de las espiras del resorte (grados)
T_{max}	Máximo esfuerzo cortante al que está sometido el resorte Pa
A	Área transversal del alambre del resorte (m^2)
K_s	Factor de corrección por el esfuerzo cortante
K_c	Factor de corrección por la curvatura del resorte
C	Índice del resorte = $\frac{D}{d}$

2.2.2. Esfuerzo máximo y pandeo

Producto de las cargas de compresión, se genera en las espiras del resorte un estado de esfuerzos cortantes y torsionales tal y como se muestra en la figura 2.3. Según el estado de carga mostrado en la figura se puede definir la ecuación para T_{max} presentada en la ecuación (2.1). Además de esto, al hacer ciertas simplificaciones se puede pasar la ecuación del esfuerzo a la presentada en la ecuación (2.2). En esta se incluye el factor K_s para corrección por la el esfuerzo a cortante y se puede calcular mediante la ecuación (2.3). El factor K_c se usará multiplicando en la ecuación (2.2), con la finalidad de introducir una corrección por la curvatura del resorte. El factor K_c calculado según se presenta en la ecuación (2.4) es más conservador que el definido en K_s . Por esta razón se utilizará la forma de el esfuerzo máximo mostrado en la ecuación (2.5)[7].

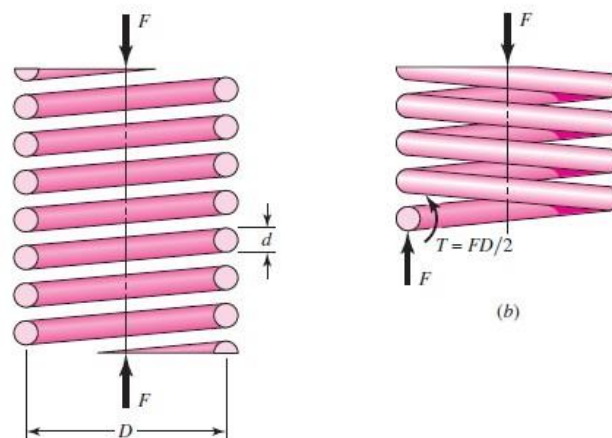


Figura 2.3: Imagen representativa de la torsión en un resorte. (Fuente: [7].)

La ecuación (2.5) muestra como el esfuerzo en las espiras del resorte se basa principal-

mente en el diámetro del resorte y del alambre, así como las características del material. Basado en el esfuerzo máximo, se puede calcular el esfuerzo de von Mises que, al no poseer esfuerzos normales se simplifica al valor presentado en la ecuación (2.6). Esto es debido a que cuando se tiene una condición de cortante puro, se puede calcular el cortante de fluencia con el esfuerzo normal de fluencia.

$$\tau_{max} = \frac{T r}{J} + \frac{F}{A} \quad (2.1)$$

$$\tau_{max} = K_s \frac{8FD}{\pi d^3} \quad (2.2)$$

$$K_s = \frac{2C+1}{2C} \quad (2.3)$$

$$K_C = \frac{2C(4C+2)}{(4C-3)(2C+1)} \quad (2.4)$$

$$\tau_{max} = K_C \frac{8FD}{\pi d^3} \quad (2.5)$$

$$\sigma_{vonMises} = \sqrt{\frac{3}{2}} \tau_{max} \quad (2.6)$$

Además de verificar el esfuerzo máximo en el resorte, se debe asegurar su estabilidad contra pandeo. Para esto se tiene que verificar que el resorte cumpla con la condición presentada en la ecuación (2.7), en donde α es una constante multiplicativa que depende del tipo de terminación del resorte.

$$L_{libre} < \frac{\pi D}{\alpha} \left(\frac{2(E-G)}{2G+E} \right)^{1/2} \quad (2.7)$$

La ecuación (2.7) se utiliza debido a que los resortes helicoidales suelen ser bastante esbeltos. Esto significa que existe una alta posibilidad de que fallen en alguno de sus modos de pandeo a una carga menor a la cual el material de que están hechos entraría en fluencia. Una longitud libre menor a la mostrada en esta ecuación ayudará a garantizar que el resorte tenga el suficiente diámetro para no fallar con facilidad al pandeo. La ecuación depende de tipo de apoyo del resorte, ya que los grados de libertad de los extremos evitarán, o facilitarán, algunos de los modos de pandeo del resorte [7].

2.3. Criterio de diseño para fatiga no armónica

Para el caso de fatiga no armónica uno de los criterios de diseño más utilizados comúnmente es el del conteo de ciclos *rain flow* [20] como es definido en el ASTM E1049-85 utilizado en conjunto con el criterio de acumulación de daño de Miner [21]. El diseño consta principalmente de dos etapas: una primera etapa en el que las fuerzas a las que se expone el componente son convertidas a ciclos con magnitud equivalente y una segunda en la cual se utiliza la acumulación de daño.

Se comenzará por detallar el proceso de conteo de ciclos *rain flow*. El objetivo principal de esto es reducir el espectro de esfuerzos variables a un espectro equivalente de esfuerzos simples reversibles los cuales generan la misma cantidad de daño en el material

[22]. Para realizar el conteo de ciclos se comienza por definir el ámbito de los esfuerzos en consideración X , Y que es el ámbito anterior adyacente a X y S que es el punto inicial del historial de esfuerzos graficado en estudio[21]. Los pasos para realizar el conteo de ciclos se describen a continuación. Asimismo en la figura 2.4 se muestra el diagrama del proceso para dar mayor claridad al lector.

1. Se comienza a partir de el punto S .
2. Se anota el siguiente pico o valle que se encuentre en el historial, si no hay suficientes datos en todo el historial salte al paso 7.
3. Si hay menos de tres puntos en el ámbito, repita el paso 1. Al tener tres puntos haga los ámbitos X y Y usando los tres puntos más recientes que no hayan sido descartados
4. Compare los valores absolutos de los ámbitos X y Y
 - a) Si $X < Y$, vaya al paso 2
 - b) Si $X \geq Y$, vaya al paso 5
5. Si el ámbito Y contiene el punto S , vaya al paso 6. Si no lo contiene cuente Y como un ciclo con la magnitud de su ámbito; descarte el pico y valle de Y y continúe al paso 3.
6. Cuente Y como un medio ciclo; descarte el primer punto de Y y mueva el punto inicial S al segundo punto del ámbito Y , repita desde el paso 3.
7. Cuente cada ámbito que no fue contado previamente como un semi-ciclo

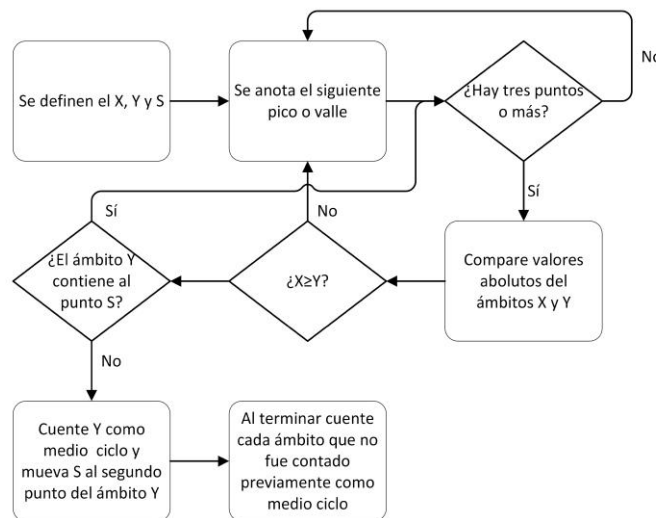


Figura 2.4: Flujograma del conteo de ciclos *rain flow*. (Fuente: el autor.)

Un ejemplo de como funciona este método se muestra en la figura 2.5(a). En este se toma A como el punto S , el ámbito $|B - A|$ como Y y X sería el ámbito $|C - B|$. Esto debido a que, como se definió con anterioridad S es el punto inicial, Y el ámbito anterior a X y X sería el ámbito en estudio. Al compara este X y Y se nota que $X > Y$ por lo que se utiliza el criterio de decisión en el paso 4(b) del algoritmo. Como el ámbito de Y

contiene a S que es el punto inicial se toma Y como un medio ciclo y se descarta el punto inicial como se muestra en la figura 2.5(b). Posteriormente se sigue el proceso trasladando el punto S al punto B de la gráfica ya que el punto A fue descartado. Esto transforma la gráfica de la forma a la que se muestra en la figura 2.5(c). Posteriormente se redefinen X y Y , en el cual se repite el mismo procedimiento que lleva a la eliminación del punto B y se añade otro medio ciclo. El algoritmo se continúa al redefinir el punto S en C y se repiten los pasos según se ejemplifica en el diagrama de la figura 2.4.

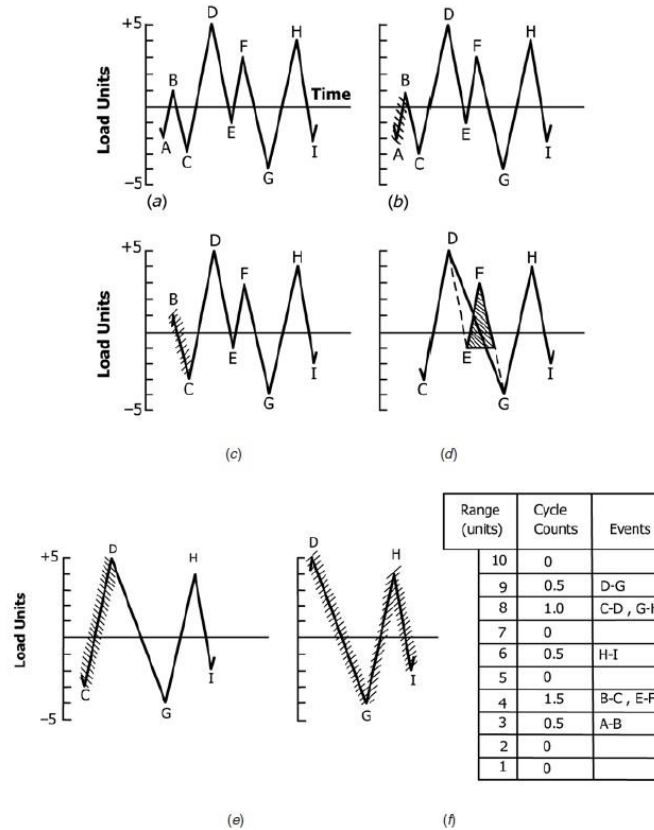


Figura 2.5: Imagen representativa del conteo de ciclos *rain flow*. (Fuente: [21].)

El algoritmo se sigue hasta que finalmente se llega a un resultado similar al mostrado en la figura 2.5(f). En el que ya no queda ningún segmento del patrón de fuerzas que procesar y lo que se tiene es un conteo de ciclos y las magnitudes que tuvieron los mismos. Una vez identificados los ciclos y sus magnitudes se puede proceder con el método de acumulación de daño de Miner para fatiga no armónica [23]. Para este, se define N como el número total de ciclos que soporta un elemento bajo una fatiga con un esfuerzo dado S . Estos datos se tienen experimentalmente para esfuerzo armónico. De manera generalizada, para un material se puede definir calcular N mediante la ecuación (2.8) en donde a y b son constantes del material [24].

$$N = aS^{-b} \quad (2.8)$$

Una vez establecido esto, definimos n_i como el número de ciclos que se obtuvieron mediante el conteo de *rain-flow* con un valor de esfuerzo S_i , el cual tendría una capacidad de

vida útil total igual a N_i . A estos se le puede atribuir un a fracción de vida útil consumida equivalente. De esta manera se tiene que para que una pieza fallará cuando la sumatoria de todas las fracciones de vida consumida sea igual a la unidad, tal como lo expresa la ecuación (2.9) que representa el momento de falla [25].

$$\frac{n_1}{N_1} + \frac{n_2}{N_2} + \frac{n_3}{N_3} + \dots = 1 \Rightarrow \sum \frac{n_i}{N_i} = 1 \quad (2.9)$$

La ecuación de Miner representa una relación en la cual un un ciclo con un valor de esfuerzo dado S_i causa un daño igual sin importar cuando se de este en el historial de carga. Utiliza un método de acumulación de energía basado en la cantidad de trabajo absorbido requerido para fatigar el material. Sus principales puntos débiles son la independencia de la secuencia y nivel de carga, así como la falta de contabilizar las interacciones entre las cargas [25].

$$\frac{n_1}{N_1} + \frac{n_2}{N_2} + \frac{n_3}{N_3} + \dots = F_{Miner} \Rightarrow \sum \frac{n_i}{N_i} = F_{Miner} \quad (2.10)$$

La ecuación (2.10) muestra la definición del factor de Miner que es una representación de la fracción de vida útil consumida del componente. Esto implica que entre más se acerque el número a 0 se tendrá un diseño que se encuentra más lejos de la falla por fatiga. El resultado del método de Miner se calcula de tal forma que entre menor sea el número calculado por medio de la ecuación (2.10) se tendrá un diseño que está más lejos de la falla por fatiga. Esto es de utilidad al ser usado en un algoritmo de optimización ya que se puede utilizar como un valor representativo para poder comparar la resistencia mecánica a la fatiga de dos componentes dados. Al tener el número de Miner de dos elementos, aquel que tenga un menor número tendrá una mejor resistencia a la fatiga ante un patrón de fuerzas dado.

2.4. Diseño por robustez

El diseño por robustez consiste en que un producto o elemento mantenga las funciones y desempeño deseado ante variaciones externas, o que sea afectado de manera mínima. Es decir, un producto es diseñado robustamente si es poco sensible ante ruido incontrolable de alguna de las variables de diseño [26]. Esto se puede explicar con el ejemplo mostrado en la figura 2.6, donde se evidencia como la solución de una optimización convencional no es coincidente con la de diseño robusto. La solución del diseño robusto busca el punto que minimiza la variación de la respuesta ante un cambio en las variables de entrada de la función. Un ejemplo de esto, en el caso de los resortes es minimizar el cambio en la masa y la resistencia a la fatiga ante cambios no controlables en las dimensiones del diámetro del resorte y el alambre. Estas variaciones no controlables son atribuibles a las tolerancias del material y del método de manufactura y que al escoger un punto de diseño robusto se minimiza el impacto de estas variaciones.

Existen varias metodologías para realizar un diseño robusto, pero una de las más

frecuentes en el diseño mecánico es la formulación de la sensibilidad mínima [27]. Este

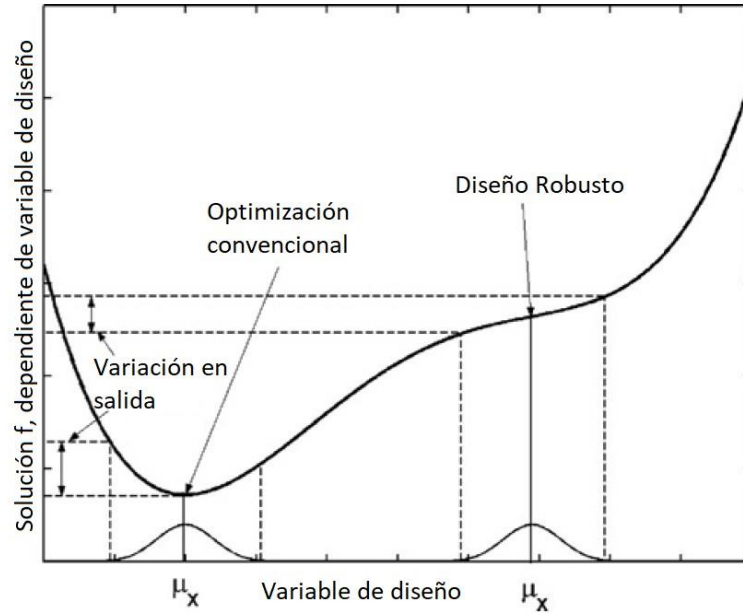


Figura 2.6: Gráfico que ejemplifica un diseño robusto. La imagen muestra la diferencia entre un punto de optimización convencional y una optimización por robustez de diseño. (Fuente: [16].)

método utiliza teoría de que al minimizar la sensibilidad a una de las variables, implica una reducción en la probabilidad de falla en un diseño [28]. Si se define una función F que representa la respuesta de interés del sistema, la cual depende del diseño y las variables de diseño sin incertidumbre b y las variables con incertidumbre x , tal como se presenta en la ecuación 2.11 [28].

$$F = F(b, x) \quad (2.11)$$

La idea es determinar un diseño objetivo b^* tal que la sensibilidad de F respecto a x sea un mínima, por lo que el problema se reduce al presentado en la ecuación (2.12). La ecuación (2.12) queda sujeta a localizarse entre valores requeridos y críticos, así como también cumplir con los otros criterios de diseños definidos por g_i como se muestra en la ecuaciones (2.13) y (2.14) [28].

$$\min_{dx} \frac{dF(b, x)}{dx} \quad (2.12)$$

$$F(b, x) \leq F^{diseno} < F^{cr} \quad (2.13)$$

$$g_i(b, x) \leq 0, i = 1, \dots, m \quad (2.14)$$

Este método de diseño por robustez también se puede definir como minimizar la sensibilidad a las variables de diseño. Esta sensibilidad, por ende se puede definir por medio de la ecuación (2.15), en la cual S es el factor de sensibilidad, F es la función objetivo y D_p son las variables requeridas [26].

$$S = \frac{\Delta Fr / Fr}{\Delta Dp / Dp} \approx \frac{\delta Fr / \delta Dp}{Fr / Dp} \quad (2.15)$$

La investigación pretende tomar ventaja de este método de diseño por robustez para poder obtener por medio de la optimización un diseño que sea más resistente a los cambios en las variables de diseño. El diseño robusto presenta una alternativa a los criterios de diseño convencionales que toma en cuenta factores que se presentan en la manufactura de componentes y ayuda a reducir los requerimiento de tolerancias [27].

2.5. Algoritmos para optimización multi-objetivo

Para iniciar este segmento se pretende realizar un resumen de algunas de las publicaciones e investigaciones similares que se han realizado en el tema de optimización. Esto con la finalidad de validar y respaldar las decisiones tomadas a lo largo de este segmento y también darle un marco de referencia al contenido del mismo. Cabe destacar, que este segmento no pretende ser exhaustivo en cuanto a las fuentes existentes, sino apuntar a hacia los principales documentos existentes que ayudan a sustentar el mismo.

El tema de optimización de un diseño con base en la resistencia a parámetros de interés es uno ampliamente estudiado. Entre los autores y casos similares al de este trabajo se tiene la reducción de elementos de un compresor basado en el diseño robusto por Zhang et al [8]. Este artículo tiene la gran similitud de que uno de los objetivos es realizar una optimización por robustez semejante a la de este trabajo. Además de este, otro trabajo similar es el de Jian et al [29], en el cual se optimiza un válvula de control utilizando un algoritmo genético con dos objetivos. En la referencia [29] el énfasis realizado es principalmente hacia la estabilidad y constante del resorte. Sin embargo, se utilizan fuerzas variables que son una referencia de alto valor con la finalidad de que sean ejemplo y dan base a las fuerzas a las cuales se someterán los componentes en esta investigación. En el artículo de Kim et al [9] se presenta una optimización de una válvula de alivio utilizando un algoritmo genético similar al que se utilizará en este trabajo, pero con dos funciones objetivo. El artículo presenta como variables objetivo la reducción de la masa del sistema y la respuesta dinámica del mismo. La investigación, además, presenta un análisis dinámico del sistema que es de gran utilidad debido a la semejanza que posee con el que se realizará en este artículo, además de poseer la masa como una de las funciones objetivo.

La investigación de Zhou et al [10] referente a la optimización de un embrague de automóvil posee una aproximación similar a la que se pretende realizar en en este trabajo. En ella se optimiza un embrague de diafragma de un automóvil con dos funciones objetivo: la masa y la frecuencia del sistema. En este, en vez de utilizar un algoritmo genético se utiliza uno híbrido de selección paralela con una función compartida, que es básicamente otro algoritmo determinista para hacer una optimización multi-objetivo.

Finalmente se tiene el artículo de Nikola et al [30] en el cual se realiza la optimización multi-objetivo de una suspensión de un vehículo. Este utiliza el algoritmo NSGA II con tres variables objetivo: la aceleración, el desplazamiento de la suspensión y la deflexión de las llantas. Este a su vez realiza un estudio dinámico con la finalidad de conocer la respuesta del sistema y poder optimizar respecto a la misma. Debido a esto, el artículo también posee similitudes con esta investigación que son un buen punto de referencia para basar las decisiones a tomar en la implementación metodológica de esta investigación.

2.5.1. Optimización multiobjetivo

La optimización multiobjetivo es un área que busca dar un respaldo matemático a problemas que contengan más de una función objetivo que tienen que ser optimizadas simultáneamente. La optimización multiobjetivo busca realizar la búsqueda de un conjunto de soluciones finales óptimas entre las cuales se escoge una opción final basada en un criterio de decisión [31]. Este tipo de optimización busca encontrar la frontera óptima en el espacio de solución, denominada la frontera Pareto óptima. Un punto x^* se puede definir como perteneciente de la frontera Pareto óptima dominante S si no existe otro punto x tal que $f_i(x) \leq f_i(x^*)$ para todos los puntos del espacio solución, en donde el resultado de la función f se desea minimizar. Se puede decir que un punto que no es completamente dominado por otro punto del espacio solución cuando al mejorar los valores de una de las funciones tendrá el efecto de empeorar el valor de idoneidad en alguna de las otras funciones, de esta manera existe un punto tal que sea objetivamente mejor en todas las magnitudes de sus valores de idoneidad [32]. Una ejemplificación clara de este concepto se puede apreciar en la figura 2.7, donde si se quisiera minimizar f_1 y f_2 , los puntos sobre la línea azul estarían sobre la frontera Pareto óptima [31].

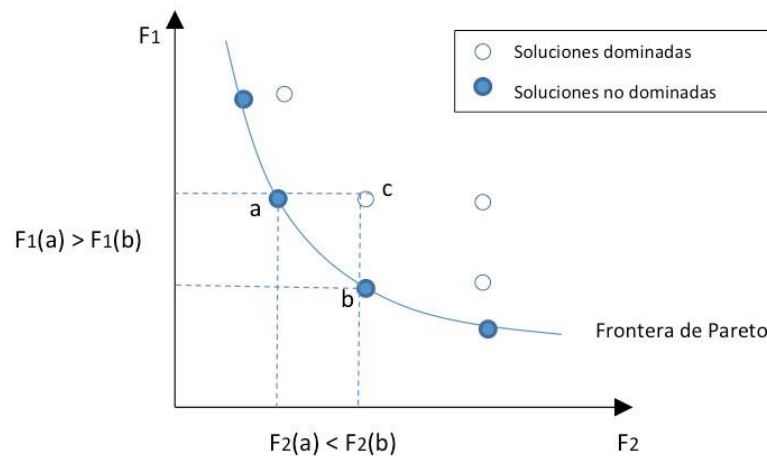


Figura 2.7: Imagen representativa de una frontera Pareto-óptima dominante. (Fuente: [33].)

Cabe destacar que mientras que un problema de optimización con una sola función objetivo que solo tiene una solución dominante sobre todos los otros puntos del espacio solución un problema de optimización multi-objetivo tiene un conjunto de soluciones dominantes que son aquellas soluciones que no son completamente dominadas por ningún otro punto del espacio solución. En casos en los que se tienen varias funciones objetivo, como es el caso que desarrolla en esta investigación, se tiene un conjunto de soluciones óptimas entre las cuales se puede escoger. La forma de la frontera Pareto óptima dependerá de la cantidad de funciones objetivo. Como ejemplo de esto un caso con dos funciones objetivo presentará una curva como la mostrada en la figura 2.7, mientras que una con tres funciones objetivo tendrá una curva tridimensional o superficie.

2.5.2. Algoritmos evolutivos

Un algoritmo evolutivo se puede definir como un algoritmo genérico meta-heurístico basado en la población. Esto quiere decir que representa una forma de solucionar un problema en el cual los métodos tradicionales de optimización son muy lentos, o no logran

llegar a un resultado. A pesar de esto, este tipo de algoritmos no garantizan obtener el mínimo global en algunos problemas [34].

Este tipo de algoritmos se basan en métodos de optimización estocástica, es decir que utilizan variables generadas aleatoriamente para encontrar una solución. Estos se basan principalmente en una serie de pasos de optimización combinatoria en la cual el resultado se logra buscando sobre un gran número de soluciones posibles. Una de las ventajas principales que poseen es la posibilidad de obtener soluciones viables con menor poder computacional que métodos directos no estocásticos. Además, debido a la metodología en la cual se buscan los individuos de la población, este tipo de algoritmo tiene la capacidad de evadir mínimos locales al hacer un muestreo de la población, y no utilizar gradientes para encontrar un punto óptimo [34].

De manera generalizada, se tiene que un algoritmo evolutivo debe presentar como mínimo los pasos básicos generales que se presentan en la figura 2.8 [35]. Como se puede observar en la figura, los algoritmos evolutivos comienzan con una población generada aleatoriamente de las posibles combinaciones de variables y posteriormente se analiza su ajuste y posición en el espacio solución. Con base en este análisis se crea una nueva generación a partir de las mejores soluciones, introduciendo combinaciones entre ellas. Cabe destacar, además, que se introducen agentes de aleatoriedad conocidos como mutaciones para asegurarse que el espacio buscado no se estanque, y garantiza tener variabilidad entre las soluciones padres e hijas [36]. Debido a su semejanza al proceso de reproducción genética se le llama a cada combinación de variables un individuo y al conjunto de los individuos se le llama población. De manera similar se dice que las combinaciones de estos individuos pueden generar hijos de los cuales los individuos originales serán los padres.

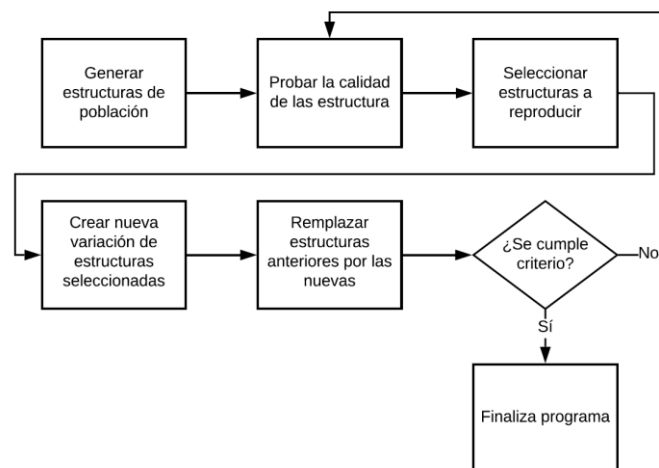


Figura 2.8: Diagrama básico de un algoritmo evolutivo. (Fuente: el autor)

Como su nombre lo dice, un algoritmo evolutivo se basa en el proceso de selección natural en una población de soluciones candidatas. Esta población se evalúa según una función de aptitud para establecer mejores candidatos y estos tendrán hijos que formarán la siguiente generación. Además de esto se introduce en el algoritmo un porcentaje de mutación y de cruce o recombinación entre las variables para mantener la diversidad de la misma [37].

La figura 2.9 muestra un diagrama simplificado de este tipo de algoritmo en el cual se tiene una solución inicial, posteriormente se evalúan las funciones objetivo y según los resultados de las mismas se genera una nueva población. Este ciclo se repite hasta alcanzar el criterio de convergencia, que usualmente es un número definido de generaciones [37].

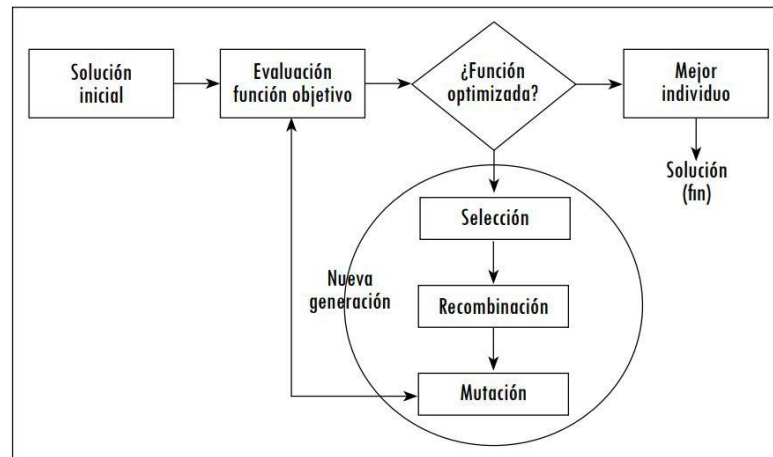


Figura 2.9: Diagrama simplificado de un algoritmo evolutivo. (Fuente: [38].)

Los algoritmos evolutivos son un proceso iterativo. La figura 2.9 muestra con claridad el ciclo utilizado para crear nuevas generaciones y como la recombinación y la mutación son las principales operaciones para crear diversidad en la población con base en la cual se está optimizando. Las nuevas generaciones se evalúan y ordenan según la función objetivo de interés para determinar cuáles son los óptimos y de ahí sacar la nueva generación [39].

2.5.3. Algoritmo genético ordenado no dominado (NSGA II)

Los algoritmos genéticos son una subdivisión de los algoritmos evolutivos. Uno de los más predominantes entre ellos es el algoritmo genético en ordenamiento no dominado II (NSGA-II por sus siglas en inglés). En este algoritmo, antes de hacer la selección de estructuras a reproducir presentado en la figura 2.8, la población se clasifica según qué tan dominante es cada una de las soluciones. Todas las soluciones no dominantes se incluyen en una categoría, y las dominantes en otra [36].

Posteriormente, después de haber clasificado los resultados, se le asigna una mayor cantidad de copias o valores *hijos* a los valores en la clasificación dominante con respecto a la no dominante. Esto se hace de esta manera ya que se asume que tendrán mayor posibilidad de generar buenos resultados. Además de esto, una de las principales características del NSGA-II que lo distingue de los otros algoritmos genéticos que no solamente ordena los resultados de una iteración según su dominancia, sino que también utiliza la distancia de hacinamiento o apilamiento de los datos. La distancia de hacinamiento toma en cuenta no solo que los puntos seleccionados se encuentren en la curva pareto-óptima, sino también la distancia entre los puntos consecutivos en la curva [40].

Al tomar en cuenta la distancia de hacinamiento el algoritmo asegura tener diversidad en el espacio solución, y busca que los puntos solución finales estén distribuidos a lo largo de la frontera Pareto óptima. Esto se hace realizando que puntos muy cercanos en

la curva no lleguen a llenar todo el espacio de resultado. Una manera gráfica de visualizar este concepto se presenta en la figura 2.10. En la figura 2.10 se observa como los valores dominantes con los cuales se creará la siguiente generación se escogen según esta distancia de separación mínima de los valores en la frontera Pareto-dominante hace que se mantenga una diversidad entre generaciones [41].

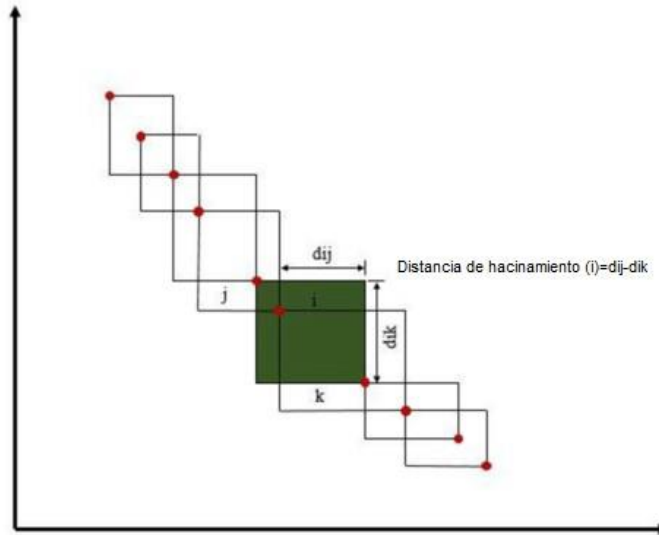


Figura 2.10: Diagrama representativo de la distancia de hacinamiento o apilamiento. (Fuente: [41].)

Esta combinación de características hacen que el algoritmo NSGA-II sea uno de los que da resultados más prometedores, debido su rápida convergencia y relativamente baja demanda computacional [42]. Una imagen representativa del algoritmo se presenta en la figura 2.11 en donde se presenta de manera simplificada los pasos del algoritmo. Además, se notan los pasos adicionales requeridos respecto a un algoritmo evolutivo básico como el mostrado el la figura 2.8 [43].

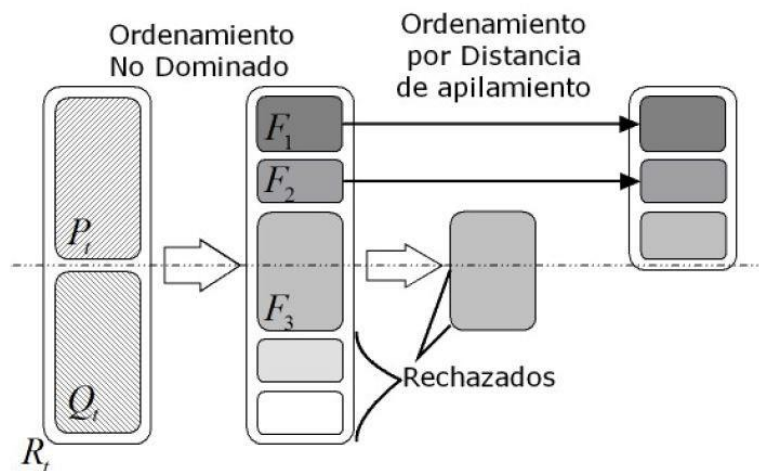


Figura 2.11: Diagrama representativo del algoritmo NSGA-II. (Fuente: [43].)

El algoritmo utiliza los siguientes pasos generalizados:

1. Creación de la población inicial por medio de alguna metodología.
2. Se ordena la población con base en la no-dominancia en los diferentes frentes, esta no-dominancia se obtiene al comparar sus valores de las funciones objetivo para determinar los individuos más fuertes.
3. Se seleccionan los individuos según su posición en el ordenamiento y su distancia de apilamiento.
4. Se crea una población de descendientes utilizando la selección por torneo, el cruce y la mutación.
5. Se combinan los padres e hijos y se hace el ordenamiento no dominado, identificando los frentes.
6. Se escoge la población que seguirá en el proceso tomando en cuenta las distancias de apilamiento para mantener la diversidad.

2.5.4. Selección, recombinación y mutación

En este segmento se pretende adentrar dentro de los concepto de selección, recombinación y mutación. Esto debido a la necesidad de entender estos conceptos para poder comprender en su totalidad la sección de la metodología en donde se explicará específicamente cómo se realizarán estos pasos.

Primero se procederá a explicar en qué consiste el proceso de selección para la recombinación. La operación de selección se basa en el llamado *fitness value* o valor de idoneidad de un individuo de la población. Este valor indica que tan fuerte es un candidato para ser el óptimo de la población, entre mayor sea su número de ajuste, mayor será la probabilidad de ser escogido para seguir adelante y reproducirse [44].

La selección utiliza métodos probabilísticos para determinar cuales serán los individuos padres. Esto se realiza de esta forma para mantener la diversidad de la población y fomentar que las generaciones futuras no se enfoquen únicamente en una región de la frontera pareto óptima. Existen varios métodos de selección, pero en esta sección se adentrará dentro de los dos más comúnmente usados que son el de la ruleta y el de torneo; se comenzará por explicar el de ruleta. El método de ruleta consiste en realizar la selección por medio del equivalente probabilístico a un juego de ruleta, en la cual a los individuos más aptos se les asignará un segmento más grande en la ruleta y, por ende, una mayor probabilidad de ser escogidos para la recombinación [44].

La figura 2.12 representa gráficamente el método de selección por ruleta. En esta imagen los a_j son los diferentes miembros de la población y su tamaño de segmento circular esta asociado según la proporción del valor de idoneidad de cada individuo con respecto a la sumatoria de todos los valores de ajuste de la población. Esto se puede expresar además por medio de la ecuación en donde $ps(a_j)$ es la probabilidad de ser seleccionado del individuo j ésimo en la población P con los individuos a de la forma $P = \{a_1, a_2, a_3, \dots, a_n\}$. El término de $f(a_j)$ representa el valor de idoneidad del individuo específico de la población [44].

$$ps(a_j) = \frac{f(a_j)}{\sum_{i=1}^n f(a_i)}, j = 1, 2, \dots, n \quad (2.16)$$

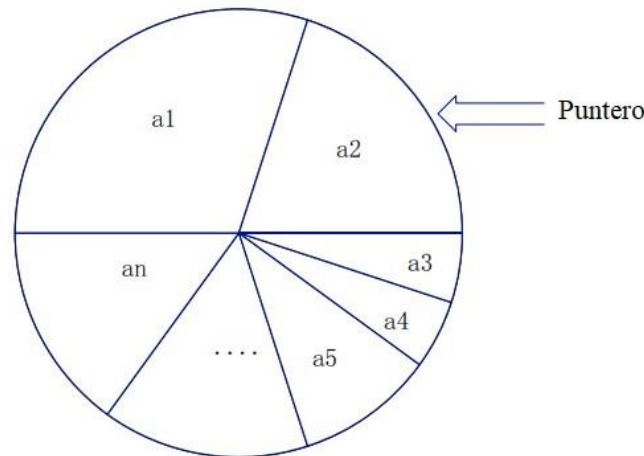


Figura 2.12: Diagrama representativo de la distancia de la selección por ruleta. (Fuente: [44].)

El otro método de selección más comúnmente usado es la selección por torneo. El método de torneo tiene la idea de seleccionar el individuo con el mayor valor de idoneidad entre un grupo que ha sido seleccionado aleatoriamente. El tamaño del torneo puede escogerse dependiendo del problema en estudio pero siempre debe ser mayor a dos [45].

En este tipo de selección no hay un método o formulación matemática para escoger entre el grupo, como si lo había en el ruleta. Simplemente se escogen aleatoriamente individuos para posteriormente escoger el competidor con mayor valor de idoneidad. La forma de selección escogiendo aleatoriamente varios individuos de la población que formaran parte del torneo. De entre estos se escoge el individuo con el mayor valor de idoneidad y este será el escogido para continuar. Ese proceso se repite N veces, siendo N el número de individuos de la población [45]. La figura 2.13 muestra un ejemplo de como funciona este método de selección. En el caso del ejemplo se seleccionan aleatoriamente los individuos A, E y T, para posteriormente seleccionar A como el ganador del torneo. Este individuo se mantendrá para la siguiente generación ya que su valor de idoneidad es el mayor de los tres.

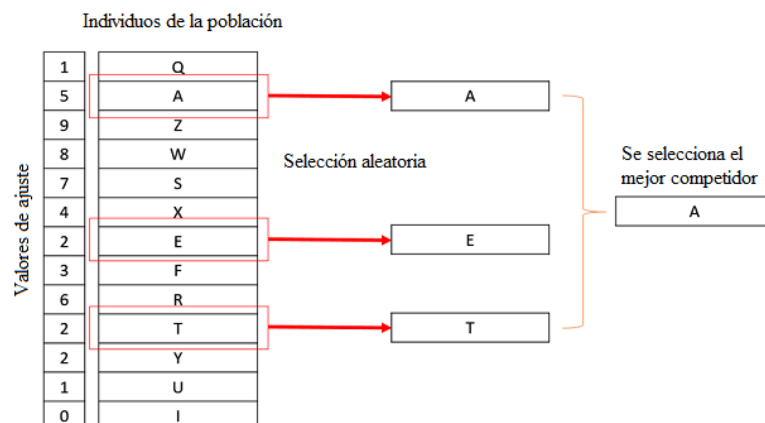


Figura 2.13: Imagen representativo de la selección por torneo. (Fuente: [45].)

El método de torneo, de manera generalizada, proporciona una convergencia más rápida y estable en algoritmos genético con una población suficiente. Como un ejemplo de esto en la figura 2.14 se muestra el caso de la ecuación (2.17) siendo optimizada por los dos métodos para minimizarla. Como se sabe el mínimo de esta ecuación se encuentra en 0, pero al comparar la figura 2.14.(a) con la 2.14.(b) se puede notar como el problema converge mas rápidamente utilizando muchas menos generaciones [44].

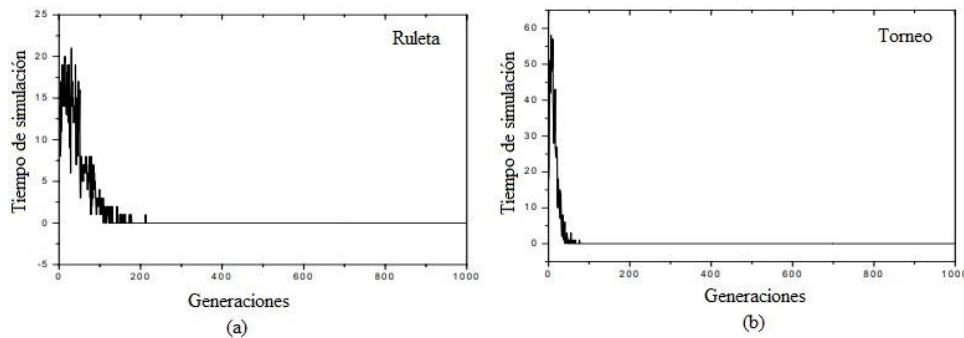


Figura 2.14: Ejemplo de un problema de optimización utilizando selección de ruleta (a) y torneo (b). (Fuente: [44].)

$$f(x) = x_1^2 + x_2^2 \quad (2.17)$$

Ahora se procederá a explicar el proceso de recombinación que consiste en un operador genético utilizado para combinar la información genética de dos individuos de la población con la finalidad de crear uno o más nuevos *hijos*. Es una forma estocástica de generar nuevas combinaciones en el espacio solución a partir de los individuos existentes y es el análogo a la reproducción para la creación de hijos en la naturaleza [46]. La operación de recombinación no utiliza información sobre los valores de ajuste o funciones objetivo, sino que es principalmente una acción en el subespacio de la población. Esto quiere decir que por la acción de recombinación solo se pueden obtener soluciones en las cuales los *chromosomas* o variables de los nuevos individuos son una combinación lineal de los anteriores o padres [46].

Como un ejemplo de esto se presenta un sistema binario en el que los cromosomas tienen solo dos opciones *a* o *b* y cuatro cromosomas, se tiene un individuo $S_1 = [aabb]$ y otro $S_2 = [abaa]$. Debido a que al recombinación no genera nuevos componentes podemos asegurar que el primer componente del individuo hijo sera *a* y los otros 3 componentes serán una combinación de las opciones de las otras tres variables de los padres, este ejemplo de la recombinación se presenta en la figura 2.15 [47]. La imagen mostrada en la figura 2.16 tambien ejemplifica lo que sucede durante la operación genética de recombinación al tener dos individuos con opciones de los valores del 1 al 99 y tres cromosomas para ayudar a brindar más claridad a este concepto. Este segundo ejemplo para los cromosomas del hijo se obtuvo el primer cromosoma del padre *A* y los otros dos cromosomas del padre *B* y es un ejemplo claro de lo indicado anteriormente que un hijo solo podrá obtener cromosomas que son una combinación de la de sus padres mediante el proceso de recombinación.

Finalmente, se tiene mutación que es un operador genético que se utiliza para mantener la diversidad de los cromosomas en la población de una generación a la siguiente. La mu-

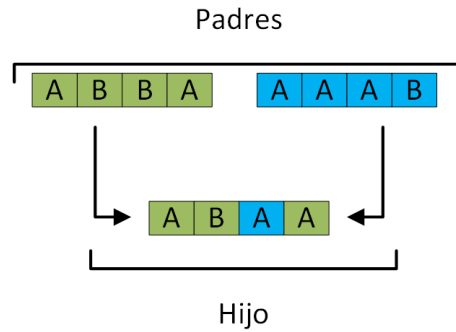


Figura 2.15: Imagen representativa de la recombinação. (Fuente: el autor.)

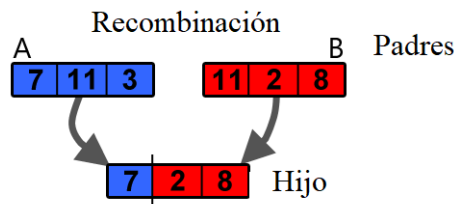


Figura 2.16: Imagen representativa de la recombinação. (Fuente: [48].)

tación altera uno o más genes en un individuo o cromosoma de manera aleatoria. Este, a diferencia de la recombinação, puede generar individuos con cromosomas completamente diferentes, al ser un cambio aleatorio [49].

Existen varias estrategias para la mutación por lo que se pretenden describir las más prominentemente utilizadas con los algoritmos genéticos. La primera es la mutación uniforme, en esta simplemente uno de los valores de los genes escogido se reemplaza con un valor aleatorio dentro de los posibles. Esta es relativamente simple de ejecutar, sin embargo posee la desventaja que no utiliza ninguna información de la población de la cual se está mutando y puede proveer un punto demasiado lejos de la frontera pareto-óptima [50].

La segunda estrategia que se explicará es la mutación polinomial, esta en vez de realizar un cambio de variable completamente aleatorio, se realiza una perturbación al cromosoma para obtener un resultado relativamente cercano o colindante al del padre. Como ejemplo de esto, si se tiene un padre con un valor de $\rho = 3$ la imagen de la figura 2.17 muestra la distribución de probabilidad que tendría el hijo utilizando una distribución polinomial. Cabe destacar que el ancho y ámbito de esta distribución se puede modificar según la ecuación de distribución polinomial que se utilice [50].

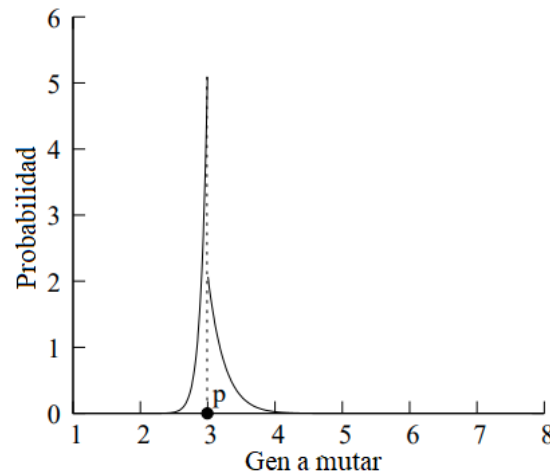


Figura 2.17: Imagen representativa de la distribución de la mutación polinomial. (Fuente: [50].)

Lo explicado en este capítulo será referido en secciones posteriores de este trabajo con la finalidad de lograr la comprensión plena por parte del lector. Cabe destacar, además, que esta sección no pretende englobar la totalidad de los temas desde lo más básico. Además de esto el marco teórico presenta una base de fundamentos que ayudarán a justificar las decisiones tomadas a lo largo de este documento.

Capítulo 3

Metodología

Este capítulo presenta una explicación detallada de los pasos y estrategias utilizadas para la implementación del algoritmo. Además, presenta y justifica los tipos de casos de estudio que se analizarán en la parte de resultados de este documento. La idea principal de este capítulo es que futuros lectores puedan recrear y entender el procedimiento utilizado en este trabajo. La metodología descrita en este capítulo sirve como un marco que le da validez y justifica los resultados presentados posteriormente en este documento.

3.1. Sinopsis metodológica

La implementación del algoritmo inicia por determinar las condiciones de operación de la válvula y las restricciones físicas que se utilizarán para la optimización del diseño del resorte. Se comienza por establecer un modelo dinámico con el cual se podrá calcular la respuesta del resorte ante las fuerzas externas. Posteriormente se definen las funciones de optimización y restricciones a utilizar. Luego de esto se procede a determinar los casos de carga que se analizarán en los resultados. Una vez definidos estos pasos, se continua por la creación del algoritmo y su validación con funciones conocidas. Finalmente se analizan los casos de carga del caso en estudio. La figura 3.1 representa los pasos a seguir en la metodología de este trabajo.

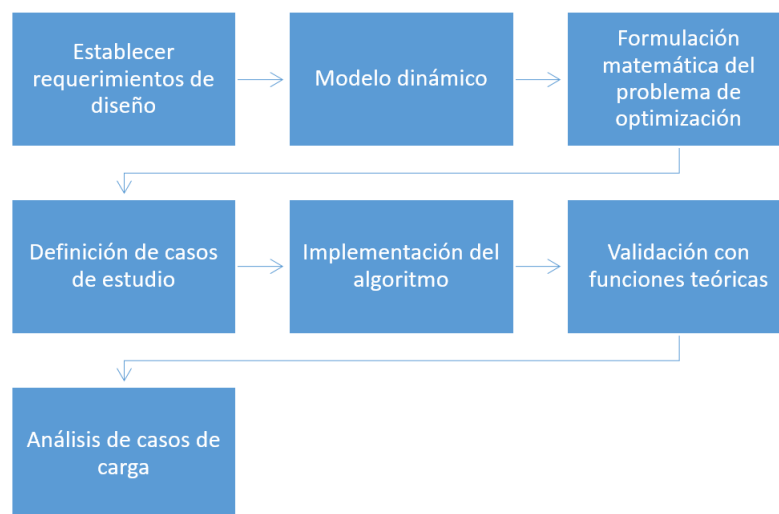


Figura 3.1: Flujo metodológico. (Fuente: el autor.)

3.2. Modelo dinámico

El primer paso para el proceso de optimización del resorte sujeto a carga variable, es la elaboración de un modelo dinámico que permita determinar las cargas que experimenta el resorte. De acuerdo con varios autores [9], [51], [29] se ha elaborado modelos simplificados de un grado de libertad para estudiar la dinámica en válvulas de control. Para el caso de esta investigación, se parte de esos modelos, y se modela el sistema como uno de varios grados de libertad para considerar un amplio dominio de frecuencias.

Ahora se procede a describir y presentar las formulaciones necesarias para el análisis dinámico de la válvula. Se parte del diagrama de la válvula mostrado en la figura 3.2 y se convierte en un sistema de ecuaciones diferenciales para obtener las fuerzas que actúan sobre el resorte y la posición del mismo. Cabe destacar que $L_{comprimida}$ es la longitud del resorte con una precompresión x_0 . La ecuación (3.1) se puede obtener al realizar análisis de cuerpo libre derivado de la figura 3.2. En esta ecuación X es el desplazamiento del extremo del resorte tomando en cuenta la precompresión, \dot{X} es la velocidad que tiene el

pistón y extremo del resorte y \ddot{X} es la aceleración del mismo. Además el término P_1 es la fuerza causada por la presión aguas arriba de la válvula y P_2 es la fuerza causada por la presión aguas abajo en las cuales se tomará la componente de la fuerza que contribuye al movimiento del pistón. El término c es el coeficiente de fricción del ensamble del pistón con el cilindro del mismo. Si se agrupan las fuerzas externas de las presiones en un término $F_{externa}$ se llega a la forma mostrada en la ecuación (3.2). Esta forma será la desarrollada tomando la sumatoria de las fuerzas externas que actúan sobre el pistón como una de las entradas al sistema.

$$(P_1 - P_2 \cos(\theta)) - kx - c\dot{x} = m\ddot{x} \quad (3.1)$$

$$F_{externa} = m\ddot{x} + c\dot{x} + Kx \quad (3.2)$$

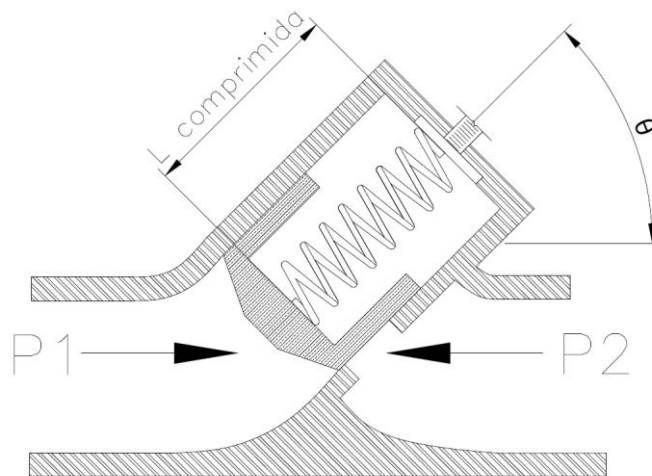


Figura 3.2: Imagen representativa de la configuración de la válvula. (Fuente: el autor.)

A partir de la ecuación (3.2) el sistema se pretende discretizar para realizar un análisis modal con múltiples grados de libertad. Esto con la finalidad de poder determinar los efectos de las frecuencias de la $F_{externa}$ en el sistema. Para esto, se separa el resorte a un sistema equivalente de masa amortiguador resorte con varios grados de libertad, tal como se muestra en la figura 3.3.

Esto se hace de esta manera debido a que en caso de que la primera frecuencia natural del sistema esté por debajo de la frecuencia de excitación no se capturaría la totalidad de los modos y podrían existir fuerzas internas mayores a la externa. De esta manera se incrementarán los grados de libertad hasta que la frecuencia natural máxima del sistema de masa amortiguador resorte sea mayor a la frecuencia de la fuerza de excitación externa.

Este método dará una cantidad de perfiles fuerza iguales a los grados de libertad utilizados, uno para cada segmento del resorte que representen los componentes de ese grado de libertad. Para el caso de optimización se escogerá el perfil de fuerzas que ocasione una menor resistencia a la fatiga, ya que ahí se encuentra el punto crítico de diseño del resorte. Para obtener esto, se quiere llegar a una función de la forma matricial presentada en la ecuación (3.3). En la ecuación M es la matriz de masa, C es la matriz de constantes de amortiguación, K la constantes de rigidez del resorte y F la de las fuerzas externas. Inicialmente se igualan estas a cero y se resuelve el sistema para encontrar las frecuencias naturales del como se presenta en la ecuación (3.4). Al resolver esta ecuación se tiene que los valores de las frecuencias naturales se pueden encontrar como $\omega = \sqrt{\lambda}$.

$$M\ddot{X} + C\dot{X} + KX = F \quad (3.3)$$

$$[K]\{a\} = \lambda[M]\{a\} \quad (3.4)$$

Entonces, en una primer instancia se divide el sistema tal que tenga una rigidez equivalente y una masa equivalente, de tal forma que la constante de rigidez de cada sección del resorte que fue discretizada cause que la totalidad del sistema mantenga la rigidez total para que los sistemas sean equivalentes. Para esto el sistema de la figura 3.3 de n grados de libertad se construye de tal forma que la sumatoria de los k_i forman una magnitud a la totalidad de la K del resorte y lo mismo se puede decir de los m_i que subdividen la masa total del resorte en puntos discretos. Las ecuaciones necesarias para estos conceptos se presentan en las ecuaciones (3.5) y (3.6). En donde K_{total} y M_{total} son la masa y constante del resorte completo mientras que K_i y M_i son las masas y constantes del resorte equivalente para cada sistema discreto.

$$\frac{1}{K_{total}} = \sum_{i=1}^n \frac{1}{K_i} \quad (3.5)$$

$$M_{total} = \sum_{i=1}^n M_i \quad (3.6)$$

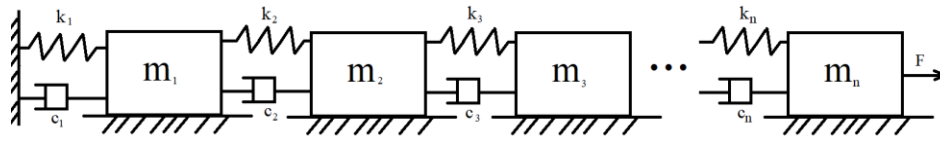


Figura 3.3: Imagen representativa del sistema de n grados de libertad (Fuente: el autor.)

El proceso iterativo de encontrar cuantos grados de libertad se requieren se inicia con un sistema discreto con 3 grados de libertad. Posteriormente, se calculan las frecuencias naturales de este sistema discreto y se comparan la frecuencia máxima natural del sistema contra la frecuencia relevante máxima de la carga externa. Si la frecuencia natural máxima es menor, se seguirán agregando grados de libertad al sistema hasta que se cumpla este criterio. Una vez determinado el número de grados de libertad del proceso se procede a desacoplar el sistema de ecuaciones diferenciales de tal manera que se puedan resolver como ecuaciones independientes y no simultáneamente.

Se procede a describir y desarrollar la metodología utilizada para calcular las frecuencias naturales del sistema generalizado para cualquier número de grados de libertad. Para comenzar, se separan los elementos de la figura 3.3 en diagramas de cuerpo libre. Se presentan 3 casos: el del primer elemento del sistema, el de un elemento del sistema i que se encuentra en el medio y el del último elemento n del sistema.

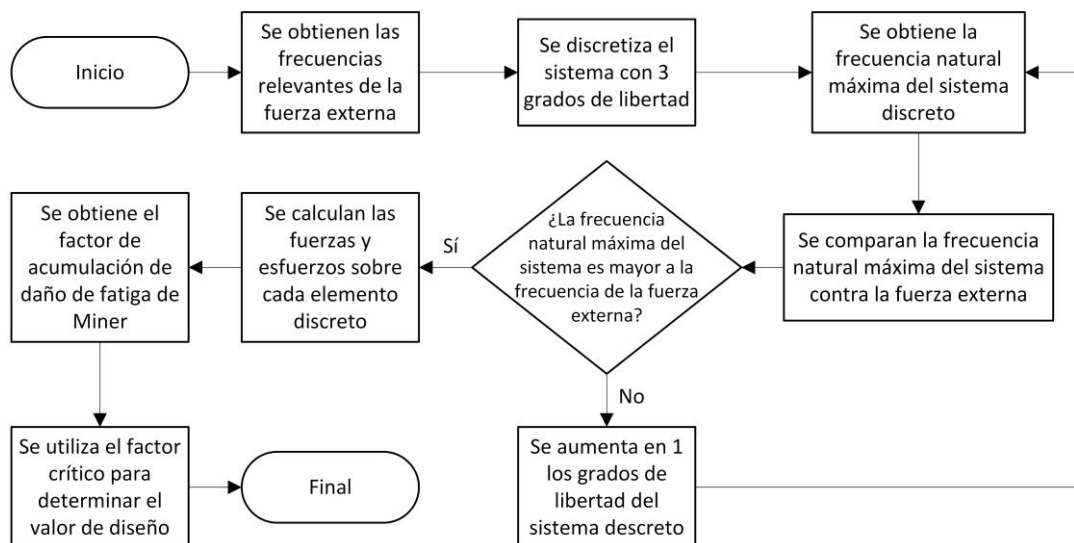


Figura 3.4: Diagrama representativo del flujo del programa de cálculo de fatiga. (Fuente: el autor.)

El diagrama de cuerpo libre en la figura 3.5 presenta el primer elemento discreto del sistema, la figura 3.6 presenta el diagrama de un elemento en el medio y la figura 3.7 presenta el último elemento discreto del sistema. Cabe destacar que en estos diagramas se utiliza la nomenclatura mencionada con anterioridad para los k_i y los m_i , mientras que c_i es el factor de amortiguación para el elemento discreto y \dot{x}_i^j es la velocidad del i ésimo elemento.

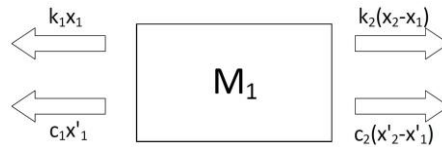


Figura 3.5: Diagrama de cuerpo libre del primer elemento. (Fuente: el autor.)

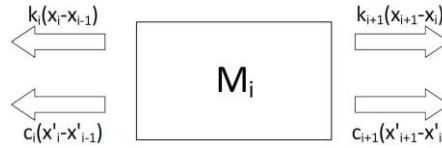


Figura 3.6: Diagrama de cuerpo libre de un elemento i en el medio. (Fuente: el autor.)

Los diagramas de cuerpo libre presentados permiten ensamblar el sistema de ecuaciones mostrado en la ecuación (3.7), el cual se encuentra generalizado para n grados de libertad. Para encontrar las frecuencias naturales del sistema se puede ignorar la fuerza externa y los factores de amortiguamiento debido a que estos no afectan la frecuencia natural. Al colocar el sistema de forma matricial se generan las matrices M y K mostradas en las ecuaciones (3.8) y (3.9).

La matriz M es una matriz diagonal que tiene los valores discretos de cada masa en cada valor de la diagonal, mientras que la matriz K tiene una forma diferente de calcular la primera y última fila. Sin embargo, el resto de las filas se pueden llenar con la diagonal con el valor $k_{i-1} + k_i$, en valor anterior a la diagonal con $-k_{i-1}$ y el posterior con $-k_i$. Es importante notar que para este caso los valores de k_i y m_i son iguales para todos los elementos por lo que las matrices presentadas se simplifican al tener valores de igual magnitud.

$$\begin{aligned}
 \square & 0 = -m_1 \ddot{x}_1 - c_1 \dot{x}_1 + c_2 (\dot{x}_2 - \dot{x}_1) + k_1 x_1 + k_2 (x_2 - x_1) \\
 \square & 0 = -m_i \ddot{x}_i - c_i (\dot{x}_i - \dot{x}_{i-1}) + c_{i+1} (\dot{x}_{i+1} - \dot{x}_i) \dots \\
 & \dots + k_1 x_1 + k_{i+1} (x_{i+1} - x_i) - k_i (x_i - x_{i-1}) \\
 \square & F_{externa} = \ddot{x} + k_n (x_n - x_{n-1}) + c_n (\dot{x}_n - \dot{x}_{n-1})
 \end{aligned} \tag{3.7}$$

$$M = \begin{matrix} \square & & & & \square \\ & m_1 & 0 & 0 & \dots \\ \square & 0 & m_2 & 0 & \dots \\ & \vdots & \vdots & \ddots & \vdots \\ & & & & 0 & 0 & \dots & m_n \end{matrix} \tag{3.8}$$

$$K = \begin{matrix} \square & & & & \square \\ \square & k_1 + k_2 & -k_2 & 0 & 0 & \dots \\ \square & -k_2 & k_2 + k_3 & -k_3 & 0 & \dots \\ & \vdots & \vdots & \vdots & \vdots & \dots \\ \square & \vdots & \vdots & \ddots & \vdots & \square \end{matrix} \tag{3.9}$$

$$\begin{array}{cccccc}
 \begin{array}{c} \square \\ \cup \end{array} & 0 & \dots & -k_{n-1} & k_{n-1} + k_n & -k_n \begin{array}{c} \square \\ \cup \end{array} \\
 & 0 & 0 & \dots & -k_n & k_n
 \end{array}$$

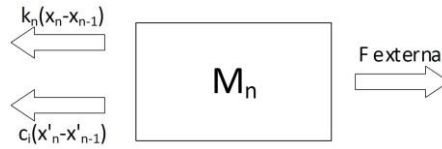


Figura 3.7: Diagrama de cuerpo libre del último elemento. (Fuente: el autor.)

Una vez identificadas las matrices de masa y rigidez a resolver, se utiliza el problema de autovalores y autovectores presentado en la ecuación (3.4). Al resolver este problema se obtienen los autovectores que representan la matriz modal y los autovalores que representan el cuadrado de las frecuencias naturales. Una vez obtenidas las frecuencias naturales se compara la frecuencia natural máxima contra las fuerzas externas.

Una vez identificados los grados de libertad requeridos se procede a calcular las fuerzas que existe en cada uno de los elementos discreto. Esto requiere resolver el sistema de ecuaciones diferenciales presentado en la ecuación (3.7). Para resolver este se desacoplará el sistema de ecuaciones de tal forma que se puedan resolver como ecuaciones diferenciales individuales.

El desacople del sistema de ecuaciones se logra diagonalizando las matrices de masa y rigidez. La manera de convertir estas matrices en matrices diagonales es multiplicándolas tal como se muestra en las ecuaciones (3.10) para la matriz de masa, (3.11) para la matriz de rigidez, (3.12) para la matriz de fuerza y (3.13) para la matriz de amortiguación para obtener las matrices desacopladas. En estas ecuaciones X y X^i es la matriz modal normalizada y su transpuesta. La matriz modal se obtiene del problema de autovectores según se explicó anteriormente y normalizándolo con base a la magnitud máxima de cada columna. El valor de Q es la matriz que reemplaza las fueras en el sistema desacoplado. La constante de amortiguación C es calculada según se presenta en la ecuación (3.13) en donde ζ es la constante de amortiguación con un valor de 0,05 para el caso de elemento metálico.

$$M_{Desac} = X^i M X \quad (3.10)$$

$$K_{Desac} = X^i K X \quad (3.11)$$

$$Q = X^i F_{ext} \quad (3.12)$$

$$C_{Desac} = 2\zeta M_{Desac} \omega_n \quad (3.13)$$

Mediante este desacople, se pueden resolver individualmente cada una de las filas de la ecuación de manera independiente según se presenta en la ecuación (3.14), al ser matrices diagonales. En esta ecuación los valores de q se pueden atribuir al desplazamiento o efecto de cada uno de los modos de vibración sobre el movimiento del sistema final y las matrices con el subíndice *Desac* son las matrices desacopladas. Una vez resueltas las ecuaciones independientes se puede devolver al espacio solución para obtener los desplazamientos y velocidades reales de cada elemento según se presenta en la ecuación (3.15) en donde los q_i

son los valores obtenidos de q para cada elemento, X_i son los valores de la i ésima columna de la matriz modal normalizada y X es la matriz con los desplazamientos.

$$M_{Desac}\ddot{q} + C_{Desac}\dot{q} + K_{Desac}q = Q \quad (3.14)$$

$$X = q_1X_1 + q_2X_2 + \dots + q_nX_n \quad (3.15)$$

Esta ecuación será utilizada para poder calcular los desplazamientos de los elementos para el modelo dinámico. Una vez calculados los desplazamientos se obtendrán las fuerzas que afectan cada uno de los mismos y, por ende, se podrán calcular los esfuerzos a los cuales están sometidos.

3.3. Optimización del resorte

Antes de comenzar con la implementación del algoritmo, se comenzará la sección metodológica por explicar y justificar las funciones objetivo que se buscarán optimizar. Estas funciones serán las que regirán los valores de ajuste con los cuales se indetificará la dominancia de un miembro de la población contra el otro según se explica en la sección de marco teórico para un algoritmo genético.

Cabe destacar que por la manera en la que se pretende realizar el algoritmo, todas las funciones se presentarán como problemas de minimizar un valor. En los casos en la que la función se quisiera maximizar, se le agregará un valor negativo al frente con la finalidad de convertirlo en un problema de minimización.

3.3.1. Funciones objetivo

Masa del resorte

La primera función objetivo que se va a definir va a asociada a la masa del resorte, por lo que se puede decir que al minimizar el volumen se estará minimizando la masa. La ecuación (3.16) presentada en la función f_1 define la primera función a minimizar que es la de la masa del resorte las variables de esta ecuación se definieron tanto en el marco teórico como en la sección de simbología 2.3.

Cabe destacar que la manera en la que se calcula la masa del resorte en la función (3.16) se presume un resorte de extremos planos. En esta función lo que se hace es calcular el área transversal del alambre del resorte y se multiplica por la longitud del mismo con un número de espiras igual a $n_a + 2$.

$$\min f_1(d, D, n_a) = \frac{\pi^2}{4} d^2 D (n_a + 2) \quad (3.16)$$

Resistencia contra fatiga

En el caso del diseño contra fatiga se utilizará el método de conteo de ciclos rain flow y el criterio de acumulación de daño por fatiga de Miner. Este método da como resultado la fracción de vida útil que fue gastada por uno de los ciclos de fatiga, según se detalló en la ecuación (2.10). Este resultado fue el utilizado para decidir la dominancia

en el sorteo de los elementos en el algoritmo. Al contrario del caso anterior este número se desea minimizar, por lo que no requiere de la introducción de un signo negativo al frente.

La función objetivo dos contra la fatiga no armónica busca minimizar la fracción de vida útil utilizada por los ciclos de fatiga, según se define con el método de Miner. Por lo que la función se convierte en la presentada en la ecuación (3.17) en la cual N_{Miner} es la fracción de vida útil utilizada basado en la acumulación de daño. Para el caso de los esfuerzos se evaluarán los esfuerzos calculados a partir del modelo dinámico y se utilizará el patrón que cause el factor de Miner crítico para la optimización.

$$\min f_2(d, D, n_a, F, S_u) = N_{Miner} \quad (3.17)$$

La forma de calcular el valor N_{Miner} para cada individuo comienza por obtener las fuerzas que lo afectan según se explicó en el modelo dinámico. Una vez obtenidos los patrones de fuerzas, se calculan los esfuerzos de Von Mises según se muestra en la ecuación (2.6). Cabe destacar que se obtiene una cantidad de patrones de esfuerzos igual a la cantidad de grados de libertad del modelo dinámico. A los esfuerzos obtenidos se les realiza el método de conteo de ciclos de rainflow según se describe en el marco teórico lo cual da como resultado la cantidad de ciclos y sus respectivas magnitudes de los esfuerzos. Una vez obtenido el conteo de ciclos, se calculan los N_i para cada uno de ellos utilizando la ecuación (2.8) para poder introducirlos en la ecuación (2.10). De esta manera se obtiene un factor de Miner para cada patrón de esfuerzos de cada individuo y finalmente se escoge el factor de Miner mayor de cada individuo de la población como su valor de idoneidad para la función objetivo de fatiga, ya que representa el caso crítico.

Diseño por robustez

El principio para obtener la tercera función objetivo estará basado en minimizar la sensibilidad del sistema a las variaciones incontrolables de las variables de diseño principales. Este utiliza la metodología Taguchi para el diseño robusto, para esto se pretende minimizar la variación de los resultados del sistema ante una posible variación de los diámetros del alambre, del resorte y el esfuerzo último del material. Para calcular el factor de sensibilidad del diseño robusto esto se ocupa realizar la derivada numérica del cambio en la variable de diseño contra el cambio en las variables de entrada. La idea de esto es disminuir el efecto de las variaciones no controlables escogiendo un punto de diseño tal que tenga un impacto mínimo en los resultados.

Para las funciones objetivo se definen las variables con variabilidad o ruido no controlable como d , D y S_u . Esto debido a que las tolerancias de manufactura en el diámetro del alambre y el resorte son variables de alta importancia con variaciones no controlables que introducen incertidumbre al sistema. Además el esfuerzo último es uno de los factores principales a la hora de determinar la resistencia del resorte. La finalidad de considerar esta incertidumbre en las variables es escoger un punto de diseño en el cual la variación del diámetro del resorte, del alambre y el esfuerzo último del material impacten lo menos posible en la masa del resorte y su resistencia a la fatiga.

Cabe destacar que, aunque la masa es derivable según se presentan en las ecuaciones

(3.18) y (3.19) se utilizará la definición numérica de la derivada y se calcularán de igual manera que se utilizará la fatiga. Esto con la finalidad de sumar los factores de sensibilidad según se explicará más adelante en un único valor a minimizar.

$$\frac{\partial f_1}{\partial d} = \frac{\pi^2 d D (n + 2)}{2} \quad (3.18)$$

$$\frac{\partial f_1}{\partial D} = \frac{\pi^2 d^2 (n + 2)}{4} \quad (3.19)$$

La metodología explicada para determinar la resistencia a la fatiga no es derivable analíticamente y, por ende, no se puede aplicar la definición analítica del factor de sensibilidad. Por esta razón se pretende utilizar el factor de sensibilidad de manera numérica. Para esto se definirán los deltas a utilizar en las variables de entrada y se evaluará el espacio solución en esos puntos. Posteriormente se determinará la magnitud del cambio en el espacio solución dividido entre la variación de la variable de entrada.

Para el caso en estudio las variaciones principales serán en cuanto a los posibles cambios de dimensión que tendría el diámetro del resorte y del alambre. Además de esto se tomará la resistencia última del material del resorte S_U como uno de los factores con variabilidad. Esto representará la variabilidad que existirá en los materiales y el proceso de manufactura [8]. La función objetivo 3 entonces, de manera generalizada, se convierte en minimizar el factor de sensibilidad del elemento en estudio y con base a esto se sortearán los pares genéticos más fuertes.

La función objetivo 3 se presenta en la ecuación (3.20) la cual además depende de las funciones objetivo 1 y 2 y sus variaciones. En esta función se tiene que f_1 es la función objetivo de la masa, f_2 es la función objetivo de la fatiga, b es el conjunto de variables de las que dependen estas funciones que se consideran sin incertidumbre y x aquellas variables que se considera su variabilidad.

En el caso de estudio se considerará que x estará formado por el diámetro del resorte D , el diámetro del alambre d y la resistencia última del material del resorte S_U . Las demás variables no se considerarán como variables significativas que generen ruido en la solución. Es decir, se enfocará el diseño a que sea robusto ante la variación existente en las dimensiones del resorte, el alambre y las características del material.

$$\min f_3(f_1, f_2, b, x) = \frac{\partial F(b, x)}{\partial x} = S \quad (3.20)$$

El factor de sensibilidad tomará en cuenta las variaciones en el diámetro del alambre del resorte y el diámetro exterior del resorte. La variabilidad asociada a estos será utilizando una distribución cuadrada y asumiendo un cambio de cada una de esas variables de un $\pm 20\%$. El ámbito del cambio en las variables de entrada asignado fue escogido según las tolerancias de manufactura comúnmente utilizadas en resortes [52].

Una vez definido esto, el factor presentado en la ecuación (3.20) al realizarlo numéricamente se convierte en minimizar la suma de los cuadrados de los factores de sensibilidad numéricos. Estos están definidos en la ecuación presentada en la función (3.33). En esta se tiene que los S_i son los factores de sensibilidad, f_1 es la función objetivo de la masa, mientras que f_2 es la función objetivo de la fatiga. Además b se define como el conjunto

de variables de las cuales dependen las funciones que se considera que no tienen incertidumbre. Los factores de sensibilidad S_i se definen en la ecuación (3.22). En esta ecuación se define el cambio porcentual de los resultados de masa según la variación de d y D . Además, se define el cambio porcentual de los resultados de la resistencia a la fatiga según la variación de d , D y S_u .

$$\min f_3 = \sum_{i=5}^4 S_i \quad (3.21)$$

$$\begin{aligned} S_1 &= \frac{f_1(d, D, b) - f_1(d + \Delta d, D, b)}{f_1(d, D, b)} \frac{d}{\Delta d} \\ S_2 &= \frac{f_1(d, D, b, S_u) - f_1(d, D + \Delta D, b)}{f_1(d, D, b)} \frac{D}{\Delta D} \\ S_3 &= \frac{f_2(d, D, b, S_u) - f_2(d + \Delta d, D, S_u, b)}{f_2(d, D, S_u, b)} \frac{d}{\Delta d} \\ S_4 &= \frac{f_2(d, D, S_u, b) - f_2(d, D + \Delta D, S_u, b)}{f_2(d, D, b)} \frac{D}{\Delta D} \\ S_5 &= \frac{f_2(d, D, S_u, b) - f_2(d, D, S_u + \Delta S_u, b)}{f_2(d, D, b)} \frac{S_u}{\Delta S_u} \end{aligned} \quad (3.22)$$

En la ecuación (3.22) se tiene que los factores Δd , ΔD y ΔS_u corresponden a las incertidumbres asociadas a los parámetros de diseño y resistencia del material. Estos valores son determinados por el usuario, y para los casos en los que se va a evaluar este algoritmo van a ser tomados con base en las incertidumbres obtenidas usualmente en la industria.

3.4. Restricciones

Las restricciones del modelo serán basadas en restricciones dimensionales del elemento mecánico y fuerzas a las que estará sometido. Entre las restricciones principales y valores a verificar se tienen la compresión máxima del resorte, la resistencia al pandeo del mismo, la resistencia a la fluencia y la longitud del resorte. Cabe destacar que aunque estos no

son parámetros a optimizar, se deben verificar con la finalidad de asegurar la correcta operación y seguridad del resorte.

Además de estos factores, se tienen los valores variables que se delimitarán, que son las entradas a evaluar en el espacio solución. Los valores variables principales que se tienen serán parámetros dimensionales del resorte, entre estos definimos: la longitud libre del resorte, el diámetro máximo del resorte, diámetro del alambre del resorte y el número de espiras activas. Los ámbitos de estas variables serán determinadas como un patrón de entrada, los casos que se correrán como ejemplo en esta investigación se utilizarán valores de referencia y diseños existentes con la finalidad de validar el programa.

Las restricciones del algoritmo se presentan de manera numérica en la ecuación (3.23). La ecuación presenta la definición de las restricciones asociadas al índice del resorte, al pandeo y la constante del resorte. Además se incluye una restricción de que los elementos de interés son aquellos con un factor de acumulación de daño menor a uno, ya que el componente falla contra fatiga para valores mayores a este. Finalmente se toma una restricción para evaluar que la longitud libre del resorte esté dentro de un ámbito determinado. Esto se verifica confirmando que al utilizar un ángulo recomendado de las espiras del resorte entre $6,35^\circ$ y 9° [53] se pueda obtener la longitud deseada. La longitud se calcula a partir del ángulo del resorte θ , calculando el paso del resorte utilizando la ecuación (3.24) y luego calculando la longitud total por medio de la ecuación (3.25).

$$\begin{aligned}
 \text{Restricciones del algoritmo} = & \begin{cases} 4 \leq C = \frac{D}{\alpha} \leq 12 \\ L_{\text{libre}} < \frac{\pi D}{\alpha} \left(\frac{2(E-G)}{2G+E} \right)^{1/2} \end{cases} \quad (3.23)
 \end{aligned}$$

$$\begin{cases} N_{\text{Miner}} < 1 \\ L_{\text{Min}} < L_{\text{libre}} < L_{\text{Max}} \\ K_{\text{Min}} < K < K_{\text{Max}} \end{cases}$$

$$\text{Paso} = 2D \cos(\theta) \quad (3.24)$$

$$L_{\text{libre}} = n_a \text{Paso} \quad (3.25)$$

Un factor importante a destacar de las restricciones de la constante del resorte y la longitud libre es que sus valores mínimo y máximo están delimitados por las otras variables como el diámetro del alambre, del resorte y el número de espiras activas. Los límites permitidos absolutos para la constante del resorte estarán dados por los que se presentan en la ecuación (3.26). Por otro lado, los límites permitidos para la longitud libre se presentan en la ecuación (3.27) y están basados en los ángulos del resorte permitidos.

$$\frac{Gd_{\text{Min}}^4}{8D_{\text{Max}}^3 n_{a \text{ Max}}} \leq K \leq \frac{Gd_{\text{Max}}^4}{8D_{\text{Min}}^3 n_{a \text{ Min}}} \quad (3.26)$$

$$1,987D_{\text{Min}}n_{a \text{ Min}} < L_{\text{libre}} < 1,975D_{\text{Max}}n_{a \text{ Max}} \quad (3.27)$$

Estas restricciones se definieron con la finalidad de asegurar el adecuado funcionamiento del resorte y de poder tener control sobre parámetros de interés en el diseño. El algoritmo funciona de tal forma que si una de estas restricciones no se cumple se le asignará un valor de idoneidad arbitrariamente alto al individuo de la población. De esta manera se asegura de que este no sea utilizado como uno de los individuos óptimos de la generación.

3.5. Casos de carga

El método numérico será sometido a varios casos de carga con la finalidad de verificar su funcionamiento. Para los casos de carga se pretende tomar en cuenta tres patrones de fuerza. Como primera instancia se considerará una una fuerza simétrica y armónica con un patrón sinusoidal debido a la simplicidad del caso. Luego se estudiará un caso práctico de una onda de presión a la que se encuentra sometido un resorte de una válvula de control. Como referencia se toman los ejemplos presentados en la investigación de Jian et al [29], como el que se presenta en la figura 3.8. Finalmente, se pretende realizar el análisis del caso que posee una mayor inestabilidad, que es aquél en el cual las fuerzas a las que se somete el elemento mecánico varían aleatoriamente.

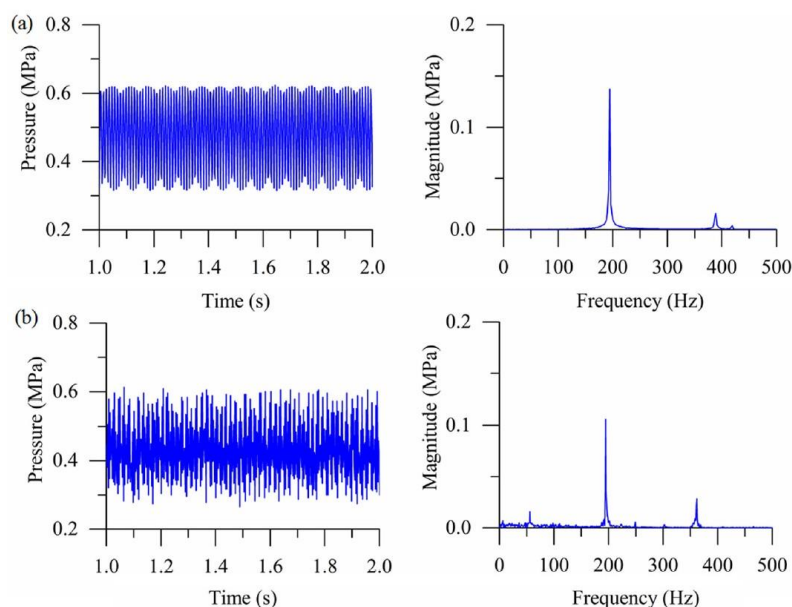


Figura 3.8: Ondas de presión a utilizar para los casos de carga. (Fuente: [29].)

Carga sinusoidal

El primer caso de estudio a realizar es el de un resorte sometido a carga sinusoidal simétrica y armónica. El patrón para la fuerza externa en este carga se define por la ecuación (3.28), en la cual N es la amplitud de la onda en N y ω su frecuencia en rad/s . Se utiliza como un caso de referencia simplificado de un caso real. El caso en específico utiliza una amplitud de 50 N y una frecuencia de 20 Hz, estas magnitudes se determinaron basado en referencias de ondas de presión en tuberías [51] y tomando en cuenta las dimensiones de las dimensiones del resorte en estudio. Al graficar esta carga sinusoidal se tiene como resultado el patrón de la fuerza mostrado en la figura 3.9. El patrón mostrado se considerará como un ciclo de fatiga para el factor de acumulación de daño.

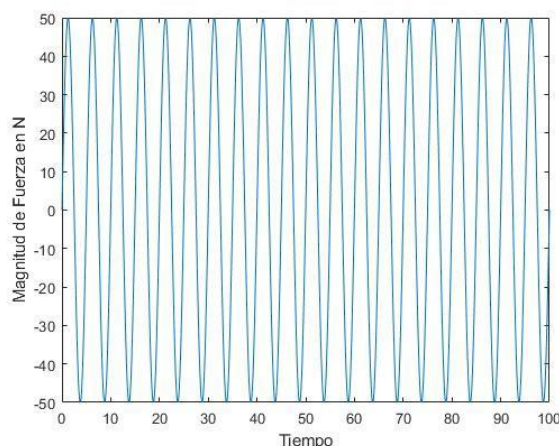


Figura 3.9: Patrón de fuerza sinusoidal. (Fuente: el autor).

La carga sinusoidal que se utilizará será una de la forma presentada en la ecuación (3.28), en la cual N es la amplitud de la onda en N y ω su frecuencia en rad/s . Para el caso de estudio se utilizó una amplitud de 50 N y una frecuencia de 1256 rad/s . Además de esto, los ámbitos de valores y demás variables utilizadas para este caso de estudio se presentan en el cuadro 3.1. La forma de la onda utilizada se presenta en la figura

$$F(t) = Nsen(\omega t) \quad (3.28)$$

Carga aleatoria

Otro de los casos de estudio a realizar es el de someter el resorte a una carga aleatoria de entrada. Para esto se generó una carga aleatoria con valores posibles en el ámbito de los ± 120 N con una distribución rectangular de probabilidad, la magnitud de esta fuerza fue basada en la amplitud de la onda presentada por [29] con el cambio de que se alteró a una distribución aleatoria. El resultado de la fuerza generada la cual se tomará en cuenta para este caso de estudio se presenta en la figura 3.10. Este patrón de fuerza para el caso de fatiga es el que se tomará en cuenta como un ciclo para los cálculos del factor de acumulación de daño según se explica en la sección de metodología. Las variables utilizadas para este caso, al igual que anterior, se muestran en el cuadro 3.1

Carga semejando una onda de presión

Mediante este caso de carga se pretende aproximar lo más posible el diseño del resorte a las fuerzas existentes en una línea de presión. Para esto se define una fuerza externa basada en los patrones de fuerza mostrada en la figura 3.8 utilizando las magnitudes y frecuencias para convertir la onda del espacio de la frecuencia a una onda equivalente en el espacio temporal. Esto se realiza aplicando la formulación matemática mostrada en la ecuación (3.29), donde N_i es la amplitud de la frecuencia y ω_i representada en la figura 3.8. Para este caso en específico se toma un valor de N_0 de 30 N, N_1 de 20 N y N_2 de 3 N. Además el caso se utilizó con un ω_1 de 200 Hz y ω_2 360 Hz basado en la onda de presión caracterizada la publicación de Jian [29]. Al sustituir estos valores convierte se obtiene la forma mostrada en la ecuación (3.30)

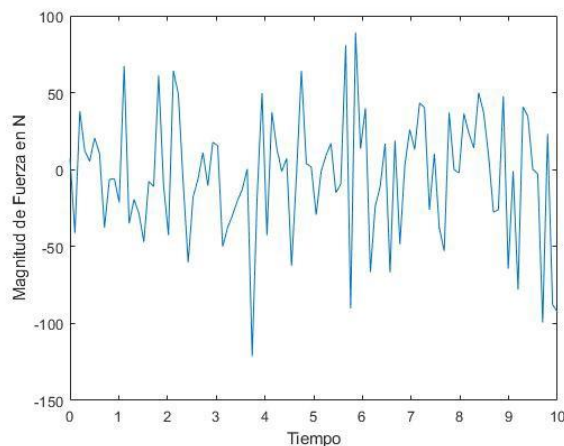


Figura 3.10: Gráfica de fuerza aleatoria utilizada para el caso de estudio. (Fuente: el autor).

$$F(t) = N_0 + N_1 \text{seno}(\omega_1 t) + N_2 \text{seno}(\omega_2 t) + \dots + N_i \text{seno}(\omega_i t) \quad (3.29)$$

$$F_{\text{externa}}(t) = 30 \text{ N} + 20 \text{ N} \text{seno}(1257t) + 3 \text{ N} \text{seno}(2262t) \quad (3.30)$$

La fuerza resultante usando el patrón de frecuencias definido se muestra en la figura 3.11. Este patrón de fuerza se considerará como uno de los ciclos de fatiga en cuanto el factor de acumulación de daño de Miner se refiere. La figura 3.12 muestra un periodo de tiempo más corto para poder apreciar con mayor claridad los aportes de los diferentes armónicos. Los demás valores usados para las variables de diseño y del programa para este caso de estudio se definen en el cuadro 3.1. Los rangos de las variables d , D y n_a se escogieron de tal forma que estuvieran cercanos a los utilizados en válvulas de control existentes [5], sin embargo estos rangos pueden ser cambiados con facilidad en el algoritmo en caso de que se requiera modificar el caso de estudio.

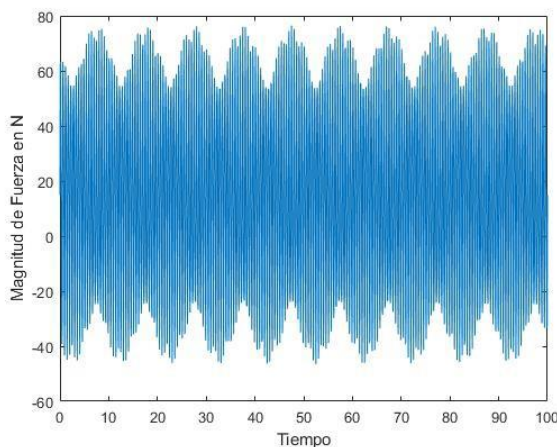


Figura 3.11: Patrón de fuerza simulando una onda de presión. (Fuente: el autor).

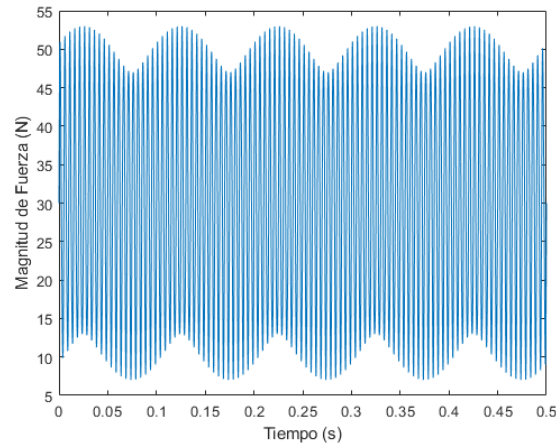


Figura 3.12: Detalle de patrón de fuerza simulando una onda de presión. (Fuente: el autor).

Cuadro 3.1: Valores de las variables utilizadas para los casos de carga. (Fuente: el autor).

Variables	Valores utilizados	Unidades
d	4 a 8	mm
D	4 a 12	cm
n_a	6 a 11	N/A
ΔD	10 %	N/A
Δd	10 %	N/A
ΔS_u	5 %	N/A
Tamaño de población	70	N/A
Número de generaciones	100	N/A
Probabilidad de mutación	10 %	N/A
Probabilidad de cruce	90 %	N/A

3.6. Implementación del algoritmo

Como se ha mencionado con anterioridad, el método numérico a utilizar para lograr la optimización del diseño del resorte es el algoritmo genético de sorteo no dominante. Basado en los pasos del algoritmo explicados en la sección del marco teórico, para este caso en específico se propone el flujo general presentado en la figura 3.13.

En el flujo del programa se tiene una generación de la población uno, la cual es validada para después ser evaluada en las funciones objetivo. Posterior a esto se crea una generación de hijos con la finalidad de entrar al ciclo iterativo. El ciclo iterativo del algoritmo pretende preservar tanto la élite de las poblaciones mientras se mantiene la diversidad. De manera general, el algoritmo utilizará los siguientes pasos, los cuales tendrán una explicación más detallada a lo largo de este segmento del trabajo [54].

1. Inicialización de la población: se inicia la población basado en los ámbitos de las variables del problema y las restricciones
2. Ordenamiento no-dominante: se realiza el acomodo de la población basada en el sorteo no dominante. El ordenamiento será explicado más adelante en este segmento

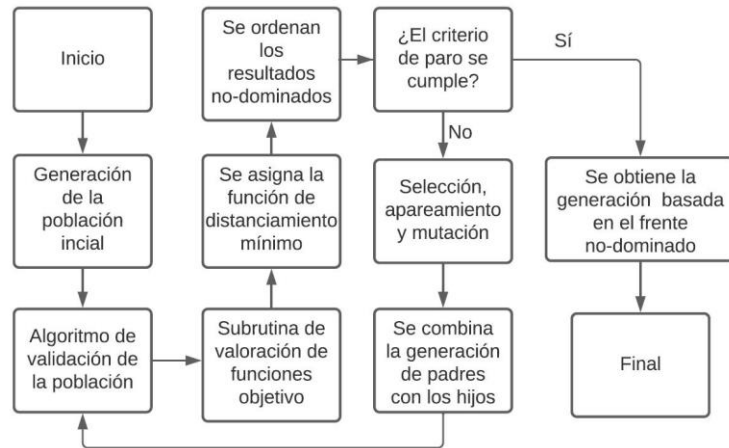


Figura 3.13: Diagrama de flujo del proceso total. (Fuente: el autor)

del trabajo.

3. Distanciamiento mínimo de aglomeración de población: una vez que la población es ordenada, se le asigna un valor de aglomeración a cada valor. Esto lo que hace es encontrar la distancia euclidiana entre cada punto individual en cada frente basado en los objetivos. Siempre se seleccionan los individuos a los extremos de los frentes.
4. Selección de candidatos: La selección se hace basada en el operador de selector de agrupamiento en un torneo binario, la cual se basa en el ámbito de no dominancia de cada individuo y en la distancia de agrupamiento [55].
5. Operadores genéticos: el algoritmo genético utilizará cruce binario simulado (SBX) y mutación polinomial [56]. El cruce binario simula el fenómeno observado en la naturaleza para la creación de *hijos* a partir de padres. Además de esto existe una posibilidad que se realice una mutación polinomial, lo cuál se detallará más adelante en la metodología.

3.7. Subrutinas del programa

3.7.1. Creación de la población inicial

La primera subrutina a utilizar es la de generación de la población inicial. La subrutina recibe como entrada el tamaño de la población que se desea generar, el número de variables de decisión, el ámbito mínimo y el ámbito máximo de cada variable como vectores. La subrutina genera un vector con valores aleatorios dentro del ámbito de la función, esto se hace utilizando una metodología similar a la mostrada en la ecuación (3.31). En esta $min(j)$ es el ámbito mínimo de la variable j -ésima y $max(j)$ es el ámbito máxima de la variable j -ésima. De esta manera se genera la población inicial aleatoria para la cantidad de variables j . La expresión $rand(1)$ genera un número aleatorio entre 0 y 1 con una distribución de probabilidad uniformemente distribuida.

$$f(i,j) = \min(j) + (\max(j) - \min(j)) * \text{rand}(1) \quad (3.31)$$

3.7.2. Validación de la población

La subrutina de validación de la población es el siguiente paso a seguir en el algoritmo. Esta rutina tiene como función evaluar si los individuos de la población cumplen con las restricciones del algoritmo. La subrutina tiene como resultado un uno de no cumplirse alguna de las restricciones y un cero de cumplirse todas. Como se muestra en la figura 3.14, la subrutina evalúa cada individuo e identifica aquellos que se encuentran incumpliendo con alguna las restricciones definidas con anterioridad en la ecuación (3.23).

Después de la subrutina se utiliza el método de penalización estática para modificar los valores de ajuste calculadas de cada individuo funciones según se muestra en la ecuación (3.32). En esta ecuación (3.32) f_j es la función objetivo sin penalizar, f_i es la función con el factor de penalización y C es una constante arbitrariamente grande en este caso se utilizó un valor de 100 000. En la ecuación se define P como el resultado de la rutina de validación en la cual tiene un valor de uno si el individuo de la población no cumple con las restricciones y cero si cumple con todas las restricciones. Este término se utiliza para penalizar el valor de un individuo que no cumpla con las restricciones para que no pertenezca a la frontera pareto óptima al modificar sus valores de ajuste manualmente.

$$f_i = f_j + CP \quad (3.32)$$

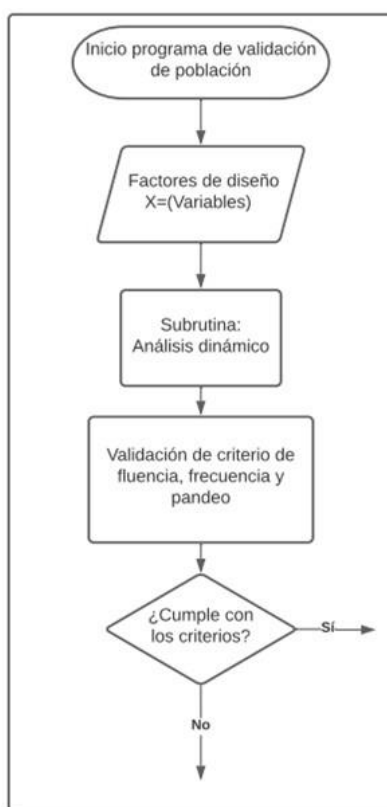


Figura 3.14: Flujograma del programa de validación de la población. (Fuente: el autor.)

3.7.3. Evaluación de las funciones objetivo

La subrutina de evaluación de funciones objetivo, como su nombre lo indica, contiene los pasos necesarios para lograr evaluar cada uno de los individuos de la población en todas las funciones objetivo. En el caso de este trabajo son tres principales según se discutió con anterioridad: la masa, la resistencia a la fatiga y la robustez del diseño. Esta subrutina tiene una serie de pasos que se pueden resumir con el diagrama de flujo mostrado en la figura 3.15. La figura muestra como con base en los valores de las variables de cada objetivo se calculan las magnitudes de interés a optimizar y obtiene para cada individuo el valor que tiene para cada una de las funciones objetivo.

La subrutina inicia calculando los valores de ajuste de los individuos para la primera función objetivo. Esto se hace al substituir los valores específicos de cada individuo en la ecuación (3.16) que se definió con anterioridad. Este valor de masa será el utilizado al sortear la dominancia de los valores con base en el primer objetivo.

La evaluación de la segunda función objetivo se realizó siguiendo el flujo presentado en la figura 3.4. El primer paso del flujo es encontrar las frecuencias relevantes de la fuerza de excitación externa, esto se realiza convirtiendo la fuerza externa del espacio del tiempo al de la frecuencia con una transformación de Fourier. De la transformación se obtienen cuales de las frecuencias son las que tienen amplitudes mayores, lo cual se realiza normalizando las amplitudes de las frecuencias y seleccionando aquellas que tengan una contribución igual o mayor al 20% de la amplitud máxima encontrada.

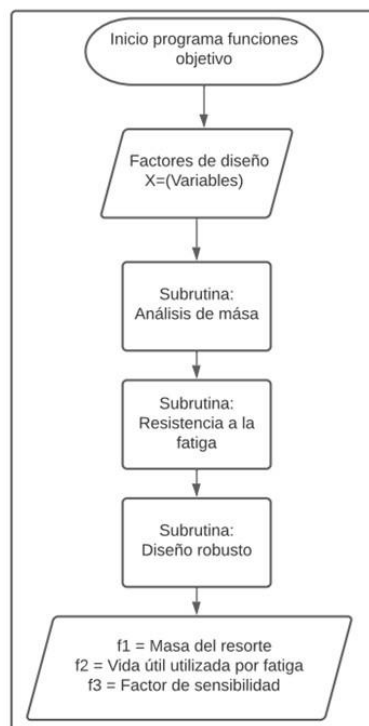


Figura 3.15: Diagrama de la subrutina de evaluación de funciones objetivo. (Fuente: el autor.)

Una vez obtenido los desplazamientos y velocidades se substituyen en el sistema de ecuaciones presentado en la ecuación (3.7) para obtener las fuerzas internas de cada elemento.

Posteriormente, se calcula el factor de acumulación de daño de Miner y se utiliza el factor crítico (el mayor) para el resorte.

El cálculo de fatiga se realiza para cada individuo de la población según se describió anteriormente. Luego, con base en los valores calculados de factores de acumulación de daño se asigna su valor de idoneidad para la función objetivo dos según se determinó en la sección de definición de las funciones objetivo. Finalmente para esta subrutina se evalúa la tercera función objetivo, la de la robustez del diseño. Para esto se asigna un porcentaje de variación para las variables d y D . En este caso se dice que la variación no controlable con una distribución rectangular con $\pm 15\%$ del valor original basado en las incertidumbres de manufactura de los elementos, sin embargo este valor se podría ajustar según sea necesario a otros casos específicos.

Sustituyendo esta variación en la ecuación (3.33) se obtienen los factores de sensibilidad mostrados en la ecuación (3.34). Estas ecuaciones son las utilizadas en la subrutina de evaluación de funciones objetivo y el valor calculado de f_3 para cada individuo de la población es el asignado como el valor de idoneidad en la tercera función objetivo.

$$f_3 = \sqrt{S_1^2 + S_2^2 + S_3^2 + S_4^2} \quad (3.33)$$

$$S_1 = \frac{f_1(d) - f_1(1,3d)}{f_1(d)} \cdot 0,3$$

$$S_2 = \frac{f_1(D) - f_1(1,3D)}{f_1(D)} \cdot 0,3$$

Factores $S_i =$

$$S_3 = \frac{f_2(d) - f_2(1,3d)}{f_2(d)} \cdot 0,3$$

$$S_4 = \frac{f_2(D) - f_2(1,3D)}{f_2(D)} \cdot 0,3$$

$$(3.34)$$

De esta manera esta subrutina proporciona los valores de ajuste de las tres funciones objetivo para cada uno de los individuos de la población. Esta información se utilizará más adelante para ordenar los individuos en los diferentes frentes dominantes y escoger cuales serán aquellos que tendrán hijos para la siguiente generación.

3.7.4. Ordenamiento no dominante y distanciamiento mínimo

El objetivo de este subprograma es ordenar los valores de los individuos de la población según sus valores de ajuste según su dominancia. Para esto, se utiliza el algoritmo de sorteo rápido [39] que identifica la cantidad de individuos de la población que domina a un individuo y la cantidad que son dominados por el mismo. Esto se realiza según se describe a continuación para cada individuo p de la población P :

1. Se inicializa el conjunto S_p que contendrá todos los individuos dominados por p .
2. Se inicializa la variable n_p que contendrá el conteo de todos los individuos que dominan p .
3. Para cada individuo q contenido en P pero diferente de p

- Si p domina a q , se incluye a q en el conjunto S_p
 - Si q domina a p , se aumenta la variable $n_p = n_p + 1$
4. Entonces si $n_p=0$, no existen individuos que dominen p y se le asigna a p un ámbito de 1. Además se incluye p en el conjunto de individuos no dominados en el primer frente.
 5. Este procedimiento se lleva a cabo para todos los individuos

Posteriormente, se realiza un procedimiento con S_p de cada individuo para determinar a que frente pertenecen el resto no están en el primer frente F_1 . Esta información será de utilidad ya que al seleccionar los individuos de la siguiente generación si el tamaño de la población es mayor al primer frente se tomarán individuos de los siguientes frentes dominantes. Para determinar el resto de los frentes se realiza el procedimiento detallado a continuación para cada individuo mientras el i -ésimo frente F_i no esté vacío.

1. Se inicializa la variable Q como conjunto vacío, esta almacenará los individuos del frente $i + 1$
2. Para cada individuo p del frente F_i se le asigna un contador n_q y se le resta uno al contador de q por cada vez que un individuo se repita en el subconjunto S_p de los diferentes individuos del frente
3. Si $n_q = 0$ implica que no hay otros individuos, además de los del frente F_i que lo dominan, por lo que se asignan esos individuos al conjunto Q
4. Finalmente se incrementa el contador de F_i y se le asigna como Q , para reiniciar el proceso con el siguiente frente.

Posteriormente a esto a cada individuo se le asigna una magnitud de distanciamiento con la finalidad de mantener la diversidad en los resultados obtenidos. La distancia se asigna por cada frente y se utiliza para comparar las distancias dentro de los individuos de cada frente, sin embargo estas carecen de significado al compararlas con individuos de otros frentes.

El valor de distanciamiento describe la distancia euclidiana entre cada individuo del frente basado en los valores de sus objetivos en un espacio en el cual los ejes ortogonales son la cantidad de objetivos. Al asignar este valor los individuos de los extremos siempre son seleccionados y se le asigna una magnitud de distanciamiento infinita.

La magnitud de distanciamiento se calculó según se describe en la ecuación (3.35)

en la cual inicialmente se ordenan los individuos de la población de mayor a menor con base en los valores de la función objetivo f_k . Esto da un ordenamiento llamado $H(p_i)$ siendo $i = 1$ el individuo con menor valor de la función objetivo dentro del frente e $i = n$ siendo el mayor valor. En la ecuación (3.35) $f_k(H(p_i))$ es el valor de idoneidad del individuo p_i para la k -ésima función objetivo. Esta función se aplica del segundo individuo del ordenamiento $H(p_i)$ hasta el individuo $n - 1$, ya que el primer y último individuo tienen valor de distanciamiento infinito y siempre son seleccionados al competir contra un individuo de su mismo frente.

$$Dist(p_i) = H(p_i) + \frac{f_k(H(p_i)) - f_{k+1}(H(p_i - 1))}{f_{kmax} - f_{kmin}} \quad (3.35)$$

3.7.5. Selección, apareamiento y mutación

La selección en este caso se usa utilizando un torneo binario, según se definió en el marco teórico. Lo que se hace es seleccionar dos individuos aleatoriamente de la población y compararlos entre sí. El ganador será seleccionado para continuar a la siguiente generación de la población y tendrá la posibilidad de una mutación o crear hijos con otro individuo. Un individuo p perteneciente al frente i y con un valor de distanciamiento d_i se selecciona sobre otro individuo q si cumple con las siguientes condiciones, definiendo p y q como individuos de la población P :

- En el caso de que el frente i de p sea un número menor al frente al que pertenece q , p es seleccionado.
- En caso de que p y q pertenezcan al mismo frente se selecciona el individuo que tenga la mayor magnitud de distanciamiento

Una vez seleccionados los individuos ganadores del torneo se tienen dos posibilidades para ellos, la posibilidad de que tengan algún descendiente mediante una recombinación con otro individuo de la población o la posibilidad de que se cree un nuevo individuo basado en una mutación del individuo original. La probabilidad escogida para la recombinación es de un 90 % y el otro 10 % es la probabilidad de que el individuo sufra una mutación.

Para la recombinación se utiliza el cruce binario simulado según se presenta en la ecuación (3.36) en donde $c_{i,k}$ es iésimo hijo de la población con el componente késimo. Cabe destacar que bajo esta metodología dos individuos padres generarán dos individuos hijos. En esta β_k es un número generado aleatoriamente por la densidad probabilística presentada en la ecuación (3.37) [55] [57].

Para este caso se puede obtener la distribución de β requerida tomando un valor u aleatorio entre 0 y 1 y evaluando la ecuación (3.38), y η_c es un factor utilizado el cual determinará que tan cerca o lejos estarán los hijos de los padres originales y se le conoce como el índice de recombinación. Para el algoritmo utilizado se utiliza un índice de recombinación de 0,2 basado en las pruebas realizadas y recomendaciones de la bibliografía consultada [55] [58].

$$c_{1,k} = \frac{1}{2} [(1 + \beta_k)p_{1,k} + (1 - \beta_k)p_{2,k}] \quad (3.36)$$

$$c_{2,k} = \frac{1}{2} [(1 + \beta_k)p_{1,k} + (1 - \beta_k)p_{2,k}]$$

$$p(\beta) = \begin{cases} \frac{1}{2} (\eta_c + 1) \beta & \text{Si } 0 \leq \beta \leq 1 \\ \frac{1}{2} (\eta_c + 1) \beta^{\eta_c} & \text{Si } \beta > 1 \end{cases} \quad (3.37)$$

$$SBX : \beta(u) = \begin{cases} (2u)^{1/(\eta+1)} & \text{Si } u \leq 1/2 \\ (2(1-u))^{-1/(\eta+1)} & \text{Si } u > 1/2 \end{cases} \quad (3.38)$$

Con la finalidad de brindar al lector mayor claridad de este proceso la figura 3.16

presenta el diagrama de flujo para calcular los cromosomas de los hijos utilizando el cruce binario simulado. El procedimiento de calcular los cromosomas de los hijos se repite hasta alcanzar el número d cromosomas de los padres, de tal forma que los nuevos individuos

Cuadro 3.2: Valores de las variables utilizadas para correr el algoritmo. (Fuente: el autor.)

Variable	Valor utilizado
Criterio de paro	150 generaciones
Tamaño de población	100 individuos
Tasa de cruce	90 %
Tasa de mutación	10 %
Constante η de mutación polinomial y cruce binario	20

de la población quedan completamente definidos. En el caso en estudio, al tener tres cromosomas por individuo este proceso se repite tres veces.

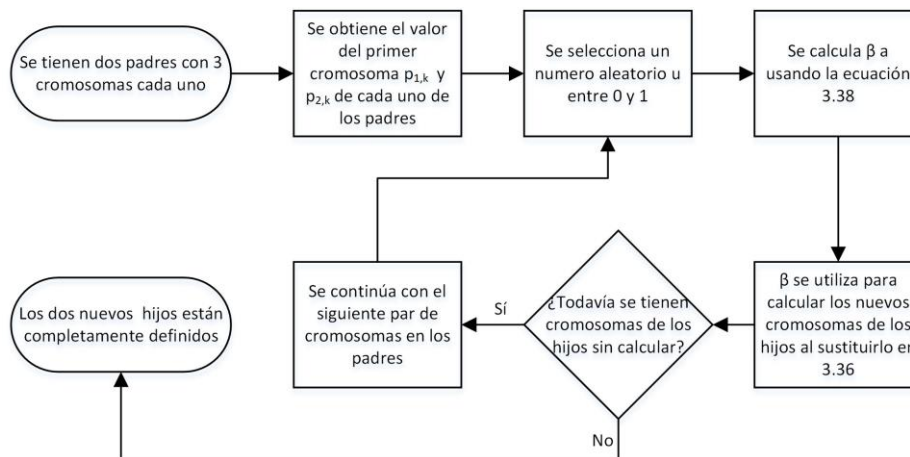


Figura 3.16: Diagrama de flujo para calcular los nuevos cromosomas (Fuente: el autor).

3.8. Parámetros del algoritmo

El criterio de paro del algoritmo se realiza con un número de generaciones a completar. Una vez completado el número de generaciones indicadas con el tamaño de población escogido el algoritmo se detendrá. Los resultados principales obtenidos de una corrida son los individuos pertenecientes a la frontera pareto-óptima que son los del primer frente. Además de esto, una vez terminado se tiene el valor de los cromosomas o variables de cada uno de estos individuos.

Esta información no solo permite ver el comportamiento entre las funciones objetivos, sino que también provee que variables de diseño utilizar para el diámetro del alambre del resorte y del número de espiras para que individuo de la frontera. De esta manera se sabe que al escoger uno de estas opciones de combinaciones de variables se estará escogiendo una de las combinaciones óptimas no dominadas de la población.

Los parámetros utilizados para los casos de estudio se presentan en el cuadro 3.2. Estos se definieron según las pruebas realizadas para validar el algoritmo en anexo A.1, así como referencias bibliográficas [39].

3.9. Comentarios finales

Durante esta sección del documento se describió los grupos conceptuales principales que componen entre ellos. Se establecieron los requerimientos de diseño para el resorte, tanto a nivel geométrico como materiales, restricciones y fuerzas externas. Además se propuso un modelo dinámico y se detalló la manera de calcular los grados de libertad y desacoplar las ecuaciones para resolver el sistema. Se definieron las formulaciones matemáticas del problema de optimización que se utilizarán para describir el problema de optimización así como su procedencia. Otro componente importante que se definió a lo largo de este capítulo metodológico son los parámetros utilizados para la implementación del algoritmo con su respectiva justificación de como fueron elegidos. Finalmente se definieron los casos de estudio que se analizarán en las siguientes secciones del trabajo y las fuerzas que estos utilizan. El código fuente que se creó y se utilizará a lo largo de este trabajo y se desarrolló a partir de este procedimiento metodológico se presenta en el Anexo A.4.

Capítulo 4

Resultados y discusión

Durante este capítulo del trabajo se presentarán y discutirán los resultados obtenidos para el algoritmo implementado. Primero se presentará la evaluación del rendimiento del algoritmo al compararlo con funciones teóricas tanto en los resultados obtenidos como en las gráficas de las fronteras pareto-óptimas. Posteriormente se realizarán corridas con las funciones objetivo de interés y se analizarán los resultados obtenidos.

4.1. Estructura general del código

El diagrama presentado en la figura 4.1 presenta la estructura general del código. Los cuadros de la derecha del diagrama son explicativos. En el caso de que tengan una flecha, explican las variables a introducir al subprograma o una definición del mismo. Por otro lado, los que terminan con un punto indican el tipo de información que se está transmitiendo de un subprograma al siguiente.

La estructura del código fuente utiliza la programación por objetos y consiste un subprograma que ejecuta el algoritmo que a su vez llama a otros subprogramas. En el primer subprograma llamado *CorrerAlgoritmo* se definen las variables de la cantidad de generaciones, tamaño de la población, la cantidad de funciones objetivo y los rangos de las variables del resorte. Otro de los subprogramas importantes a mencionar es el de *EvaluarObjetivo* donde se definen las restricciones, fuerzas externas, funciones objetivo y propiedades del material. En el anexo A.2. se incluye una guía detallada de como utilizar y definir las variables para poder utilizar el algoritmo creado. Además, cabe destacar que el algoritmo se construyó tomando como referencia otros algoritmos existentes como se muestra en [59] y [60].

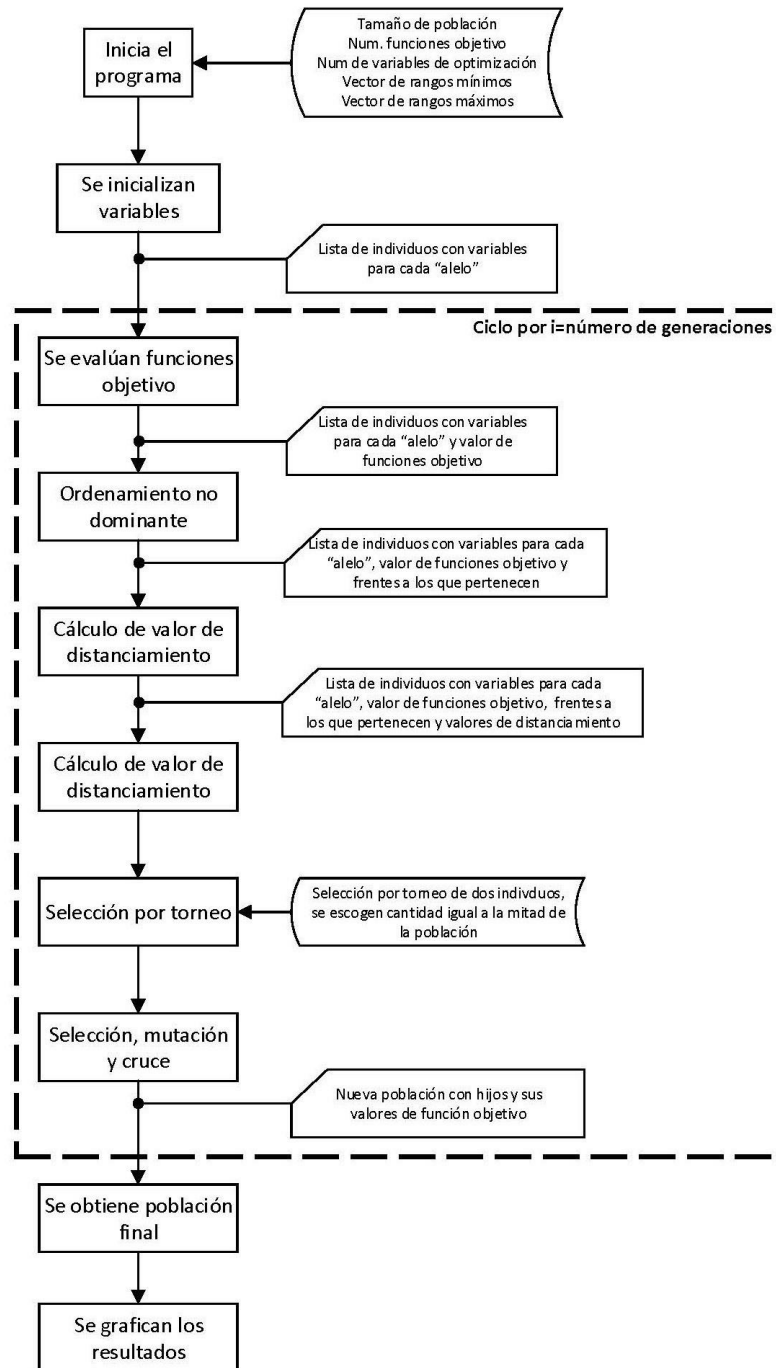


Figura 4.1: Estructura general del código creado (Fuente: el autor).

Entre los principales retos a la hora de desarrollar este algoritmo fue asegurar la estabilidad del mismo y la convergencia de los resultados mediante las verificaciones que se realizaron. Además está la dificultad de lograr tener una respuesta adecuada del modelo dinámico ante los diferentes patrones de fuerza y poder aplicar diferentes restricciones al algoritmo de manera simultánea y efectiva. Estos y otros retos se detallan y discuten a lo largo de este capítulo.

4.2. Casos de estudio resueltos con el algoritmo

Una vez comprobada la efectividad del algoritmo en esta sección se pretenden analizar y discutir los resultados obtenidos para varios casos de diseño. Estos casos de diseño varían tanto en el tipo y magnitud de la carga así como en las dimensiones del resorte diseñado. La idea de estos casos es someter a prueba la capacidad de optimización del algoritmo con las funciones objetivo de interés y analizar con detalle los resultados obtenidos.

4.2.1. Carga de onda de presión

Los resultados obtenidos para este caso de estudio se presentan en las figuras 4.2. La figura representa la frontera pareto óptima antes los tres objetivos de optimización: la masa, la resistencia a la fatiga y el factor de diseño por robustez. Además de esto en la figura se presentan tres de los individuos pertenecientes a la frontera pareto-óptima con la finalidad de recalcar el efecto que tiene sobre el diseño del resorte cada posición de la frontera. Algo importante a recalcar de la gráfica es el impacto que tiene el patrón de la fuerza externa en la frontera pareto óptima debido a que, a pesar de no tener un impacto directo sobre la función objetivo de la masa, tiene un gran impacto en el comportamiento de la resistencia a la fatiga y la robustez del diseño. Esto se hará más evidente al comparar este caso de estudio con los otros.

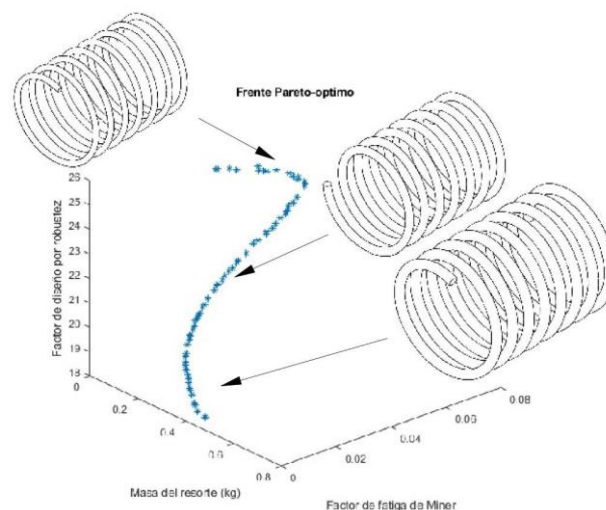


Figura 4.2: Resultado de frontera pareto-óptima para fuerza onda de presión (Fuente: el autor).

Durante este caso de estudio se notó además como entra las soluciones encontradas el diámetro del alambre es uno de los factores que más influencia el valor de idoneidad de los

individuos de la población. El gráfico de dispersión mostrado en la figura 4.3 muestra los valores para cada una de las variables de diseño para los individuos de la frontera pareto óptima de manera que cada punto es un individuo colocado en el espacio tridimensional según el valor de su diámetro del resorte, del alambre y número de espiras. Se puede notar en la misma figura como los individuos idóneos para este caso de estudio tienden a tener valores cercanos a los 4,5 cm de diámetro del resorte, sin embargo están distribuidos a lo largo de todo el rango posible para el diámetro del alambre. La cantidad de espiras activas posee una dispersión mayor al del diámetro del resorte, sin embargo la mayoría de los individuos de la población rondan un valor cercano a las 7 espiras activas.

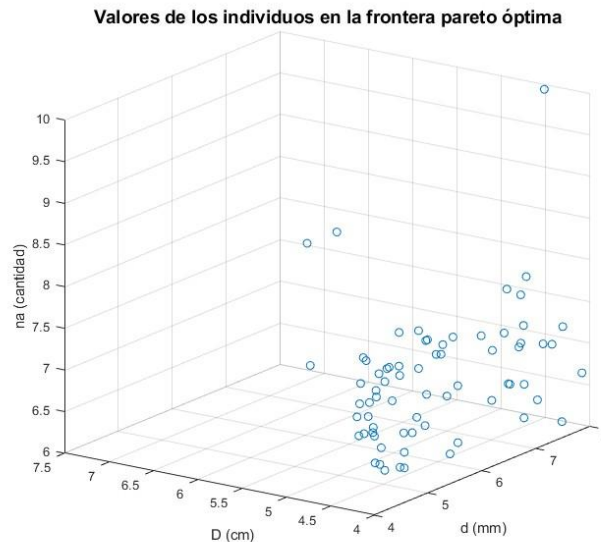


Figura 4.3: Distribución de las variables para solución de la onda de presión (Fuente: el autor).

El cuadro 4.1 presenta los valores numéricos para los parámetros de diseño de algunos individuos representativos de la frontera pareto óptima. El cuadro muestra evidencia de la relación existente entre la masa y la robustez del diseño en la cual aparenta que a mayor masa se tiene un menor factor de diseño por robustez y, por ende, un diseño más robusto. Además de esto se puede notar como no se tiene un patrón claro sobre el factor de Miner, ya que es altamente dependiente de cuando un diseño del resorte tenga una frecuencia natural que se acerque a las frecuencias principales de la fuerza externa.

Mediante el uso de la figura 4.3 se puede evidenciar como los puntos de la frontera óptima tienden a tener valores del diámetro del resorte cercanos al límite inferior del ámbito permitido para la variable. Esto indica que a diámetros de resorte menores se tiene una mejor proporción de masa contra resistencia a la fatiga y robustez del diseño en las condiciones de diseño para el caso de estudio y la fuerza a la que está siendo sometido.

Entre los casos de estudio este es el que representa un aplicación más cercana a las condiciones de diseño a las cuales se podría encontrar una válvula de control. Es importante destacar que debido a las altas frecuencias de la fuerza de excitación externa la cantidad de grados de libertad del modelo dinámico fue mayor a los otros casos desarrollados. Esto implica que el tiempo de ejecución del algoritmo también fue mayor debido que para cada individuo de la población se debió resolver un sistema con más grados de libertad. Es por

Cuadro 4.1: Ejemplos de resultados para el caso de estudio con fuerza de presión (Fuente: el autor).

Individuo	d (mm)	D (mm)	n_a	Masa (g)	Factor de Miner	Factor de sensibilidad
A	4,51	40	7,0	141	6,01E-02	24,5
B	6,32	42	7,0	279	1,58E-02	21,2
C	8,00	50	8,3	470	9,3E-04	18,5

esta razón que el método numérico desarrollado será más eficiente para resolver casos en los cuales las fuerzas de excitación tengan una frecuencia menor.

A nivel de la toma de decisiones para el diseño se puede notar como entre el espacio solución mostrado en la figura 4.2 las opciones para los componentes de la frontera pareto-óptima pueden variar hasta un 600% en su valor de masa con una variación de hasta 38% de la variación en la robustez del diseño. Los valores de la resistencia a la fatiga van desde individuos que tienen un valor cercano a cero hasta otros que se encuentran cerca del valor máximo de uno. Esto tiene implicaciones a nivel de diseño en cuanto a la toma de decisiones ya que aunque se favorecen los individuos que son impactados la menor cantidad posible por la fatiga es importante el efecto que tiene sobre la masa y la robustez de diseño. En la industria se podría utilizar una tabla y un criterio en el cual se define una resistencia mínima a la fatiga y se verifican que la masa y la robustez del diseño sea aceptable.

4.2.2. Carga armónica simétrica

Los resultados obtenidos en este caso se muestran en la figura 4.4. En esta figura se grafica los individuos de la frontera pareto-óptima con los tres ejes representando sus valores respectivos en cada una de las funciones objetivo. Algunos de los individuos representativos se muestran en el cuadro 4.2. Los individuos D, E y F del cuadro muestran con claridad como al tener un mayor diámetro del alambre y espiras se tiene una mayor masa, mayor resistencia a la fatiga y el diseño tiende a ser más robusto.

Además de esto la figura 4.4 muestra como los diferentes puntos de la frontera pareto-óptima representan cambios relevantes a la geometría del resorte. Al comparar la geometría de la frontera del caso de carga armónica con respecto a los otros casos de estudio se evidencia con claridad como el patrón de las fuerzas de entrada tiene un impacto significativo sobre los resultados. En la figura además se muestran tres individuos de la frontera con la finalidad de representar el impacto que existe sobre el diseño del resorte

Cuadro 4.2: Ejemplos de resultados para el caso de estudio con fuerza sinusoidal armónica (Fuente: el autor).

Individuo	d (mm)	D (mm)	n_a	Masa (g)	Factor de Miner	Factor de sensibilidad
D	4,21	40	6,5	123	4,89E-02	25,1
E	6,28	40	8,4	286	1,73E-02	21,3
F	7,67	40	7,5	410	1,54E-03	19,1

La carga simétrica es el patrón de carga más simplificado dentro de los que se presentan en los casos de estudio. Sin embargo, es un buen punto de comparación y de prueba del algoritmo con uno de los patrones de fuerza comúnmente utilizado en el diseño de componentes. La gráfica de la figura 4.5 presenta la población obtenida para este caso de

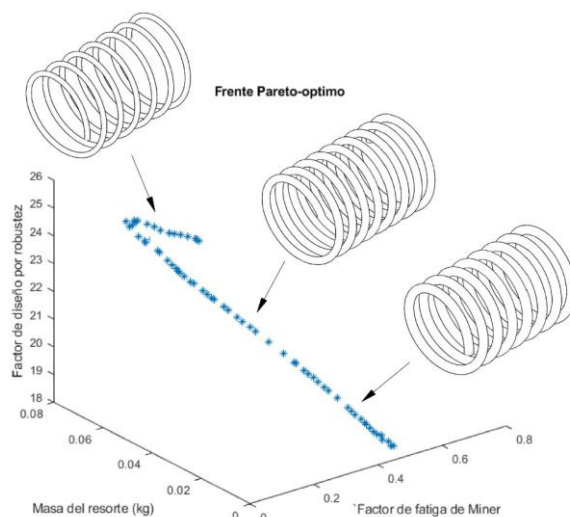


Figura 4.4: Resultado de frontera pareto-óptima para fuerza sinusoidal (Fuente: el autor).

estudio para este patrón de fuerza. Como se puede observar del gráfico de dispersión la gran parte de la población se encuentra en la región con un número de espiras entre 7 y 8, un diámetro del resorte entre 4 cm y 6 cm y un diámetro del alambre entre 4 mm y 8 mm. De esto se puede deducir que mientras que se puede obtener un diseño óptimo con un número amplio de diámetros del alambre, los valores del diámetro del resorte y número de espiras óptimos se encuentran en un ámbito relativamente pequeño.

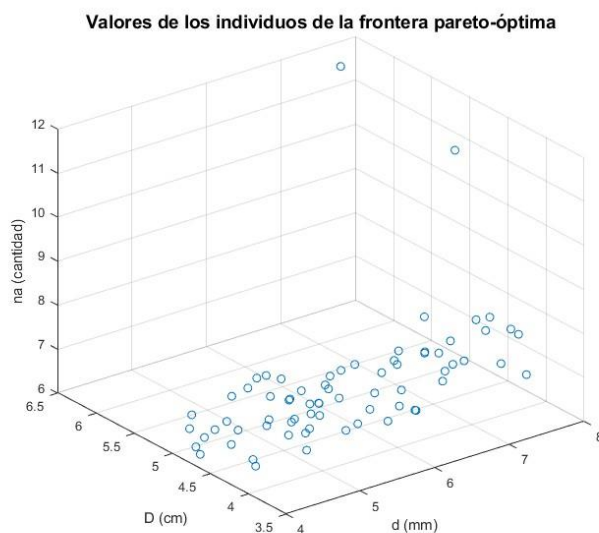


Figura 4.5: Distribución de las variables para solución de la fuerza sinusoidal. (Fuente: el autor).

A pesar de ser un patrón relativamente simple existen varias situaciones en las cuales una válvula podría estar sometida a una fuerza simétrica sinusoidal. Tal es el caso en las tuberías cuando se tiene un pico de presión que es amortiguado a lo largo de la tubería, un ejemplo de esto es cuando se tiene un golpe de ariete amortiguado a lo largo de una línea de presión o cuando se tiene una oscilación de presión debido a una bomba centrífuga aguas

arriba de la tubería [61]. Al momento de tomar una decisión de diseño para este caso en específico los resultados obtenidos son especialmente útiles para decidir en el diámetro del alambre y el número de espiras. Como se puede notar del cuadro 4.2 y de la figura 4.5 el diámetro del resorte tiende a tener un valor cercano a 4 cm por lo que al redondear los valores a un número que sea factible en la manufactura las variables del número de espiras y diámetro del alambre son los que tendrán más variación entre los individuos de la población.

De manera generalizada, las fuerzas externas se pueden descomponer en sus frecuencias principales y las magnitudes de las mismas. Es por esto que este caso de estudio tiene la capacidad de ser útil como una primera aproximación a un problema de diseño en el cual a pesar de no saber con detalle el patrón de la fuerza externa se tiene conocimiento de sus características principales. Se puede correr el algoritmo definiendo la fuerza de manera relativamente sencilla con la amplitud de la fuerza y su frecuencia y tener resultados de como se verían los individuos de la frontera pareto-óptima. Además de esto, al hacer corridas con diferentes valores muestra el comportamiento de los resultados de diseño y el modelo dinámico ante frecuencias de fuerza externa diferente. Estas frecuencias, según se ha notado en los casos de estudio del resorte, tienen un impacto significativo sobre la forma de la frontera pareto-óptima y sus valores.

4.2.3. Carga aleatoria

Los resultados de la frontera pareto-óptima obtenida al realizar la corrida con esta fuerza aleatoria se presenta en la figura 4.6. Además de esto, en la gráfica se muestran tres de los resortes representativos de la población a lo largo de la frontera. La gráfica evidencia una relación directa en la cual a menores valores de masa se obtiene un mayor factor de sensibilidad. Sin embargo esta relación no es tan definida con la fatiga puesto que los diferentes valores de masa tendrán un efecto significativo en la resonancia del sistema y los distintos modos de vibración.

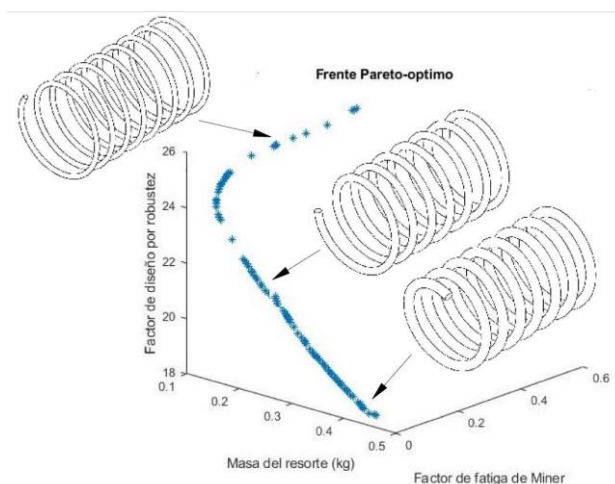


Figura 4.6: Resultado de frontera pareto-óptima para fuerza aleatoria. (Fuente: el autor).

El cuadro 4.3 presenta los valores para cada una de las variables de diseño de algunos de los individuos representativos pertenecientes a la población de la frontera pareto-óptima. El cuadro ejemplifica la relación mencionada anteriormente en la cual al aumentar

Cuadro 4.3: Ejemplos de resultados para el caso de estudio con fuerzas aleatorias (Fuente: el autor).

Individuo	d (mm)	D (mm)	na	Masa (g)	Factor de Miner	Factor de sensibilidad
G	4	40	8,1	126	4,09E-01	25,4
H	6	40	7,0	258	3,00E-02	21,6
I	8	40	7,3	462	7,41E-09	18,4

la masa se tiene una disminución del factor de sensibilidad como se observa al comparar el individuo A contra el individuo C. Además de esto, se puede notar que para estos individuos, y de manera generalizada en este caso de estudio, los valores de la frontera óptima tienen un diámetro del resorte de 40 mm. . Basado en estos resultados se infiere que al minimizar el diámetro para este caso de diseño se logran mejores valores de las funciones objetivo de interés y se logra la mejor relación entre masa, resistencia y robustez.

La carga aleatoria posee un comportamiento similar en cuanto a la varianza en las magnitudes de las variables de diseño a los casos anteriores, en las cuales el diámetro del alambre es el factor que varía significativamente entre los individuos de la población. La figura . Además de esto se puede notar que a aumentar la masa el resorte rápidamente se empieza a ver susceptible ante la excitación de la frecuencia externa. Esto se evidencia por medio de la frontera pareto-óptima presentada en la figura 4.6.

La figura 4.7 presenta las combinaciones de las variables de diseño de los individuos pertenecientes a la frontera pareto-óptima. La figura muestra como, a pesar de que se tiene una mayor dispersión en el espacio de variables de entrada, se tiene una tendencia similar a las fuerzas anteriores en la cual el diámetro del alambre es la variable que posee mayor cambios en el ámbito de diseño. Esto tiende a indicar que es una de la variables principales que rige el movimiento del punto de diseño a lo largo de la frontera pareto-óptima.

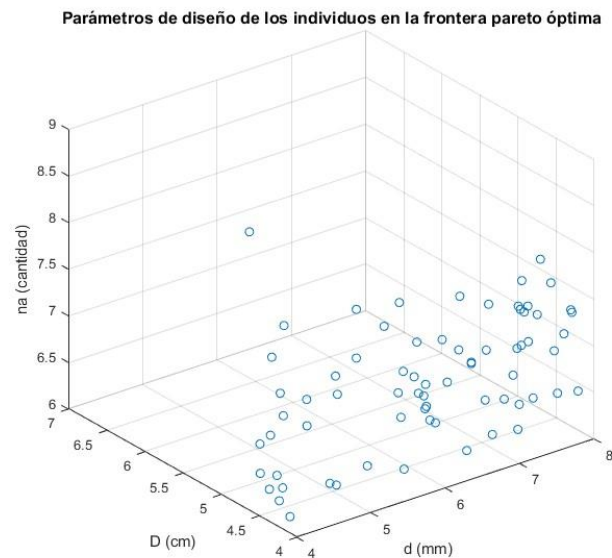


Figura 4.7: Distribución de las variables para solución la fuerza aleatoria. (Fuente: el autor).

La carga aleatoria representa un tipo de fuerza a la cual los componentes mecánicos

se encuentran sometidos comúnmente en el diseño de elemento mecánicos. En el caso de válvulas se pueden tener incrementos súbitos de presión que generen un impacto en la tubería. Además es común utilizar resortes en aplicaciones para amortiguar algún impacto o fuerza no controlada como es el caso de los compensadores en vehículos automotrices. Este tipo de aplicaciones le da relevancia a correr un caso de estudio como este. Además de ser un caso en el que las fuerzas externas tienen un patrón relativamente más complejo, sirve para determinar el comportamiento que el método numérico tiene ante este tipo de caso y los resultados que genera el mismo.

Los resultados obtenidos para este caso se asemejan en cierta forma al comportamiento obtenido en los otros dos casos de estudio, sin embargo cabe destacar que la variación u oportunidad de selección entre los individuos de la frontera pareto-óptima es mejor. Como ejemplo de esto se tiene que la diferencia entre el individuo con menos masa al de mayor masa es de alrededor de 315% con una variación de la robustez del diseño de hasta 40%, comparado con los otros dos casos que tenían alrededor de 600% de variación en la masa entre individuos de la frontera. Esto implica en que el ámbito de selección de individuos se tiene una gama más reducida de opciones en cuanto a la masa, mientras que la robustez del diseño y el factor de fatiga mantienen tamaños de ámbito similares a los otros casos de estudio.

4.3. Comentarios generales sobre el algoritmo

Durante la ejecución de los casos de estudio se identificó la dependencia del algoritmo y sus resultados con respecto al patrón de fuerzas utilizado. Se notó como el patrón de fuerzas es uno de los factores relevantes que definen la geometría de la frontera pareto óptima. Las fuerzas y sus frecuencias son uno de los factores predominantes en la respuesta del modelo dinámico del resorte. Esta respuesta dinámica estará relacionada directamente a la resistencia a la fatiga del resorte y por ende también tendrá un efecto sobre la robustez del diseño. Al comparar las figuras 4.2, 4.4 y 4.6 se evidencia como a pesar de que la masa tiene una misma tendencia al no ser influenciada por la fuerza externa, la frecuencia natural del resorte y su respuesta ante una frecuencia de excitación externa sí tienen un impacto significativo sobre los resultados obtenidos.

Otro de los factores que se notó al hacer las corridas de verificación, casos de estudio y durante el desarrollo del algoritmo es el impacto que tienen la complejidad de las restricciones sobre el tiempo de corrida del algoritmo. Durante las corridas del algoritmo se notó que al agregar restricciones se aumentaba el tiempo de corrida en aproximadamente un 15% al compararlo con el algoritmo sin restricciones. Esto debido a que representa un chequeo adicional que se realiza a cada individuo de la población. De manera similar es útil destacar que la complejidad de las funciones objetivo es otro de los factores de relevancia que afectan el tiempo de corrida del algoritmo ya que son ejecutadas para cada individuo de la población individualmente y pueden implicar un gran número de pasos. La complejidad de las ecuaciones y restricciones son dependientes del modelo de diseño utilizado que represente el caso de estudio, es ahí donde utilizar un método de discretización del modelo mecánico a uno matemático que simplifique en la medida de lo posible la capacidad computacional requerida tendrá un efecto significativo en la velocidad y eficiencia del algoritmo.

Las corridas de verificación se utilizaron para ajustar los parámetros del algoritmo

de población, cantidad de generaciones y porcentajes de mutación y cruce. Durante este proceso se notó la importancia que tiene la cantidad de la población en lograr una frontera pareto-óptima con diversidad y que realmente refleje la frontera del espacio solución. Por otra parte, se confirmó como la cantidad de generaciones es uno de los factores principales que afectan la convergencia ya que deben pasar suficientes generaciones para que los individuos lleguen hasta la frontera pareto-óptima. Durante la verificación se notó que de manera generalizada a partir de aproximadamente 100 generaciones no había cambios significativos en la frontera pareto-óptima por lo que se utilizó ese valor en los casos de estudio. Los porcentajes de mutación y cruce se definieron basados en la literatura, y tuvieron un buen rendimiento al ser comparados con los resultados de las ecuaciones teóricas.

El tipo de resultado obtenido en los casos de estudio presentado tiene el potencial de ser de gran utilidad para la industria de manufactura que utilicen resortes helicoidales, tal como el de las válvulas de control. Los resultados de la frontera pareto-óptima pueden servir como una guía para los diseñadores de estos componentes para poder seleccionar las dimensiones del resorte que darán un desempeño óptimo del mismo. Como ejemplo de esto, para casos frecuentes en la industria se pueden hacer tablas en las cuáles se presenten las dimensiones de cada uno de los individuos de la frontera pareto-óptima con sus valores en las funciones objetivo.

Los resultados obtenidos además proveen una alternativa de diseño a la metodología que es utilizada actualmente en la industria. Este método de diseño puede utilizarse para verificar los diseño existentes, pero también para encontrar oportunidades de mejora en los mismos como un punto de comparación que representa ventajas en cuanto al costo pero tiene un mejor rendimiento ante la fatiga y robustez del diseño. La optimización, según se evidencia en los casos de estudio, propone una alternativa que puede ahorrar en costos de material y de manufactura, al proponer combinaciones de diseño que poseen una menor masa y sensibilidad ante cambios dimensionales en las tolerancias del diseño.

Cabe destacar que los casos de estudio demostraron el potencial del algoritmo para ahorrar en gastos de manufactura y materiales. Las ventajas que tiene utilizar alguno de los individuos de la frontera pareto-óptima contra otras opciones que no pertenecen a la misma se pueden resumir en tres puntos. Como primera instancia los resortes requieren menos masa para alcanzar un mayor resistencia a la fatiga por lo que su costo es más bajo. Luego se tiene una mayor resistencia a la fatiga para un mismo costo, lo cual también conlleva a fallas menos frecuentes del componente a la fatiga. Finalmente, se tiene un diseño más robusto que es menos susceptible a variaciones de la materia prima y por ende requerirá de menores controles para lograr la tolerancia deseada.

La relativa facilidad con la que se pueden cambiar los parámetros de entrada del algoritmo para tener los diferentes casos de diseño muestra la utilidad del algoritmo para poder comparar los elementos idóneos ante parámetros de diseño distintos. La gran diferencia en los resultados presentados para los tres patrones de fuerzas muestran con claridad el gran impacto que tienen las condiciones externas sobre los parámetros óptimos de diseño. Es ahí donde tener la posibilidad de correr varios escenarios en los cuales se podría requerir que opere un resorte facilita las decisiones de diseño y ayuda a seleccionar un elemento que opere dentro de los parámetro óptimos.

Capítulo 5

Conclusión

Este capítulo del trabajo pretende sintetizar las contribuciones intelectuales y el valor científico aportado por este documento. Además, los aportes descritos se relacionarán directamente con los objetivos propuestos de la y se comentarán las posibles acciones futuras a cubrir por otras investigaciones en el futuro. Finalmente se brindan recomendaciones a seguir en caso de querer seguir desarrollando la implementación de este algoritmo.

5.1. Conclusiones y aportes de este trabajo

Durante esta investigación se logró desarrollar con efectividad un método numérico para la optimización de resortes helicoidales a compresión. Este método numérico utiliza el algoritmo genético NSGA II con la finalidad de poder dar al diseñador las combinaciones óptimas de diámetro del resorte, diámetro del alambre y número de espiras que minimicen la masa del resorte, mientras se maximiza su resistencia a la fatiga y robustez del diseño. El método numérico fue verificado con ecuaciones de prueba de diferentes complejidades en cuales se sabe su resultado teórico, por lo que se puede comparar el desempeño del algoritmo contra los resultados teóricos. Este procedimiento comprueba el rendimiento del algoritmo y es evidencia del correcto funcionamiento y convergencia del algoritmo propuesto. Los resultados de esta verificación se presentan en el anexo A.1.

La investigación además logra cuantificar y definir los criterios de diseño y condiciones a las cuales está sometido el resorte. Como parte de esto se elaboró un modelo dinámico de varios grados de libertad con la finalidad de calcular la respuesta a la frecuencia del resorte. El modelo funciona de tal forma que se asigna un número de grados de libertad que logre capturar los efectos de las principales frecuencias de la fuerza externa. Esta aproximación de utilizar un modelo dinámico provee un cálculo que captura el comportamiento del resorte de una manera más realista que un modelo estático. Un ejemplo de esto es que en los casos de estudio, a pesar de que muchos de los individuos de la población no entraban en fluencia, tenían un factor de Miner bajo por lo que fallarían por fatiga.

Además de definir un modelo dinámico se realiza un análisis de resistencia a la fatiga con el método de acumulación de daño de Miner. Al utilizar este método en conjunto con el modelo dinámico se puede estimar la resistencia del resorte a la fatiga tomando en cuenta las frecuencias de las fuerzas externas. En los casos de estudio se nota como la resistencia a la fatiga es uno de los factores que tiene más variación con respecto a cambios en el diseño. Estos cambios en el diseño debido a la respuesta dinámica muestran la gran influencia que tiene la forma de la fuerza externa sobre la optimización del diseño del resorte. La masa y rigidez del resorte poseen un papel fundamental en el cálculo de los modos de vibración

principales del resorte helicoidal, y estos están directamente ligados a la respuesta dinámica y resistencia a la fatiga del resorte. Uno de los individuos podría presentar una mayor resistencia ante un caso estático, pero si alguno de sus primeros modos de vibración se encuentra en fase con las frecuencias de la fuerza externa tendrá un bajo rendimiento ante la acumulación de daño por fatiga. Es ahí donde un modelo dinámico como el utilizado aporta un valor adicional y toma en cuenta no solo la resistencia del resorte ante una posible carga máxima, sino que también como sus modos de vibración interactúan con la carga externa.

Otro de los aportes de esta investigación que la diferencian de otras similares es la de utilizar un enfoque de diseño por robustez para el resorte. El algoritmo implementado logró aplicar de manera efectiva la optimización de un diseño robusto. Además de esto el método numérico busca minimizar los cambios en la masa y la resistencia a la fatiga ante variaciones incontrolables del diámetro del resorte, el diámetro del alambre y el esfuerzo de fluencia del material. Esto se ve como una minimización de la sumatoria de los cuadrados de los factores de sensibilidad para cada una de estas variables en los casos de estudios presentados en los resultados del trabajo. El usar una aproximación de diseño robusto para los resortes helicoidales es una alternativa al factor de seguridad convencional que toma en consideración la manufactura de los resortes y una mejor adaptabilidad a la variación en tolerancias de materia prima. Además, combinar el concepto de diseño por robustez con el algoritmo genético utilizado tiene una gran ventaja al poder evaluar los factores de sensibilidad de un gran número de individuos del espacio solución y realmente saca provecho a este criterio de diseño al automatizar el cálculo de los factores de sensibilidad y comparar un gran número de los individuos de la población.

El algoritmo y casos de estudio de este trabajo fueron enfocados principalmente al resorte de una válvula de compresión. Sin embargo, puede ser modificado con facilidad para optimizar resortes helicoidales en otras funciones al cambiar las fuerzas de entrada y las condiciones de diseño seleccionadas por el usuario. A través del algoritmo se logró obtener el diseño para tres casos de estudio particulares con la finalidad de evaluar su uso. Los casos evaluados fueron ante fuerzas sinusoidales, una fuerza aleatoria y una fuerza que simula una onda de presión con base en la recompilación bibliográfica realizada. Estos casos aproximan posibles fuerzas externas a las que podría estar sometido el resorte en un caso ideal y ayudan a dar una idea de las posibles relaciones que existen entre la forma de la fuerza externa y el efecto que tiene esta sobre el diseño del resorte. Además muestran la facilidad con la que se pueden ajustar los parámetros de entrada para tener el espacio solución que es comparable entre diferentes casos pudiendo representar de manera relativamente ágil varias condiciones de diseño del resorte. La capacidad de correr varios tipos de fuerza y parámetros de manera relativamente simple es de gran interés para el diseñador debido al impacto significativo que tienen los distintos parámetros de entrada sobre el diseño sobre el resorte helicoidal.

Los casos de estudio mostraron como el patrón de las fuerzas de entrada tiene un impacto significativo sobre la forma y valores de la frontera Pareto óptima. La evidencia principal de esto se puede notar al comparar las figuras 4.2, 4.4 y 4.6, así como los valores de las tablas 4.1, 4.2 y 4.3. Además de esto los casos de estudio hicieron evidente la tendencia encontrada que a mayor masa se suele tener un menor factor de sensibilidad y por ende más robustez. Esto se infiere que es debido a que elementos de mayor tamaño podrán absorber o ser afectados en menor manera ante variaciones relativamente pequeñas en sus dimensiones y características del material. Sin embargo, no se puede decir lo mismo de

el factor de fatiga de Miner ya que es altamente sensible a la respuesta a la frecuencia del resorte según se descubrió en los casos de estudio mostrados en los resultados. Esto se debe principalmente a que aunque se tenga una mayor masa en el resorte, si este tiene una frecuencia natural de alguno de sus principales modos de vibración que es un submúltiplo de las frecuencias de la fuerza de excitación el elemento mecánico entrará en resonancia e incrementará las fuerzas internas a las que está sometido por lo que no necesariamente una mayor masa se traslada a una mayor resistencia a la fatiga.

Los resultados analizados en los casos de estudio mostraron como los valores para las variables de diseño del resorte de los individuos pertenecientes a la frontera pareto-óptima tienden a tener una mayor varianza en el diámetro del alambre comparado con el diámetro del resorte y el número de espiras. De esto se concluye que en los casos de estudio analizado cambios en el diámetro del alambre tendrán un mayor efecto sobre la robustez del diseño, la masa y la resistencia a la fatiga que cambios en las otras variables por lo que alteraciones en esta dimensión implican un mayor cambio del punto de diseño.

Durante la corrida de los casos de estudio y verificación del algoritmo se notó la importancia que tienen la cantidad de generaciones y población sobre el resultado final. Un número alto de población ayuda a tener una frontera más diversa ya que existirán más individuos en la frontera pareto-óptima. Por otra parte, la cantidad de generaciones es la que ayudará a alcanzar la convergencia ya que se hará el proceso iterativo un mayor cantidad de veces.

La investigación logró demostrar la versatilidad del algoritmo NSGA II y su adaptabilidad para poder ser utilizado exitosamente en la optimización de válvulas de control. El método numérico definido puede ser ampliado con relativa facilidad a una mayor cantidad de funciones objetivo y variables de diseño de ser requerido. Además de esto el método demostró ser efectivo para optimizar los modelos numéricos propuestos para definir el problema de diseño que representan estos resortes.

El tipo de resultados obtenidos en los casos de estudio pueden ser utilizados para crear tablas de diseño o material similar con la finalidad de ser usado por los diseñadores sin la necesidad de realizar nuevas corridas del algoritmo. Un ejemplo en cual esto sería de utilidad es al afrontar casos de diseño similares y repetitivos. Tal es el caso en las industrias en las que se diseñen productos que incluyan resortes se pueden crear tablas o material de apoyo para tomar decisiones rápidas y óptimas, como por ejemplo en la industria de las válvulas de control. De esta manera al tener un criterio del valor deseado de una de las funciones objetivo, esto implica un punto en la frontera pareto-óptima con dimensiones del resorte óptimas. Además, se podrá corroborar el valor de las otras funciones objetivo de interés para verificar que sean aceptables.

El método numérico creado además tiene implicaciones en cuanto al ahorro en la manufactura y en la reducción de la velocidad de fatiga. Al utilizar los individuos de la frontera pareto-óptima se estarán escogiendo los individuos que requieren una menor masa y cuya combinación de variables ofrecen una mayor resistencia a la fatiga basado en las formulaciones matemáticas propuestas. Una menor masa conlleva a menos material utilizado y, por ende, un ahorro en los costos de la materia prima para los resortes. Además de esto una mayor robustez del diseño implica que se requerirá de un control menos estricto de las tolerancias en las dimensiones del resorte analizadas y la resistencia del material lo cuál conlleva una manufactura de menor costo.

La investigación propuesta además ayuda a evidenciar la necesidad cada vez más imperativa de los profesionales en ingeniería mecánica de tener capacidades de programación más allá de los métodos numéricos tradicionales. Herramientas como los algoritmos evolutivos y otros métodos de optimización son necesarios con la finalidad de poder dar un nuevo enfoque con valor agregado a los problemas de diseño que se presentan en la industria. Conocer estas técnicas para poder dar una justificación objetiva a cambios de diseño son altamente relevantes para poder demostrar los beneficios y ventaja competitiva del cambio que se realizará. Para poder desarrollar este tipo de herramientas en el área de diseño de componentes no es suficiente únicamente tener conocimientos de programación, sino que es necesario tener un dominio del tema en el que se desea aplicar para poder dirigir correctamente el diseño del del algoritmo que se está desarrollando.

Conjuntos de herramientas como la que se desarrolló en este trabajo también son de gran utilidad para poder realizar un cambio de diseño que tenga un impacto positivo en la empresa, pero que también logre reducir minimizar los riesgos del cambio. La frontera Pareto-óptima es una manera de presentar la relación del balance que se tiene entre la disminución y aumento en los valores de la masa, resistencia y robustez que usualmente es el punto de pivote principal para tomar una decisión al diseñar un elemento mecánico. Es decir se quiere escoger el punto que tenga suficiente masa para que el resorte sea funcional y seguro pero no más de esa cantidad ya que implicaría gastos adicionales.

5.2. Recomendaciones y trabajo futuro

Una de las recomendaciones y posible ampliación del trabajo hacia el futuro es crear una metodología para cuantificar las fuerzas externas que son aplicadas para el resorte. Para el método numérico aplicado en este caso se tomaron las fuerzas externas como una entrada conocida en el algoritmo. Por esta razón, se puede tomar como un trabajo futuro realizar un método numérico o análisis para cuantificar las fuerzas que se le aplicarían al resorte según la geometría de la válvula.

Se sugiere que para un trabajo futuro se amplíe el ámbito de aplicación y casos de estudio para otras situaciones comunes en las cuales se utilicen resorte de compresión helicoidales. Como primera instancia se identifican casos como los resortes de suspensión en automóviles y amortiguadores de maquinaria o equipo. Estos son casos en los cuales el método numérico implementado sería aplicable y son semejantes a las válvulas de control ya que la respuesta dinámica es sumamente importante. Es por esto que se puede explorar utilizar un acercamiento similar al realizado en este trabajo.

Se recomienda reafirmar algunos de los casos propuestos con un modelo experimental para darle un grado más de veracidad de los cálculos realizados. Además, en una investigación futura se pueden utilizar análisis de elementos finitos para tener un nivel de mayor precisión en el cálculo de esfuerzos en modalidad elástica y mayores grados de libertad para calcular la respuesta a la fatiga. Este tipo de ejercicio ayudaría a reafirmar los resultados obtenidos y le daría aún mayor validez al modelo dinámico propuesto.

Evaluar la posibilidad de aplicar técnicas de programación en paralelo para evaluar simultáneamente las funciones objetivo y restricciones del algoritmo pueden ser una herramienta de utilidad para agilizar el tiempo de corrida del método numérico. Se propone

explorar esta opción más profundamente como una posible mejora a desarrollar del método numérico en caso de que el tiempo de ejecución sea una característica de interés a mejorar.

Una opción a desarrollar para futuros trabajos con la finalidad de ampliar el ámbito de aplicación de este algoritmo es desarrollar un modelo dinámico que tome en cuenta fenómenos como la vibraciones no lineales o auto-excitadas. Este tipo de vibraciones se pueden presentar en tuberías y válvulas y desarrollar un modelo que las tome en cuenta podrá agregar más exactitud a la representación matemática del sistema mecánico en estudio. Además de esto agregar estos factores dentro del modelo tomaría en cuenta fuerzas adicionales que aparecen en la interacción de estos fenómenos vibratorios con el modelo dinámico que se propuso en este trabajo.

Cabe destacar que aunque se notaron algunas tendencias con los casos de estudio en la relación de la masa con la robustez de diseño y la resistencia a la fatiga, debido a que se corrieron únicamente tres casos de estudios no hay suficientes datos para generalizar estas. Es recomendable en trabajos futuros que estas relaciones sean de especial interés correr más casos de prueba para confirmar si estas relaciones son aplicables a un grupo más amplio de casos. Esto también tendría la ventaja de poder tomar decisiones de diseño más informadas ante posibles variaciones al saber la relación entre los parámetros de entrada.

Bibliografia

- [1] M. Watch, “Control valves market report 2021-2027: Market size, industry outlook, market forecast, demand analysis, market share,” 2020.
- [2] L. Wood, “Global control valve market to 2025: Focus on oil gas, water wastewater treatment, energy power, chemicals,” 2020.
- [3] K. Sotoodeh, *Subsea Valves and Actuators for the Oil and Gas Industry*. Elsevier Science, 2021.
- [4] L. Wood, “Control valves industry report.”
- [5] E. Electric, “Daniel liquid control valves,” 2020.
- [6] —, “Daniel series 700 - model v707 spring loaded back pressure valve,” 2019.
- [7] R. Budynas, J. Shigley, and J. Nisbett, *Shigley's Mechanical Engineering Design*, ser. McGraw-Hill series in mechanical engineering. McGraw-Hill, 2008, no. v. 10.
- [8] Y. Zhang, “Reliability-based robust design optimization of vehicle components, Part II: Case studies,” *Frontiers of Mechanical Engineering*, vol. 10, pp. 145–153, 2015.
- [9] H. Kim, S. Kim, Y. Kim, and J. Kim, “Optimization of Operation Parameters for Direct Spring Loaded Pressure Relief Valve in a Pipeline System,” *Journal of Pressure Vessel Technology*, vol. 140, no. 5, p. 051603, Oct. 2018.
- [10] J. Zhou, C. Wang, and J. Zhu, “Multi-objective optimization of a spring diaphragm clutch on an automobile based on the non-dominated sorting genetic algorithm (nsga-ii),” *Mathematical and Computational Applications*, vol. 21, p. 47, 12 2016.
- [11] X. Qimin, L. Liwei, and X. Qili, “The optimal design and simulation of helical spring based on particle swarm algorithm and matlab,” *Wseas transactions on circuits and systems*, 2009.
- [12] H. Bock, X. Hoang, R. Rannacher, and J. Schlöder, *Modeling, Simulation and Optimization of Complex Processes: Proceedings of the Fourth International Conference on High Performance Scientific Computing, March 2-6, 2009, Hanoi, Vietnam*. Springer Berlin Heidelberg, 2012.
- [13] R. Arora, *Optimization: Algorithms and Applications*. CRC Press, 2015.
- [14] A. S. Corporation and B. Division, *Mechanical Springs: Their Engineering and Design*. Literary Licensing, LLC, 2013.
- [15] J. Shunta, *Achieving World Class Manufacturing Through Process Control*. Prentice Hall PTR, 2015.

- [16] D. Montgomery, *Design and Analysis of Experiments*, ser. Student solutions manual. John Wiley & Sons, 2008.
- [17] K.-L. Tsui, “An overview of taguchi method and newly developed statistical methods for robust design,” *IIE Transactions*, vol. 24, no. 5, pp. 44–57, 1992.
- [18] F. Tayyebi, M. Bi Jan Khan, and H. Ramezani Etedali, “Experimental study of parameters affecting the performance of discharge control valve: effects of the spring fatigue and valve assembly,” *Irrigation and drainage structures engineering research*, vol. 20, no. 77, pp. 73–88, 2020.
- [19] E. Electric, “User manual - daniel series 700 valve,” 2019.
- [20] A. Ugural, *Mechanical Engineering Design: Third Edition*. CRC Press, 2020.
- [21] ASTM, *ASTM E1049-85 Standard Practices for Cycle Counting in Fatigue Analysis*, 2017.
- [22] Y.-L. Lee and T. Tjhung, *Metal Fatigue Analysis Handbook*. Boston: Butterworth-Heinemann, 2012.
- [23] Y. Lee, J. Pan, R. Hathaway, and M. Barkey, *Fatigue Testing and Analysis: Theory and Practice*. Elsevier Science, 2005.
- [24] A. Carpinteri, M. De Freitas, and A. Spagnoli, *Biaxial/Multiaxial Fatigue and Fracture*, ser. ISSN. Elsevier Science, 2003.
- [25] A. Fatemi and L. Yang, “Cumulative fatigue damage and life prediction theories: a survey of the state of the art for homogeneous materials,” *International Journal of Fatigue*, vol. 20, no. 1, pp. 9–34, 1998.
- [26] P. Gu, B. Lu, and S. Spiewak, “A new approach for robust design of mechanical systems,” *CIRP Annals-Manufacturing Technology*, vol. 53, no. 1, pp. 129–133, 2004.
- [27] K. Dehnad, *Quality Control, Robust Design, and the Taguchi Method*. Springer US, 2012.
- [28] A. D. Belegundu and S. Zhang, “Robustness of Design Through Minimum Sensitivity,” *Journal of Mechanical Design*, vol. 114, no. 2, pp. 213–217, 06 1992.
- [29] H. Jian, W. Wei, H. Li, and Q. Yan, “Optimization of a pressure control valve for high power automatic transmission considering stability,” *Mechanical Systems and Signal Processing*, vol. 101, pp. 182–196, 2018.
- [30] J. Y. Wang, H. T. Wang, and L. M. Zheng, “Chaos Control of Vehicle Nonlinear Suspension System with Multi-Frequency Excitations by Nonlinear Feedback,” *Applied Mechanics and Materials*, vol. 55-57, pp. 1156–1161, 2011.
- [31] K.-H. Chang, “Chapter 5 - multiobjective optimization and advanced topics,” in *Design Theory and Methods Using CAD/CAE*, K.-H. Chang, Ed. Boston: Academic Press, 2015, pp. 325 – 406.
- [32] L. H. Lee, E. P. Chew, S. Teng, and D. Goldsman, “Finding the non-dominated pareto set for multi-objective simulation models,” *IIE Transactions*, vol. 42, no. 9, pp. 656–674, 2010.

- [33] J. Pinilla and J. Orjuela Castro, "Optimización multiobjetivo en la gestión de cadenas de suministro de biocombustibles. una revisión de la literatura," *Ingeniería*, vol. 20, 03 2015.
- [34] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Chapter 10 - metaheuristic algorithms: A comprehensive review," pp. 185–231, 2018.
- [35] C. Pouw, *Development of a Multiobjective Design Optimization Procedure for Marine Propellers*. TU Delft, 2007.
- [36] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. USA: John Wiley Sons, Inc., 2001.
- [37] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms*, ser. Decision Engineering. Springer London, 2010.
- [38] F. Nieto and J. Montoya-Torres, "Implementación de un procedimiento basado en algoritmos evolutivos para programar la producción de marquillas estampadas por transferencia térmica," *Ingeniería y Universidad*, 01 2008.
- [39] K. Deb, "Multi-objective optimization using evolutionary algorithms: an introduction," *KanGAL Report*, vol. 2011003, 2011.
- [40] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [41] M. Abdollahi-Moghaddam, M. Rejvani, and P. Alamdari, "Determining optimal formulations and operating conditions for al2o3/water nanofluid flowing through a microchannel heat sink for cooling system purposes using statistical and optimization tools," *Thermal Science and Engineering Progress*, vol. 8, 2018.
- [42] S. Verma, M. Pant, and V. Snasel, "A comprehensive review on nsga-ii for multi-objective combinatorial optimization problems," *IEEE Access*, vol. 9, 2021.
- [43] Abiri, Ebrahim, Bezareh, Zobeideh, Darabi, and Abdolreza, "The optimum design of ram cell based on the modified-gdi method using non-dominated sorting genetic algorithm ii (nsga-ii)," *Journal of Intelligent Fuzzy Systems*, 2017.
- [44] J. Zhong, X. Hu, J. Zhang, and M. Gu, "Comparison of performance between different selection strategies on simple genetic algorithms," pp. 1115–1121, 2005.
- [45] B. Miller and D. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Urbana*, vol. 51, p. 61801, 1995.
- [46] "Nature-inspired optimization algorithms," in *Nature-Inspired Optimization Algorithms*, X.-S. Yang, Ed. Oxford: Elsevier, 2014.
- [47] S. Luke and L. Spector, "A comparison of crossover and mutation in genetic programming," *Genetic Programming*, vol. 97, pp. 240–248, 1997.
- [48] A. Fernandez, "Understanding genetic algorithms. a use case in the organizational field," 2020.
- [49] S. Patil and M. Bhende, "Comparison and analysis of different mutation strategies to improve the performance of genetic algorithm," *IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 4669–4673, 2014.

- [50] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," *International Journal of Artificial Intelligence and Soft Computing*, vol. 4, no. 1, pp. 1–28, 2014.
- [51] C. Hós, A. Champneys, K. Paul, and M. McNeely, "Dynamic behaviour of direct spring loaded pressure relief valves in gas service: II reduced order modelling," *Journal of Loss Prevention in the Process Industries*, vol. 36, pp. 1–12, 2015.
- [52] A. Springs, "Standard spring dimension tolerances for custom and stock springs."
- [53] S. A. Nama, "Effect of pitch angle on static characteristics of a helical machined spring," *Iraqi J Mech Mater Eng*, vol. 15, no. 3, pp. 44–58, 2015.
- [54] A. Seshadri, *A Fast Elitist Multiobjective Genetic Algorithm*. Science Direct, 2019.
- [55] K. Deb, R. B. Agrawal *et al.*, "Simulated binary crossover for continuous search space," *Complex systems*, vol. 9, no. 2, pp. 115–148, 1995.
- [56] M. R. O. Kakde, "Survey on multiobjective evolutionary and real coded genetic algorithms," in *Proceedings of the 8th Asia Pacific symposium on intelligent and evolutionary systems*. Citeseer, 2004, pp. 150–161.
- [57] H.-G. Beyer and K. Deb, "On self-adaptive features in real-parameter evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 3, 2001.
- [58] K. Deb, K. Sindhya, and T. Okabe, "Self-adaptive simulated binary crossover for real-parameter optimization," in *Proceedings of the 9th annual conference on genetic and evolutionary computation*, 2007, pp. 1187–1194.
- [59] A. Seshadri, "Elitist multiobjective genetic algorithm," 2017.
- [60] M. Kalami, "Nsga-ii," 2015.
- [61] H. Cao, M. Mohareb, and I. Nistor, "Finite element for the dynamic analysis of pipes subjected to water hammer," *Journal of Fluids and Structures*, vol. 93, p. 102845, 2020.
- [62] F. Kursawe, "A variant of evolution strategies for vector optimization," in *Parallel Problem Solving from Nature*, ser. Lecture Notes in Computer Science, H.-P. Schwefel and R. Männer, Eds. Berlin, Heidelberg: Springer, 1991, pp. 193–197.
- [63] C. Poloni, A. Giurgevich, L. Onesti, and V. Pediroda, "Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 403–420, 2000.
- [64] J. D. Schaffer, "Some experiments in machine learning using vector evaluated genetic algorithms," Vanderbilt Univ., Nashville, TN (USA), Tech. Rep., 1985.
- [65] R. Viennet, C. Fonteix, and I. Marc, "New multicriteria optimization method based on the use of a diploid genetic algorithm: Example of an industrial problem," in *European Conference on Artificial Evolution*. Springer, 1995, pp. 120–127.
- [66] A. Osyczka and S. Kundu, "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm," *Structural optimization*, vol. 10, no. 2, pp. 94–99, 1995.

Anexos

Anexo A.1. Validación del algoritmo

Rendimiento y efectividad del algoritmo

Para verificar el rendimiento y efectividad del algoritmo se comparará con diferentes funciones de prueba en las cuales se sabe el resultado teórico y se someterán al mismo algoritmo a utilizar. Estas funciones y resultados funcionan a manera de referencia y le dan validez a los resultados obtenidos para la funciones objetivo que se quieren utilizar.

Se comenzará por comparar los resultados obtenidos con problemas de optimización con dos funciones objetivo y sin tener restricciones. La primera función que se usó a manera de verificación es la llamada función de Kursawe. Esta función está definida según lo presenta la ecuación (5.1). Los resultados teóricos de esta función se pueden observar en la gráfica 5.1 con los cuales se compararán los resultados obtenidos del algoritmo utilizado en este trabajo [62].

$$\text{Minimizar} = \begin{cases} f_1(x) = \prod_{i=1}^2 [-10 * \exp(-0,2 \sqrt{\frac{1}{x_i^2 + x_{i+1}^2}})] \\ f_2(x) = \sum_{i=1}^3 [|x_i|^{0,8} + 5 * \sin(x^3)] \end{cases} \quad \text{Donde} = \begin{cases} -5 < x_i < 5 \\ 1 \leq i \leq 3 \end{cases} \quad (5.1)$$

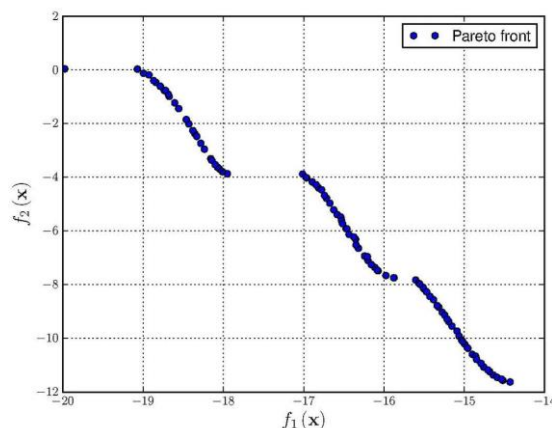


Figura 5.1: Gráfica de la frontera pareto-óptima del problema de Kursawe teórica. (Fuente: [62].)

El resultado para esta función que se obtuvo con el algoritmo propuesto en este trabajo se presenta en la figura 5.2. Como se puede observar al comparar esta figura con la figura 5.1, se obtuvo un resultado concordante tanto en la forma de la gráfica como en los valores obtenidos para la función 1 y 2. Se puede decir que el algoritmo propuesto logró optimizar de manera correcta esta función al tener resultados equivalentes a los teóricos.

Luego de esto se procedió a evaluar la función Poloni de dos objetivos. Esta se puede definir según se muestra en la ecuación (5.2) y las variables utilizada en este problema de optimización se definen en la ecuación eq:Polini2ObjVariables. Los resultado teóricos de esta ecuación se presenta en la gráfica de la figura 5.3 [63].

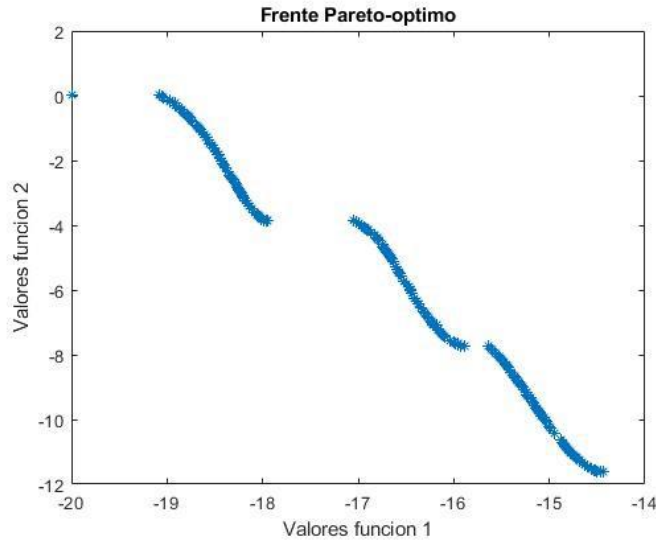


Figura 5.2: Resultados obtenidos con el algoritmo para el problema de optimización Kursawe. (Fuente: el autor.)

$$\text{Minimizar } \begin{cases} f_1(x, y) = [1 + (A_1 - B_1(x, y))^2 + (A_2 - B_2(x, y))]^2 \\ f_2(x, y) = (x + 3)^2 + (y + 1)^2 \end{cases} \quad (5.2)$$

Donde $\pi \leq x, y \leq \pi$

$$\text{Definiendo } \begin{cases} \square A_1 = 0,5\sin(1) - 2\cos(1) + \sin(2) - 1,5\cos(2) \\ \square A_2 = 1,5\sin(1) - \cos(1) + 2\sin(2) - 0,5\cos(2) \\ \square B_1 = 0,5\sin(x) - 2\cos(x) + \sin(y) - 1,5\cos(y) \end{cases} \quad (5.3)$$

$$\square B_2 = 1,5\sin(x) - \cos(x) + 2\sin(y) - 0,5\cos(y)$$

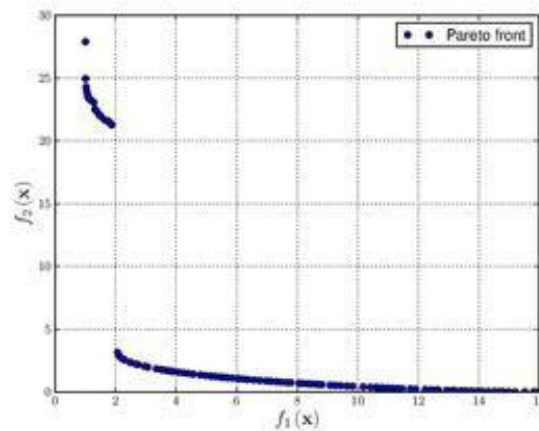


Figura 5.3: Gráfica de la frontera Pareto-óptima del problema Poloni de dos objetivos teórica. (Fuente: [63].)

El resultado obtenido corriendo la función con el algoritmo creado se presenta en la

figura 5.3. Como se puede observar al comparar esta figura con la figura 5.3 se nota como

los resultados son básicamente los mismos. El frente pareto-óptimo encontrado concuerda con los valores y formas del resultado encontrado por los autores y se puede decir que el programa creado pudo optimizar de manera efectiva este problema.

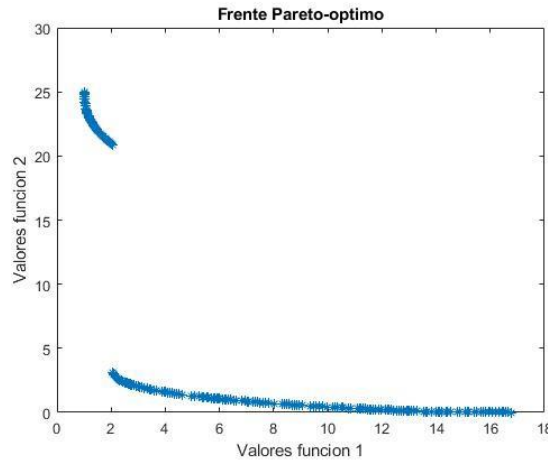


Figura 5.4: Gráfica de la frontera pareto-óptima del problema Poloni de dos objetivos obtenido con el algoritmo creado. (Fuente: el autor).

Posteriormente se procedió a evaluar las funciones de Schaffer número 1 y número 2 que se presentan en las ecuaciones (5.4) y (5.5) respectivamente [64]. Como se observa de la ecuación Schaffer número 2 incluye una función que cambia según el ámbito de la variable x, por lo que además agrega otro nivel de dificultad al algoritmo de optimización.

$$Minimizar = \begin{cases} f_1(x) = x^2 \\ f_2(x) = (x - 2)^2 \end{cases} \quad \text{Donde} = -10^5 \leq x \leq 10^5 \quad (5.4)$$

$$Minimizar = \begin{cases} f_1(x) = \begin{cases} -x, & \text{si } x < 1 \\ x - 2, & \text{si } 1 < x < 3 \\ 4 - x, & \text{si } 3 < x < 4 \\ x - 4, & \text{si } x > 4 \end{cases} \\ f_2(x) = (x - 5)^2 \end{cases} \quad \text{Donde} = -5 \leq x \leq 10 \quad (5.5)$$

Los resultados teórico de estos problemas de optimización se presentan en la figuras 5.5 y 5.6. Se puede observar la clara diferencia en los frentes pareto óptimo y como las condiciones de la función f_1 para el problema de optimización Shaffer No.2 causa que frontera pareto-óptima tenga una discontinuidad que el algoritmo propuesto tendrá que captar.

Las figuras 5.7 y 5.8 muestran los resultados obtenidos utilizando el algoritmo creado. Como se puede observar se obtuvieron equivalentes a los resultado teórico de esta función tanto en la forma de los gráficos como en los valores.

Ahora se probará algoritmo con una función que agrega un grado de complejidad di-

ferente. El problema de optimización de Viennet es uno que en vez de tener dos funciones

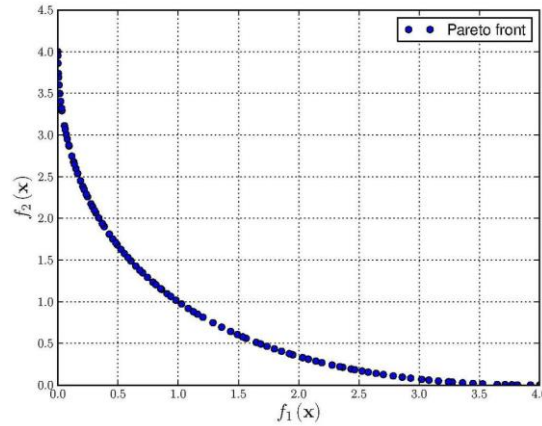


Figura 5.5: Gráfica de la frontera pareto-óptima del problema de la función Schaffer No.1 teórica. (Fuente: [64].)

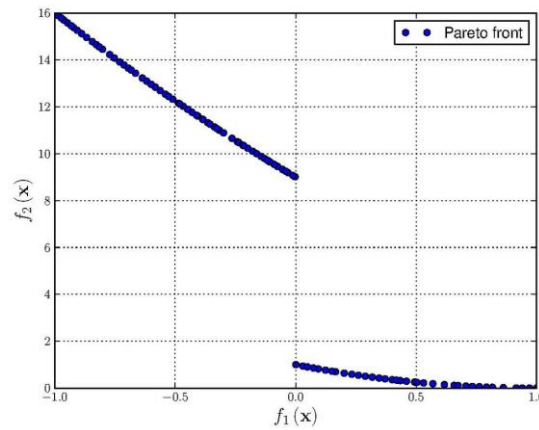


Figura 5.6: Gráfica de la frontera pareto-óptima del problema de la función Schaffer No.2 teórica. (Fuente: [64].)

objetivo, tiene tres. Este no solo crea una frontera pareto-óptima más compleja en tres dimensiones, sino que también se asemeja un mayor medida al problema que se desea optimizar en este trabajo que también posee tres funciones objetivo.

Las funciones a optimizar para este problema de optimización se presentan en la ecuación (5.6). En esta se observa como f_1 , f_2 y f_3 dependen de ambas variables y tienen un ámbito que va desde -3 a 3. El hecho de tener una función adicional hace que el frente pareto-óptimo tenga un eje adicional y sea en tres dimensiones. Las fronteras con mayor cantidad de funciones objetivos no se puede representar gráficamente [65].

$$\begin{aligned}
 \text{Minimizar} = & \begin{cases} f_1(x, y) = 0,5(x^2 + y^2) + \sin(x^2 + y^2) \\ f_2(x) = (x - 5)^2 \\ f_3(x, y) = \frac{1}{x^2 + y^2 + 1} - 1,1 \exp(-(x^2 + y^2)) \end{cases} & \text{Donde} = -3 \leq x, y \leq 3
 \end{aligned}
 \tag{5.6}$$

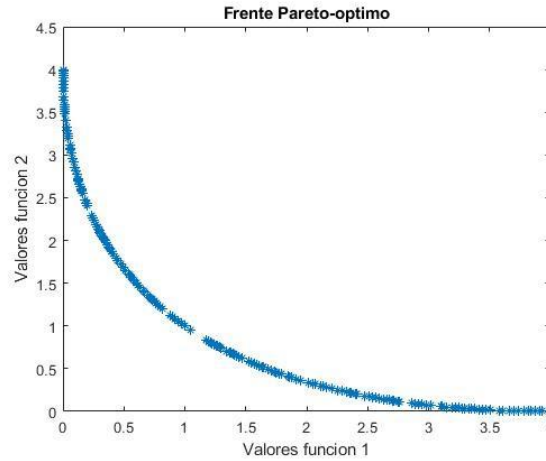


Figura 5.7: Gráfica de la frontera Pareto-óptima del problema de la función Schaffer No.1 obtenida. (Fuente: el autor).

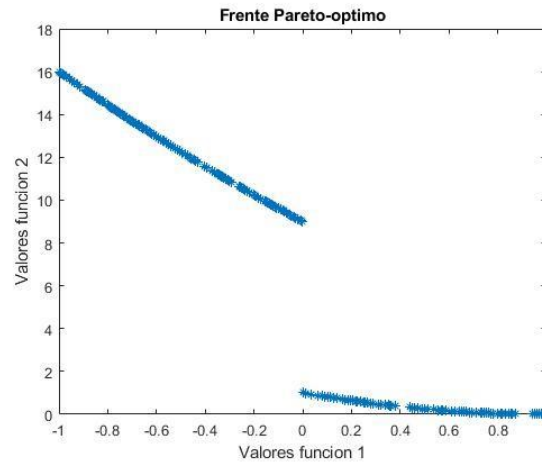


Figura 5.8: Gráfica de la frontera Pareto-óptima del problema de la función Schaffer No.2 obtenida. (Fuente: el autor).

Los resultados teóricos de este problema de optimización se presentan en la gráfica de la figura 5.9 donde se observa la relativa complejidad de esta frontera al compararla con las mostradas anteriormente. Por otra parte, en la figura 5.10 se presenta el resultado obtenido con el algoritmo creado mostrando la gráfica en 3 dimensiones.

Las gráficas de los resultados obtenidos con el algoritmo creado muestran que se obtuvo un resultado equivalente al teórico tanto en la forma de la frontera, como en los valores de las funciones obtenidas. Este resultado ayuda a aportar significativamente a la validez de los resultados obtenidos para las funciones objetivo deseadas ya que se asemeja en gran medida al problema que se desea optimizar.

Hasta el momento se había probado el algoritmo únicamente con problemas de optimización sin restricciones. Ahora se procederá a realizar la verificación del algoritmo con algunas funciones de prueba que involucren restricciones. Las restricciones, según se detalló en el marco metodológico se implementarán utilizando un método de penalización estática.

El primer problema de optimización con restricción es el llamado "Constr-Ex". Este

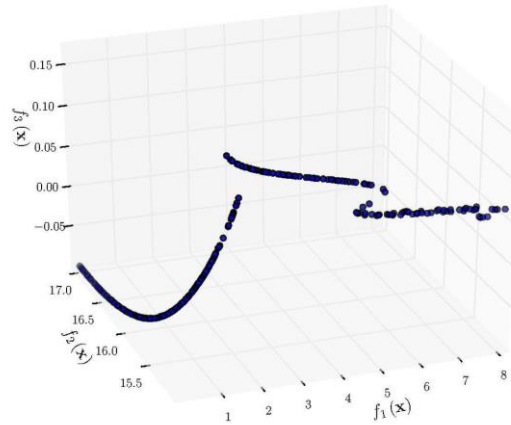


Figura 5.9: Gráfica de la frontera pareto-óptima del problema de Viennet teórica. (Fuente: [65].)

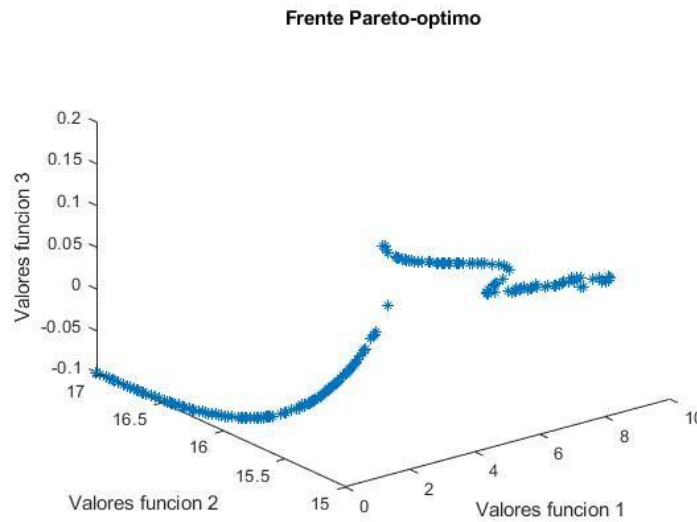


Figura 5.10: Gráfica de la frontera pareto-óptima del problema de Viennet práctica. (Fuente: el autor).

problema se define en las funciones presentadas en la ecuación (5.7) en donde las funciones están sujetas a las condiciones g_1 y g_2 . Además se tiene que las variables x y y tienen un ámbito de variables según se presenta en la ecuación (5.8).

$$\text{Minimizar} = \begin{cases} f_1(x) = x \\ f_2(x, y) = \frac{1+y}{x} \end{cases} \quad \text{Sujeto a} = \begin{cases} g_1(x, y) = y + 9x \geq 6 \\ g_2(x, y) = -y + 9 \geq 1 \end{cases} \quad (5.7)$$

$$\text{ámbito de variables} = \begin{cases} 0,1 \leq x \leq 1 \\ 0 \leq y \leq 5 \end{cases} \quad (5.8)$$

Los resultados teóricos de este problema se presentan en la figura 5.11 mientras que los resultados obtenidos con el algoritmo se presentan en la figura 5.12. Se puede observar como el algoritmo propuesto logra obtener una frontera pareto-óptima más definida y continua que la que se presenta en la teórica, además de obtener la misma forma y ámbito de valores que la teórica. Basado en los resultado obtenidos el algoritmo logró optimizar de manera exitosa este problema con restricción.

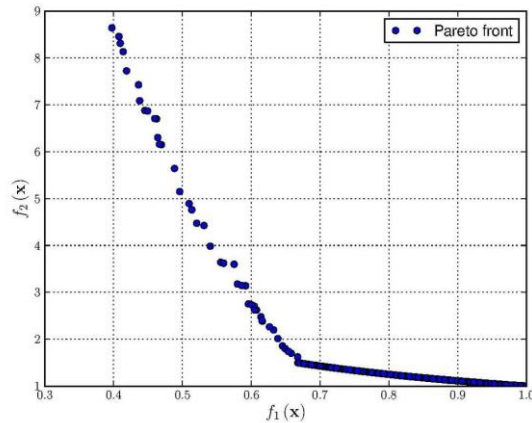


Figura 5.11: Gráfica de la frontera pareto-óptima del problema Constr-Ex teórica. (Fuente: [39].)

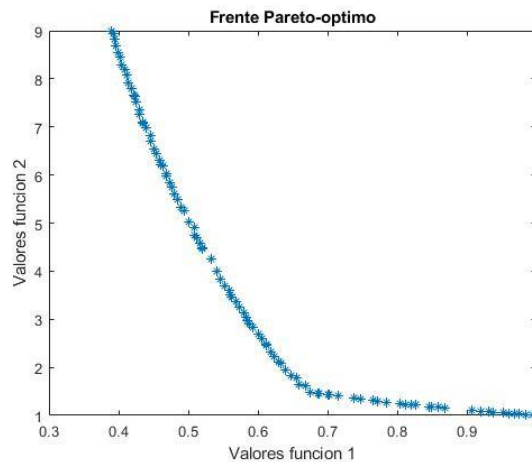


Figura 5.12: Gráfica de la frontera pareto-óptima del problema Constr-Ex obtenida con el algoritmo. (Fuente: el autor).

Finalmente se comprobó el funcionamiento adecuado del algoritmo comparando el problema de optimización con uno que tiene múltiples restricciones y variables. El problema de la función de Oszycza y Kundu posee seis diferentes restricciones y seis variables de optimización según se presenta en la ecuación (5.9). Este problema al tener tantas restricciones crea una frontera compleja que pone a prueba el algoritmo al determinar si encontrará o no la geometría y valores de la frontera [66].

$$\begin{aligned}
 \text{Minimizar } &= \begin{cases} f_1(x) = -25(x_1 - 2)^2 - (x_2 - 2)^2 - (x_3 - 1)^2 - (x_4 - 4)^2 - (x_5 - 1)^2 \\ f(x) = -25 \sum_{i=1}^5 x_i^2 \end{cases} \\
 \text{Sujeto a } &= \begin{cases} g_1(x) = x_1 + x_2 - 2 \leq 0 \\ g_2(x) = 6 - x_1 - x_2 \leq 0 \\ g_3(x) = 2 - x_2 + x_1 \leq 0 \\ g_4(x) = 2 - x_1 - 3x_2 \leq 0 \\ g_5(x) = 4 - (x_3 - 3)^2 - x_4 \leq 0 \\ g_6(x) = (x_5 + 3)^2 + x_6 - 4 \leq 0 \end{cases} \quad \text{Donde } = \begin{cases} 0 \geq x_1, x_2, x_6 \geq 10 \\ 1 \geq x_3, x_5 \geq 5 \\ 0 \leq x_4 \leq 6 \end{cases}
 \end{aligned} \tag{5.9}$$

Los resultados teóricos para el problema se presentan en la figura 5.13 mientras que los resultados obtenidos con el algoritmo de optimización propuesto se presentan en la figura 5.14. Como se observa en la gráfica obtenida con el algoritmo al compararla con la teórica se puede notar que se obtuvieron los mismos ámbitos de valores y forma de la gráfica. Se concluye a partir de esto que algoritmo logró optimizar efectivamente el problema de Oszczka y Kundu.

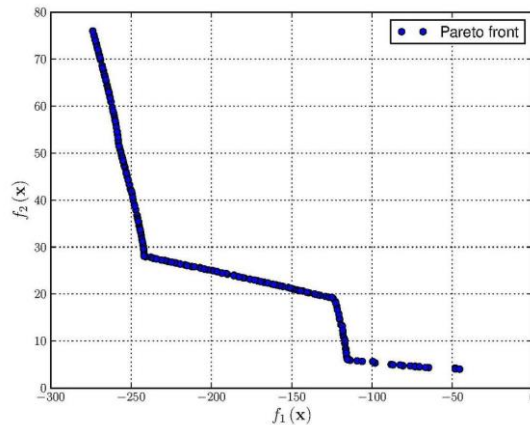


Figura 5.13: Gráfica de la frontera pareto-óptima del problema Oszczka y Kundu teórica. (Fuente: [66].)

Una vez evaluado el algoritmo bajo diferentes casos y complejidades se comprueba la validez y efectividad del mismo. Durante las pruebas realizadas se notó también que en los casos evaluados la convergencia se daba con mayor rapidez al tener una población relativamente pequeña con un número de generaciones alta, mientras que un número bajo de generaciones causaba una inestabilidad en la convergencia del algoritmo.

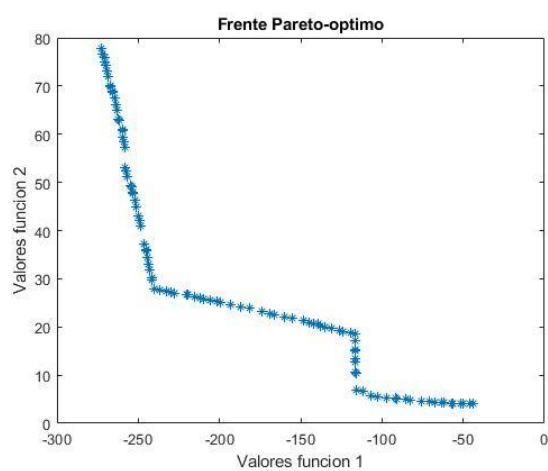


Figura 5.14: Gráfica de la frontera Pareto-óptima del problema Osyczka y Kundu obtenida con el algoritmo (Fuente: el autor).

Anexo A.2. Guía para correr el método numérico

El contenido de este anexo pretende guiar al lector con la finalidad de poder entender donde se definen las variables en el código y cómo utilizarlo. A continuación se definen los pasos a seguir y sus valores recomendados:

1. Seleccionar la cantidad de generaciones y el tamaño de la población a utilizar definiendo la variable de cantidad de generaciones en el archivo de “CorrerAlgoritmo”.
2. Definir los límites de las variables a utilizar para los valores de d , D y n_a en el archivo “CorrerAlgoritmo”. Los valores inferiores se deben incluir en el vector llamado “ValoresInferiores” en el orden $(d_{Min}, D_{Min}, n_a_{Min})$, mientras que los superiores en el “ValoresSuperiores” en el orden $(d_{Max}, D_{Max}, n_a_{Max})$
3. En la función “EvaluarObjetivo” se definen las siguientes variables:
 - Su el esfuerzo último de material en Pascales
 - $K_{inferior}$ y $K_{superior}$ que es el límite de la constante del resorte inferior y superior en N/m
 - p el paso del resorte en mm
 - F el vector de las fuerzas de fatiga en N
 - t el vector de los tiempos correspondientes a los valores de las fuerzas en segundos
 - $f_{MaterialFatiga}$ el valor adimensional de penalización para calcular la constante b de las gráficas S vs N
 - Variacion que es la variación de las variables dimensionales para el diseño robusto
 - VariaciónSu que es la variación del esfuerzo último para el diseño robusto
 - $mPiston$ que es la masa de la cabeza del pistón pegada al resorte en kg
4. Posteriormente se procede a definir las restricciones del modelo como $Restricc_i$. En el código del anexo A.1 se presentan las restricciones comentadas a lo largo del documento.
5. Finalmente se define la variable $etaCombinacion$ y $etaMutacion$ que son las constantes para la combinación y mutación polinomial
6. Una vez definidas todas las variables se ejecuta la función “CorrerAlgoritmo”
7. Una vez terminadas las iteraciones se tiene como resultado la figura de la frontera pareto-óptima y un archivo llamado “Solución” que se crea en la carpeta donde se ejecutó la función.
8. El archivo “Solución” contiene los valores de cada individuo de la frontera pareto-óptima. Cada fila corresponde a un individuo diferente, mientras que la primera columna es el valor de d en metros, la segunda el valor de D en metros y la tercera el número de espiras. La cuarta columna presenta la masa del individuo en kg, la quinta el factor de fatiga de Miner y la sexta el factor de diseño por robustez. La última columna presenta el valor de la distancia de hacinamiento del individuo.

Anexo A.3. Artículo técnico

A continuación se presenta el manuscrito del resumen extendido presentado en la tercera Jornada de Investigación de la Universidad de Costa Rica del año 2021. El manuscrito representa una ponencia de investigación realizada que contiene un resumen de la metodología así como una representación de los resultados parciales obtenidos.

Optimización multi-objetivo del diseño de resortes considerando diseño robusto

Multi-objective optimization of springs considering robust design

Erick Van Patten Rivera

*Estudiante de Posgrado, Escuela de Ingeniería Mecánica
Universidad de Costa Rica
erick.vanpatten@ucr.ac.cr*

Denis Abarca Quesada

*Profesor, Escuela de Ingeniería Mecánica
Universidad de Costa Rica
denis.abarca@ucr.ac.cr*

Juan Gabriel Monge Gapper

*Profesor, Escuela de Ingeniería Mecánica
Universidad de Costa Rica
juan.mongegapper@ucr.ac.cr*

Palabras clave: *Algoritmo Genético, Análisis contra fatiga, Diseño robusto Optimización multi-objetivo, Resorte Helicoidal*

Keywords: *Fatigue Analysis, Genetic Algorithm, Helical springs, Multiobjective optimization, Robust design*

I. INTRODUCCIÓN

La industria de hidrocarburos es una de las más relevantes económicamente a nivel mundial debido a lo indispensable que es para el transporte, proceso y generación de energía [1]. En esta industria la regulación de caudal y presión en las tuberías es un factor crítico y que es comúnmente desempeñado por válvulas de control[2]. El correcto funcionamiento de esta válvula es dependiente de un resorte helicoidal, por lo que propiciar un diseño robusto y resistente a la fatiga es vital para el funcionamiento y seguridad de la válvula. Además de esto mantener al mínimo la masa y costo de producción del resorte tiene ventajas económicas relevantes debido al alto volumen y costo de producción de estos resortes.

Por lo tanto, durante esta investigación se pretende implementar un algoritmo genético que optimice la robustez del diseño, la resistencia a la fatiga y minimice la masa. Esto con la finalidad de no solo asegurar la operación segura y estable del resorte, sino también que se está logrando el diseño más económicamente atractivo al mismo tiempo asegurando tener la menor masa posible que cumpla con las condiciones requeridas.

II. METODOLOGÍA

El proceso de optimización se realiza utilizando el algoritmo evolutivo NSGA II. El flujo específico del algoritmo que se pretende utilizar se presenta de manera simplificada en la figura 1. El algoritmo genético NSGA II es un algoritmo de optimización estocástica que simula la evolución en la naturaleza para encontrar los individuos de la frontera Pareto-óptima de un problema [3]. Los puntos de diseño que se encuentran en esta frontera poseen los mejores valores para cada una de las funciones de optimización con respecto a los otros puntos del espacio solución [3].



Fig. 1: Imagen con el diagrama de flujo simplificado del programa (Fuente: El autor).

El caso en estudio sobre los resortes helicoidales se tiene que las funciones objetivo utilizadas son: la minimización de la masa, la maximizar la resistencia contra fatiga y la robustez del diseño. La primera función consiste en minimizar la masa del resorte. Para esto básicamente se utiliza la ecuación del volumen de un resorte multiplicada por la densidad del material, por lo que depende de los diámetros del resorte, del número de espiras activas y el diámetro del alambre.

La segunda función objetivo es maximizar la resistencia a la fatiga, esta función está relacionada con el método de conteo de ciclos y acumulación de daño de Miner. El factor de acumulación de daño de Miner representa la fracción de la vida útil del resorte consumida por uno de los ciclos presentados. Para esta función objetivo se realiza un sistema de masa-amortiguador-resorte con suficientes grados de libertad determinados de tal forma que la frecuencia máxima natural máxima del sistema discreto sea mayor a la frecuencia máxima principal de la fuerza externa al realizarle una transformación de Fourier [4].

Finalmente la tercera función objetivo utiliza el método Taguchi para asegurar un diseño robusto [5]. Este enfoque basado en la robustez del diseño busca minimizar la varianza del resultado obtenido en las dos funciones objetivo anteriores del resorte ante la posible variación no controlable del diámetro del alambre y el diámetro del resorte. Por lo tanto, busca minimizar el impacto en los resultados obtenidos ante un posible cambio de las variables de entrada.

III. RESULTADOS

La figura 2.a) muestra uno de los resultados obtenidos al introducir una fuerza aleatoria al sistema y poner un rango de variables para los diámetros del resorte de 3 cm a 8 cm, diámetro del alambre 4 mm a 8 mm y de espiras activas de 7 a 11. Por otra parte, en la figura 2.b) se muestran los resultados obtenidos para las mismas condiciones pero con una fuerza sinusoidal.

Se puede notar como la fuerza de excitación tiene un alto impacto en los resultados finales obtenidos y como el algoritmo permite al diseñador encontrar los puntos en la frontera óptima. Los puntos ubicados en esta frontera tendrán la mejor relación de resistencia a la fatiga y robustez del diseño con la menor masa posible, por lo que facilita en gran medida la elección de las dimensiones del resorte al saber que las combinaciones de variables extraídas del algoritmo serán las óptimas.

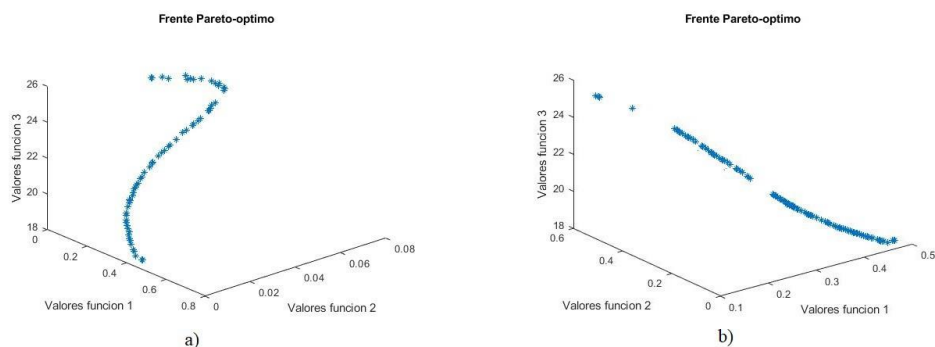


Fig. 2: Resultados para un resorte ante una fuerza aleatoria (Fuente: El autor).

REFERENCIAS

- [1] L. Wood, *Control Valves Industry Report*. oct. de 2019.
- [2] E. Electric, *Daniel, Series 700 Control Valve, 6 IN (DN150)*. Emerson Electric, 2021.
- [3] K. Deb y D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. USA: John Wiley Sons, Inc., 2001.
- [4] R. Budynas, J. Shigley y J. Nisbett, *Shigley's Mechanical Engineering Design*. McGraw-Hill, 2008.
- [5] D. Montgomery, *Design and Analysis of Experiments*, ép. Student solutions manual. John Wiley & Sons, 2008.

*Este documento fue elaborado bajo la licencia abierta Creative Commons en su modalidad
Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0)*

Anexo A.4. Código del algoritmo

A continuación se presenta el código fuente de los subprogramas creados:

```
----- Subprograma 1 -----
%Universidad de Costa Rica
%Posgrado de Ingeniería Mecánica
%Maestría Profesional En Ingeniería Mecánica
%Nombre de archivo:"CorrePrograma.m"
%Descripción básica: El propósito de este archivo es ejecutar el
    algoritmo
%NSGA II y establecer los parámetros de entrada: Tamaño de
    población, número de
%generaciones, número de funciones objetivo, número de variables
    de
%decisión y los rangos de estas variables.
clear
clc
TamañoPoblacion = 70;%Establece tamaño de la población, unidades
    : Cantidad adimensional
NumDeGeneraciones = 150; %Establece número de generaciones,
    unidades: Cantidad adimensional
numDeObj = 3; %Establece número funciones objetivo, unidades:
    Cantidad adimensional
numVariablesDecision = 3; %Establece número de variables de
    decisión, unidades: Cantidad adimensional
minRango = [0.004 0.08 7]; %Establece rangos mínimos de
    variables, unidades: Cantidad adimensional
maxRango = [0.0013 0.015 11]; %Establece rangos máximos de
    variables, unidades: Cantidad adimensional
nsga2 (TamañoPoblacion , NumDeGeneraciones , numDeObj,
    numVariablesDecision ,minRango,maxRango);%Corre el programa de
    optimización
```

```
----- Subprograma 2 -----
%Universidad de Costa Rica
%Posgrado de Ingeniería Mecánica
%Maestría Profesional En Ingeniería Mecánica
%Nombre de archivo:"evalua_objetivos.m"
%Descripción básica: El propósito de esta función es evaluar
    para cada uno
%de los miembros de la población los valores obtenidos para la
    función
%objetivo y asignarle un valor. Las variables de entrada son x
    que es un
%vector con las entradas de las variables de decisión de un
    individuo, M
%que es la cantidad de funciones objetivos y V la cantidad de
    variables de
%discreto. Devuelve el vector f que contiene los valores de la
    función
```

%objetivo

```
function f = evalua_objetivos(x, M, V)
```

```
f = [];
```

```
%Definición de variables
```

```
d=x(1); %Diametro de alambre en m
```

```
D=x(2); %Diametro de resorte en m
```

```
na=x(3); %Número de espiras activas del resorte
```

```
Lmin=0.10; %Longitud del Resorte en m
```

```
Lmax=0.20; %Longitud del resorte en m
```

```
na=x(3); %Numero de espiras activas?
```

```
Su=300e6; %Es fuerza Ultimo Del Material En Pascales
```

```
Se=0.65*S_u; %Resistencia A Fatiga Con Ciclos Infinitos
```

```
fMaterialFatiga=0.9; %Penalización De Es fuerza Por Fatiga
```

```
kmin=8492; %Constante del resorte mínima en N/m
```

```
kmax=8492; %Constante del resorte mínima en N/m
```

```
m=(2*pi^2*(na+2)*(D/2)*(d/2)^2)*7850; %masa del resorte en Kg
```

```
Fc=100; %frecuencia de la honda
```

```
p=15; %paso del resorte en mm
```

```
long=linspace(0,1,200);
```

```
F=50*sin(20*pi*2*long); %Patron de fuerzas para la fatiga
```

```
t=linspace(0,100,length(F)); %Vector De Tiempo Para La Fuerza Externa
```

```
a=(fMaterialFatiga*S_u)^2/Se; % Valor del material a para la fatiga, adimensional
```

```
b=-1/3*log(fMaterialFatiga*S_u/Se); %Valor del material b para la fatiga, adimensional
```

```
C=D/d; %Índice del resorte
```

```
Variacion=0.1; %Variación de la variables para el diseño robusto
```

```
%Define restricciones
```

```
Const1=0; %Constante para penalización fija
```

```
Restricc1 = C>=4 || C<=12; %Restricción del índice del resorte, adimensional
```

```
Restricc2= p*na<pi*D/alfa*(2(E-G)/(2G+E))^2; %Restricción de longitud libre del resorte, metros
```

```
%Verifica si alguna de las restricciones no se cumple
```

```
Restricciones= Restricc1&&Restricc2&&Restricc3;
```

```
if not(Restricciones)
```

```
    Const1=1; %Asigna valor de 1 a la penalización si alguna restricción no se cumple
```

```
end
```

```
%Calcula función objetivo 1
```

```
f(1)=m+100000*Const1; %Masa del resorte en Kg
```

```
%Calcula función objetivo 2
```

```
[FuerzasFatiga]=ModeloDiscretizadoAFuerzas(F,t,k,m);%Calcula las
    fuerzas de fatiga para cada elemento discreto en N
FuerzasFatiga=FuerzasFatiga(:,all(~isnan(FuerzasFatiga)));%
    Verifica que el vector este lleno completamente
Esfuerzos=FuerzaAEsfuerzoResorte(FuerzasFatiga,D,d);%Calcula
    esfuerzos para todos los elementos discretos de la fatiga
f(2)=SubrutinaFatigaNoArmonica(Esfuerzos,a,b)+100000*Const1; %
    Calcula el factor critico de miner para las fuerzas
    calculadas

%Verificación de restriccion de miner.
if f(2)>1
    Const1=1;%Asigna valor de 1 a la penalización si alguna
        restriccion no se cumple
    f(2)=f(2)+100000*Const1;
end

% Función objetivo 3

EsfuerzosVardi=FuerzaAEsfuerzoResorte(FuerzasFatiga,D,(Variacion
    +1)*d); %Calcula esfuerzo con la variación de d
fAlternodi=SubrutinaFatigaNoArmonica(EsfuerzosVardi,a,b)+100000*
    Const1; %Calcula factor de miner para variación de d
EsfuerzosVarDi=FuerzaAEsfuerzoResorte(FuerzasFatiga,(Variacion
    +1)*D,d); %Calcula esfuerzo con la variación de D
fAlternodi=SubrutinaFatigaNoArmonica(EsfuerzosVarDi,a,b)+100000*
    Const1; %Calcula factor de miner para variación de D
mdi=(2*pi^2*(na+2)*(D/2)*((Variacion+1)*d/2)^2)*7850; %Calcula
    variación de masa con variación de d
mDi=(2*pi^2*(na+2)*((Variacion+1)*D/2)*(d/2)^2)*7850; %Calcula
    variación de masa con variación de D
Sens1=((m-mDi)/m)/Variacion; %Factor de sensibilidad de masa
    para cambio en d
Sens2=((m-mDi)/m)/Variacion; %Factor de sensibilidad de masa
    para cambio en D
Sens3=((f(2)-fAlternodi)/f(2))/(Variacion); %Factor de
    sensibilidad de fatiga para cambio en d
Sens4=((f(2)-fAlternodi)/f(2))/(Variacion); %Factor de
    sensibilidad de fatiga para cambio en D
f(3)=sqrt(Sens1^2+Sens2^2+Sens3^2+Sens4^2)+100000*Const1; %Funci
    ón objetivo 3, suma de cuadrados
```

----- Subprograma 3 -----

```
%Universidad de Costa Rica
%Posgrado de Ingeniería Mecánica
%Maestría Profesional En Ingeniería Mecánica
%Nombre de archivo:"nsga2.m"
%Descripción básica: es ejecutar el algoritmo de nsga II, esta
    es la función que llamará a los otros subprogramas. La
    variable pobes el tamaño de la población y genla
```

```

    cantidad de gneeraciones
function nsga2(pop,gen,numObjetivos,numVariables,minRango,
    maxRango)

M=numObjetivos; %Definimos M como el numero de funciones
    objetivo
V=numVariables; %Definimos V como el número de variables
minrango=minRango; %Vector con los rangos minimos de las
    variables
maxrango=maxRango; %Vector con el rango máximo de las variables

cromosoma= inicializa_variables(pob, M, V, min rango , max rango)
    ;

cromosoma= ordenamiento.no_dominante(cromosoma, M, V); %Ordena
    la población con el el algoritmo de ordenamiento

for i = 1 : gen %Comienza el proceso del algoritmo evolutivo:
    selección, recombinación, mutación
    grupoSeleccion = round(pob/2); %selecciona grupo del torneo
    torneo = 2; %Selecciona tama o de torneo

    cromosoma padre= seleccion torneo(cromosoma, grupoSeleccion ,
        torneo); %Se hace el torneo binario para la seleccion

    %Perfrom crossover and Mutation operator
    %The original NSGA-II algorithm uses Simulated Binary
        Crossover (SBX) and
    %Polynomial mutation. Crossover probability pc = 0.9 and
        mutation
    % probability is pm = 1/n, where n is the number of decision
        variables.
    %Both real-coded GA and binary-coded GA are implemented in
        the original
    %algorithm, while in this program only the real-coded GA is
        considered.
    %The distribution indeices for crossover and mutation
        operators as mu = 20
    %and mum = 20 respectively.
    mu = 20; % Factor eta para recombinación
    mum = 20; % Factor eta para mutación
    cromosoma_hijos= ...
        operador_genetico(cromosoma padre, ...
            M, V, mu, mum, min rango , max rango);

    [main_pop,temp] = size(cromosoma); %población temporal con
        el doble de individuos

```

```

[hijos_pob ,temp] = size(cromosoma hijos); %población
temporal con el doble de individuos

clear temp
cromosoma_intermedio(1:main_pop,:) = cromosoma;
cromosoma_intermedio(pob_princi+ 1 : pob_princi+ hijos_pob ,1
: M+V) = ...
cromosoma_hijos;

%Non-domination-sort of intermediate population
%The intermediate population is sorted again based on non-
domination sort
%before the replacement operator is performed on the
intermediate
%population.
cromosoma_intermedio= ...
ordenamiento_no_dominante(cromosoma_intermedio , M, V); %
ordenamiento no dominante de poblacion temporal
%Perform Selection
%Once the intermediate population is sorted only the best
solution is
%selected based on it rank and crowding distancia. Each
frente is filled in
%ascending order until the addition of population size is
reached. The
%last frente is included in the population based on the
individuos with
%least crowding distancia
cromosoma= remplazar_cromosoma(cromosoma_intermedio , M, V,
pop);

save solucion.txt cromosoma-ASCII %guarda resultados en solución
%Graficadores de resultados
if M == 2
figure(1);
plot(cromosoma(:,V + 1),cromosoma(:,V + 2),'*');
title('Frente Pareto-optimo');
xlabel('Valores funcion 1');
ylabel('Valores funcion 2');
elseif M ==3
figure(1);
% plot3(Funcion1,Funcion2,Funcion3,'*')% nueva linea
plot3(cromosoma(:,V + 1),cromosoma(:,V + 2),cromosoma(:,V +
3),'*');% Vieja linea
title('Frente Pareto-optimo');
xlabel('Valores funcion 1');
ylabel('Valores funcion 2');
zlabel('Valores funcion 3');
end

```

----- Subprograma 4 -----

```

%Universidad de Costa Rica
%Posgrado de Ingeniería Mecánica
%Maestría Profesional En Ingeniería Mecánica
%Nombre de archivo:” ordenamiento_no_dominante .m”
%Descripción básica: El propósito de este subprograma es
    realizar el ordenamiento no dominante de las variables. Los
    individuos del primer frente tienen un rango de 1, el Segundo
    un rango de 2 y así consecutivamente. Además de esto la
    función calcula la distancia de hacinamiento de los
    individuos.
function f = ordenamiento_no_dominante(x, M, V)

[N, m] = size(x);
clear m

frente= 1; %inicializa primer frente

F(frente).f = [];
individuo = [];

for i = 1 : N %empieza el ordenamiento
    for j = 1 : N
        dom_inferior= 0; %conteo de variables que dominan el
            individuo
        dom_igual= 0; %conteo de variables que son iguales que el
            individuo
        dom_superior= 0; %conteo de variables que son dominadas
            por el individuo
        %se comienza por encontrar el primer frente
        for k = 1 : M
            if (x(i,V + k) < x(j,V + k))
                dom_inferior= dom_inferior+ 1;
            elseif (x(i,V + k) == x(j,V + k))
                dom_igual= dom_igual+ 1;
            else
                dom_superior= dom_superior+ 1;
            end
        end
        end

        if dom_inferior== 0 && dom_igual~= M
            individuo(i).n = individuo(i).n + 1;
        elseif dom_superior== 0 && dom_igual~= M
            individuo(i).p = [individuo(i).p j];
        end
    end

    if individuo(i).n == 0
        x(i,M + V + 1) = 1;
        F(frente).f = [F(frente).f i];
    end
end

```

end

```

%Se buscan los siguientes frentes
while ~isempty(F(frente).f)
    Q = [];
    for i = 1 : length(F(frente).f)
        if ~isempty(individuo(F(frente).f(i)).p)
            for j = 1 : length(individuo(F(frente).f(i)).p)
                individuo(individuo(F(frente).f(i)).p(j)).n =
                    ...
                    individuo(individuo(F(frente).f(i)).p(j)
                        ).n - 1;
                if individuo(individuo(F(frente).f(i)).p
                    (j)).n == 0
                    x(individuo(F(frente).f(i)).p(j),M + V +
                        1) = ...
                    frente+ 1;
                Q = [Q individuo(F(frente).f(i)).p(j)];
            end
        end
    end
end

frente= frente+ 1;
F(frente).f = Q;
end

[indice_frentes temporal] = sort(x(:,M + V + 1));
%Se ordenan los individuos basado en el n;umero de frente
for i = 1 : length(indice_frentes)
    ordenamiento_basado_en_frentes(i,:) = x(indice_frentes(i),:)
    ;
end
indice_actual = 0;

%Se calcula la distancia de hacinamiento en cada individuo del
frente
for frente= 1 : (length(F) - 1)
    % objetivo = [];
    distancia = 0;
    y = [];
    indice_anterior= indice_actual + 1;
    for i = 1 : length(F(frente).f)
        y(i,:) = ordenamiento_basado_en_frentes(indice_actual +
            i,:);
    end
    indice_actual = indice_actual + i;
    %Se ordenan los individuos según los valores de idoneidad
    de las funciones bojetivo
    sorteo_basado_en_objetivo = [];
    for i = 1 : M
        [sorteo_basado_en_objetivo , indice_de_objetivos] = ...
            sort(y(:,V + i));
    end
end

```



```

sorteo_basado_en_objetivo = [];
for j = 1 : length(indice_de_objetivos)
    sorteo_basado_en_objetivo(j,:) = y(
        indice_de_objetivos(j),:);
end
f_max = ...
    sorteo_basado_en_objetivo(length(indice_de_objetivos)
        ), V + i);
f_min = sorteo_basado_en_objetivo(1, V + i);
y(indice_de_objetivos(length(indice_de_objetivos)),M + V
    + 1 + i)...
    = Inf;
y(indice_de_objetivos(1),M + V + 1 + i) = Inf;
for j = 2 : length(indice_de_objetivos) - 1
    sig_obj = sorteo_basado_en_objetivo(j + 1,V + i);
    previous_obj = sorteo_basado_en_objetivo(j - 1,V +
        i);
    if (f_max - f_min == 0)
        y(indice_de_objetivos(j),M + V + 1 + i) = Inf;
    else
        y(indice_de_objetivos(j),M + V + 1 + i) = ...
            (sig_obj - previous_obj)/(f_max - f_min);
    end
end
end

distancia = [];
distancia(:,1) = zeros(length(F(frente).f),1);
for i = 1 : M
    distancia(:,1) = distancia(:,1) + y(:,M + V + 1 + i);
end
y(:,M + V + 2) = distancia;
y = y(:,1 : M + V + 2);
z(indice_anterior:indice_actual,:) = y;
end

f = z();

```

----- Subprograma 5 -----

```

%Universidad de Costa Rica
%Posgrado de Ingeniería Mecánica
%Maestría Profesional En Ingeniería Mecánica
%Nombre de archivo:"FuerzasFatiga.m"
%Descripción básica: El propósito de este calcular las fuerzas
    de fatiga basado en el modelo dinámico

function [FuerzasFatiga]=ModeloDiscretizadoAFuerzas(Fuerz, tiempo
    ,k,m)

carga=Fuerz;

```

```

t = linspace(0,max(tiempo)-min(tiempo),max(size(carga))); %
    Espacio temporal
kResorte = k; %Constante del resorte en N/m
mResorte = m; %masaDelSistemaEnKg
GradosLibert=2; % Inicia grados de libertad
FrecMaxNat=0;
Modos=[];
matrizModal=[];
QSub=[];
Sistema .m = 0;
Sistema .k = 0;
Sistema .c = 0;
%Calculo de max frecuencia natural del sistema
dt=(t(2)-t(1));
Fs=1/dt;
TamanoCarga=max(size(carga));
Fourier=fft(carga); %Transformada de fourier para calcular
    maximas frecuencias de la fuerza externa
P2=abs(Fourier/TamanoCarga);
P1=P2(1:TamanoCarga/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f=Fs*(0:(TamanoCarga/2))/TamanoCarga;%

for i=1:max(size(P1))%
    if abs(P1(i))/max(P1)<0.5
        P1(i)=0;
    else
        freqNaturalesRelevantes(i)=f(i);
    end
end

FrecuenciaF=1/max(freqNaturalesRelevantes); %Frecuencias de las
    fuerzas a esas magnitudes en HZ

while FrecMaxNat<max(FrecuenciaF)% Calcula cantidad de grados de
    libertad requeridos
matrizMasa=[];
matrizRigidez=[];
matrizModal=[];
autovalores=[];
frecuenciasNaturales=[];

GradosLibert=GradosLibert+1;
mDis = mResorte/ GradosLibert;
kDis = GradosLibert * kResorte;

matrizMasa = mDis*eye(GradosLibert);

```

```

matrizRigidez = zeros(GradosLibert , GradosLibert);

matrizRigidez(1,[1 2]) = [2*kDis -kDis];
matrizRigidez(end, [end-1 end]) = [-kDis kDis];

for i=2:GradosLibert-1
    matrizRigidez(i,[i-1 i i+1]) = [-kDis 2*kDis -kDis];
end

[matrizModal,autovalores] = eig(matrizRigidez, matrizMasa);
frecuenciasNaturales = max((sqrt(autovalores)));
FrecMaxNat= max(frecuenciasNaturales);

end

for i=1:GradosLibert %Este for normaliza los modos dividiendo
    entre el valor mas alto
        matrizModal(:,i)=matrizModal(:,i)/max(abs(matrizModal(:,i)))
    ;
end
%Separa el Sistema de ecuaciones diferenciales para poder ser
    calculados
matrizMasaDiag = matrizModal'*matrizMasa*matrizModal;
matrizRigidezDiag = matrizModal'*matrizRigidez*matrizModal;
wn = max(sqrt(matrizRigidezDiag/matrizMasaDiag));
mDesacoplada = max(matrizMasaDiag);
kDesacoplada = max(matrizRigidezDiag);
zeta = 0.05;
cDesacoplada = 2*zeta*mDesacoplada.*wn;
Funitario=zeros(GradosLibert ,1);
Funitario(end,1)=1;
F = Funitario*carga;
Q = matrizModal'*F;

for i=1:GradosLibert %Este for normaliza los modos dividiendo
    entre el valor mas alto
        QSub=Q(i,:);
        Sistema.m = mDesacoplada(1);
        Sistema.k = kDesacoplada(1);
        Sistema.c = cDesacoplada(1);
        [q,tt] = resolverSistemaCargaAleatoria(Sistema,t,QSub);
        qSub(i,:)=q;
    end

xx=zeros(GradosLibert ,max(size(qSub)));
for i=1:GradosLibert
    xx=xx+(qSub(i,:).*matrizModal(:,i));
end

```

```

FuerzasFatiga=zeros(GradosLibert ,max( size (qSub)));

% Calcula las fuerzas de fatiga
FuerzasFatiga (1,:) = kDis * (xx(2,:) - xx(1,:)) - kDis * (xx(1,:)); %
    Fuerzas en primer m
for i=2:GradosLibert -1
    FuerzasFatiga (i,:) = kDis * (xx(i+1,:) - xx(i,:)) - kDis * (xx(i,:) - xx
        (i-1,:)); % Fuerzas en iesimo m
end
Finterpoladas=interp1(t,carga ,tt , 'PCHIP');
FuerzasFatiga(end,:) = Finterpoladas - kDis * (xx(end,:) - xx(end-1,:))
    ; % Fuerzas en m final

```

----- Subprograma 6 -----

```

%Universidad de Costa Rica
%Posgrado de Ingeniería Mecánica
%Maestría Profesional En Ingeniería Mecánica
%Nombre de archivo: "Inicializa_variables.m"
%Descripción básica: El propósito de este subprograma es
    incilizar las variables para obtener la primera generación.

function f = inicializa_variables(N, M, V, min rango , max rango)

```

THE

```

% POSSIBILITY OF SUCH DAMAGE.

```

```

min = min rango ;
max = max rango ;
K = M + V;

```

```

%Inicializa cada cromosoma

```

```

for i = 1 : N
    for j = 1 : V

```

```

        f(i,j) = min(j) + (max(j) - min(j))*rand(1);

```

```

%Agrega el valor de cada función objetivo a la matriz para
    asociarla a los individuos

```

```

        f(i,V + 1: K) = evalua_objetivos(f(i,:), M, V);

```

```

    end

```

----- Subprograma 7 -----

```

%Universidad de Costa Rica

```

```

%Posgrado de Ingeniería Mecánica

```

```

%Maestría Profesional En Ingeniería Mecánica

```

```

%Nombre de archivo: "Inicializa_variables.m"

```

```

%Descripción básica: El propósito de este subprograma es generar
    los hijos con base en los padres utilizando la recombinación
    simulada binaria y mutación.

```

```

function f = operador genetico(cromosoma_padre , M, V, mu, mum,
    limite_inf , limite_sup)

```

```

[N,m] = size(cromosoma_padre);

clear m
p = 1;
recombinacion= 0;
mutacion= 0;

for i = 1 : N
    if rand(1) < 0.9 %90% de probabilidad de mutación
        hijo_1 = [];
        hijo_2 = [];
        padre_1 = round(N*rand(1)); % Selecciona el primer padre
        if padre_1 < 1
            padre_1 = 1;
        end
        padre_2 = round(N*rand(1)); % Selecciona el Segundo
        padre
        if padre_2 < 1
            padre_2 = 1;
        end
        end

        while isequal(cromosoma_padre(padre_1,:), cromosoma_padre
            (padre_2,:)) %verifica que los padres no sean el
            mismo
                padre_2 = round(N*rand(1));
                if padre_2 < 1
                    padre_2 = 1;
                end
            end
        end

        padre_1 = cromosoma_padre(padre_1,:); %Obtiene informaci
            ón del cromosoma del padre 1
        padre_2 = cromosoma_padre(padre_2,:); %Obtiene informaci
            ón del cromosoma del padre 2

        for j = 1 : V %Realiza la recombinación
            u(j) = rand(1);
            if u(j) <= 0.5
                bq(j) = (2*u(j))^(1/(mu+1));
            else
                bq(j) = (1/(2*(1 - u(j))))^(1/(mu+1));
            end

            hijo_1(j) = ... %Se genera el hijo 1
                0.5*(((1 + bq(j))*padre_1(j)) + (1 - bq(j))*
                padre_2(j));
            % Generate the jth element of second hijo
            hijo_2(j) = ... %Se genera el hijo 2
                0.5*(((1 - bq(j))*padre_1(j)) + (1 + bq(j))*
                padre_2(j));
        end
    end
end

```

```

        if hijo_1(j) > limite_sup(j) %verifica que el hijo 1
            está en el rango de estudio
            hijo_1(j) = limite_sup(j);
        elseif hijo_1(j) < limite_inf(j)
            hijo_1(j) = limite_inf(j);
        end
        if hijo_2(j) > limite_sup(j) %verifica que el hijo 2
            está en el rango de estudio
            hijo_2(j) = limite_sup(j);
        elseif hijo_2(j) < limite_inf(j)
            hijo_2(j) = limite_inf(j);
        end
    end
end

%Se calculan las funciones objetivo para los hijos
hijo_1(:,V + 1: M + V) = evalua_objetivos(hijo_1, M, V);
hijo_2(:,V + 1: M + V) = evalua_objetivos(hijo_2, M, V);
recombinacion= 1;
mutacion= 0;
else %0% probabilidad de mutación
    padre_3= round(N*rand(1));
    if padre_3 < 1
        padre_3= 1;
    end
    hijo_3= cromosoma_padre(padre_3,:);
    for j = 1 : V
        r(j) = rand(1); %selecciona un padre aleatorio
        if r(j) < 0.5
            delta(j) = (2*r(j))^(1/(mum+1)) - 1;
        else
            delta(j) = 1 - (2*(1 - r(j)))^(1/(mum+1));
        end

        %Se crea el hijo
        hijo_3(j) = hijo_3(j) + delta(j);

        if hijo_3(j) > limite_sup(j) %verifica que el hijo 2
            está en el rango de estudio
            hijo_3(j) = limite_sup(j);
        elseif hijo_3(j) < limite_inf(j)
            hijo_3(j) = limite_inf(j);
        end
    end
end

hijo_3(:,V + 1: M + V) = evalua_objetivos(hijo_3, M, V);
% Evalua las funciones objetivo del hijo 3
mutacion= 1;
recombinacion= 0;
end

```

%Se rellena la informacion de los hijos segun la mutacion o recombinaion

```

    if recombinacion
        hijo(p,:) = hijo_1;
        hijo(p+1,:) = hijo_2;
        was_crossover = 0;
        p = p + 2;
    elseif mutacion
        hijo(p,:) = hijo_3(1,1 : M + V);
        mutacion= 0;
        p = p + 1;
    end
end
end

f = hijo;

```

----- Subprograma 8 -----

```

%Universidad de Costa Rica
%Posgrado de Ingeniería Mecánica
%Maestría Profesional En Ingeniería Mecánica
%Nombre de archivo:" FuerzaAEsfuerzoResorte"
%Descripción básica: %La idea de esta funcion es pasar el ciclo
    de fuerzas del resorte a
esfuerzos. La entrada es la fuerza en Newton, D y d en metros.
    La salida
son los valores de esfuerzo en MPa

```

```

function [Esfuerzos]=FuerzaAEsfuerzoResorte(F,D,d)
FSize=size(F);
Esfuerzos=zeros(FSize);
C=D/d;
Kb=(4*C+2)/(4*C-3);
Esfuerzos =(Kb*(8*D).*F)/(pi*d^3);

```

----- Subprograma 9 -----

```

%Universidad de Costa Rica
%Posgrado de Ingeniería Mecánica
%Maestría Profesional En Ingeniería Mecánica
%Nombre de archivo:" TransfFourier.m"
%Descripción básica: %La idea de esta funcion es realizar la
    transformada de fourier de un grupo de datos entratne

```

```

function [Y,f] = TransfFourier( X, n, dim, muestreo)
% Realiza la transformada de fourier
Y = fft(X,n,dim);
Y = fftshift(Y,dim);

%Determina el vector de frecuencia
nMuestras = size(Y, dim);
df = muestreo/ nMuestras;
f = 0 : df : df*(nMuestras-1);
f = fftshift(f);
ind = f>=muestreo/2-eps(muestreo/2);

```



```
f(ind) = f(ind)-muestreo;  
f = shiftdim(f, 2-dim);  
  
end
```