

OPTIMIZACIÓN DE LA GEOMETRÍA DE UN EJE SOMETIDO A CARGAS  
CÍCLICAS UTILIZANDO EL ALGORITMO METAHEURÍSTICO: PARTICLE  
SWARM OPTIMIZATION (PSO)

HERMAN ADOLFO NAVARRO OCHOA  
JOSE ALEJANDRO RODRÍGUEZ DÍAZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNICA  
BUCARAMANGA

2021

OPTIMIZACIÓN DE LA GEOMETRÍA DE UN EJE SOMETIDO A CARGAS  
CÍCLICAS UTILIZANDO EL ALGORITMO METAHEURÍSTICO: PARTICLE  
SWARM OPTIMIZATION (PSO)

HERMAN ADOLFO NAVARRO OCHOA  
JOSE ALEJANDRO RODRÍGUEZ DÍAZ

Trabajo de grado para optar título de Ingeniero Mecánico

Director:

Jabid Eduardo Quiroga Méndez  
Ingeniero Mecánico, PhD en Ingeniería Civil

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNICA  
BUCARAMANGA

2021

## DEDICATORIA

*Principalmente a mis padres y a mi hermana; así como al resto de la familia y todos aquellos que me han apoyado incondicionalmente y por supuesto a mis amigos a lo largo de la universidad, esto no habría sido posible sin ellos.*

***Herman Adolfo Navarro Ochoa***

*A mis padres, mi hermana y mi familia,  
han sido la razón de mis logros.  
A mis compañeros de carrera y amigos de infancia  
fueron apoyo incondicional en todo momento.  
Y a Dios que ha sido siempre fiel y oportuno,  
de él todo, mío nada.*

**Jose Alejandro Rodríguez Díaz**

## **AGRADECIMIENTOS**

Agradecemos principalmente al Ing. Jabid Eduardo Quiroga Méndez, nuestro director de proyecto por brindarnos su ayuda y poner a disposición sus conocimientos durante el desarrollo de esta investigación. Asimismo, agradecemos a algunos miembros de la Escuela de Ingeniería Mecánica por su apoyo durante el proceso:

Ing. Rafael Alexis Pinto Flórez.          Egresado de Escuela de Ingeniería Mecánica.

Ing. William Pinto.                          Profesor de la Escuela de Ingeniería Mecánica.

Ing. Gilberto Parra.                          Profesor de la Escuela de Ingeniería Mecánica.

Finalmente agradecemos a la Escuela de Ingeniería Mecánica, por su excelente formación y exigencia brindada a lo largo de estos años.

## CONTENIDO

INTRODUCCIÓN .....	13
1. OBJETIVOS .....	15
1.1 OBJETIVO GENERAL .....	15
1.2 OBJETIVOS ESPECÍFICOS.....	15
2. MARCO DE REFERENCIA.....	16
2.1 MARCO TEÓRICO .....	16
2.1.1 Frente Óptimo de Pareto.....	16
2.1.1.1 Óptimo de Pareto en la optimización multiobjetivo. ....	17
2.1.2 Problemas de Optimización Multiobjetivo POM. ....	18
2.1.3 Análisis de Varianza ANOVA.....	19
2.1.4 Algoritmos Genéticos AG's. ....	21
2.1.5 Algoritmo PSO. ....	23
2.1.6 Algoritmo Original. ....	24
2.1.7 Proceso de optimización con PSO.....	25
2.1.8 Formas de búsqueda. ....	27
2.1.9 Formulación Matemática.....	28
2.1.10 Alcances. ....	32
2.2 MARCO DE ANTECEDENTES.....	34
2.2.1 Optimización geometría de un eje aplicando algoritmos genéticos. ....	34
2.2.2 Proceso de optimización con PSO.....	35
2.2.3 IAO vs PSO.....	38
2.2.4 Optimización multiobjetivo de un eje con algoritmo basado en quimiotaxis de bacterias .....	40
3. METODOLOGÍA .....	42
4. CARACTERIZACIÓN DEL PROBLEMA DEL EJE.....	43
5. IMPLEMENTACIÓN Y APLICACIÓN DEL ALGORITMO.....	47
5.1 Implementación PSO en Matlab .....	47

5.2 Aplicación al problema .....	48
5.3 Resultados .....	50
5.3.1 Óptimo de Pareto.....	50
5.3.1.1 Elección De La Mejor Respuesta.....	52
5.3.2 Implementación ANOVA. ....	55
5.3.2.1 Análisis De La Variable De Respuesta Convergencia. ....	55
5.3.2.2 Análisis De La Variable De Respuesta Tiempo. ....	63
6. DISCUSIÓN .....	71
6.1 Análisis de la optimización .....	71
6.2 Comparación de resultados .....	80
7. CONCLUSIONES .....	83
BIBLIOGRAFÍA.....	85
ANEXOS.....	87

## LISTA DE TABLAS

<b>Tabla 1.</b> Datos proporcionados por el problema.....	44
<b>Tabla 2.</b> Resultados escogidos a partir del frente óptimo de soluciones. ....	52
<b>Tabla 3.</b> Duplicado del experimento para la Masa y su media .....	56
<b>Tabla 4.</b> Resultados de masa para las 32 combinaciones. ....	57
<b>Tabla 5.</b> Duplicado del experimento para el tiempo y su media .....	64
<b>Tabla 6.</b> Resultados de tiempo para las 32 combinaciones. ....	65
<b>Tabla 7.</b> Datos obtenidos para el análisis de convergencia.....	72
<b>Tabla 8.</b> Resultados obtenidos mediante NSGA. ....	80

## LISTA DE FIGURAS

<b>Figura 1.</b> Gráfico de un óptimo de Pareto en optimización multiobjetivo.....	17
<b>Figura 2.</b> Muestras de población con distintas medias, sin embargo, con la misma medida de varianza ponderada entre grupos. ....	20
<b>Figura 3.</b> Diagrama de flujo del algoritmo PSO.....	27
<b>Figura 4.</b> Movimiento de una partícula en el espacio de soluciones. ....	30
<b>Figura 5.</b> Algoritmo PSO Básico.....	31
<b>Figura 6.</b> Ejemplo de caja. ....	36
<b>Figura 7.</b> Esquema de una caja blanca. ....	37
<b>Figura 8.</b> Esquema de una caja negra. ....	37
<b>Figura 9.</b> Funciones objetivo para la población final de bacterias. ....	42
<b>Figura 10.</b> Eje sometido a cargas cíclicas. ....	45
<b>Figura 11.</b> Frente óptimo de Pareto obtenido con la última iteración. ....	51
<b>Figura 12.</b> Zonas de optimización del algoritmo PSO. ....	53
<b>Figura 13.</b> Gráfica de efector principales para masa.....	62
<b>Figura 14.</b> Gráfica de interacción para masa. ....	63
<b>Figura 15.</b> Gráfica de efectos principales para Tiempo.....	69
<b>Figura 16.</b> Gráfica de interacción para Tiempo. ....	70
<b>Figura 17.</b> Gráfica de Pareto de efectos estandarizados para Tiempo. ....	71
<b>Figura 18.</b> Análisis de convergencia en función del valor Gbest. ....	73
<b>Figura 19.</b> Masa vs. Deflexión, pesos: [0 1] .....	74
<b>Figura 20.</b> Masa vs. Deflexión, pesos: [1 0] .....	74
<b>Figura 21.</b> Masa vs. Deflexión, pesos: [1 1] .....	75
<b>Figura 22.</b> Masa vs. Deflexión, pesos: [1 10] .....	76
<b>Figura 23.</b> Zona de optimización D2 vs. L2.....	77
<b>Figura 24.</b> Curvas de nivel D2 vs. L2. ....	78
<b>Figura 25.</b> Zona de optimización D1 vs. D3. ....	79
<b>Figura 26.</b> Curvas de nivel D1 vs. D3.....	79
<b>Figura 27.</b> Esquema de la configuración del eje PSO y NSGA. ....	82

## LISTA DE ANEXOS

<b>ANEXO A.</b> PSO .....	87
<b>ANEXO B.</b> Posiciones .....	90
<b>ANEXO C.</b> P mejores .....	91
<b>ANEXO D.</b> Partículas.....	92
<b>ANEXO E.</b> Calcular.....	93
<b>ANEXO F.</b> Cal Velocidades .....	94
<b>ANEXO G.</b> Función Adaptación.....	95
<b>ANEXO H.</b> G mejor.....	96
<b>ANEXO I.</b> Velocidad .....	97
<b>ANEXO J.</b> Fatiga .....	98
<b>ANEXO K.</b> Y máx .....	101
<b>ANEXO L.</b> Restricciones .....	104
<b>ANEXO M.</b> Redondeo paso.....	105
<b>ANEXO N.</b> qt.....	106
<b>ANEXO O.</b> qs.....	107
<b>ANEXO P.</b> Kts.....	108
<b>ANEXO Q.</b> Kt.....	109
<b>ANEXO R.</b> Kfs .....	110
<b>ANEXO S.</b> Kf .....	111
<b>ANEXO T.</b> Only FO.....	112
<b>ANEXO U.</b> Graphs Contour .....	113
<b>ANEXO V.</b> Concentrador Esfuerzos .....	116

## RESUMEN

**TÍTULO:** Optimización de la geometría de un eje sometido a cargas cíclicas utilizando el algoritmo metaheurístico: Particle Swarm Optimization (PSO)\*

**AUTOR:** Herman Adolfo Navarro Ochoa, Jose Alejandro Rodríguez Díaz\*\*

**PALABRAS CLAVE:** Optimización, fatiga, PSO, diseño mecánico, POM, ejes.

**DESCRIPCIÓN:** Los problemas de optimización multiobjetivo, frecuentes en la ingeniería, suponen un reto en el hallazgo de soluciones, puesto que su búsqueda se encuentra limitada por los procesos de cálculo tradicionales. El PSO surge como una herramienta de tipo procesamiento paralelo, es un algoritmo inteligente que explora el espacio de búsqueda en múltiples direcciones, teniendo en cuenta el rango y las restricciones propias de las funciones evaluando de manera autónoma la aptitud de las suposiciones en cada iteración.

Gracias a los avanzados sistemas de cómputo numérico, algoritmos como el PSO se pueden modelar y simular proponiendo un método de solución alternativo a problemas que involucran múltiples funciones objetivo aún si son contradictorias entre sí. En esta investigación se presenta la aplicación del PSO, mediante el software Matlab, como solución a un problema de diseño mecánico, concretamente un eje sometido a cargas cíclicas bajo restricciones de fatiga buscando optimizar de manera simultánea su masa y deflexión máxima. Se realizó un diseño de experimentos con la técnica de análisis estadístico de varianza (ANOVA) con el fin de mejorar la búsqueda del algoritmo a partir de sus parámetros de entrada, además de comparar su eficiencia respecto de otros algoritmos bio-inspirados (NSGA y Quimiotaxis de bacterias).

---

\* Trabajo de grado.

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Mecánica. Director: Jabid Eduardo Quiroga Méndez, Ingeniero Mecánico y PhD en Ingeniería Civil.

## ABSTRACT

**TITLE:** Optimization of the geometry of a shaft subjected to cyclic loads using the metaheuristic algorithm: Particle Swarm Optimization (PSO)\*

**AUTHOR:** Herman Adolfo Navarro Ochoa, Jose Alejandro Rodríguez Díaz \*\*

**KEY WORDS:** Optimization, fatigue, PSO, mechanical design, POM, shafts.

**DESCRIPTION:** Multi-objective optimization problems, common in engineering, pose a challenge in finding solutions, since their search is limited by traditional calculation processes. The PSO emerges as a parallel processing type tool, it is an intelligent algorithm that explores the search space in multiple directions, taking into account the range and restrictions of the functions, autonomously evaluating the suitability of the assumptions in each iteration.

Thanks to advanced numerical computing systems, algorithms such as PSO can be modeled and simulated proposing an alternative solution method to problems that involve multiple objective functions even if they are contradictory to each other. In this research, the application of PSO is presented, using Matlab software, as a solution to a mechanical design problem, specifically a shaft subjected to cyclical loads under fatigue restrictions, seeking to simultaneously optimize its mass and maximum deflection. A design of experiments was carried out with the statistical analysis of variance (ANOVA) technique in order to improve the search for the algorithm based on its input parameters, in addition to comparing its efficiency with respect to other bio-inspired algorithms (NSGA and Chemotaxis bacteria).

---

\* Degree work

\*\* Faculty of Physical and Mechanical Engineering. School of Mechanical Engineering. Director: Jabid Eduardo Quiroga Méndez, Mechanical Engineer and PhD in Civil Engineering.

## INTRODUCCIÓN

Existen algunos problemas complejos, cuya resolución a menudo se ve insatisfecha por la matemática lineal, lo que hace necesario el uso de métodos heurísticos, los cuales proponen estrategias que guían al descubrimiento y se destacan por su uso en problemas de optimización con múltiples variables, en los que coexisten varias soluciones factibles.

El PSO (Particle Swarm Optimization) es una herramienta que busca describir un comportamiento natural, específicamente el movimiento de los individuos pertenecientes a una misma especie cuando se mueven en conjunto, como una bandada de aves o un cardumen de peces. Matemáticamente los individuos representan posibles soluciones dentro de un espacio de búsqueda correspondiente al dominio de las funciones que se quieren optimizar, estos individuos (también llamados partículas) se desplazan por el espacio en búsqueda de las mejores posiciones hasta encontrar, cada una, el mejor valor en su recorrido. Este algoritmo es del tipo procesamiento paralelo, lo cual quiere decir que es capaz de explorar el espacio de búsqueda en múltiples direcciones, asegurando varios resultados adecuados en un corto período de tiempo. En el campo ingenieril es muy común encontrarse con situaciones que requieren la determinación de dos o más parámetros relacionados entre sí, de manera simultánea, las cuales se interpretan matemáticamente como funciones multiobjetivo.

Uno de los propósitos de la mecánica es la transmisión de potencia y para ello es necesario el uso de ejes que permitan la interacción de varios elementos en diferentes puntos, esto propone un desafío en el diseño puesto que se generan diferentes restricciones de dimensionamiento para cada sección del eje a lo largo de su longitud; se propone el uso del algoritmo PSO como solución a un problema base que involucra condiciones mecánicas típicas, con el fin de encontrar las

configuraciones óptimas para su geometría, restringiendo su peso y deflexión de manera simultánea y con óptimos resultados. En las últimas décadas, han surgido diferentes herramientas computacionales, que permiten desarrollar éste tipo de algoritmos. En este trabajo se utiliza el software Matlab®, donde se implementan las ecuaciones propias del algoritmo junto con las del problema, que incluyen la teoría de falla por fatiga y el análisis de resistencia de materiales, obteniendo un conjunto de soluciones en forma de frente óptimo de Pareto, que posteriormente se comparan con resultados de investigaciones similares (que responden al mismo problema) y se analizan para obtener un único resultado bajo ciertos criterios. Finalmente, con el objetivo de generar algunas recomendaciones, se propone el análisis de varianza ANOVA para determinar en qué proporción afectan ciertos factores el costo computacional y la precisión del algoritmo.

En el año 2005 se desarrolló este problema usando el algoritmo genético NSGA (Non-Dominated Sorting Genetic Algorithm) y tiempo después mediante un algoritmo basado en Quimiotaxis de bacterias, generando un conglomerado de soluciones factibles, analizadas mediante un método estadístico. Se ha aplicado el PSO a otros problemas en el campo de la ingeniería como la identificación de parámetros de un sistema tipo caja gris en comparación a la respuesta de una función escalón en lazo abierto de un motor DC. Este algoritmo también ha sido objeto de comparación con otros métodos como el IAO (Optimización por Análisis de Intervalos), demostrando ser más rápido e igualmente preciso. La aplicación de este tipo de algoritmos a situaciones reales abre una amplia ventana a la exploración y explotación del espacio de búsqueda de casi cualquier problema de tipo industrial que requiera optimización.

# 1. OBJETIVOS

## 1.1 OBJETIVO GENERAL

Contribuir a la exploración y explotación del espacio de búsqueda en el desarrollo de problemas multiobjetivo, optimizando los diámetros y las longitudes de un eje sometido a condiciones de carga cíclica, bajo restricciones de peso y deflexión lateral máxima, mediante la aplicación del método metaheurístico: Particle Swarm Optimization (PSO).

## 1.2 OBJETIVOS ESPECÍFICOS

1. Identificar los conceptos básicos, metodología, recursos, formulación matemática y alcances del algoritmo de optimización PSO.
2. Diseñar, programar e implementar el algoritmo de PSO en la herramienta Matlab de tal forma que permita variar ciertos parámetros tales como: cantidad inicial de partículas, dominio de búsqueda, número de iteraciones, velocidad de cálculo.
3. Implementar el algoritmo propuesto en el objetivo anterior para el siguiente problema: Optimización de la geometría de un eje sometido a cargas cíclicas con restricciones de peso y deflexión lateral máxima; con el propósito de encontrar el conjunto de soluciones que optimicen las funciones objetivo.
4. Implementar el análisis de varianza ANOVA para determinar los parámetros de impacto sobre el algoritmo.

5. Verificar que las soluciones dadas por el algoritmo satisfagan las funciones objetivo y sobre estas aplicar el método estadístico de la eficiencia de Pareto para encontrar el frente óptimo de geometrías y finalmente evaluar su desempeño respecto de los resultados encontrados y publicados en el artículo de referencia.

## 2. MARCO DE REFERENCIA

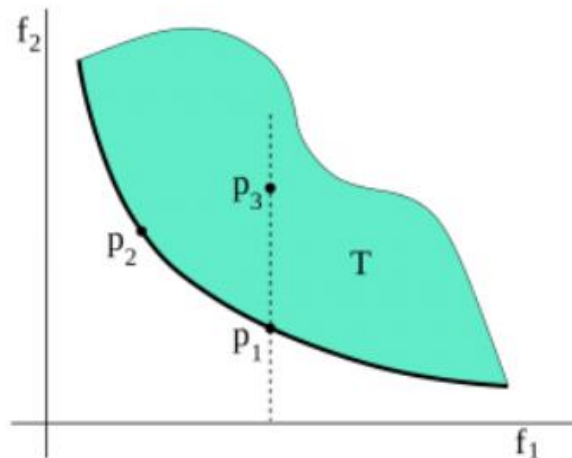
### 2.1 MARCO TEÓRICO

**2.1.1 Frente Óptimo de Pareto.** El óptimo de Pareto es una estrategia económica propuesta y desarrollada por Vilfredo Pareto en su libro “Manuale di economia politica” (1906) que advierte la posibilidad de una mejora dentro de una sociedad siempre y cuando ninguno de los individuos se vea afectado por el progreso o provecho del otro. Este concepto está basado en criterios de utilidad y tiene como objetivo desarrollar un proceso natural hasta alcanzar un punto óptimo en donde se encuentre el mayor bienestar alcanzable de una sociedad, esto quiere decir que si crece la utilidad de un individuo no disminuirá la utilidad de otro y por lo tanto aumenta el bienestar general.

El óptimo de Pareto tiene un sinnúmero de aplicaciones dentro de la economía, la distribución eficiente de recursos y la política, sin embargo, el concepto es aceptable y adaptable a procesos matemáticos de optimización multiobjetivo, es decir, en problemas donde se requiere satisfacer varias funciones objetivo a la vez y de forma conjunta. Es habitual que los objetivos se encuentren en conflicto entre sí, para el caso de la investigación en este proyecto las funciones objetivo se encuentran ligadas y adicionalmente, el algoritmo a desarrollar debe ser rápido y dar soluciones de calidad.

**2.1.1.1 Óptimo de Pareto en la optimización multiobjetivo.** Es habitual que, para simplificar el conflicto, los problemas tiendan a modelarse para cumplir uno sólo de los objetivos y el resto de ellos pueden ser tratados como restricciones del problema. También es posible en algunos casos simplificar a una sola función por medio de la suma ponderada de las demás. Existe un concepto ligado a los problemas de este tipo matemático desarrollado por Francis Ysidro Edgeworth en 1881, economista y estadístico británico, que fue generalizado después por Vilfredo Pareto en 1896.

**Figura 1.** Gráfico de un óptimo de Pareto en optimización multiobjetivo.



Fuente: <https://economipedia.com/definiciones/optimo-de-pareto.html>

La figura explica de manera gráfica el concepto. Los ejes  $f_1$  y  $f_2$  son las funciones que se desean optimizar y los puntos  $P$  son las posibles soluciones numéricas que pueden satisfacer las funciones. El Óptimo de Pareto plantea en este caso que una solución  $P_i$  es óptima cuando no existe otra que pueda minimizar una función sin perjudicar la otra.  $P_2$  puede ser solución si se desea minimizar más a  $f_1$  y  $P_1$  puede

ser solución si se desea minimizar más a  $f_2$ , para cualquiera de estas premisas P3 no será una solución óptima.

Para los problemas de esta naturaleza es muy posible que las soluciones sean vectores, por lo cual una solución  $x^*$  será óptima para las funciones objetivo si no existe otra solución  $x$ , que haga a un vector  $i = f(x)$  dominar sobre otro vector  $j = f(x^*)$  en al menos una función, y que al hacerlo, no decremente algún criterio sin causar un incremento simultáneo en al menos otro criterio.

**2.1.2 Problemas de Optimización Multiobjetivo POM.** En la vida cotidiana podemos encontrar diferentes situaciones problema que se reconocen como Problemas Multiobjetivo, es decir, que una única solución no será siempre satisfactoria del todo, ya que existen múltiples criterios que deben satisfacerse o ser tenidos en cuenta, estos pueden tender a entrar en conflicto y no existe una solución que satisfaga todas las condiciones simultáneamente, allí es donde entran a jugar las preferencias del decisor y las restricciones propias del problema.

Existen problemas de único objetivo, que consisten en optimizar una sola variable, por ejemplo, las empresas transportadoras, buscan optimizar el tiempo de viaje de la mercancía, costos de operación de un vehículo, emisiones de CO2 y otros factores más, pero siempre uno a la vez.

Los problemas modernos de ingeniería involucran la optimización, es decir, que las funciones objetivo deben ser maximizadas o minimizadas con el fin de encontrar las soluciones óptimas y es frecuente encontrar situaciones reales, donde se requiere optimizar dos y más funciones, las cuales pueden incluso llegar a ser contradictorias

entre sí, estos problemas se conocen como Problemas de Optimización Multiobjetivo<sup>1</sup> y existen diferentes técnicas para resolverlos.

Cuando la atención no se puede centrar en uno solo de los objetivos, una solución óptima para un objetivo puede no serlo para los otros, sería impropio escoger dicha alternativa como la más óptima y se genera la necesidad de saber cómo se encuentran relacionadas las funciones objetivo entre sí.<sup>2</sup>

Como se muestra en el Frente Óptimo de Pareto, existe una frontera entre las soluciones factibles y no factibles, y se debe recurrir a las diferentes herramientas computacionales y algoritmos con el fin de proporcionar a partir de allí, las soluciones adecuadas al problema.

**2.1.3 Análisis de Varianza ANOVA.** El análisis de varianza, también llamado análisis factorial, fue propuesto por el estadístico y biólogo Ronald Aylmer Fisher en 1930 y es una herramienta estadística útil para la determinación del efecto o influencia de uno o más factores, cada uno con dos o más niveles, sobre la media de una variable continua.

Usada a menudo en el análisis y diseño de experimentos, esta herramienta permite evaluar los posibles efectos de los factores sobre la varianza de una variable. Su funcionamiento básico consiste en calcular la media de cada uno de los grupos, y a continuación, comparar la varianza entre estas medias. Se parte de la hipótesis que los distintos grupos proceden de la misma población, con la misma media y varianza

---

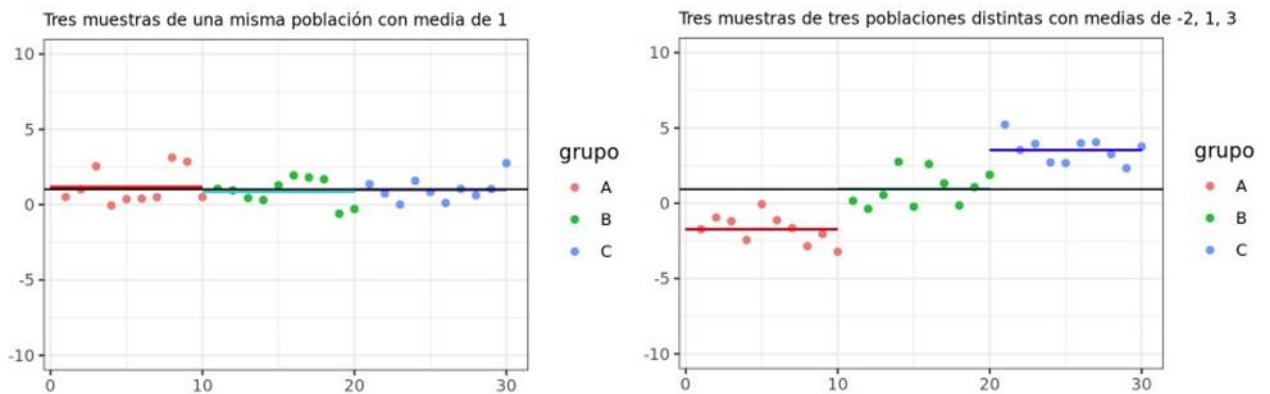
<sup>1</sup> GUZMÁN María Alejandra, DELGADO Alberto. Optimización de la geometría de un eje aplicando algoritmos genéticos. *Revista ingeniería e investigación* (58). VOL. 25 No 2, agosto 2005 15-23.

<sup>2</sup> Problema Multiobjetivo. Capítulo 4.

Es decir, se compara la varianza explicada según el grupo (intervarianza), con la varianza promedio dentro de los grupos (intravarianza, que es la no explicada por la variable grupo). La varianza ponderada entre grupos será la misma que la varianza promedio dentro de los grupos, así, a medida que las medias de los grupos estén más alejadas las unas de las otras, la varianza entre medias se incrementará y dejará de ser igual a varianza promedio de los grupos.

A continuación, se muestra un ejemplo de los tipos de población y sus medias dentro del análisis de varianza.<sup>3</sup>

**Figura 2.** Muestras de población con distintas medias, sin embargo, con la misma medida de varianza ponderada entre grupos.



Fuente: [https://www.cienciadedatos.net/documentos/19\\_anova](https://www.cienciadedatos.net/documentos/19_anova)

En el ANOVA, uno de los elementos más importantes es el *Fratio*, que es el ratio entre la varianza de las medias de los grupos y el promedio de la varianza dentro de los grupos. Cuando se cumple la hipótesis nula, que dice que los distintos grupos proceden de la misma población, con la misma media y varianza, *F* adquiere el valor de 1, ya que la *intervarianza* y la *intravarianza* serían iguales. Cuanto más diferentes

<sup>3</sup> JOAQUÍN AMAT, Rodrigo. ANOVA análisis de varianza para comparar múltiples medias. Cienciadatos.net. Enero 2016.

sean entre sí las medias de los grupos, mayor será la varianza entre medias en comparación al promedio de la varianza dentro de los grupos, obteniéndose valores superiores a 1, y con menor probabilidad de que la distribución adquiriera valores tan extremos.

$S_1^2$  es la varianza de una muestra de tamaño  $N_1$ , extraída de una población normal de varianza  $\sigma_1^2$ , lo mismo sucede para  $S_2^2$ ,  $N_2$  y  $\sigma_2^2$  de otra muestra independiente, F sería el cociente entre ellos:

$$F = \frac{S_1^2 / \sigma_1^2}{S_2^2 / \sigma_2^2}$$

F es una variable de Snedecor, esto quiere decir que dicha distribución se utiliza para contrastar la igualdad de las varianzas (homocedasticidad), con ( $N_1$  y  $N_2$ ) grados de libertad. Existen diferentes tipos de ANOVA, dependiendo si los datos son independientes, o si son datos pareados, si se compara una variable dependiente en comparación con un factor, o si se compara esa misma variable con dos factores, y con la variación de si estos factores son o no, dependientes entre sí.

**2.1.4 Algoritmos Genéticos AG's.** Los algoritmos genéticos han supuesto un foco de atención para los investigadores de diferentes ramas de la ingeniería, debido a la cantidad de artículos y aplicaciones que han surgido sobre el tema en los últimos años<sup>4</sup>. Se trata de métodos heurísticos de búsqueda inspirados en los procesos presentes en la naturaleza, tratando de copiar sus métodos, con el fin de aplicarlos a la resolución de nuestros propios problemas.

---

<sup>4</sup> KURI MORALES. Ángel y GALAVIZ CASAS. José. Algoritmos Genéticos.

Una de las razones de su éxito es que se pueden aplicar a problemas donde es muy difícil implementar la matemática tradicional, otra, es su versatilidad y plasticidad inherente a estos métodos, es decir, que se pueden aplicar a diferentes problemas sin importar su naturaleza, y sin hacer modificaciones sustanciales al procedimiento general, es decir que no se supone prácticamente nada acerca del problema a resolver, a diferencia de los métodos tradicionales, en donde se exige que el problema consista de una función claramente definida y tenga ciertas propiedades.

Los Algoritmos Genéticos<sup>5</sup>, al trabajar de manera análoga con el comportamiento natural, consideran una población inicial de individuos, cada uno de los cuales representa una posible solución al problema planteado. Se le asignará a cada individuo un valor, o “puntuación”, que representa el grado de efectividad de un organismo para competir por los recursos, cuanto mayor sea este valor, aumentarán las probabilidades de que ese individuo sea seleccionado para reproducirse, cruzándose así, con otro individuo seleccionado bajo el mismo sistema. Este cruce producirá nuevos individuos que compartirán características de sus padres y mutará en cierto grado, mostrando nuevas características; cuanto menor sea el valor asignado, menor será la posibilidad de que ese individuo sea seleccionado para reproducirse, y no podrá transmitir su material genético a las siguientes generaciones. Así, una nueva generación habrá reemplazado a la anterior, con la interesante propiedad que contiene mayor proporción de buenas características, favoreciendo a los individuos mejor adaptados a lo largo de las generaciones y a través de las áreas más prometedoras del espacio de búsqueda, hasta converger en las soluciones óptimas del problema.

Una de las grandes ventajas de los AG's es que son técnicas robustas que pueden analizar grandes cantidades de datos de manera simultánea, y se puede aplicar a problemas de diferentes áreas, no se garantiza que el algoritmo encuentre la

---

<sup>5</sup> KURI. Ángel. Algoritmos Genéticos.

solución óptima, pero si existe evidencia de que son capaces de encontrar soluciones aceptables en un tiempo bastante competitivo en comparación con otros algoritmos. Existen problemas que requieren el desarrollo de técnicas especializadas, las cuales son capaces de encontrar soluciones igualmente válidas o incluso mejores respecto a los AG's en menor tiempo, sin embargo, se ha demostrado que estas dos técnicas se pueden hibridar obteniendo mejoras en la resolución de estos problemas.

**2.1.5 Algoritmo PSO.** El Algoritmo inteligente *Particle Swarm optimization (PSO)* es una técnica de cálculo y optimización basada en comportamientos naturales y poblaciones. Considerado también como un método heurístico usualmente implementado en problemas de computación general. Este algoritmo ha demostrado ser capaz de resolver problemas matemáticos de diferente naturaleza con una gran simplicidad y una rapidez considerable, lo que le ha permitido abrirse paso en el mundo de la ingeniería y en el sector industrial.

Fue introducido por primera vez en 1995 por James Kennedy y Russell Eberhart<sup>6</sup> y está inspirado en el paradigma de un enjambre (“swarm”), que no es más que el comportamiento de animales en grupo (bandada de pájaros, cardúmenes de peces, grupos de abejas, etc.). La idea nace de la comunicación que existe entre los individuos pertenecientes a un mismo enjambre; si se toma el ejemplo de la bandada de pájaros es sencillo encontrar la relación entre su comportamiento y el objetivo del algoritmo. Los pájaros al alzar su vuelo en grupo se mueven muy estrechamente entre sí, sin embargo, aunque esto ocurre en el despegue e instantes posteriores a este, no se chocan y por el contrario parecen describir movimientos cada vez más ordenados hacia lugares aparentemente más cómodos.

---

<sup>6</sup> KENNEDY, James y EBERHART, Russel. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*. 1995, Vol. 4, Pp. 1942-1948

La razón de este fenómeno que ocurre en muchas otras especies de la naturaleza se debe a la “actualización” constante, individual y grupal, de la posición en el espacio, un solo pájaro será capaz de encontrar un mejor lugar por el cual moverse con el pasar del tiempo y entenderá a su vez cuales son los lugares por los que no debe moverse y comunicará su experiencia al grupo nutriendo así el comportamiento y el vuelo de todos los individuos. Lo mismo ocurre en el proceso de búsqueda de alimento en los grupos de hormigas, aunque su visión es bastante limitada son capaces de encontrar alimento y transportarlo entre varias gracias a que cada una tiene la libertad de explorar el espacio y comunicar a las demás los mejores lugares donde ha estado y en los cuales ha encontrado comida.

El comportamiento del algoritmo describe un proceso similar basado en el cambio de la posición a través del tiempo de cada una de las posibles soluciones y la mejora continua de su ubicación en el espacio de búsqueda por medio de la experiencia individual y la comunicación grupal hacia el mejor valor que pueda encontrarse y que satisfaga las funciones objetivo.

Se inicializa con una población de soluciones aleatorias que son llamadas “partículas” las cuales se mueven en un espacio de búsqueda, limitado por ciertas restricciones matemáticas propias de las ecuaciones a optimizar, con una determinada velocidad que se va ajustando en el tiempo según el comportamiento histórico guiando el proceso hacia la mejor área del espacio.

**2.1.6 Algoritmo Original.** El fenómeno de la búsqueda de la mejor solución del enjambre está descrito por tres magnitudes cinemáticas: la posición, la velocidad y la aceleración. Estas magnitudes van cambiando con el tiempo y se van actualizando conforme ocurre la socialización de grupo inclinando el proceso a encontrar el valor óptimo.

$$V_{id} = V_{id} + C_1 rand( ) (p_{id} - x_{id}) + C_2 Rand( ) (p_{id} - x_{id}) \quad 1$$

$$x_{id} = x_{id} + V_{id}$$

2

Donde  $C_1$  y  $C_2$  son constantes siempre positivas,  $\text{rand}()$  y  $\text{Rand}()$  son dos valores aleatorios en un rango de 0 a 1,  $x_i$  representa la  $i$ -ésima partícula  $(x_{i1}, x_{i2}, \dots, x_{iD})$ ,  $P_i$  representa la mejor posición anterior de cada partícula, la letra  $g$  representa la mejor posición entre todas las partículas de la población y  $V_i$  la tasa a la que cambia la posición de la partícula.

Según Yuhui Shi<sup>7</sup> la ecuación 1 describe cómo la velocidad es dinámicamente actualizada y la ecuación 2 describe la actualización de la posición de la partícula. Sin embargo, en la ecuación de la velocidad se pueden distinguir tres partes, la primera es el impulso y se refiere a que la velocidad cambia desde la velocidad actual, la segunda parte se refiere al aprendizaje de la partícula respecto a su propio movimiento y la tercera, respecto del movimiento del grupo. Dicha ecuación tendrá un límite ajustado por el usuario del que dependerá la exactitud de la búsqueda, al ajustar esa velocidad como una velocidad máxima las partículas tendrán la posibilidad de viajar más lejos o más limitadas y encontrar soluciones globales o locales según el ajuste.

**2.1.7 Proceso de optimización con PSO.** Según investigaciones previas<sup>8</sup> se ha logrado determinar que el algoritmo de optimización por enjambre de partículas (PSO) presenta una estructura general, clara y organizada para su correcta implementación y desarrollo (Figura 3).

---

<sup>7</sup> YUHUI, Shi. Particle Swarm Optimization. Ed. *Electronic Data Systems, Inc. IEE Neural Networks Society*. 2004, pp. 8-13.

<sup>8</sup> DUARTE, César y QUIROGA, Jaid. Algoritmo PSO para identificación de parámetros en un motor DC. *Revista facultad de ingenierías de la Universidad de Antioquia*. 2010, nro. 55, pp. 116-124

Es posible establecer una serie de pasos y criterios que permiten encaminar la búsqueda desde las variables iniciales hasta la obtención de un resultado que satisfaga las funciones objetivo.

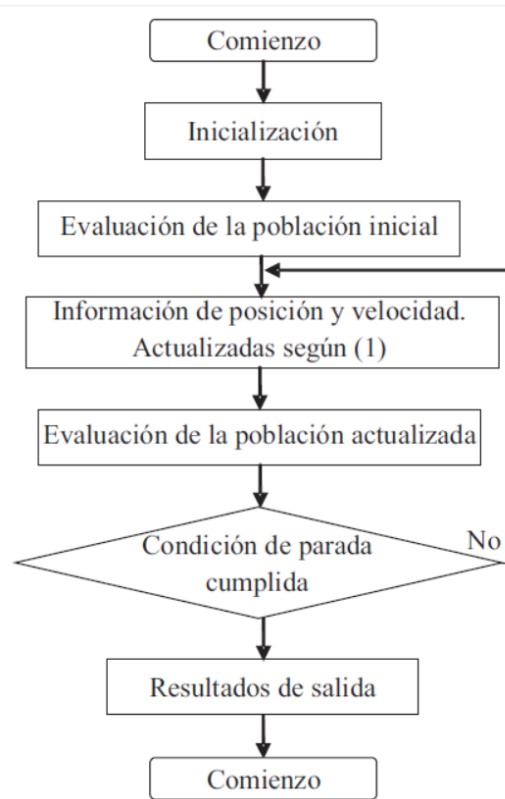
Es necesario entender que todo el proceso está ligado a las ecuaciones que rigen el movimiento de las partículas del enjambre en el espacio de búsqueda:

$$V_i^{k+1} = w * V_i^k + c_1 * rand_1 (pbest_i - x_i^k) + c_2 * rand_2 * (gbest_i - x_i^k)$$

$$x_i^{k+1} = x_i^k + V_i^{k+1}$$

- ✓ Es necesario establecer límites numéricos para todos los parámetros iniciales y así establecer un punto de partida (población inicial,  $V$ ,  $pbest_i$ ,  $rand_1$ ,  $rand_2$  y parada).
- ✓ La función objetivo evaluara el  $gbest$ .
- ✓ En este punto los parámetros iniciales que describen el movimiento de las partículas se actualizan y se reescriben teniendo en cuenta los límites previos.
- ✓ Es un proceso casi repetitivo de (3) puesto que volverán a actualizarse los valores previamente actualizados.
- ✓ Se evalúa si se ha cumplido ya la reglamentación de parada en el punto (1), si no, se continua el proceso iterativo.
- ✓ Se concluye con el valor de interés final:  $gbest$ .

**Figura 3.** Diagrama de flujo del algoritmo PSO.



Fuente: Jabid Quiroga, C. D. (2010). Algoritmo PSO para identificación de parámetros en un motor DC. Facultad de ingenierías, Universidad de Antioquia, 119p.

**2.1.8 Formas de búsqueda.** El algoritmo puede ser usado con diferentes estructuras. Las partículas pueden actualizar su información de manera individual (versión PSO global) o apoyadas en las partículas más cercanas, conjunto al que se le denomina vecindario (versión PSO local). Estos vecindarios también pueden ser comunicados en algunas formas geométricas, es decir, que las partículas entran en interacción con otras cercanas asemejando formas piramidales, en estrella, de anillo o incluso aleatoriamente.

Tres años después de ser propuesto el algoritmo original le fue añadido un parámetro extra con el objetivo de equilibrar la búsqueda local como también la búsqueda global.

$$V_{id} = wV_{id} + C_1rand( ) (p_{id} - x_{id}) + C_2Rand( ) (p_{id} - x_{id})$$

La ecuación () es la misma que (1) con la diferencia del parámetro  $w$  agregado. Esta letra simboliza el peso inercial dentro del fenómeno matemático y decrece a medida que el algoritmo se desarrolla en su proceso iterativo, este valor es importante porque garantiza, en cierta medida, su convergencia y es necesario inicializarlo con un valor. Mientras más alto sea su valor, favorecerá la búsqueda global, por el contrario, si su valor es pequeño la búsqueda se inclinará hacia el tipo local.

**2.1.9 Formulación Matemática.** Como se mencionó, existen tres factores que modifican la posición “actual” de la partícula, la primera es la posición actual, que significa la capacidad que se tiene para enfrentar el problema, la segunda es el aprendizaje, respecto a experiencias anteriores, y la tercera es el conocimiento respecto al movimiento de sus vecinos y el grupo. El algoritmo PSO se planteó para espacios de búsqueda de variables continuas  $n$ -dimensionales y a continuación, se aprecia la fórmula de actualización de ubicación y velocidad de las partículas.

Vector  $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$  almacena la posición actual de la partícula en el espacio de búsqueda.

Vector  $pBest_i = (p_{i1}, p_{i2}, \dots, p_{in})$  almacena la mejor solución encontrada por la partícula hasta el momento.

Vector velocidad  $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$  almacena la dirección según la cual se moverá la partícula.

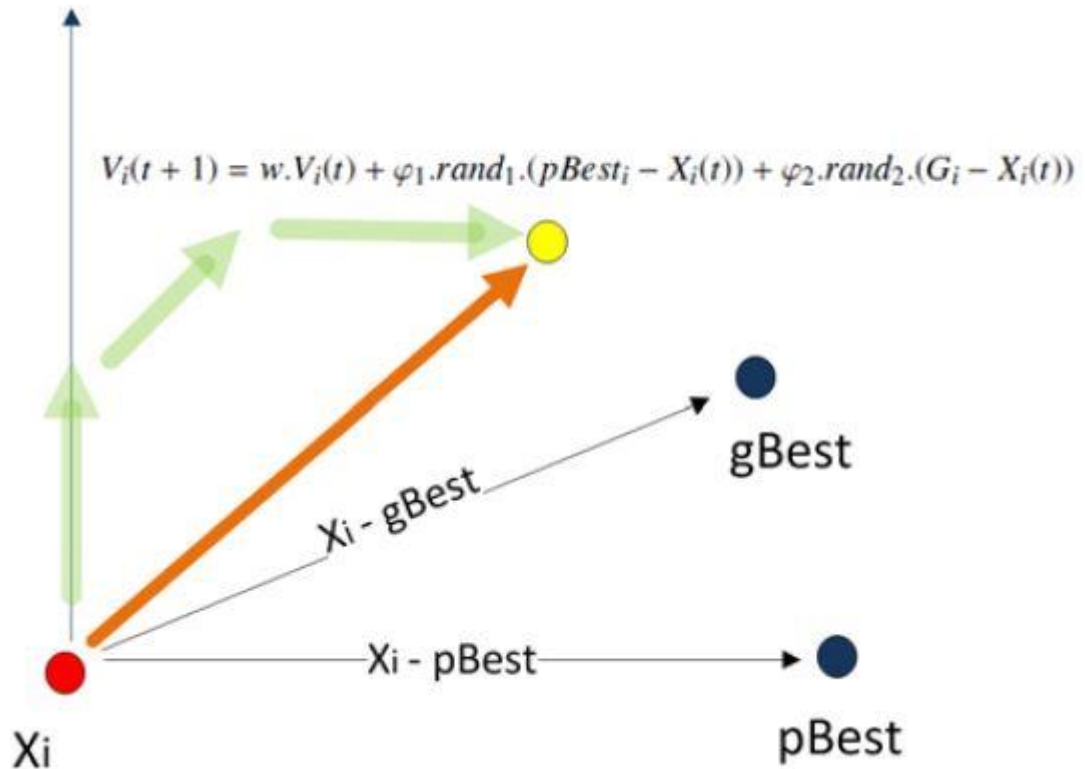
El valor fitness  $fitness_{xi}$  almacena el valor de aptitud de la solución actual (vector  $x_i$ ).

El valor fitness  $fitness_{pBesti}$  almacena el valor de aptitud de la mejor solución local encontrada hasta el momento (vector  $pBesti$ ).

La población se inicializa generando las posiciones y velocidades de las partículas aleatoriamente, y los individuos comienzan a moverse por el espacio de búsqueda a través de un proceso iterativo, la posición del individuo se va actualizando mediante el valor  $fitness_{xi}$ . Además, si el nuevo fitness del individuo resulta ser el mejor encontrado hasta el momento, se actualizan también los valores de posición  $pBesti$  y fitness  $fitness_{pBesti}$ .

Teniendo en cuenta lo que se ha explicado anteriormente del algoritmo, y entendiendo la aproximación matemática, se representa a continuación el movimiento de una partícula en el espacio de soluciones. Se aprecia como las posiciones y velocidades iniciales se calculan al azar hasta que a través de varias generaciones convergen en las mejores soluciones.

**Figura 4.** Movimiento de una partícula en el espacio de soluciones.



Fuente: Optimización Multi Objetivo. Aplicaciones a problemas del mundo real. Mg. Javier López.

El PSO es un proceso metaheurístico, común en los procesos multiobjetivo de optimización combinatoria, es decir que hace muy pocas suposiciones por no decir ninguna, con respecto al problema que busca resolver y se desempeñan muy bien espacios altamente dimensionales. Se pueden aplicar estos métodos a problemas donde las funciones presenten ruido o estén parcialmente definidas, cabe resaltar que ninguna de estas técnicas garantiza que la solución alcanzada sea la óptima, sin embargo, se resalta su sencilla implementación y su convergencia rápida a muy buenas soluciones<sup>9</sup>.

<sup>9</sup> LÓPEZ, Javier. Optimización Multi-objetivo. Aplicaciones al mundo real. Tesis doctoral en ciencias informáticas. La Plata. Editorial de la Universidad de La Plata. 2013.

Existen distintas versiones del algoritmo, ligeramente diferentes entre sí, en las cuales puede que los individuos tengan o no, información, la versión canónica (PSO global) toma la totalidad de los individuos como vecindario de la partícula; en cambio, en el PSO local, se puede restringir el vecindario de la partícula a un subconjunto de la población, se deben aplicar los parámetros adecuados con el fin de evitar la convergencia prematura y conseguir que las mejores ubicaciones se diseminen de forma más lenta.

**Figura 5.** Algoritmo PSO Básico.

---

**Algorithm 2:** Algoritmo PSO Básico

---

```

Pop = CrearPoblacion(N);
while no se alcance la condición de terminación do
    for i = 1 to size(Pop) do
        evaluar partícula  $x_i$  del cúmulo Pop;
        if fitness( $x_i$ ) es mejor que fitness(pBesti) then
             $pBest_i \leftarrow x_i$ ;
            fitness(pBesti)  $\leftarrow fitness(x_i)$ ;
        for i = 1 to size(Pop) do
            elegir  $g_i$  de acuerdo al criterio de vecindario usado;
             $v_i \leftarrow w \cdot v_i + (\varphi_1 \cdot rand_1) \cdot (pBest_i - x_i) + \varphi_2 \cdot rand_2 \cdot (g_i - x_i)$ ;
             $x_i \leftarrow x_i + v_i$ ;
    Result: la mejor solución encontrada
  
```

---

Fuente: Optimización Multi Objetivo. Aplicaciones a problemas del mundo real. Mg. Javier López.

El funcionamiento del algoritmo es bastante sencillo, se comienza con una cantidad inicial de partículas, una de ellas se identificará como la mejor solución (*gBest*), en función de su valor de aptitud (*fitness value*), esta partícula actuará atrayendo a las otras a su posición, y al mismo tiempo todas las partículas se ven atraídas también por sus propias soluciones, sobrepasando la posición de su objetivo y explorando el espacio de búsqueda más allá de la mejor posición actual, todas las partículas pueden sobrepasar la mejor posición y convertirse en un nuevo (*gBest*), así atraerá

nuevas partículas a su posición, haciendo que las otras cambien de dirección hacia ella, y asumiendo la continuidad del espacio, es probable que una buena solución esté rodeada de soluciones igualmente buenas o mejores.

**2.1.10 Alcances.** Gracias a su sencillo concepto y su corta formulación el PSO puede ser adaptado a todos los problemas matemáticos para los que se usen técnicas similares evolutivas, sociales, heurísticas o genéticas. Este algoritmo funciona de manera especial para los problemas que requieren optimización y sus variantes.

Los problemas de optimización con restricciones son quizás el área más explorada y de mayor uso para el PSO, sin embargo requiere de ciertas estrategias para garantizar su funcionamiento dentro de las fronteras que envuelvan el problema, una de ellas es convertir el problema con restricciones en uno sin restricciones aplicando sanciones una vez el algoritmo exceda los límites propuestos, otra estrategia es conservar las soluciones que sean viables y poner atención en reparar las que no lo sean y por último existen los llamados híbridos que funcionan de la misma manera que el algoritmo original pero anclado o sujeto a estrategias matemáticas que faciliten la decodificación estratégica de información estableciendo la dirección de búsqueda, por ejemplo matrices de restricciones que permitan constituir una lista de desempeño (Ray y Liew 2001).

Por otro lado, existen problemas de optimización que requieren la búsqueda de un mínimo y un máximo para las mismas funciones, son llamados problemas Min/Max y se manejan como dos algoritmos ejecutados de manera independiente, sin embargo, ambos PSO cooperan entre sí manejándose como un entorno cambiante del otro.

Una variante de la optimización y quizás la más extensa, por su metodología en combinación con otras técnicas estadísticas, es la optimización de problemas

multiobjetivo (MOO). Son procedimientos que deben llevarse a cabo teniendo el cuidado de optimizar las funciones objetivo sin perjudicar la solución de las demás, esto teniendo en cuenta las restricciones que existan y el nivel de dependencia de las variables entre sí. En estos casos es posible que se encuentren varias soluciones aptas que cumplan matemáticamente con la optimización y por ello es necesario aplicar técnicas como el frente óptimo de Pareto para la selección de las mejores o, en caso de que se requiera una única solución, la mejor de todas. Es necesario decir que existen diferentes enfoques para este tipo de problemas, convertir los MOO en problemas de un solo objetivo es posible mediante la suma de todas las funciones objetivo para conseguir una ponderada que los incluya todos (esto, siempre y cuando el problema lo permita). Otro enfoque es mantener una diversidad de poblaciones usando un cúmulo de partículas líderes de mejor rendimiento y luego migrar hacia partículas seleccionadas del mismo cúmulo que sigan a ciertas líderes seleccionadas a su vez de manera aleatoria sin prestar atención las mejores dentro del vecindario que las rodea. Un último enfoque para la solución de estos problemas es llamado partícula de clasificación no dominada en donde básicamente no existe comparación con el historial propio de la partícula, sino que se compara con el historial de la población total. El algoritmo PSO tiene aplicaciones en problemas mucho más complejos como el seguimiento dinámico, un área poco explorada hasta el momento, en donde las funciones cambian con el tiempo y por lo tanto los valores que las satisfacen, o incluso en problemas con aplicaciones reales: el área de la salud, en donde se ha usado para el análisis de la evolución de redes neuronales, historiales de pesos y hasta para el registro de imágenes biomédicas en 3D, en el área de la electrónica para controlar potencias, voltajes o comparar las respuestas de los motores DC físicos con modelos matemáticos y aún en el área de la ingeniería mecánica para la optimización de geometrías en el diseño.

## 2.2 MARCO DE ANTECEDENTES

### 2.2.1 Optimización geometría de un eje aplicando algoritmos genéticos<sup>10</sup>.

Los Algoritmos genéticos (AG's), que se basan principalmente en los procesos naturales de selección y transmisión de la información genética, simulando los procesos de selección natural con el fin de obtener los mejores resultados tras varias generaciones. Se utilizan principalmente cuando los espacios de búsqueda son bastante amplios, cuando no se sabe si la solución óptima va a estar en un pico o en un valle y cuando se requiere encontrar la solución en un corto tiempo.

En el algoritmo genético simple los individuos se cruzan con cierta probabilidad, después se muta el individuo resultante con otra probabilidad y eventualmente se vuelve a cruzar con otros individuos, ya cruzados y mutados. En el algoritmo NSGA (Non-dominated Sorting Genetic Algorithm) antes de que los individuos se seleccionen para cruzarse, se realiza una clasificación en frentes óptimos de Pareto sobre la base de la determinación de la no dominación de cada individuo; y de acuerdo con el frente en el cual quede clasificada una solución, se le asigna un valor ficticio de adecuación, siendo mayor este valor para las soluciones clasificadas en el primer frente de Pareto y menor para las soluciones clasificadas en el último frente Pareto.

Se presenta un problema de diseño de un eje mecánico, sometido a cargas típicas de un problema de ingeniería, cuyas restricciones se explicarán en su debido momento; Se tiene en cuenta el criterio de la resistencia de materiales, tanto para cargas estáticas como para cargas cíclicas, considerando los concentradores de esfuerzo, buscando identificar y prevenir las posibles fallas. También se debe tener en cuenta el criterio de rigidez, en el cual se consideran las cargas externas, con el

---

<sup>10</sup> GUZMÁN María Alejandra, DELGADO Alberto. Optimización de la geometría de un eje aplicando algoritmos genéticos. *Revista ingeniería e investigación* (58). VOL. 25 No 2, agosto 2005 15-23.

fin de mantener las deformaciones dentro de los rangos permitidos. Se desea optimizar el peso y la deflexión lateral máxima de manera simultánea, lo cual en principio puede parecer contradictorio, ya que, al disminuir el peso, también lo hará su volumen, haciendo que la sección transversal sea más pequeña, disminuyendo la rigidez y aumentando la deflexión lateral máxima.

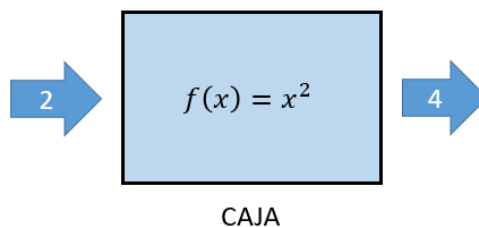
Se asignaron los valores correspondientes según recomendaciones, e inicialmente se mantiene en cero la probabilidad de mutación para probar la efectividad del algoritmo NSGA por sí mismo, lo cual muestra buenos resultados desde alrededor de la generación 20, donde las soluciones van convergiendo en las cercanías del frente óptimo de Pareto; ya en la generación 100 se evidencia claramente este fenómeno con más de 10 soluciones óptimas. También se realizó el proceso para diferentes valores de mutación (0.03, 0.08, 0.1) en 100 generaciones, evidenciando una mayor dispersión con el aumento de este parámetro. En conclusión, se puede decir que la implementación del algoritmo resulta una estrategia eficiente para este tipo de situaciones problema, y que permite hallar de manera efectiva el frente óptimo de Pareto, además, se obtienen distintas configuraciones geométricas igualmente válidas para las mismas condiciones de carga, todas ellas óptimas, tanto para probabilidad cero como diferente de cero, permitiendo al diseñador escoger la que mejor se adapte a las particularidades del problema.

**2.2.2 Proceso de optimización con PSO.** En muchas áreas de la ciencia, sobre todo en control, es usado un procedimiento conocido como identificación de sistemas cuyo objetivo es la construcción de modelos matemáticos con base en datos experimentales de un proceso. En pocas palabras es una herramienta que ayuda a construir un modelo partiendo de los resultados.

Dentro de esta ciencia existen métodos utilizados en computación para determinar parámetros que guíen el sistema que se analiza a un modelo concreto. Los métodos usados generalmente en computación reciben el nombre de pruebas de cajas y se les asigna un color según la forma en que se analice un problema: negro, gris o blanco.

Tal como nos enseñan en el cálculo diferencial las funciones transforman unos valores iniciales en otros por medio de un procedimiento matemático en el que se reemplazan unas variables y se obtienen unos resultados. Así pues, si la función que se quiere analizar es:  $f(x) = x^2$ , al insertar como dato de entrada el número 2, este será reemplazado por la  $x$  y posteriormente elevado al exponente 2 dando como resultado el número 4. La función se comporta como la caja en la cual se introduce un número, luego ocurre un proceso matemático al interior de esta y finalmente entrega un resultado (Figura 6.). Las pruebas de software parten de esta premisa e intentan probar un modelo matemático conociendo o no el procedimiento que realiza la caja y los datos de entrada o salida.

**Figura 6.** Ejemplo de caja.



La prueba de caja blanca es una prueba de software en la que se conoce el proceso por el cual se obtienen los resultados, es decir, es una prueba que se realiza sobre las funciones internas de la caja conociendo el código fuente y el objetivo es analizar los caminos o alternativas que se pueden trazar para obtener los resultados.

**Figura 7.** Esquema de una caja blanca.



La prueba de caja negra es un proceso similar al anterior, pero se realiza sobre las funciones externas del sistema, puesto que no se conoce el código fuente ni el proceso que realiza la caja en su interior para obtener los resultados. El objetivo es probar la funcionalidad del módulo y encontrar errores en el mismo.

**Figura 8.** Esquema de una caja negra.



Existe un último proceso resultado de una mezcla de los dos anteriores llamado prueba de caja gris en la que se conoce parcialmente el interior de la caja y el exterior de la misma.

Según la investigación<sup>11</sup> es posible usar PSO para la identificación de parámetros en funciones de transferencia dentro de un modelo de caja gris. Concretamente se buscó comparar la respuesta o salida (en este caso la velocidad) de un motor DC de 48 V con una potencia nominal de 24 W sometido a un tren de pulsos bipolar con frecuencia de 0,5 Hz con la respuesta de un modelo matemático de control y analizar qué tan parecidos resultaban. El objetivo era encontrar los parámetros (valores numéricos exactos) de la función que modelaba el comportamiento del motor por medio de PSO, permitiendo así modelar el fenómeno de manera casi exacta y en un tiempo muy cortos, 20 segundos para este caso.

El procedimiento fue iterativo con ayuda de la herramienta PSOt de Matlab partiendo de la limitación del espacio de búsqueda de cada parámetro por medio de una ecuación que los describía individualmente, así el código fuente realizó la iteración hasta encontrar los valores óptimos de  $K$ ,  $\tau$  y  $t_d$  que permitieran reducir al mínimo la discrepancia (función objetivo) entre la salida del sistema y la salida del modelo haciendo que fuesen lo más parecidas posible.

**2.2.3 IAO vs PSO.** El problema de optimización abre un campo de estudio bastante amplio para los algoritmos inteligentes y su aplicabilidad en computación. Generalmente el algoritmo cuya implementación sea sencilla y a su vez tenga una precisión significativa y un tiempo de cómputo corto será el de mayor aceptación dentro de la comunidad científica, sin embargo, cumplir con una de estas tres cualidades usualmente implica sacrificar alguna otra, por lo que los estudios basados en comparaciones de métodos son de gran ayuda para identificar el algoritmo que mejor se adapte a la necesidad del usuario.

---

<sup>11</sup> DUARTE, César y QUIROGA, Jabid. Algoritmo PSO para identificación de parámetros en un motor DC. *Revista facultad de ingenierías de la Universidad de Antioquia*. 2010, nro. 55, pp. 116-124

Un estudio realizado en el 2010<sup>12</sup> comparó la efectividad de dos algoritmos, uno de ellos es el de interés para el presente trabajo investigativo, Particle Swarm Optimization (PSO) de clasificación heurística y el otro llamado partición por Análisis de Intervalos (IAO) de naturaleza determinística. Ambos resultaron ser buenos en la solución de las funciones a optimizar, sin embargo, se pudo comprobar que mientras el PSO fue considerablemente más rápido sacrificó precisión y el IAO resultó ser más preciso, pero a su vez más lento.

El PSO empleado fue el algoritmo general cuyo funcionamiento secuencial se describe en 2.1.5. El IAO se implementó con una herramienta computacional llamada INTLAB para encontrar el máximo/mínimo global de algunas funciones sujetas a condición de igualdad y desigualdad.

Los algoritmos fueron puestos a prueba con diferentes funciones y sus respectivas restricciones dentro del marco de tres casos: funciones con una sola variable, con dos variables y con tres variables. Los resultados inclinaron la balanza hacia el uso del PSO. Para el caso de una sola variable el PSO fue muy cercano a la respuesta correcta e incluso llegó al resultado con un tiempo de cómputo de hasta 80 veces más rápido que el IAO. Para el caso de las funciones con dos variables al PSO le faltó precisión, pero manejó una aproximación con una tolerancia de  $10^{-6}$  (bastante cercano) con velocidades de cómputo desde 40 veces hasta 600 veces más rápido que su contrincante. Finalmente, para el caso de tres variables el resultado fue similar, aunque con mayor precisión en comparación con el caso dos, de 20 a 50 veces más rápido para el PSO.

---

<sup>12</sup> VANEGAS ARDILA, Diego; BARRAGÁN NIÑO, Karol y CORREA CELY, Rodrigo. Comparación de las técnicas de optimización por análisis de intervalos y la de enjambre de partículas para funciones con restricciones. *Ingeniería y Universidad*. 2011, vol. 15, nro. 1, pp. 47-60.

Esto nos indica que, frente a otros algoritmos, la optimización por enjambre de partículas puede ayudar a solucionar problemas con una precisión exacta y en los casos en donde quizás pueda ser inexacta sigue siendo considerablemente cercano al valor correcto y matemáticamente aceptable dentro de la optimización, todo esto respaldado por sus dos mayores ventajas: su simple aplicación y su alta velocidad de cómputo para brindar respuestas.

**2.2.4 Optimización multiobjetivo de un eje con algoritmo basado en quimiotaxis de bacterias<sup>13</sup>.** Existe evidencia de la quimiotaxis de bacterias como estrategia optimizada de movimientos, ya que ayuda a la bacteria a sobrevivir encontrando fuentes de nutrientes y evitando ambientes hostiles. Considerando una distribución no homogénea de quimioatrayentes y asumiendo que las direcciones aleatorias sucesivas son independientes y se encuentran distribuidas de manera uniforme, la bacteria: Gira y nada en línea recta en una dirección aleatoria, y si en esa dirección hay gradiente positivo del quimioatrayente, continuará hasta que empiece a decrecer, se mide el decremento y la bacteria gira nuevamente y nada en una dirección aleatoria, si en esta nueva dirección hay gradiente negativo del quimioatrayente, la bacteria se detiene, después gira un tiempo y vuelve a nadar en otra nueva dirección.

En las últimas décadas se ha estudiado el modelamiento de este proceso y sus campos de aplicación, implementándolo en movimientos de robots, redes neuronales e incluso hibridándolo con otros algoritmos, entre ellos el PSO, con el fin de mejorar alguna o varias características. En este caso se aplican los algoritmos a la resolución de Problemas Multiobjetivo, y aplicando en conjunto el concepto de

---

<sup>13</sup> CONGRESO IBEROAMERICANO DE INGENIERIA MECANICA (8, Cusco, 23 al 25 de octubre de 2007). Optimización multiobjetivo de un eje con algoritmo basado en quimiotaxis de bacterias. Pontificia Universidad Católica del Perú, 2007, 9.

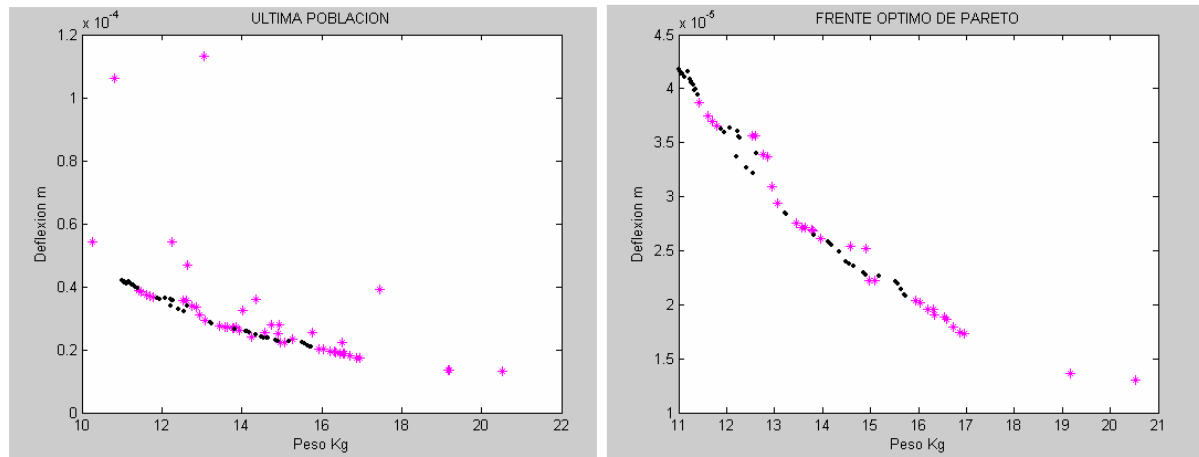
Óptimo de Pareto, se consigue encontrar varias soluciones óptimas, diferentes entre sí.

Se ha formulado el algoritmo de optimización, a partir de estudios previos, sin embargo, cabe mencionar que se parte del NSGA-II para la ubicación de las bacterias, se definen los parámetros adicionales y se implementan las funciones de señal (distancia) entre las bacterias, reproducción (que elimina la mitad de ellas y reproduce las restantes), dispersión y verificación. Se realizan pruebas con tres conocidos problemas multiobjetivo, los primeros dos, con dos funciones cada uno, y el tercero, con dos funciones y dos restricciones, obteniendo excelentes resultados, ya que la última población de bacterias resultaba coincidiendo con el frente óptimo de Pareto para todos los casos, casi en su totalidad.

Al ser satisfactorios los resultados se procede a aplicar el algoritmo quimiotáctico al conocido problema ingenieril del eje mecánico, que ya se ha mencionado con anterioridad, cuyos dos objetivos son conseguir la menor deflexión lateral posible y reducir el peso, debido a la presencia de cargas fluctuantes, se considera la resistencia a la fatiga. Las otras restricciones se encuentran en los diámetros mínimos del eje y la deflexión lateral máxima, se conocen condiciones de carga y la mayoría de los parámetros utilizados, como dimensiones de la polea, potencia del motor, materiales del eje.

En la imagen de la izquierda se presentan los valores de las funciones objetivo obtenidas por medio del algoritmo NSGA en color negro y la totalidad de bacterias de la última población, obtenidos con el algoritmo basado en la quimiotaxis en color magenta; y en la imagen de la derecha se presentan las funciones objetivo para las bacterias ubicadas en el frente óptimo de Pareto obtenido con la aplicación del algoritmo basado en quimiotaxis de bacterias, demostrando la eficiencia y los excelentes resultados de las herramientas bio-inspiradas de optimización eficiente, siempre y cuando los problemas sean planteados y restringidos de forma correcta.

**Figura 9.** Funciones objetivo para la población final de bacterias.



Fuente: Optimización multiobjetivo de un eje con algoritmo basado en quimiotaxis de bacterias. Guzmán, M.A., Delgado, A., De Carvalho, J.

### 3. METODOLOGÍA

Inicialmente se realiza una revisión bibliográfica, que comprende elementos como: los problemas multiobjetivo, algoritmos genéticos, análisis de varianza, frente óptimo de Pareto y principalmente todo lo relacionado al algoritmo PSO, ya que éste es el centro del proyecto, se evalúa qué otros problemas se han resuelto, y se define la posibilidad de aplicarlo a nuestra situación particular. Una vez realizado este proceso se comienza con la formulación del algoritmo en el programa Matlab, esto consiste en el desarrollo de un código fuente, que sea capaz de evaluar las funciones objetivo deseadas de manera simultánea, y que a su vez, acote el espacio de búsqueda según las restricciones previamente planteadas.

Una vez el programa funciona correctamente, se realizan las pruebas correspondientes, con el fin de que cuando se ejecute, reporte las respuestas

óptimas para el diseño del eje, a partir de allí se define el frente óptimo de Pareto y se reportan las mejores respuestas. Se realiza también el análisis correspondiente de las gráficas que ratifican el correcto funcionamiento del problema, como las curvas de nivel y las líneas de campo, acompañadas de algunas otras, cada una con su respectiva descripción. También se realiza el análisis estadístico ANOVA, con el fin de refinar la búsqueda y evaluar cuáles son las variables que, al modificarse, tienen mayor influencia en las respuestas (la masa y la deflexión del eje).

Finalmente se comparan los resultados obtenidos con otros artículos y proyectos, que han resuelto el mismo problema mediante otros métodos y se realizan las observaciones y recomendaciones correspondientes, se reportan los debidos resultados y las conclusiones más importantes.

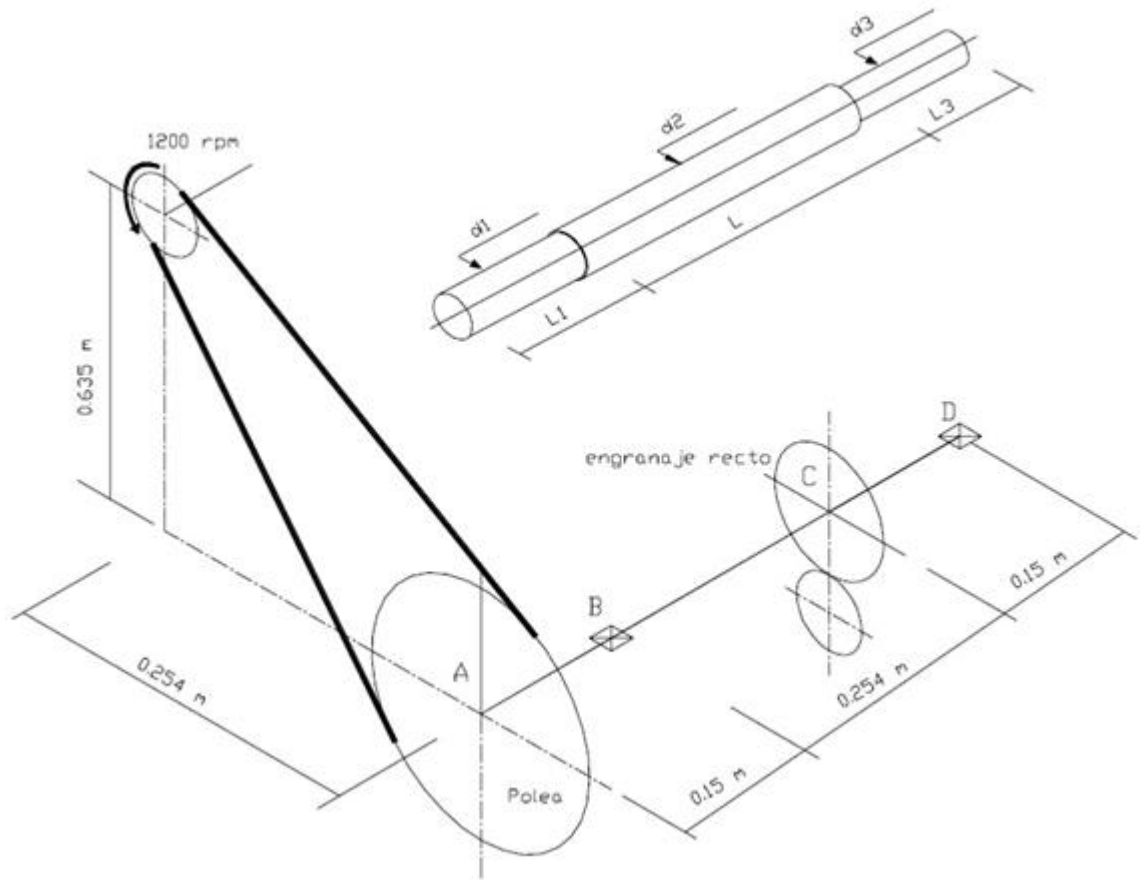
#### **4. CARACTERIZACIÓN DEL PROBLEMA DEL EJE**

Se presenta un problema de diseño de un eje mecánico, sometido a cargas típicas de un problema ingenieril, en el punto A se tiene una polea para correa trapezoidal de diámetro = 20 pulgadas (0.508 m), peso 60 libras (266.88 N), seguido de un apoyo de rodamiento de bolas en el punto B, luego un engranaje recto en C de diámetro = 10 pulgadas (0.254 m), ángulo de presión =  $20^\circ$  y peso = 25 libras (111.2 N) y finalmente otro apoyo de rodamiento de bolas en D. La variación de potencia del motor es de 20 hp (14.91 kW) a 5 hp (3.72 kW) en cada revolución. La velocidad del motor es de 1200 rpm. El eje está hecho de acero AISI 1040 CD, el factor de seguridad contra falla por fatiga  $N_f = 2.2$ , siguiendo las recomendaciones y criterios de conveniencia en diseño mecánico, la deflexión lateral máxima no puede superar las 0.005 pulgadas (0.000127 m).

**Tabla 1.** Datos proporcionados por el problema.

<b>DATOS DEL PROBLEMA</b>		
<b>PUNTO A: POLEA</b>		
Dimensión	Sistema Inglés	Sistema métrico
Diámetro	21 pulg	0,508 m
Peso	60 lb	266,88 N
<b>PUNTO B y D: RODAMIENTO DE BOLAS</b>		
<b>PUNTO C: ENGRANAJE RECTO</b>		
Dimensión	Sistema Inglés	Sistema métrico
Diámetro	10 pulg	0,254 m
Ángulo de presión		20°
Peso	25 lb	111,2 N
<b>MOTOR</b>		
Potencia (por rev)	20 hp - 5hp	14,91 kW - 3,72 kW
Velocidad		1200 rpm
<b>EJE</b>		
Material		AISI 1040 CD
Factor de seguridad		2,2
Deflexión máxima		0,005 pulg (0,000127 m)

**Figura 10.** Eje sometido a cargas cíclicas.



Fuente: GUZMÁN María Alejandra, DELGADO Alberto. Optimización de la geometría de un eje aplicando algoritmos genéticos. Revista ingeniería e investigación (58). VOL. 25 No 2, agosto 2005 15-23.

En la Figura 10 se muestra el significado de las variables  $d_1$ ,  $d_2$ ,  $d_3$ ,  $L_1$ ,  $L_3$  y  $L$ , así como un esquema del montaje de la polea, el engranaje y los apoyos sobre el eje, junto con las principales distancias entre las poleas y los cambios de sección del mismo.

Se aplican las ecuaciones de equilibrio, se hallan las reacciones en los apoyos, los diagramas de fuerzas cortantes, momentos flectores y torsores, finalmente se integra para encontrar la deflexión lateral del eje. [Shigley y Mischke, 2002] Con esto se obtienen las dos ecuaciones a optimizar,  $f_1(\bar{x})$  y  $f_2(\bar{x})$ , la primera en términos de los diámetros y longitudes y la segunda en términos de las deflexiones laterales; también se tienen en cuenta restricciones según la aplicación de la teoría de falla por fatiga de Goodman modificada<sup>14</sup>, así, el problema se plantearía de la siguiente manera:

Minimizar:

$$f_1(\bar{x}) = 0.2213 (d1^2 L1 + d2^2 L + d3^2 L3)$$

Masa del eje

$$f_2(\bar{x}) = \sqrt{(y_{xy})^2 + (y_{xz})^2}$$

Deflexión lateral máxima

Sujeto a:

Para  $i = 1, \dots, 3$

$$g_i(\bar{x}) = d_i - \left\{ 22.4 \left[ \frac{\sqrt{(K_{fi} M_{ai})^2 + \frac{3}{4}(K_{fsi} T_{ai})^2}}{24840(d_i)^{-0.097}} + \frac{\sqrt{(K_{fmi} M_{mi})^2 + \frac{3}{4}(K_{fsm} T_{mi})^2}}{75000} \right]^{\frac{1}{3}} \right\} \geq 0$$

$$g_4(\bar{x}) \equiv 0.005 - y_{lateramaxima} \geq 0$$

$$8 \leq L \leq 14$$

$$L1 = L3 = \frac{22 - L}{2}$$

$$0.5 \leq d1, d2, d3 \leq 5$$

<sup>14</sup> NORTON, Robert L., Diseño de Máquinas: un enfoque integrado. México: Pearson Educación, 2011.

## 5. IMPLEMENTACIÓN Y APLICACIÓN DEL ALGORITMO

### 5.1 Implementación PSO en Matlab

Para implementar el código en Matlab fue necesario seguir un orden lógico de programación y aplicación de las herramientas y funciones propias del problema. Para empezar, se revisó la información pertinente al algoritmo PSO, teniendo en cuenta la bibliografía y los códigos existentes de uso libre que se encuentran en la web. Cabe aclarar que, sin importar las amplias posibilidades que existen para programar una misma función, en el caso de los algoritmos genéticos se sigue un orden básico: definición del problema, definición de parámetros del PSO, inicialización, ciclo principal y resultados.

En la definición del problema se colocan las funciones a optimizar de manera simultánea, en nuestro caso son dos, el Peso y la Deflexión, ambas de distintas formas y con distintas constantes, pero dependientes finalmente de las mismas variables: diámetros y longitudes, es decir la geometría del eje. Se define también el espacio donde se localizarán las partículas inicialmente.

En la definición de los parámetros del PSO se definen las constantes propias que utiliza el algoritmo para su funcionamiento, como son: el número de partículas, número de iteraciones, constantes de aceleración y factores de inercia (máximos y mínimos), alpha (constante para el factor de estrechamiento), y para el método que implementaremos también se define w (pesos específicos que se le dará a las funciones objetivo), y rho (que garantiza una aplicación eficiente del modelo).

Después viene la inicialización del algoritmo, en ella se crean las partículas y se arrojan al espacio de búsqueda, se define también si la velocidad inicial es aleatoria o cero, para nuestro caso es cero; se declaran variables necesarias como las iteraciones, el valor Pbest (mejor posición de la partícula), y Gbest (la del líder del

cúmulo de partículas). Al ejecutar el algoritmo aparece una ventana, en la cual se muestran las partículas iniciales y su evolución durante las generaciones hasta converger en el frente de las soluciones, para ello es necesario entender qué es lo que sucede en el ciclo principal.

Para empezar, se calculan las velocidades iniciales si las hay, así como las constantes de aceleración  $C_1$  y  $C_2$ ,  $\phi$ , y  $K$  (constante de estrechamiento); a continuación, se calculan las nuevas velocidades, basándonos en la experiencia personal y grupal de las partículas, se actualizan también las posiciones, y las funciones de adaptación respectivamente. Después, se actualizan las mejores posiciones de las partículas y se define la mejor de ellas hasta ese momento; finalmente se cuenta una repetición más y se muestra en la ventana el resultado de las nuevas localizaciones de las partículas, este proceso ocurre a alta velocidad y se percibe como un movimiento continuo de dichas partículas; el proceso termina cuando se cumple con el número de iteraciones preestablecidas.

El último paso consiste en mostrar al usuario las diferentes respuestas y variables de salida requeridas, como el conjunto de soluciones, la mejor posición, el tiempo de cómputo, etc.

## **5.2 Aplicación al problema**

Para aplicar el algoritmo PSO al problema del eje es necesario describir el proceso que lleva la resolución de las ecuaciones del mismo y las respectivas restricciones propias del problema.

Como se trata de un eje sometido a carga cíclica, se parte de la ecuación de Goodman modificada, la cual plantea una restricción para cada uno de los 3 diámetros del eje, cada uno correspondiente a una sección del mismo. Ésta ecuación se encuentra en términos de algunas constantes ( $K_f$ ,  $K_{fm}$ ,  $K_{fs}$ ,  $K_{fsm}$ ),

también en términos de Momentos y Torques, Medios y Alternativos ( $M_m$ ,  $M_a$ ,  $T_m$ ,  $T_a$ ), deflexiones en los planos XY y XZ ( $Y_{xy}$   $Y_{xz}$ ) y finalmente en términos de esfuerzos, cuyos valores se conocen ( $S_y$ ,  $S_f$ ).

Para encontrar las primeras constantes mencionadas es necesario valerse del diseño de máquinas, se requiere una suposición inicial, con la que, se hallan unos términos intermedios ( $q_s$ ,  $q_t$ ,  $K_t$ ,  $K_{ts}$ ) y con ayuda de tablas<sup>15</sup> se consigue entregar valores probables para estas constantes, las cuales deben probarse en el algoritmo y modificarse según convenga. Para los Momentos y Torques, se utilizan los conceptos bien sabidos de Estática y Resistencia de Materiales, con el fin de encontrar las expresiones correspondientes para cada una de las secciones del eje en términos de las longitudes. Finalmente se reemplazan las expresiones completas para cada una de las deflexiones, teniendo en cuenta la sección crítica previamente hallada y reemplazando los valores correspondientes, una vez más, en términos de los diámetros y las longitudes. Los valores de  $S_y$  y  $S_f$  se pueden conocer de tablas y distintas fuentes bibliográficas, o simplemente tomar los mismos del problema de referencia.

Es importante mencionar que existen algunas condiciones adicionales que no requieren demostración, y lo que hacen es restringir el problema y ayudar a cumplir las condiciones, expresan un rango de valores dentro de los cuales se encontrarán los diámetros y las longitudes y relacionan la longitud de la sección media ( $L_2$ ) con las otras 2 ( $L_1$  y  $L_3$ ) que son iguales entre ellas.

Como ya se ha descrito, los Momentos, Torques y Deflexiones terminan expresándose en términos de los diámetros y las longitudes y es esto lo que permite la aplicación del PSO a la resolución de nuestro problema, en el paso anterior se definieron las dos funciones objetivo (Peso y Deflexión) con lo que, al aplicar

---

<sup>15</sup> PARADA CORRALES, Alfredo. Fatiga. Bucaramanga. 1990.

correctamente las restricciones y ecuaciones correspondientes, el algoritmo logra hallar los distintos conjuntos de óptimas soluciones una vez el algoritmo converge pasadas ciertas generaciones.

Se deben aplicar algunas restricciones adicionales, como el hecho de que los diámetros de los ejes son normalizados y se fabrican con medidas estándar, que aumentan en 1/16 de pulgada cada vez, esta restricción, para mayor comodidad, se implementa al final del proceso de búsqueda del algoritmo.

### 5.3 Resultados

**5.3.1 Óptimo de Pareto.** En la siguiente figura se muestra la última iteración para una corrida del algoritmo, en el eje X se encuentran los valores de masa y en el eje Y los de deflexión, se aprecia el frente de Pareto formado por las partículas (Línea azul); cada una de ellas representa un conjunto de soluciones para los diámetros y la longitud 2; recordando que la longitud 2 es suficiente para hallar las otras longitudes.

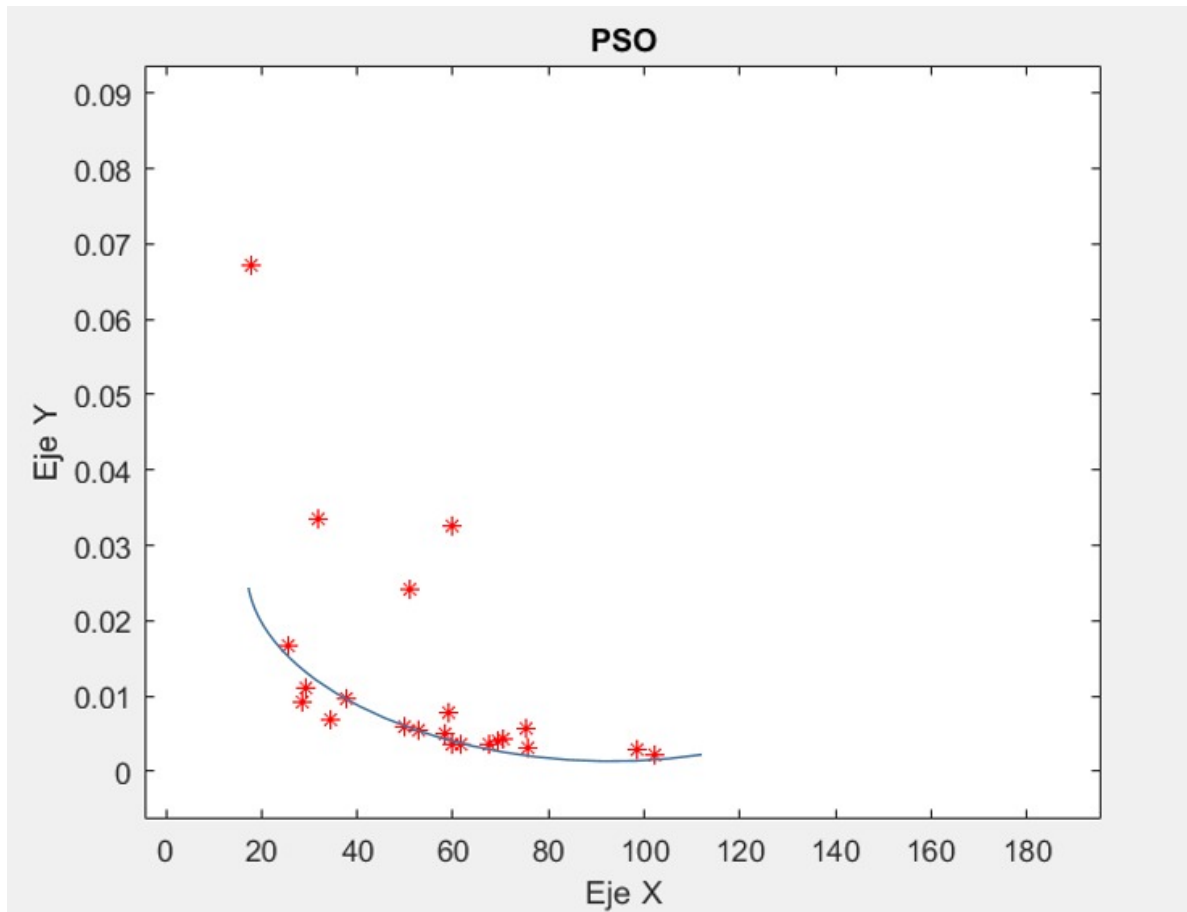
Para plantear un problema multiobjetivo es necesario escoger un método que permita mostrar los resultados y que se adapte a las necesidades del algoritmo PSO, en este caso, se escoge el Método de Pesos Aumentados de Tchebycheff, el cual permite otorgarle importancia a una de las funciones, es decir, se puede orientar la concentración de las partículas hacia una zona específica dentro del frente óptimo de Pareto, dependiendo del tipo de solución que se desee. Consiste en darle un valor de peso a cada función con el objetivo de que el algoritmo distinga la prioridad de optimización.

Para el problema del eje a tratar se decide darle un peso específico mayor a la masa, porque es el resultado con valores más significativos y fáciles de analizar, mientras que la deflexión reporta valores mucho más pequeños, dificultando su análisis, de igual manera esto compensa las restricciones del problema, ya que

pueden presentarse casos en los que la orientación de la optimización beneficie alguna función sobre la otra por las acotaciones propias del problema.

**Figura 11.** Frente óptimo de Pareto obtenido con la última iteración.

Se traza una línea de tendencia que representa los valores óptimos dentro de



todas las posibles soluciones, se escogen 9 soluciones que representan el frente óptimo de Pareto, las cuales se exponen a continuación.

**Tabla 2.** Resultados escogidos a partir del frente óptimo de soluciones.

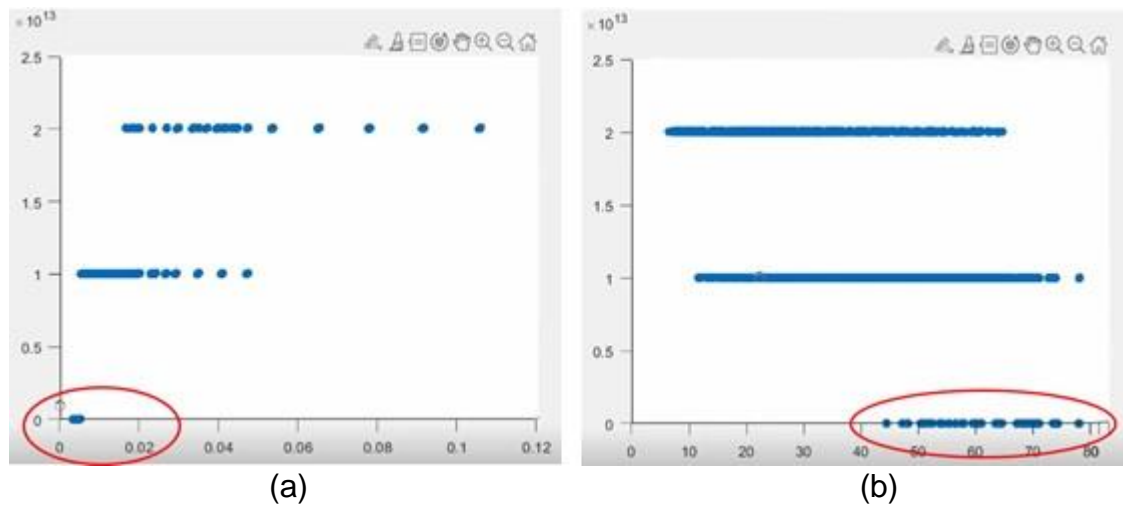
Masa = 47,2495	Masa = 48,3583	Masa = 50,8369
Deflexión = 0,004	Deflexión = 0,0047	Deflexión = 0,0042
d1 = 2,9375	d1 = 3,0625	d1 = 3
d2 = 3,125	d2 = 3,125	d2 = 3,1875
d3 = 3,25	d3 = 3,3125	d3 = 3,5625
L2 = 14	L2 = 13,1	L2 = 13
Masa = 51,9212	Masa = 52,0974	Masa = 54,1045
Deflexión = 0,0042	Deflexión = 0,0049	Deflexión = 0,0044
d1 = 3,0625	d1 = 2,4375	d1 = 3,0625
d2 = 3,25	d2 = 3,1875	d2 = 3,4375
d3 = 3,5	d3 = 4,125	d3 = 3,25
L2 = 13,2	L2 = 13	L2 = 13,6
Masa = 55,2127	Masa = 57,2704	Masa = 57,5878
Deflexión = 0,0038	Deflexión = 0,0038	Deflexión = 0,0038
d1 = 3,1875	d1 = 3,125	d1 = 2,5625
d2 = 3,3125	d2 = 3,43	d2 = 3,5625
d3 = 3,6875	d3 = 3,68	d3 = 3,75
L2 = 13,1	L2 = 13,1	L2 = 14

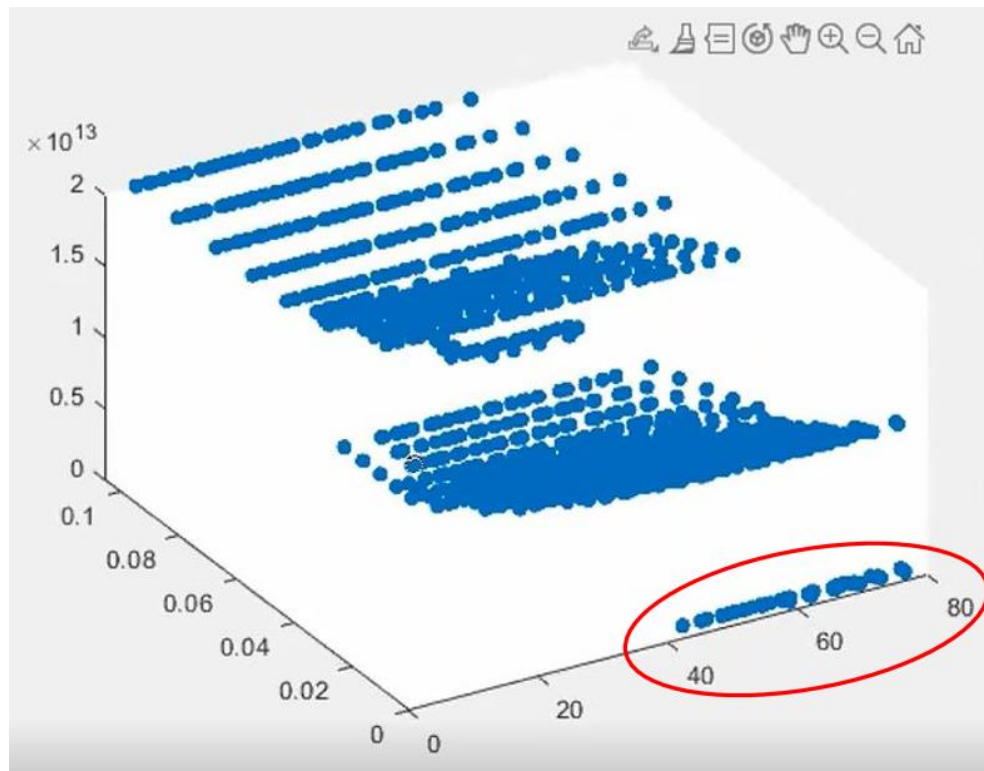
Se reportan los resultados en orden de izquierda a derecha; es evidente cómo el frente óptimo presenta resultados variados a medida que nos desplazamos sobre la curva. Los valores de masa van aumentando progresivamente, empezando en 47 Lb y alcanzando valores de 56 Lb mientras que la deflexión varía muy poco su valor, comenzando en 0,0044 in y disminuyendo hasta las 0,0038 in.

**5.3.1.1 Elección De La Mejor Respuesta.** El diseñador debe seleccionar una única configuración dentro de todas las posibles geometrías para la elaboración del eje y

es necesario establecer ciertos criterios que garanticen una correcta elección. Es posible determinar las zonas de optimización del algoritmo en función de la geometría del eje parametrizando los factores que la componen ( $d_1$ ,  $d_2$ ,  $d_3$ ,  $L_2$ ) y efectuando una serie de combinaciones factibles entre ellos (1715 combinaciones). La figura 12. muestra todos los resultados, de la función objetivo combinada, arrojados por el algoritmo para todas las combinaciones de geometría con su respectivo valor de deflexión y peso. El análisis consiste en identificar las zonas para las cuales todos los valores son mínimos, es decir, los valores de la función objetivo que no son penalizados por el algoritmo (c), la deflexión mínima dentro del rango de la deflexión [0 in - 0.05 in] (a) y un bajo peso que sostenga las consideraciones anteriores (b).

**Figura 12.** Zonas de optimización del algoritmo PSO.





(c)

Para escoger un único valor óptimo es importante aplicar el criterio ingenieril y evaluar las respectivas restricciones ya mencionadas, las mejores soluciones estarán entre las 45 lb y 65 lb aproximadamente y se tomará su valor asociado de deflexión, ya que todos resultan ser buenos, para este problema se escoge la sexta solución, que se encuentra en la tercera columna, segunda fila (Tabla 2.).

Esta selección se da bajo un criterio de montaje, en el que se prefiere que el eje conserve cierta simetría, y reporta un valor de 54 lb para la masa con una deflexión de 0,0044 in cumpliendo completamente con los requerimientos del problema.

El tiempo empleado por el algoritmo para el frente de Pareto mostrado anteriormente fue de: 28,350087 segundos.

**5.3.2 Implementación ANOVA.** El Análisis de varianza ANOVA permite identificar cuáles de los factores de entrada del algoritmo (valores que deben ser especificados antes de correrlo) son influyentes en la respuesta y el tiempo de cómputo. Para realizar esta observación se determinaron 5 parámetros de entrada cada uno con dos niveles: Número de Iteraciones (80 y 160), número de partículas (15 y 35), Peso inercial (0.6 y 0.7), valores de aceleración (1.2 y 1.5) y la velocidad inicial de las partículas (0 y 1). Para las variables dependientes o de respuesta se escogió el tiempo de cómputo y el valor de Gbest para la función objetivo masa, este último en representación de la convergencia.

**5.3.2.1 Análisis De La Variable De Respuesta Convergencia.** La convergencia es un factor que no se puede expresar numéricamente, por esta razón se escogió un valor de respuesta generado por el algoritmo que permita visualizar este fenómeno.

El Gbest para la masa es el valor en libras del menor peso histórico, o el mínimo alcanzado por las partículas en el total de iteraciones y gracias a un análisis previo (mostrado en la discusión) se logró determinar que debe estar alrededor de las 40 libras para garantizar una correcta convergencia.

Se realizó una tabla de la respuesta para la función objetivo 1 (Masa) con las posibles 32 combinaciones del diseño de experimentos, para cada una de estas combinaciones el programa se corrió por duplicado y se halló la media entre estos valores (Tabla 3) que finalmente se consignó como la respuesta en la tabla 4.

**Tabla 3.** Duplicado del experimento para la Masa y su media

MASA		MEDIA
1	2	
19,1497	16,9932	18,07145
45,0694	42,5403	43,80485
40,9178	42,2467	41,58225
41,5143	44,1388	42,82655
44,7163	42,6179	43,6671
42,6177	39,97	41,29385
38,2443	38,9599	38,6021
39,0881	39,7787	39,4334
20,6728	15,8405	18,25665
40,6194	42,1331	41,37625
42,0985	21,6782	31,88835
42,6185	41,9558	42,28715
41,9313	41,0005	41,4659
28,0149	16,6137	22,3143
42,5137	44,7955	43,6546
39,5838	40,9697	40,27675
44,7676	39,121	41,9443
41,1233	23,0299	32,0766
43,6601	42,9295	43,2948
15,3697	40,0893	27,7295
45,0472	56,6021	50,82465
39,8323	40,1367	39,9845
39,4668	40,589	40,0279
43,2926	21,0935	32,19305
44,6824	39,4529	42,06765
42,5024	43,1394	42,8209
40,8898	38,2651	39,57745
46,3927	16,4843	31,4385
40,1369	17,9777	29,0573
45,1308	40,6152	42,873
40,1509	42,5879	41,3694
41,7249	40,8138	41,26935

**Tabla 4.** Resultados de masa para las 32 combinaciones.

OrdenEst	OrdenCorrida	TipoPt	Bloques	Iteraciones	Num Partículas	Inercia	Aceleración	Velocidad	Masa
11	1	1	1	80	35	0,6	1,5	0	18,07145
1	2	1	1	80	15	0,6	1,2	0	43,80485
27	3	1	1	160	35	0,6	1,5	0	41,58225
22	4	1	1	160	15	0,7	1,2	1	42,82655
20	5	1	1	160	15	0,6	1,5	1	43,6671
28	6	1	1	160	35	0,6	1,5	1	41,29385
26	7	1	1	160	35	0,6	1,2	1	38,6021
18	8	1	1	160	15	0,6	1,2	1	39,4334
7	9	1	1	80	15	0,7	1,5	0	18,25665
19	10	1	1	160	15	0,6	1,5	0	41,37625
13	11	1	1	80	35	0,7	1,2	0	31,88835
10	12	1	1	80	35	0,6	1,2	1	42,28715
4	13	1	1	80	15	0,6	1,5	1	41,4659
15	14	1	1	80	35	0,7	1,5	0	22,3143
23	15	1	1	160	15	0,7	1,5	0	43,6546
12	16	1	1	80	35	0,6	1,5	1	40,27675
2	17	1	1	80	15	0,6	1,2	1	41,9443
16	18	1	1	80	35	0,7	1,5	1	32,0766
9	19	1	1	80	35	0,6	1,2	0	43,2948
31	20	1	1	160	35	0,7	1,5	0	27,7295
6	21	1	1	80	15	0,7	1,2	1	50,82465
32	22	1	1	160	35	0,7	1,5	1	39,9845
21	23	1	1	160	15	0,7	1,2	0	40,0279
8	24	1	1	80	15	0,7	1,5	1	32,19305
24	25	1	1	160	15	0,7	1,5	1	42,06765
17	26	1	1	160	15	0,6	1,2	0	42,8209
25	27	1	1	160	35	0,6	1,2	0	39,57745
3	28	1	1	80	15	0,6	1,5	0	31,4385
30	29	1	1	160	35	0,7	1,2	1	29,0573
5	30	1	1	80	15	0,7	1,2	0	42,873
29	31	1	1	160	35	0,7	1,2	0	41,3694
14	32	1	1	80	35	0,7	1,2	1	41,26935

A continuación, se muestran los cálculos, tablas y factores hallados con el análisis de varianza obtenidos con la herramienta minitab:

## Diseño factorial de múltiples niveles

### Resumen del diseño

Factores:	5	Réplicas:	1
Corridas base:	32	Total de corridas:	32
Bloques base:	1	Total de bloques:	1
Número de niveles: 2; 2; 2; 2; 2			

## Modelo lineal general: Masa vs. Iteraciones; Num ... eración; Velocidad

### Método

Codificación de factores (-1; 0; +1)

### Información del factor

Factor	Tipo	Niveles	Valores
Iteraciones	Fijo	2	80; 160
Num Partículas	Fijo	2	15; 35
Inercia	Fijo	2	0,6; 0,7
Aceleración	Fijo	2	1,2; 1,5
Velocidad	Fijo	2	0; 1

### Análisis de Varianza

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Iteraciones	1	115,49	115,49	2,75	0,109
Num Partículas	1	144,50	144,50	3,45	0,075
Inercia	1	86,21	86,21	2,06	0,163
Aceleración	1	278,79	278,79	6,65	0,016
Velocidad	1	149,60	149,60	3,57	0,070
Error	26	1090,02	41,92		

Total 31 1864,61

## Resumen del modelo

S	R-cuad.	R-cuad. (ajustado)	R-cuad. (pred)
6,47486	41,54%	30,30%	11,45%

## Coeficientes

Término	Coef	EE del coef.	Valor T	Valor p	FIV
Constante	37,79	1,14	33,02	0,000	
Iteraciones					
80	-1,90	1,14	-1,66	0,109	1,00
Num Partículas					
15	2,13	1,14	1,86	0,075	1,00
Inercia					
0,6	1,64	1,14	1,43	0,163	1,00
Aceleración					
1,2	2,95	1,14	2,58	0,016	1,00
Velocidad					
0	-2,16	1,14	-1,89	0,070	1,00

## Ecuación de regresión

$$\begin{aligned} \text{Masa} = & 37,79 - 1,90 \text{ Iteraciones}_{80} + 1,90 \text{ Iteraciones}_{160} + 2,13 \text{ Num Partículas}_{15} \\ & - 2,13 \text{ Num Partículas}_{35} + 1,64 \text{ Inercia}_{0,6} - 1,64 \text{ Inercia}_{0,7} + 2,95 \text{ Aceleración}_{1,2} \\ & - 2,95 \text{ Aceleración}_{1,5} - 2,16 \text{ Velocidad}_0 + 2,16 \text{ Velocidad}_1 \end{aligned}$$

## Ajustes y diagnósticos para observaciones poco comunes

Obs	Masa	Ajuste	Resid	Resid est.	
1	18,07	30,30	-12,22	-2,09	R
9	18,26	31,26	-13,01	-2,23	R
29	29,06	41,04	-11,98	-2,05	R

Residuo grande R

## Regresión factorial general: Masa vs. Iteraciones; Num ... n; Velocidad

### Información del factor

Factor	Niveles	Valores
Iteraciones	2	80; 160
Num Partículas	2	15; 35
Inercia	2	0,6; 0,7
Aceleración	2	1,2; 1,5
Velocidad	2	0; 1

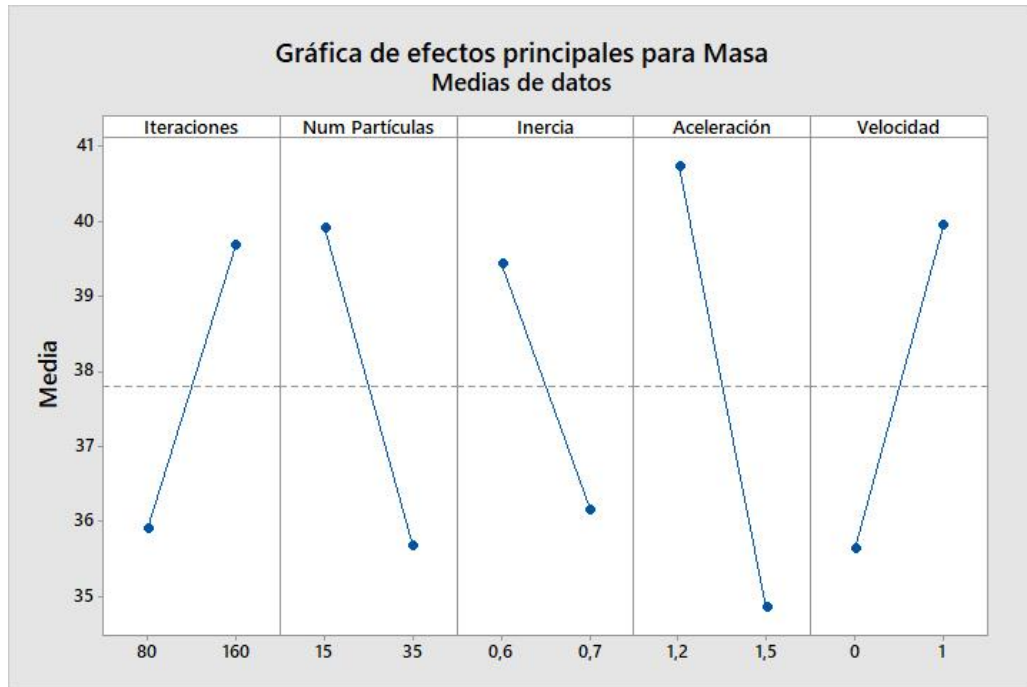
### Análisis de Varianza

Fuente	GL	SC Ajust.	MC Ajust.
Modelo	31	1864,61	60,149
Lineal	5	774,59	154,918
Iteraciones	1	115,49	115,486
Num Partículas	1	144,50	144,501
Inercia	1	86,21	86,210
Aceleración	1	278,79	278,790
Velocidad	1	149,60	149,602
Interacciones de 2 términos	10	743,58	74,358
Iteraciones*Num Partículas	1	0,90	0,896
Iteraciones*Inercia	1	2,67	2,675
Iteraciones*Aceleración	1	376,30	376,298
Iteraciones*Velocidad	1	160,21	160,212
Num Partículas*Inercia	1	21,24	21,238
Num Partículas*Aceleración	1	1,29	1,288
Num Partículas*Velocidad	1	2,45	2,448
Inercia*Aceleración	1	26,77	26,767
Inercia*Velocidad	1	7,20	7,203
Aceleración*Velocidad	1	144,56	144,558
Interacciones de 3 términos	10	115,65	11,565

Iteraciones*Num Partículas*Inercia	1	15,57	15,566
Iteraciones*Num Partículas*Aceleración	1	5,92	5,921
Iteraciones*Num Partículas*Velocidad	1	4,29	4,294
Iteraciones*Inercia*Aceleración	1	6,67	6,666
Iteraciones*Inercia*Velocidad	1	2,08	2,077
Iteraciones*Aceleración*Velocidad	1	6,96	6,957
Num Partículas*Inercia*Aceleración	1	30,78	30,776
Num Partículas*Inercia*Velocidad	1	8,90	8,901
Num Partículas*Aceleración*Velocidad	1	27,53	27,533
Inercia*Aceleración*Velocidad	1	6,96	6,955
Interacciones de 4 términos	5	149,91	29,983
Iteraciones*Num Partículas*Inercia*Aceleración	1	67,16	67,159
Iteraciones*Num Partículas*Inercia*Velocidad	1	6,73	6,729
Iteraciones*Num Partículas*Aceleración*Velocidad	1	10,40	10,397
Iteraciones*Inercia*Aceleración*Velocidad	1	56,58	56,581
Num Partículas*Inercia*Aceleración*Velocidad	1	9,05	9,048
Interacciones de 5 términos	1	80,88	80,876
Iteraciones*Num Partículas*Inercia*Aceleración*Velocidad	1	80,88	80,876
Error	0	*	*
Total	31	1864,61	

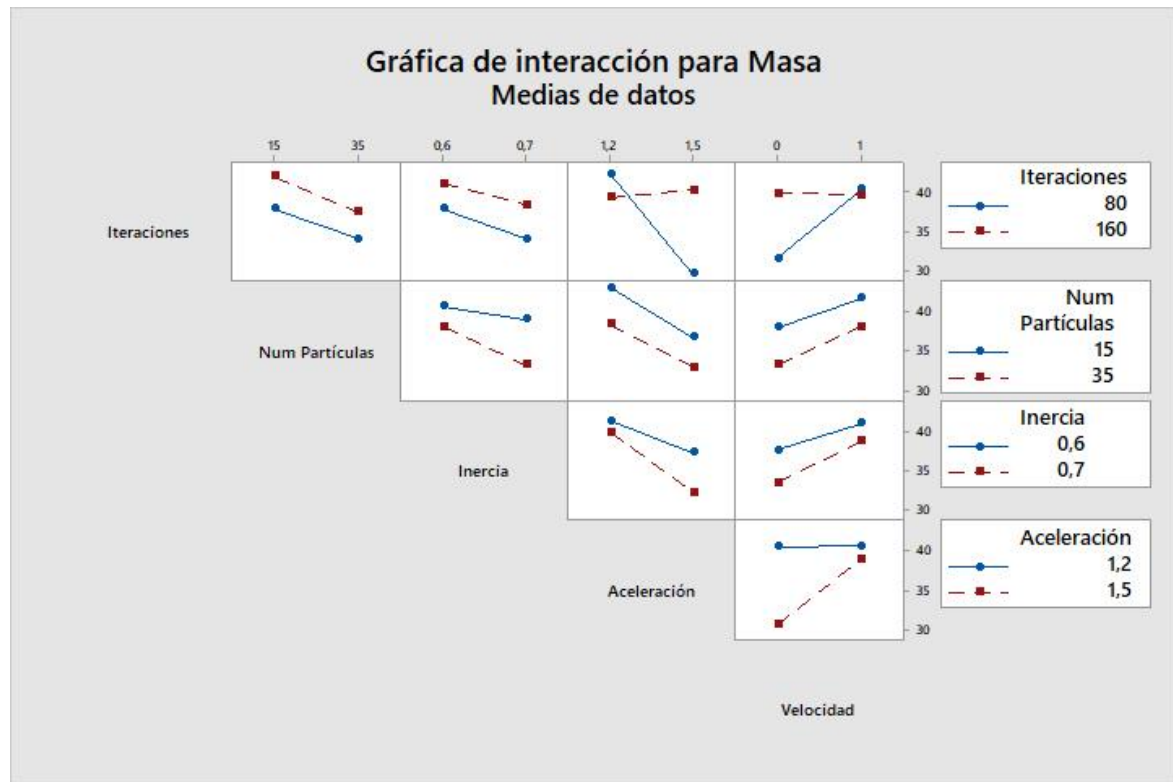
La figura 11 muestra la influencia de los cinco factores por separado sobre la respuesta masa (Eje Y). Es sabido que la tendencia correcta de optimización es hacia las 40 libras. Para las iteraciones y la velocidad se muestra una relación directamente proporcional con la masa, es decir que mientras mayor sea el valor de estos factores la masa se acerca al valor deseado, con el número de partículas, inercia y aceleración ocurre una relación inversa, a menor valor mayor valor de masa.

**Figura 13.** Gráfica de efector principales para masa.



La figura 13 muestra la interacción que existe entre los factores y la influencia que tienen sobre la masa en diferentes combinaciones. Se determina si existe una fuerte interacción entre dos factores si hay una intersección entre líneas mientras que un paralelismo evidencia una baja o nula afectación. La correlación entre el número de iteraciones y la aceleración muestra una tendencia directa entre sí, es decir, los mayores valores de ambos factores orientan al algoritmo hacia los mayores valores de masa, que en este caso son los óptimos garantizan la convergencia. Este fenómeno ocurre de la misma forma para Iteraciones vs velocidad, aceleración vs inercia y aceleración vs velocidad. El paralelismo muestra un comportamiento de igualdad entre los factores, el caso de la inercia vs iteraciones evidencia que los valores de inercia afectan de la misma forma ambos valores de iteraciones, es decir a menor inercia mayor masa alcanzan.

**Figura 14.** Gráfica de interacción para masa.



**5.3.2.2 Análisis De La Variable De Respuesta Tiempo.** Para la respuesta tiempo se realizó el mismo análisis, tomando dos valores de respuesta y hallando la media entre estos (Tabla 5), el cual fue consignado en el diseño de experimentos ANOVA (Tabla 6).

**Tabla 5.** Duplicado del experimento para el tiempo y su media

Tiempo		Media
T1	T2	
22,32	21,62	21,97
11,86	11,47	11,665
44,47	41,56	43,015
22,77	22,67	22,72
22,41	22,85	22,63
44,45	43,76	44,105
44,23	46,85	45,54
24,05	25,05	24,55
12,62	12,34	12,48
24,81	25,38	25,095
22,88	23,68	23,28
24,42	23,81	24,115
13,44	13,04	13,24
23,61	23,46	23,535
26,56	26,16	26,36
23,8	24,16	23,98
12,89	13,42	13,155
23,74	23,68	23,71
23,78	24,53	24,155
48,35	46,3	47,325
13,84	13,22	13,53
47,99	47,77	47,88
28,6	32,94	30,77
14,19	14,05	14,12
27,26	26,94	27,1
26,54	27,09	26,815
47,52	47,61	47,565
13,79	14,17	13,98
48,98	49,22	49,1
14,02	13,69	13,855
48,11	49,52	48,815
24,48	26,43	25,455

**Tabla 6.** Resultados de tiempo para las 32 combinaciones.

OrdenEst	OrdenCorrida	TipoPt	Bloques	Iteraciones	Num Partículas	Inercia	Aceleración	Velocidad	Tiempo
11	1	1	1	80	35	0,6	1,5	0	21,97
1	2	1	1	80	15	0,6	1,2	0	11,665
27	3	1	1	160	35	0,6	1,5	0	43,015
22	4	1	1	160	15	0,7	1,2	1	22,72
20	5	1	1	160	15	0,6	1,5	1	22,63
28	6	1	1	160	35	0,6	1,5	1	44,105
26	7	1	1	160	35	0,6	1,2	1	45,54
18	8	1	1	160	15	0,6	1,2	1	24,55
7	9	1	1	80	15	0,7	1,5	0	12,48
19	10	1	1	160	15	0,6	1,5	0	25,095
13	11	1	1	80	35	0,7	1,2	0	23,28
10	12	1	1	80	35	0,6	1,2	1	24,115
4	13	1	1	80	15	0,6	1,5	1	13,24
15	14	1	1	80	35	0,7	1,5	0	23,535
23	15	1	1	160	15	0,7	1,5	0	26,36
12	16	1	1	80	35	0,6	1,5	1	23,98
2	17	1	1	80	15	0,6	1,2	1	13,155
16	18	1	1	80	35	0,7	1,5	1	23,71
9	19	1	1	80	35	0,6	1,2	0	24,155
31	20	1	1	160	35	0,7	1,5	0	47,325
6	21	1	1	80	15	0,7	1,2	1	13,53
32	22	1	1	160	35	0,7	1,5	1	47,88
21	23	1	1	160	15	0,7	1,2	0	30,77
8	24	1	1	80	15	0,7	1,5	1	14,12
24	25	1	1	160	15	0,7	1,5	1	27,1
17	26	1	1	160	15	0,6	1,2	0	26,815
25	27	1	1	160	35	0,6	1,2	0	47,565
3	28	1	1	80	15	0,6	1,5	0	13,98
30	29	1	1	160	35	0,7	1,2	1	49,1
5	30	1	1	80	15	0,7	1,2	0	13,855
29	31	1	1	160	35	0,7	1,2	0	48,815
14	32	1	1	80	35	0,7	1,2	1	25,455

A continuación, se muestran los cálculos, tablas y factores hallados con el análisis de varianza obtenidos con la herramienta Minitab:

## Diseño factorial de múltiples niveles

### Resumen del diseño

Factores: 5 Réplicas: 1  
 Corridas base: 32 Total de corridas: 32  
 Bloques base: 1 Total de bloques: 1  
 Número de niveles: 2; 2; 2; 2; 2

## Regresión factorial general: Tiempo vs. Iteraciones; Num ... ; Velocidad

### Información del factor

Factor	Niveles	Valores
Iteraciones	2	80; 160
Num Partículas	2	15; 35
Inercia	2	0,6; 0,7
Aceleración	2	1,2; 1,5
Velocidad	2	0; 1

### Análisis de Varianza

Fuente	GL	SC Ajust.	MC Ajust.
Modelo	31	4793,15	154,62
Lineal	5	4508,29	901,66
Iteraciones	1	2505,61	2505,61
Num Partículas	1	1976,32	1976,32
Inercia	1	18,70	18,70
Aceleración	1	6,62	6,62
Velocidad	1	1,03	1,03
Interacciones de 2 términos	10	254,40	25,44

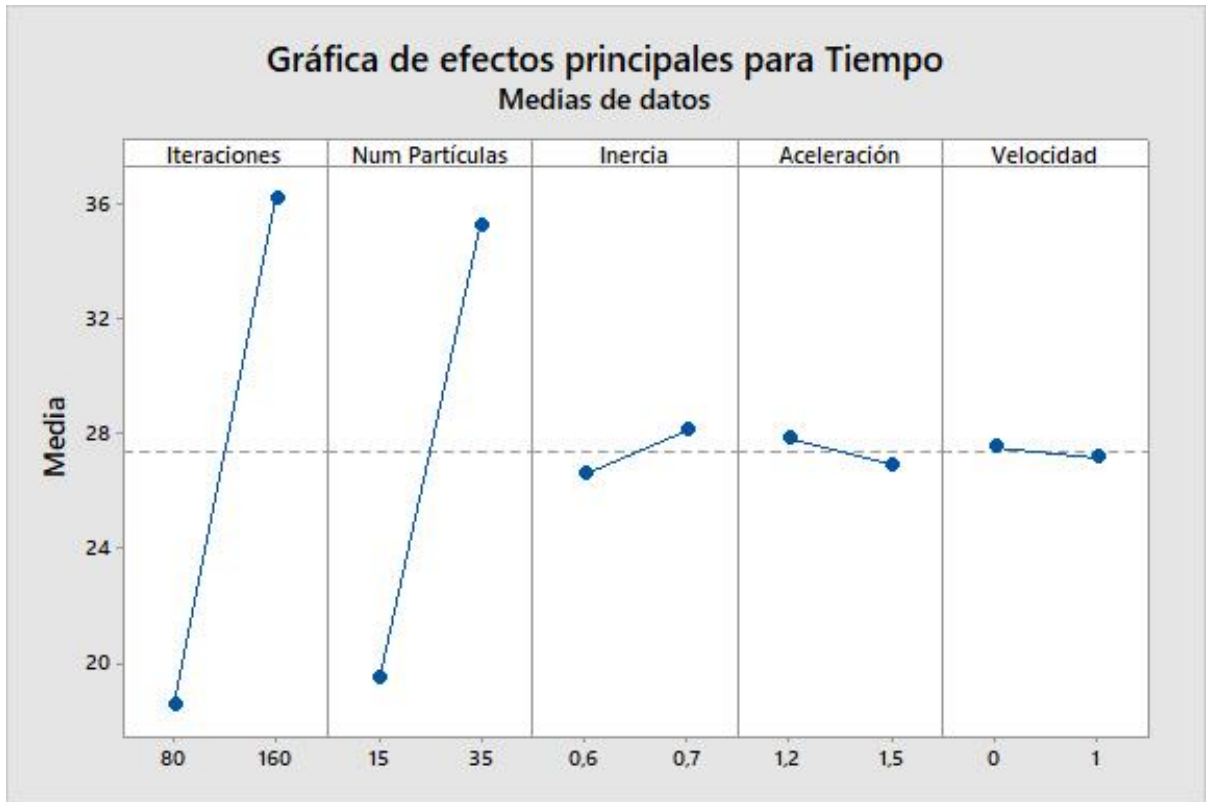
Iteraciones*Num Partículas	1	215,96	215,96
Iteraciones*Inercia	1	9,08	9,08
Iteraciones*Aceleración	1	3,23	3,23
Iteraciones*Velocidad	1	10,72	10,72
Num Partículas*Inercia	1	0,74	0,74
Num Partículas*Aceleración	1	3,41	3,41
Num Partículas*Velocidad	1	6,30	6,30
Inercia*Aceleración	1	0,64	0,64
Inercia*Velocidad	1	0,00	0,00
Aceleración*Velocidad	1	4,32	4,32
Interacciones de 3 términos	10	14,15	1,41
Iteraciones*Num Partículas*Inercia	1	0,85	0,85
Iteraciones*Num Partículas*Aceleración	1	0,01	0,01
Iteraciones*Num Partículas*Velocidad	1	2,93	2,93
Iteraciones*Inercia*Aceleración	1	2,67	2,67
Iteraciones*Inercia*Velocidad	1	0,10	0,10
Iteraciones*Aceleración*Velocidad	1	4,64	4,64
Num Partículas*Inercia*Aceleración	1	0,42	0,42
Num Partículas*Inercia*Velocidad	1	0,54	0,54
Num Partículas*Aceleración*Velocidad	1	0,75	0,75
Inercia*Aceleración*Velocidad	1	1,24	1,24
Interacciones de 4 términos	5	15,91	3,18
Iteraciones*Num Partículas*Inercia*Aceleración	1	0,59	0,59
Iteraciones*Num Partículas*Inercia*Velocidad	1	0,64	0,64
Iteraciones*Num Partículas*Aceleración*Velocidad	1	0,95	0,95
Iteraciones*Inercia*Aceleración*Velocidad	1	1,13	1,13
Num Partículas*Inercia*Aceleración*Velocidad	1	12,60	12,60
Interacciones de 5 términos	1	0,40	0,40
Iteraciones*Num Partículas*Inercia*Aceleración*Velocidad	1	0,40	0,40
Error	0	*	*
Total	31	4793,15	

## Medias

Término	Media ajustada	Error estándar de la media
Iteraciones		
80	18,51	*
160	36,21	*
Num Partículas		
15	19,50	*
35	35,22	*
Inercia		
0,6	26,60	*
0,7	28,13	*
Aceleración		
1,2	27,82	*
1,5	26,91	*
Velocidad		
0	27,54	*
1	27,18	*

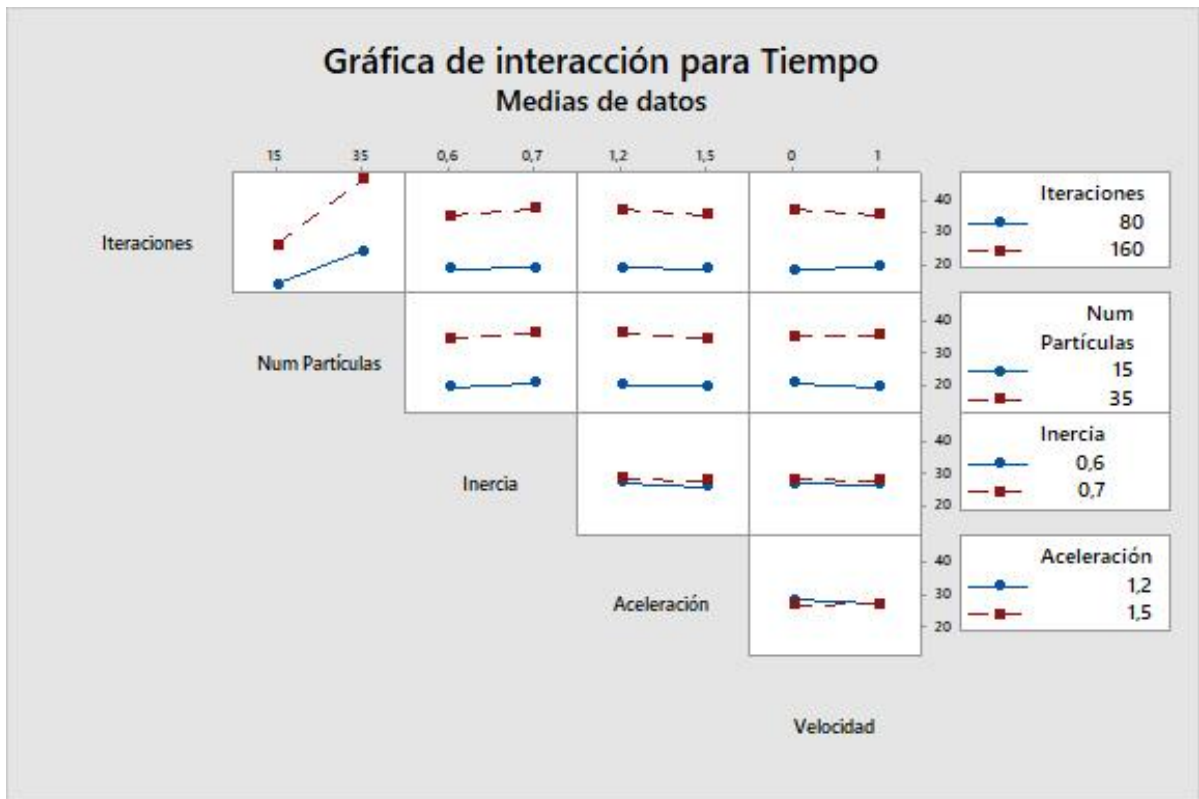
En la siguiente gráfica se aprecia cómo a medida que se incrementan las iteraciones y el número de partículas aumenta considerablemente el tiempo de cómputo, mientras que la variación de la Inercia, Aceleración y velocidad no representan mayores cambios en éste.

**Figura 15.** Gráfica de efectos principales para Tiempo.



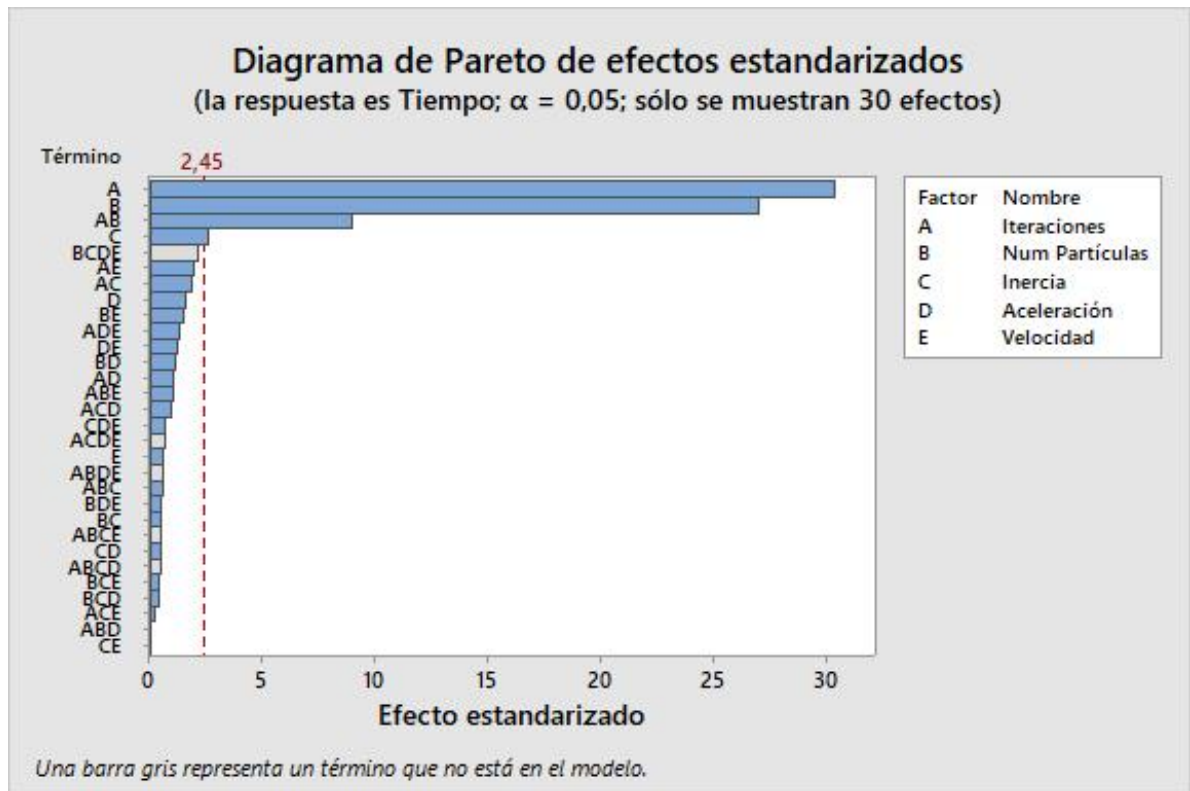
Para las Interacciones entre variables (Figura 16), es evidente el paralelismo entre la mayoría de los términos, es decir, que a medida que crece o decrece uno de los parámetros, el otro lo hace de manera proporcional, como en la relación Iteraciones vs Aceleración, o en Número de partículas vs Velocidad. En cambio, para la relación Aceleración vs. Velocidad, se aprecia cómo se cruzan las líneas de aceleración, indicando que ésta se ve afectada por el cambio de las velocidades.

**Figura 16.** Gráfica de interacción para Tiempo.



En la siguiente gráfica se evidencian los factores más influyentes en la varianza; se muestran de manera individual, y las posibles combinaciones entre ellos. Los factores que se sobrepasen la línea de referencia que está en 2,45 son estadísticamente significativos, es decir, que causan un gran efecto en la variable implicada; para este caso, las Iteraciones, el Número de partículas, la unión de los anteriores y la Inercia (en menor medida), son los que más afectan el Tiempo de respuesta del algoritmo.

**Figura 17.** Gráfica de Pareto de efectos estandarizados para Tiempo.



## 6. DISCUSIÓN

### 6.1 Análisis de la optimización

Es necesario evaluar la eficacia del algoritmo mediante un análisis de convergencia en donde se visualice que las respuestas arrojadas tienden a estabilizarse en una zona del espacio de búsqueda. Al tratarse de una función multiobjetivo el análisis de convergencia debe tener en cuenta los valores óptimos para ambas funciones objetivo y para facilitarlas éstas se parametrizan en forma de una variable adimensional llamada Gbest combinado, que representa la mejor posición histórica de las partículas (Gbest) para la masa y la deflexión simultáneamente.

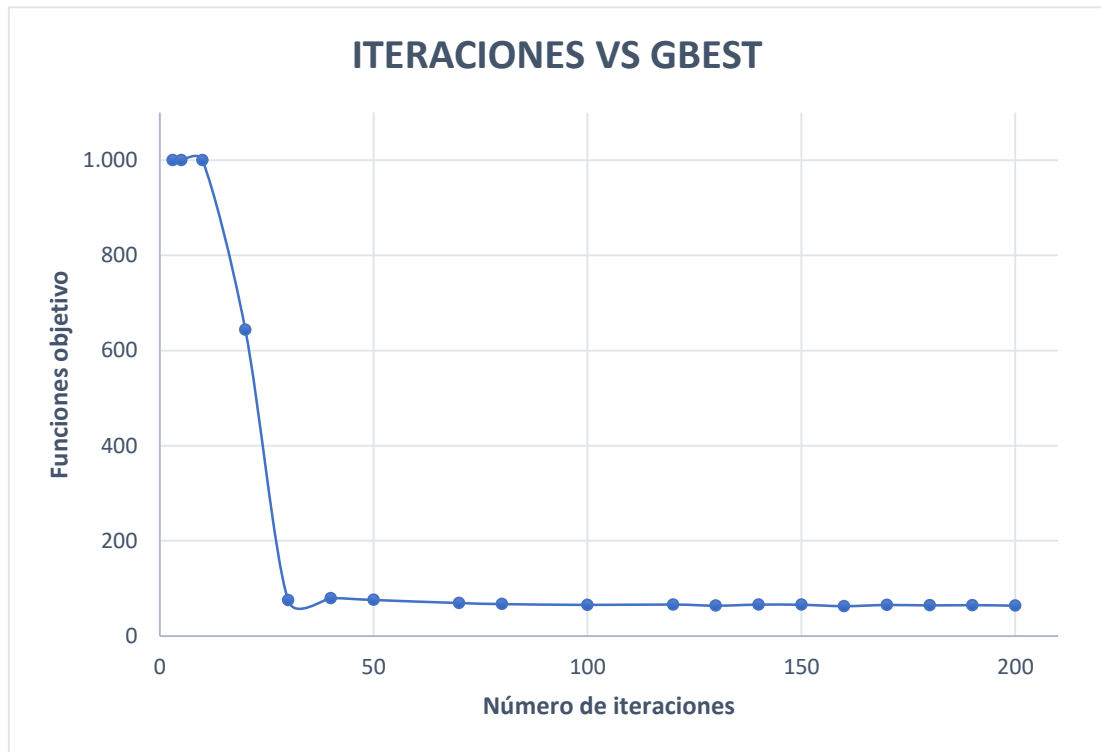
El programa se corrió cinco veces para cada número de iteraciones, se halló una media de estos valores y se grafican. Adicional a esto se halla el error entre el valor de la iteración actual con la anterior y finalmente una desviación estándar que permite ver que tan distantes son los cinco datos obtenidos en cada iteración (Tabla 5).

**Tabla 7.** Datos obtenidos para el análisis de convergencia.

PESO 1-0	n=25								
ITERACIONES	REPETICIONES					MEDIA	ERROR	DESVIACION	
	Valor RestPbest								
3	1,00E+03	1,00E+03	1,00E+03	1,00E+03	1,00E+03	1,00E+03		0	
5	1,00E+03	1,00E+03	1,00E+03	1,00E+03	1,00E+03	1,00E+03	0%	0	
10	1,00E+03	1,00E+03	1,00E+03	1,00E+03	1,00E+03	1,00E+03	0%	0	
20	121,0844	1,00E+03	1,00E+03	1,00E+03	97,4506	1,00E+03	36%	487,945833	
30	70,9789	82,8961	77,4013	70,5499	72,3706	7,48E+01	88%	5,26478847	
40	79,5814	67,3774	7,56E+01	92,8123	81,5576	7,94E+01	6%	9,26341241	
50	72,2862	7,81E+01	76,6915	72,1016	78,3797	7,55E+01	5%	3,09941717	
70	70,9682	66,2617	66,8217	71,4018	70,2952	6,91E+01	8%	2,42132169	
80	67,6687	68,3178	7,27E+01	65,9798	59,8565	6,69E+01	3%	4,65952747	
100	65,8888	63,2999	62,7478	6,46E+01	69,6168	6,52E+01	3%	2,7375412	
120	66,1912	65,884	69,2741	61,9201	66,3553	6,59E+01	1%	2,62266069	
130	61,1656	63,3155	61,5488	68,434	63,1829	6,35E+01	4%	2,90393024	
140	68,8834	68,0845	63,7523	65,7153	62,5097	6,58E+01	4%	2,72727055	
150	65,2274	64,4117	64,8766	66,3183	66,3792	6,54E+01	1%	0,87656886	
160	60,6814	68,0017	59,4285	59,5334	65,2139	6,26E+01	4%	3,84546252	
170	67,2208	64,4231	64,5081	63,9469	64,9645	6,50E+01	4%	1,28609951	
180	65,3755	66,2396	65,8347	60,6814	62,5097	6,41E+01	1%	2,42001457	
190	64,5709	66,3453	63,5246	64,5475	64,372	6,47E+01	1%	1,02861328	
200	65,0074	64,2804	64,2804	64,4865	59,4285	6,35E+01	2%	2,29349848	

La Figura 18. Muestra el valor de Gbest combinado (Eje Y) en función del número de iteraciones (Eje X) y se aprecia que para las primeras iteraciones existe una diferencia de respuestas bastante considerable, sin embargo, en el rango de 50 a 200 iteraciones el algoritmo alcanza una estabilidad. Los valores de Gbest combinado (posibles soluciones) alcanzan valores muy grandes en el rango de 3 a 40 iteraciones puesto que al no cumplir con ciertas restricciones planteadas por deflexión y fatiga son penalizados por el algoritmo, esto se debe a que un valor bajo de iteraciones no es suficiente para que el algoritmo converja adecuadamente.

**Figura 18.** Análisis de convergencia en función del valor Gbest.

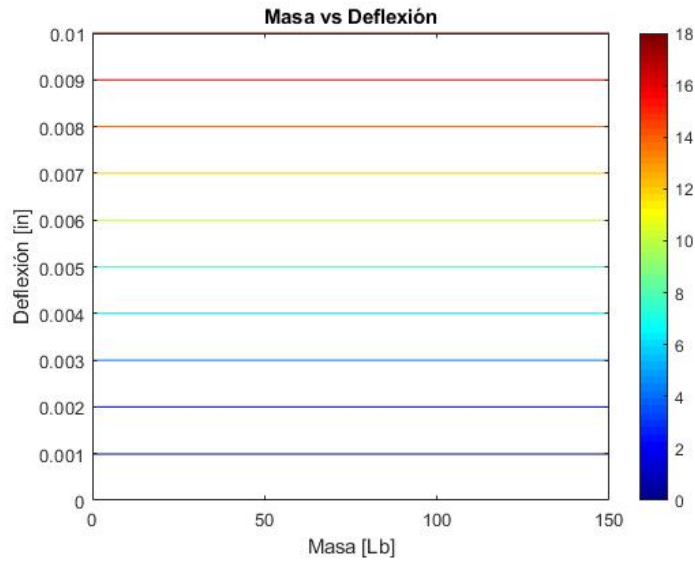


Se desea ver la tendencia de optimización para diferentes configuraciones de pesos específicos de Tchebycheff, con el fin de comprobar si la tendencia del algoritmo es la de minimizar las funciones objetivo en cualquiera de los casos. Para este análisis se proporcionan rangos óptimos para la masa y la deflexión obviando las configuraciones geométricas y se analiza el comportamiento de las funciones a optimizar según el peso que se le asigne. La tendencia de la optimización se muestra al trazar una línea perpendicular a las líneas de campo (líneas de colores) desde el color más cálido (funciones objetivo penalizadas) hacia el más frío (funciones objetivo poco penalizadas).

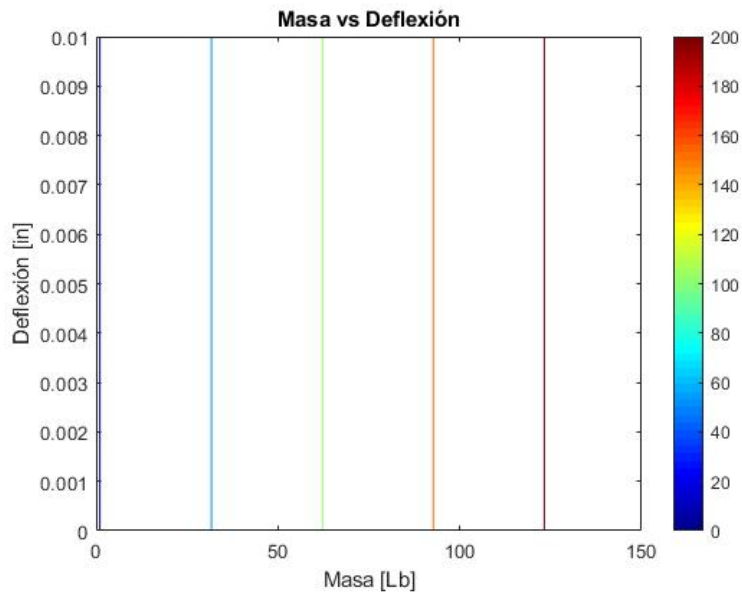
En la Figura 19. se observa la tendencia cuando se prioriza totalmente la deflexión sobre la masa, se comprueba que el algoritmo efectivamente busca minimizar el valor de la deflexión. Los valores de las funciones objetivo son menos penalizados

al tender hacia los valores más pequeños de deflexión. Lo mismo ocurre cuando se le da prioridad a la masa sobre la deflexión (Figura 20.), se evidencia que los valores de las funciones objetivo son menos sancionados cuando el algoritmo alcanza valores mínimos de masa.

**Figura 19.** Masa vs. Deflexión, pesos: [0 1]



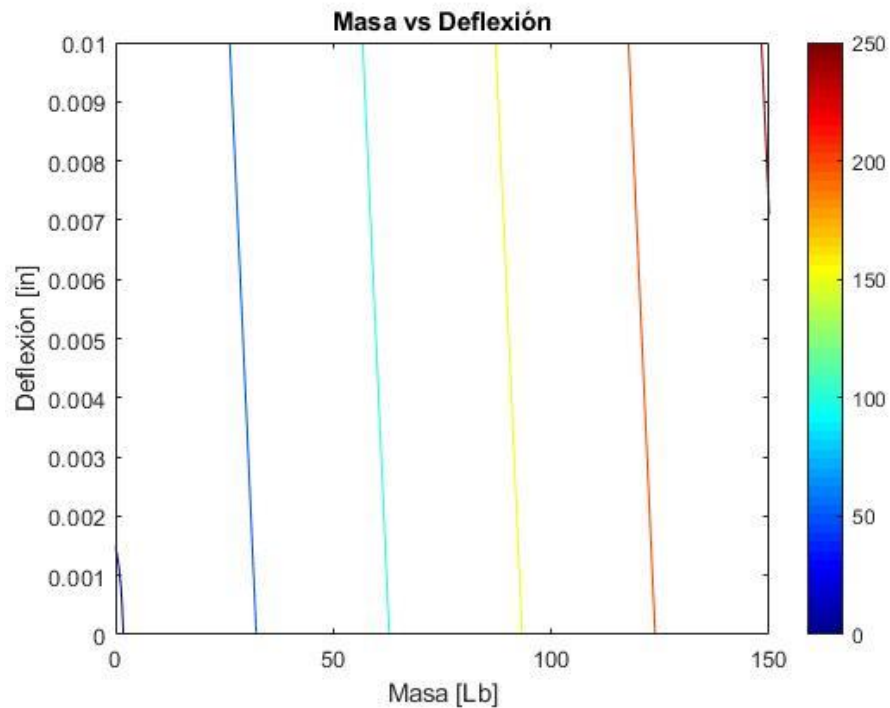
**Figura 20.** Masa vs. Deflexión, pesos: [1 0]



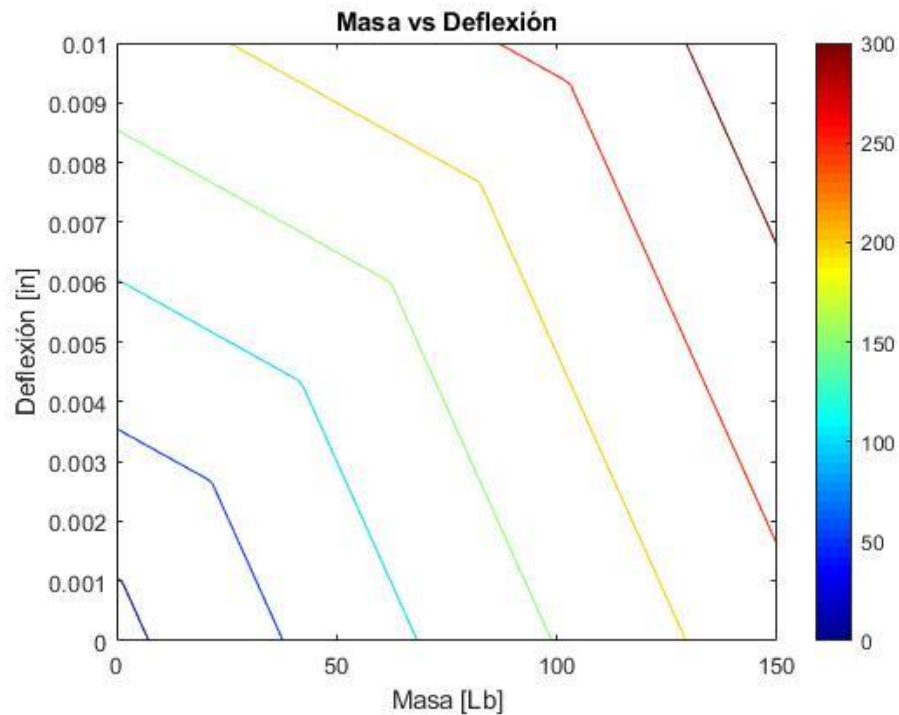
El análisis es similar para el caso en que se decida darles peso a ambas funciones, la línea de tendencia será inclinada (Figura 21.), guiando la optimización, representada por la perpendicular, hacia los valores mínimos para ambos factores por igual.

En la Figura 22. se desea ver la tendencia de la optimización, cuando se le da peso a ambas funciones, tratando de compensar la escala numérica en la que se encuentran; se puede decir, que el algoritmo busca minimizar ambas funciones, cada una con una prioridad, pero de manera conjunta.

**Figura 21.** Masa vs. Deflexión, pesos: [1 1]



**Figura 22.** Masa vs. Deflexión, pesos: [1 10]

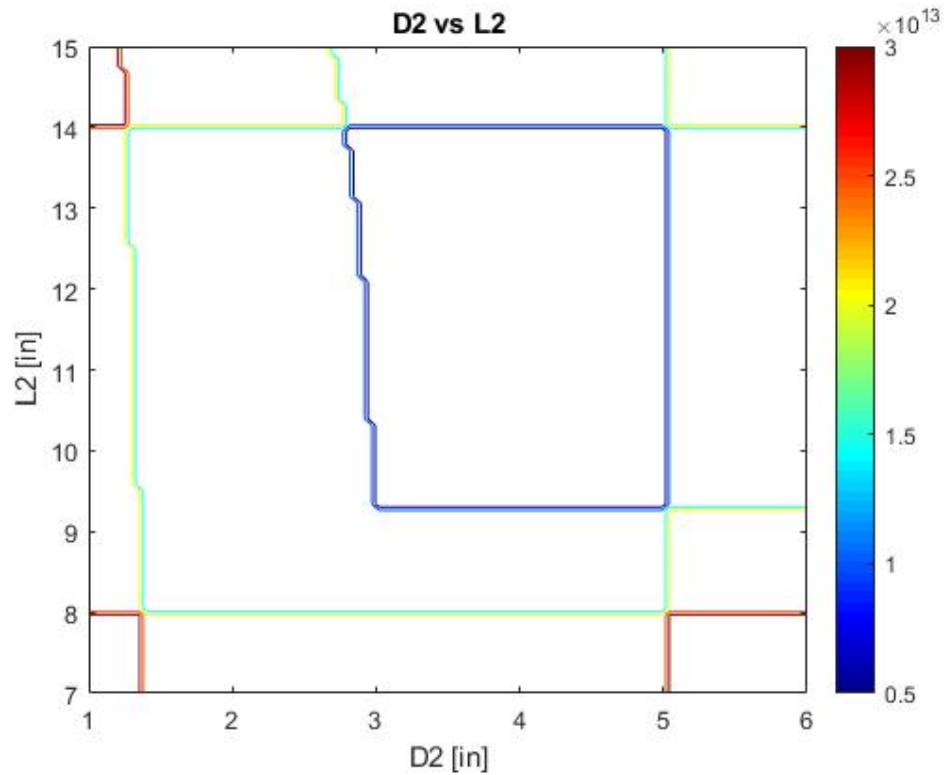


Cuando se habla de las curvas de nivel en los problemas de optimización multiobjetivo, se habla de un conjunto de niveles que representan la magnitud de los valores que puede tomar la función objetivo y se asemeja a los planos topográficos, es decir, se evidencia el relieve de la zona de búsqueda, a mayor altura, mayor valor; para este caso, se utiliza la distinción de colores para mostrar la magnitud de dichos valores. En estas gráficas se representa la relación para valores óptimos de dos factores de la geometría a la vez, para nuestro problema, son cuatro, por lo que se deben escoger dos y dejar constantes los otros.

En la Figura 23. se aprecian los valores óptimos para la Longitud 2 y el Diámetro 2 dentro del espacio de búsqueda; se acota la gráfica con una escala que excede los valores posibles de estos dos factores con el fin de visualizar las zonas donde existe una penalización (colores distintos al azul) impuesta por el algoritmo para los valores que no son óptimos, por otro lado, la zona azul encierra los valores para los cuales

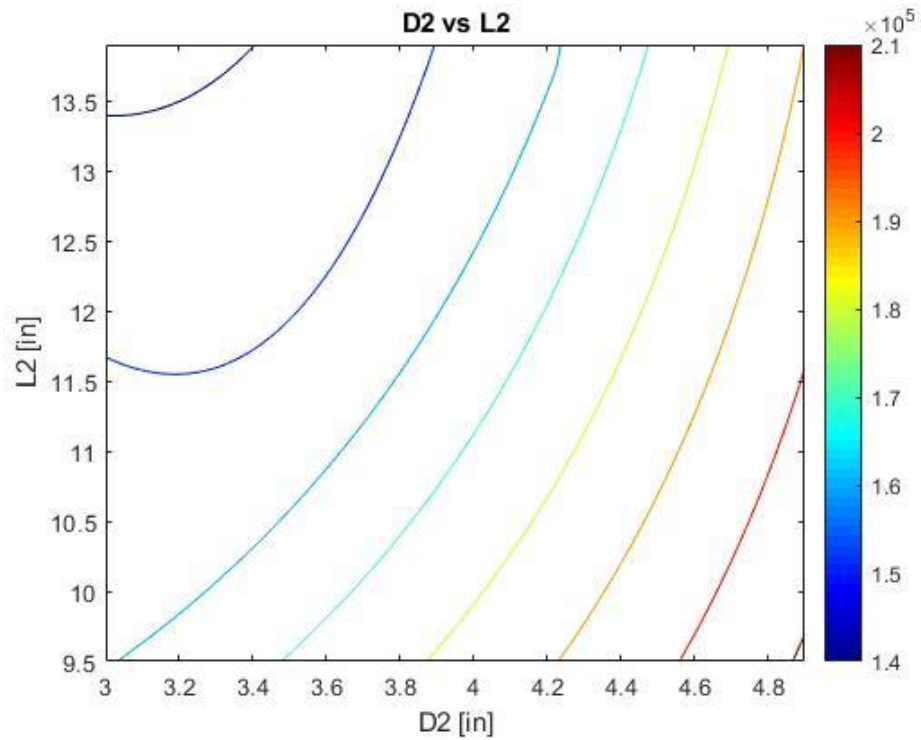
se cumple con las restricciones impuestas y optimiza de manera satisfactoria las funciones objetivo.

**Figura 23.** Zona de optimización D2 vs. L2.



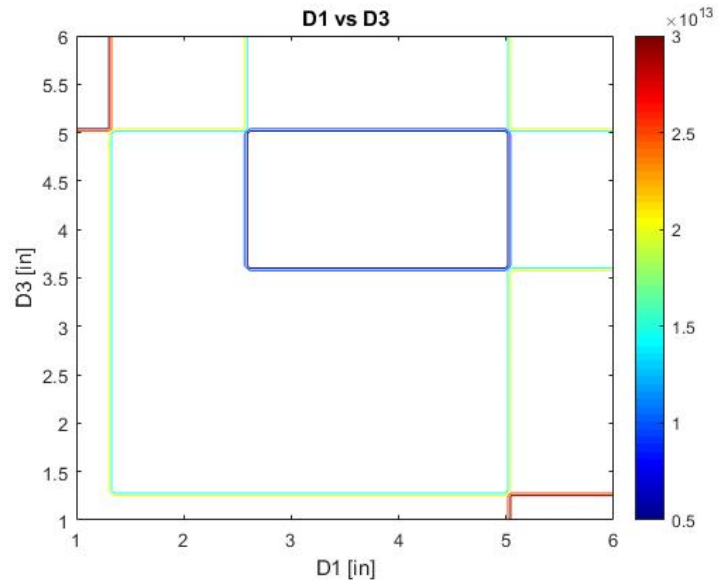
En la Figura 24. se muestra un acercamiento de la zona azul anteriormente mencionada, donde se muestran los valores óptimos que coexisten para los dos factores. En ella también se distinguen niveles, puesto que, aunque toda la zona es óptima, hay valores que minimizan de una mejor forma las funciones objetivo, en este caso la zona azul de la esquina superior izquierda contiene los mejores valores para esta configuración de factores geométricos.

**Figura 24.** Curvas de nivel D2 vs. L2.



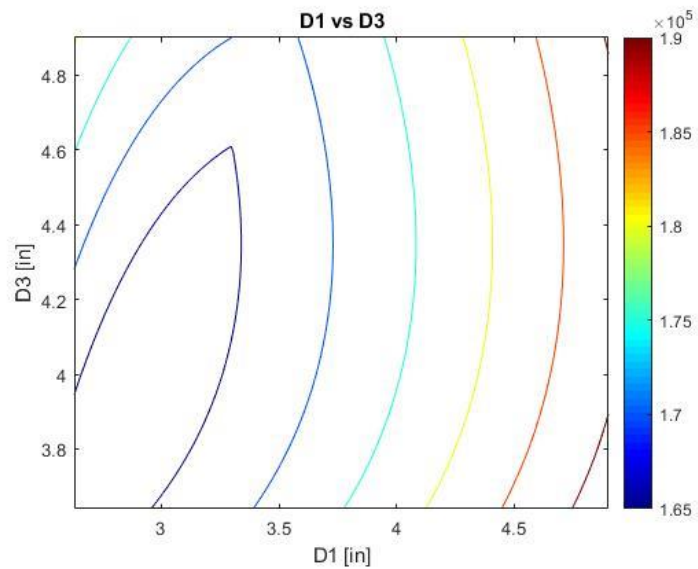
Para la Figura 25. sucede igual, se visualiza la región donde se encuentran los valores óptimos para el Diámetro 1 vs. el Diámetro 3, resultando en un espacio perfectamente rectangular, en contraste al caso anterior y se evidencian las zonas penalizadas, nuevamente las que no son de color azul.

**Figura 25.** Zona de optimización D1 vs. D3.



En la Figura 26. se hace un acercamiento para distinguir los niveles dentro de la región óptima, se evidencia una región más cerrada hacia la parte izquierda de la gráfica, donde los valores son los mejores para ambos factores.

**Figura 26.** Curvas de nivel D1 vs. D3.



## 6.2 Comparación de resultados

Uno de los objetivos principales de este trabajo, al tratarse de un problema tipo, es comparar los resultados publicados en otras investigaciones, con los obtenidos en este documento. Los resultados de las publicaciones encontradas obedecen a algoritmos bio-inspirados que se asemejan al comportamiento de PSO. En esta sección, se compara la cercanía de los resultados para diferentes configuraciones de geometría, así como también los resultados óptimos de las funciones objetivo.

La Figura 27. corresponde a los resultados escogidos a partir del algoritmo NSGA, en la cual se muestran los valores de geometría que optimizan las funciones objetivo. No es posible determinar las unidades que se usaron para la masa, por lo que se toma la determinación de reemplazar las geometrías en nuestra ecuación para dicha función y poder comparar así de manera más acertada. La comparación se realiza con los resultados consignados en la Tabla 2.

**Tabla 8.** Resultados obtenidos mediante NSGA.

$f_1(\bar{x}) = 0.0750647 * 1e+008$ $f_2(\bar{x}) = 4.22076 * 0.001$ d1 = 2.161 d2 = 2.85213 d3 = 1.38236 L1 = 10.841	$f_1(\bar{x}) = 0.23169 * 1e+008$ $f_2(\bar{x}) = 0.292313 * 0.001$ d1 = 2.66192 d2 = 5 d3 = 2.19792 L1 = 13.3624	$f_1(\bar{x}) = 0.201255 * 1e+008$ $f_2(\bar{x}) = 0.298035 * 0.001$ d1 = 2.69035 d2 = 4.6088 d3 = 2.22437 L1 = 13.2584
$f_1(\bar{x}) = 0.183833 * 1e+008$ $f_2(\bar{x}) = 0.309827 * 0.001$ d1 = 2.55308 d2 = 4.41979 d3 = 2.22356 L1 = 13.0264	$f_1(\bar{x}) = 0.0903837 * 1e+008$ $f_2(\bar{x}) = 0.964468 * 0.001$ d1 = 2.42451 d2 = 2.85218 d3 = 2.11391 L1 = 12.3519	$f_1(\bar{x}) = 0.172748 * 1e+008$ $f_2(\bar{x}) = 0.340426 * 0.001$ d1 = 2.66825 d2 = 4.22845 d3 = 2.18695 L1 = 13.1216
$f_1(\bar{x}) = 0.0970226 * 1e+008$ $f_2(\bar{x}) = 0.818535 * 0.001$ d1 = 2.60139 d2 = 2.93193 d3 = 2.14867 L1 = 12.4713	$f_1(\bar{x}) = 0.0976233 * 1e+008$ $f_2(\bar{x}) = 0.807005 * 0.001$ d1 = 2.63965 d2 = 2.92917 d3 = 2.14879 L1 = 12.5586	$f_1(\bar{x}) = 0.0845479 * 1e+008$ $f_2(\bar{x}) = 1.25182 * 0.001$ d1 = 2.14145 d2 = 2.7692 d3 = 2.13707 L1 = 12.9377

Fuente: GUZMÁN María Alejandra, DELGADO Alberto. Optimización de la geometría de un eje aplicando algoritmos genéticos. *Revista ingeniería e investigación* (58). VOL. 25 No 2, agosto 2005 15-23.

Respecto a la geometría, los diámetros y las longitudes se encuentran en la misma región óptima hallada para el PSO, y en algunos casos, en regiones muy cercanas, es decir, se alcanzan a reportar valores con una diferencia de 0,3 in para los diámetros y de hasta 2 in para las longitudes.

Una suposición hecha en el artículo del algoritmo NSGA, es que la deflexión máxima se encuentra en la sección correspondiente al diámetro 2, por lo tanto, las configuraciones obtenidas reportan éste diámetro como el mayor de los tres; para este caso, se analizó la deflexión máxima sobre toda la longitud del eje, arrojando en algunas ocasiones, una mayor inercia para la sección 3, y por lo tanto un mayor diámetro obteniendo una geometría “escalonada”.

Para hacer una comparación de las masas, se toma una configuración de geometría, de las reportadas en el artículo y se calcula con la función objetivo 1:

✓ NSGA:

- d1= 2,66825
- d2= 4,22845
- d3= 2,18695
- L2= 13,1216
- Masa= 63,37 Lb.

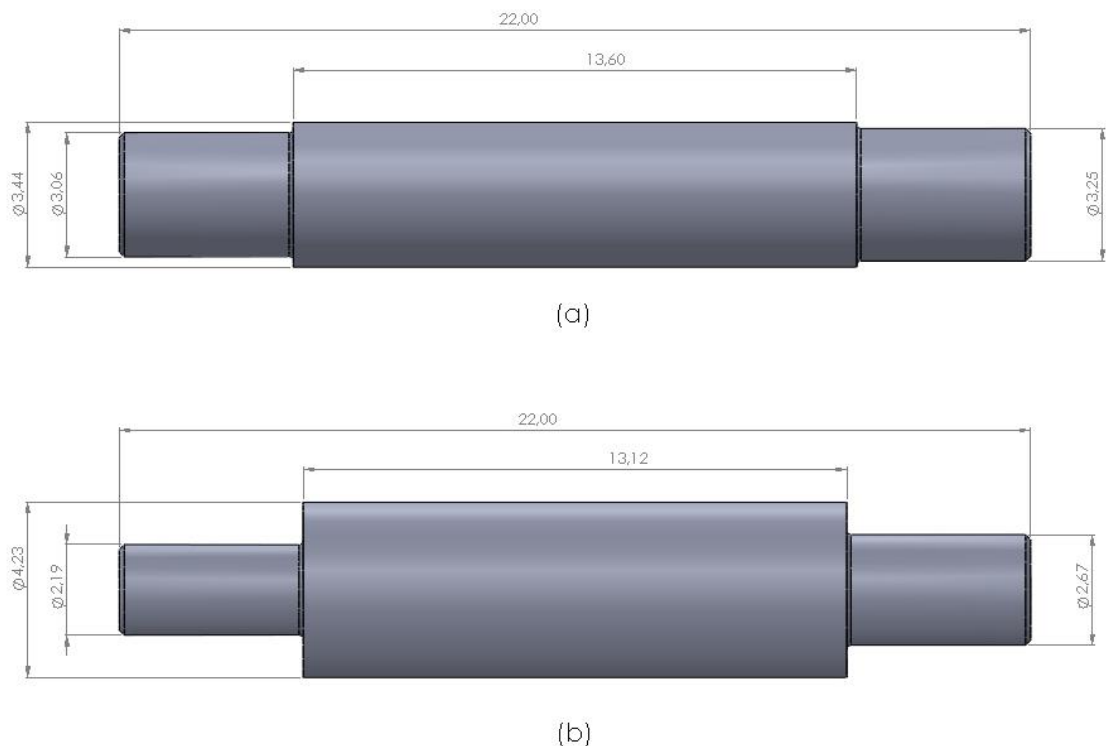
✓ PSO:

- d1= 3,0625
- d2= 3,4375
- d3= 3,25
- L2= 13,6
- Masa= 54,10 Lb.

Se aprecia una pequeña diferencia de 9 Libras entre las masas, aun así, el valor arrojado por el NSGA, se encuentra dentro de la zona óptima ya mencionada.

En base a los resultados escogidos para ambos algoritmos, se realiza un esquema para visualizar las diferencias entre ambos métodos (Figura 27). Las configuraciones geométricas arrojadas por el algoritmo genético presentan diferencias considerables entre sí, se evidencia que existe una amplia gama de soluciones para el diseñador; sin embargo, cabe mencionar que debido a la información limitada de la publicación (NSGA) para esta investigación, se supuso ciertos valores que afectan el estado de carga del eje, como la relación de velocidades (o diámetros), el ángulo de las correas en las poleas, diámetro raíz de los engranajes y algunos valores propios del material.

**Figura 27.** Esquema de la configuración del eje PSO y NSGA.



Respecto a la Quimiotaxis de bacterias no existen valores publicados, debido a que los resultados arrojados, se encuentran en el mismo frente óptimo de Pareto y en los mismos valores óptimos para la última población del algoritmo NSGA. En la Figura 9. se aprecia la superposición de ambas gráficas, los asteriscos magenta representan los valores obtenidos por la Quimiotaxis de bacterias, y los puntos negros, los obtenidos por NSGA.

## 7. CONCLUSIONES

- ✓ En este trabajo se propuso el algoritmo PSO como una estrategia de optimización para un problema real de diseño en ingeniería, en el que se involucran dos funciones objetivo, cuatro restricciones, cuatro variables y un análisis de fatiga. Se comprueba que el algoritmo permite hallar de manera eficiente las soluciones, cumpliendo con las restricciones y el criterio de Tchebycheff.
- ✓ Se comprobó que el algoritmo PSO es potencialmente más rápido que el método tradicional de diseño de ejes, permitiendo hallar diferentes configuraciones geométricas, evaluando las restricciones de manera autónoma, con sus respectivos valores de masa y deflexión. Los conjuntos de partículas más grandes probados en esta investigación fueron de 180, 190 y 200 individuos (cada uno como una solución óptima) arrojando un tiempo de cómputo de 44,7176 s, 48,8267s y 49,3116 s respectivamente.
- ✓ Para este problema de diseño mecánico se identificó la zona de valores óptimos para la variable masa (variable de mayor magnitud) ubicada entre las 40 lb y las 65 lb, manteniendo los valores de deflexión dentro sus restricciones.

- ✓ El código le permite al diseñador establecer previamente su preferencia de optimización gracias al análisis de los pesos de Tchebycheff de manera satisfactoria, es decir, puede definir con anticipación cuál factor requiere valores más ajustados al mínimo posible: la masa o la deflexión.
  
- ✓ A partir del análisis estadístico ANOVA se establecen los valores óptimos para los factores de entrada del algoritmo con los cuales converge dentro de la zona óptima (entre 40 y 65 lb): Número de partículas ( $N_p$ ) = 22, Número de iteraciones ( $MaxIter$ ) = 160, Rango de aceleración:  $C_{min} = 0,5$  y  $C_{max} = 2,5$ , Peso inercial:  $W_{min} = 0,3$  y  $W_{max} = 0,9$  y velocidad inicial = 1.
  
- ✓ Para futuras investigaciones es posible refinar los parámetros del algoritmo aumentando los rangos de los factores en el ANOVA y subdividiéndolos en más valores con un mayor número de réplicas.

## BIBLIOGRAFÍA

1. CONGRESO IBEROAMERICANO DE INGENIERIA MECANICA (8, Cusco, 23 al 25 de octubre de 2007). Optimización multiobjetivo de un eje con algoritmo basado en quimiotaxis de bacterias. Pontificia Universidad Católica del Perú, 2007, 9.
2. DUARTE, César y QUIROGA, Jabid. Algoritmo PSO para identificación de parámetros en un motor DC. *Revista facultad de ingenierías de la Universidad de Antioquia*. 2010, nro. 55, pp. 116-124.
3. GUZMÁN, María Alejandra, DELGADO, Alberto. Optimización de la geometría de un eje aplicando algoritmos genéticos. *Revista ingeniería e investigación* (58). VOL. 25 No 2, agosto 2005 15-23.
4. LÓPEZ, Javier. Optimización Multi-objetivo. Aplicaciones al mundo real. Tesis doctoral en ciencias informáticas. La Plata. Editorial de la Universidad de La Plata. 2013. pp. 102-103.
5. PARADA CORRALES, Alfredo. Fatiga. Bucaramanga. 1990.
6. QUIROGA, Jabid: CÁCERES, Edwin y PADILLA, Carlos. Optimización de trayectorias de fresado en cavidades utilizando el algoritmo luciérnaga. *Revista de la Facultad de Ingeniería U.C.V.* 2015, vol. 30, pp. 93-104.
7. SHIGLEY, J., MISCHKE, C. Diseño en Ingeniería Mecánica. McGraw Hill. Sexta Edición. México, 2002.

8. VANEGAS ARDILA, Diego; BARRAGÁN NIÑO, Karol y CORREA CELY, Rodrigo. Comparación de las técnicas de optimización por análisis de intervalos y la de enjambre de partículas para funciones con restricciones. *Ingeniería y Universidad*. 2011, vol. 15, nro. 1, pp. 47-60.
  
9. YUHUI, Shi. Particle Swarm Optimization. Ed. Electronic Data Systems, Inc. IEE Neural Networks Society. 2004, pp. 8-13.

## ANEXOS

### ANEXO A. PSO

```
clear; clc;
%% Definición del Problema
tic
%datos que deben ser agregados por el usuario
%f1=@(x1,x2) x1^2 -x1 +(x2+1)^2;           %función a minimizar
%f2=@(x1,x2) x1*x2 - x1^2 +2^x2 -x1^3;
%f2=@(x1,x2,x3) x1^2 +x1 +(x2-1)^2+(x3^2+2);
f1=@(d1,d2,d3,L1,L2,L3) ((0.2818*pi/4)*((d1^2)*L1 + (d2^2)*L2 +
(d3^2)*L3)); %peso del eje
f2=@(Yxy,Yxz) (sqrt((Y_xy)^2+(Y_xz)^2)); %Se Calcula en la
Estatica.m %deflexión lateral máxima

a=2;
b=3;

% Array =[ d1 d2 d3 L2]
Min=[a a a 8]; %Limites inferiores de las coordenadas
Max=[b b b 9]; %Limites superiores de las coordenadas
Xpaso=[1/16 1/16 1/16 0.1]; %Paso de cada parametro de entrada

%Datos globales para calculos de Estática , Resistencia y Diseño de
%Máquinas
global dataKt dataKts dataQt dataQs Sut Sfp E;
global Famax_xy Famin_xy Fcmax_xy Fcmin_xy;
global Famax_xz Famin_xz Fcmax_xz Fcmin_xz;
global Ta Tm ;
global redondeo;
redondeo = 0.05;

%Cartas de datos
dataKt=xlsread("ConcEsf","Kt");
dataKts=xlsread("ConcEsf","Kts");
dataQt=xlsread("ConcEsf","qt");
dataQs=xlsread("ConcEsf","qs");

%Propiedades
Sut=85000; %psi
Sfp=43056; %psi
E = 30e6; %psi

%Fuerzas
Famax_xy=162.09; %Libras
Famin_xy=-6.6;
Fcmax_xy=89.65;
Fcmin_xy=3.66;
```

```

Famax_xz=412.54;
Fcmax_xz=-315;
Famin_xz=98.6;
Fcmin_xz=-78.75;

Tmax=3150;           %lb*in
Tmin=787.5;         %lb*in
Ta=(Tmax-Tmin)/2;   %lb*in
Tm=(Tmax+Tmin)/2;   %lb*in

%Estatica(1,2,1.6,0.5,2,0.5)
%% Parámetros del PSO
n=size(Max,2);      %número de coordenadas (variables)
Np=22;              %número de partículas
MaxIter=160;        %número de iteraciones máximas
Cmin=[0.2 0.2];
Cmax=[2.2 2.2];
%Cmin=[0.5 0.5];    %constante de aceleración mínima
%Cmax=[2.5 2.5];    %constante de aceleración máxima
Wmin=0.3;           %factor de inercia mínimo
Wmax=0.9;           %factor de inercia máximo
alpha=1;            %factor para cálculo de estrechamiento

%Parámetros multiobjetivo

%w=[0.01 999]       %Pesos para cada función objetivo
%w=[999 0.01]
w=[1 0]
rho=1                %Factor para garantizar eficiencia del modelo
sancion=1000;

%% Inicialización
[X]= Particulas(Np,Min,Max); %se crean las partículas aleatoriamente
por el espacio de solución

[FO,FOs]=FuncionAdaptacion(f1,f2,X,Np,w,rho); %se calcula la función
para cada individuo (SE DEBE MEJORAR FUNCIÓN PARA HACERLA GENERAL)
[FO]=Restricciones(X,FOs,FO,sancion,Np);

d=1; %si d=1 entonces se define velocidad inicial aleatoria, si d=0
entonces la velocidad inicial se define en cero
V=velocidad(Np,d,n); % cálculo de las velocidades iniciales

iter=1;
Pbest=X;
[v,p]=min(FO(:,1)); %mínimo valor de la función objetivo 1
Gbest=X(p,:); %lider del cúmulo de partículas

figure(iter)
%plot(X(:,1),X(:,2),'*r') %graficar las partículas
plot(FOs(:,1),FOs(:,2),'*r')

```

```

xlim ([-2 10])
ylim ([-2 10])

%% Ciclo Principal
while iter<=MaxIter
    [C1,C2,W,K]=calcular (Wmin,Wmax,Cmin,Cmax,MaxIter,alpha,iter);
    [V]=CalVelocidades (V,C1,C2,W,Gbest,Np,X,Pbest);
    [X]=posiciones (V,X,Np,Xpaso);
    [FO,FOs]=FuncionAdaptacion(f1,f2,X,Np,w,rho); %se calcula la función
para cada individuo (se debe mejorar función para hacerla general)
    [FO] = Restricciones(X,FOs,FO,sancion, Np);
    [ResPbest,ResPbests]=FuncionAdaptacion(f1,f2,Pbest,Np,w,rho);
    [ResPbest]=Restricciones(Pbest,ResPbests,ResPbest,sancion,Np);
    [Pbest,FObest]=Pmejores (f1,f2,Pbest,ResPbest,X,FO,Np); %se debe
guardar la memoria de las mejores posiciones visitadas por las
partículas,
    [Gbest]=Gmejor (Gbest,Pbest,FObest);
    pause(0.0001)
    iter=iter+1;
    % figure(iter)
    %plot(X(:,1),X(:,2),'*r');
    plot(FOs(:,1),FOs(:,2),'*r')
    title('PSO'); xlabel('Eje X');ylabel('Eje Y') %graficar partículas
    xlim ([0 100]) %Rango de vision de masa
    ylim ([0 0.04]) %Rango de vision de deformacion maxima

end

% figure(2) %para (2*x1)^2+(x2)^2;
% plot(Gbest(:,1),Gbest(:,2),'X');title('Mejores Históricos')
% hold all;

toc

min(ResPbests(:,1))

```

## ANEXO B. Posiciones

```
function [X]=posiciones(V,X,Numparticulas,paso)

for i=1:1:Numparticulas      %actualizaci3n de las poiciones
    X(i,:)=X(i,:)+V(i,:);
    X(i,:)=redondeopaso(X(i,:),paso);
end
end
```



## ANEXO D. Partículas

```
function [X]=Particulas(Numparticulas,Min,Max) %se crean las
partículas aleatoriamente por el espacio de solución
for i=1:1:Numparticulas
    for j=1:1:size(Min,2)
        X(i,j)=(Max(1,j)-Min(1,j))*rand+Min(1,j);
    end
end
end
end
```

## ANEXO E. Calcular

```
function
[C1,C2,W,K]=calcular(Wmin,Wmax,Cmin,Cmax,Itermaximas,alpha,iteracion)

W=((Wmin-Wmax)/(Itermaximas-1))*(iteracion-1)+Wmax;           %Cálculo
de la velocidad inercial

C1=((Cmin(1,1)-Cmax(1,1))/Itermaximas)*iteracion)+Cmax(1,1);  %Cálculo
de la primera constante de aceleración

C2=((Cmax(1,2)-Cmin(1,2))/Itermaximas)*iteracion)+Cmin(1,2);  %Cálculo
de la segunda constante de aceleración

phi=C1+C2;

K=(2*alpha)/abs(2-phi-sqrt((phi^2)-4*phi));                     %factor de
estrechamiento
end
```

## ANEXO F. Cal Velocidades

```
function [V]=CalVelocidades(V,C1,C2,W,Gbest,Numparticulas,X,PosicionBest)

for i=1:1:Numparticulas           %se calculan las nuevas velocidades
basado en la experiencia personal y grupal
    V(i,:) = (W*V(i,:) + C1*rand*(PosicionBest(i,:)'-X(i,:)') +
C2*rand*(Gbest'-X(i,:)'))';
end
end
```



## **ANEXO H. G mejor**

```
function [Gbest]=Gmejor(Gbest, PosicionBest, FAbest)

[v,p]=min(FAbest)

Gbest=PosicionBest(p,:)
end
```

## ANEXO I. Velocidad

```
function [V]=velocidad(Numparticulas,d,Num)           %se crean todas las
velocidades aleatorias

if d==1
    for i=1:1:Numparticulas
        for j=1:1:Num
            V(i,j)=2*rand;
        end
    end
else
    V=zeros(Numparticulas,Num);
end
end
```

## ANEXO J. Fatiga

```
function [g1,g2,g3] = Fatiga(d1,d2,d3,L1,L2,L3)

% Restricciones por fatiga en las 3 secciones

global Famax_xy Famin_xy Fcmax_xy Fcmin_xy;
global Famax_xz Famin_xz Fcmax_xz Fcmin_xz;
global E redondeo;
global Sut Ta Tm;

%MÁXIMOS Plano X-Y

Rbmax_xy = (-Fcmax_xy*L1 - Famax_xy*(2*L1+L2))/(L1+L2);
Rdmax_xy = -Rbmax_xy - Famax_xy -Fcmax_xy ;

%Cortantes
V1max_xy=Famax_xy;
V2max_xy=V1max_xy+Rbmax_xy;
V3max_xy=V2max_xy+Fcmax_xy;

%Momentos
Mbmax_xy=V1max_xy*L1;
Mcmax_xy=Mbmax_xy + (V2max_xy*L2);
Mdmax_xy=Mcmax_xy + (V3max_xy*L1) ; %Para comprobar

%MÍNIMOS Plano X-Y

Rbmin_xy = (-Fcmin_xy*L1 - Famin_xy*(2*L1+L2))/(L1+L2);
Rdmin_xy = -Rbmin_xy - Famin_xy - Fcmin_xy ;

%Cortantes
V1min_xy=Famin_xy;
V2min_xy=V1min_xy+Rbmin_xy;
V3min_xy=V2min_xy+Fcmin_xy;

%Momentos
Mbmin_xy=V1min_xy*L1;
Mcmin_xy=Mbmin_xy+(V2min_xy*L2);
Mdmin_xy=Mcmin_xy+(V3min_xy*L1) ; %Para comprobar

%MÁXIMOS Plano X-Z

Rbmax_xz = (-Fcmax_xz*L1 - Famax_xz*(2*L1+L2))/(L1+L2) ;
Rdmax_xz = -Famax_xz - Rbmax_xz - Fcmax_xz;

%Cortantes
V1max_xz=Famax_xz;
V2max_xz=V1max_xz+Rbmax_xz;
V3max_xz=V2max_xz+Fcmax_xz;
```

```

%Momentos
Mbmax_xz=Vlmax_xz*L1;
Mcmax_xz=Mbmax_xz+(V2max_xz*L2);
Mdmax_xz=Mcmax_xz+(V3max_xz*L1);      %Para comprobar

%MOMENTOS Plano X-Z

Rbmin_xz = (-Fcmin_xz*L1 - Famin_xz*(2*L1+L2))/(L1+L2);
Rdmin_xz = -Famin_xz - Rbmin_xz - Fcmin_xz;

%Cortantes
Vlmin_xz=Famin_xz;
V2min_xz=Vlmin_xz+Rbmin_xz;
V3min_xz=V2min_xz+Fcmin_xz;

%Momentos
Mbmin_xz=Vlmin_xz*L1;
Mcmin_xz=Mbmin_xz+(V2min_xz*L2);
Mdmin_xz=Mcmin_xz+(V3min_xz*L1) ;      %Para comprobar

%Momento resultante: sacar resultante entre Mb(max y min) y Mc (max y
min)
%y comparar cuál de estas resultantes es mayor, esa es la que se
utilizará;
%para determinar los Momentos Medios y Alterativos

%MOMENTOS RESULTANTES MAXIMOS

Mra_max=0;
Mrb_max=sqrt((Mbmax_xy)^2 + (Mbmax_xz)^2);
Mrc_max=sqrt((Mcmax_xy)^2 + (Mcmax_xz)^2);
Mrd_max=0;

%MOMENTOS RESULTANTES MINIMOS

Mra_min=0;
Mrb_min=sqrt((Mbmin_xy)^2 + (Mbmin_xz)^2);
Mrc_min=sqrt((Mcmin_xy)^2 + (Mcmin_xz)^2);
Mrd_min=0;

Sf1=21975.71*d1^(-0.097);
Sf2=21975.71*d2^(-0.097);
Sf3=21975.71*d3^(-0.097);

%%MOMENTOS FINALES

```

```

%Sección 1
Mm1=(Mrb_max+Mrb_min)/2;
Ma1=(Mrb_max-Mrb_min)/2;

%Sección 3
Mm3=(Mrc_max+Mrc_min)/2;
Ma3=(Mrc_max-Mrc_min)/2;

%Sección 2
% hay 2 posibilidades para el momento en la sección 2
if Mrb_max> Mrc_max
    Mm2=Mm1;
    Ma2=Ma1;
    isB =true;
else
    Mm2=Mm3;
    Ma2=Ma3;
    isB=false;
end

Kf=K_f(d1,d2,redondeo);
Kfs = K_fs(d1,d2,redondeo);
g1=d1-((22.4)*((sqrt((Kf
*Ma1)^2)+(3/4)*(Kfs*Ta)^2)/Sf1)+(sqrt(((Kf*Mm1)^2)+(3/4)*(Kfs*Tm)^2)/Sut)
)^1/3);

if isB
    Kf=K_f(d1,d2,redondeo);
    Kfs = K_fs(d1,d2,redondeo);
else
    Kf=K_f(d3,d2,redondeo);
    Kfs = K_fs(d3,d2,redondeo);
end

g2=d2-
((22.4)*((sqrt(((Kf*Ma2)^2)+(3/4)*(Kfs*Ta)^2)/Sf2)+(sqrt(((Kf*Mm2)^2)+(3/4)
*(Kfs*Tm)^2)/Sut)))^1/3);

Kf=K_f(d3,d2,redondeo);
Kfs = K_fs(d3,d2,redondeo);
g3=d3-
((22.4)*((sqrt(((Kf*Ma3)^2)+(3/4)*(Kfs*0)^2)/Sf3)+(sqrt(((Kf*Mm3)^2)+(3/4)
*(Kfs*0)^2)/Sut)))^1/3);

end

```

## ANEXO K. Y máx

```
function [ymax] = Ymax(d1,d2,d3,L1,L2,L3)
% Se leen las variables globales de fuerzas
global Famax_xy Famin_xy Fcmax_xy Fcmin_xy;
global Famax_xz Famin_xz Fcmax_xz Fcmin_xz;
global E;

%MÁXIMOS Plano X-Y

Rbmax_xy = (-Fcmax_xy*L1 - Famax_xy*(2*L1+L2))/(L1+L2);
Rdmax_xy = -Rbmax_xy - Famax_xy - Fcmax_xy ;

%Cortantes
V1max_xy=Famax_xy;
V2max_xy=V1max_xy+Rbmax_xy;
V3max_xy=V2max_xy+Fcmax_xy;

%Momentos
Mbmax_xy=V1max_xy*L1;
Mcmax_xy=Mbmax_xy + (V2max_xy*L2);
Mdmax_xy=Mcmax_xy + (V3max_xy*L1) ; %Para comprobar

%MÁXIMOS Plano X-Z

Rbmax_xz = (-Fcmax_xz*L1 - Famax_xz*(2*L1+L2))/(L1+L2) ;
Rdmax_xz = -Famax_xz - Rbmax_xz - Fcmax_xz;

%Cortantes
V1max_xz=Famax_xz;
V2max_xz=V1max_xz+Rbmax_xz;
V3max_xz=V2max_xz+Fcmax_xz;

%Momentos
Mbmax_xz=V1max_xz*L1;
Mcmax_xz=Mbmax_xz+(V2max_xz*L2);
Mdmax_xz=Mcmax_xz+(V3max_xz*L1); %Para comprobar

%% Ahora vamos con las Deflexiones - Se invirtio el analisis

%Plano X-Y
C2_xy = 0;

C1_xy= -(Rdmax_xy/6)*(L1+L2)^3 - (Fcmax_xy/6)*(L2)^3 - (Rbmax_xy/6)*0^3
- C2_xy)/(L1+L2);
```

```

%Primer Tramo - Cuando X es menor a L1
Y_xy1=@(x) ((Rdmax_xy/6)*x^3 + C1_xy*x + C2_xy)/(E*(pi*d1^2)/64);
%Segundo Tramo - Cuando X esta entre L1 y L1+L2
Y_xy2= @(x) ((Rdmax_xy/6)*x^3 + (Fcmax_xy/6)*(x-L1)^3 + C1_xy*x +
C2_xy)/(E*(pi*d2^2)/64);
%Tercer Tramo
Y_xy3= @(x) ((Rdmax_xy/6)*x^3 + (Fcmax_xy/6)*(x-L1)^3 + (Rbmax_xy/6)*(x-
(L1+L2))^3 + C1_xy*x + C2_xy)/(E*(pi*d3^2)/64);

M=200; %Cantidad de Elementos

delta1=(L1)/M; %Tamaño de elemento tramo 1
delta2=(L2)/M; %Tamaño de elemento tramo 2
delta3=(L3)/M; %Tamaño de elemento tramo 3

xval1=0:delta1:L1; %Vector discretizado en el tramo 1
xval2=L1:delta2:L1+L2; %Vector discretizado en el tramo 2
xval3=L1+L2:delta3:L1+L2+L3; %Vector discretizado en el tramo 3

for i=1:1:M
    yval_xy1(i) = Y_xy1(xval1(i)); %Evaluando el tramo 1
    yval_xy2(i) = Y_xy2(xval2(i)); %Evaluando el tramo 2
    yval_xy3(i) = Y_xy3(xval3(i)); %Evaluando el tramo 3
end

%Plano X-Z

C2_xz = 0;

C1_xz= -(Rdmax_xz/6)*(L1 + L2)^3 - (Fcmax_xz/6)*L2^3 - C2_xz)/(L1 +
L2);

%Primer Tramo - Cuando X es menor a L1
Y_xz1=@(x) ((Rdmax_xz/6)*x^3 + C1_xz*x + C2_xz)/(E*(pi*d1^2)/64);
%Segundo Tramo - Cuando X esta entre L1 y L1+L2
Y_xz2= @(x) ((Rdmax_xz/6)*x^3 + (Fcmax_xz/6)*(x-L1)^3 + C1_xz*x +
C2_xz)/(E*(pi*d2^2)/64);
%Tercer Tramo - Cuando X esta entre L1+L2 y L1+L2+L3
Y_xz3= @(x) ((Rdmax_xz/6)*x^3 + (Fcmax_xz/6)*(x-L1)^3 + (Rbmax_xz/6)*(x-
(L1+L2))^3 + C1_xz*x + C2_xz)/(E*(pi*d3^2)/64);

for i=1:1:M
    yval_xz1(i) = Y_xz1(xval1(i)); %Evaluando el tramo 1
    yval_xz2(i) = Y_xz2(xval2(i)); %Evaluando el tramo 2
    yval_xz3(i) = Y_xz3(xval3(i)); %Evaluando el tramo 3
end

%Deflexion total en cada nodo
for i=1:1:M

```

```

    yval1(i) = sqrt((yval_xy1(i))^2+(yval_xz1(i))^2); %Evaluando el tramo
1
    yval2(i) = sqrt((yval_xy2(i))^2+(yval_xz2(i))^2); %Evaluando el
tramo 2
    yval3(i) = sqrt((yval_xy2(i))^2+(yval_xz3(i))^2); %Evaluando el
tramo 3
end

%Encontrar la deflexion maxima del eje
[ymax1,pos1]=max(yval1);
[ymax2,pos2]=max(yval2);
[ymax3,pos3]=max(yval3);

if ymax1>ymax2 & ymax1>ymax3

    ymax=ymax1;
    xmax=xval1(pos1);
else
    if ymax2>ymax3
        ymax=ymax2;
        xmax=xval2(pos2);
    else
        ymax=ymax3;
        xmax=xval3(pos3);
    end
end

end

end

```

## ANEXO L. Restricciones

```
function [FO] = Restricciones(X,FAs,FA,sancion,Numparticulas)

for i=1:1:Numparticulas
    delta = 0;

    if X(i,1)<0.5 | X(i,1)>5
        delta=delta+1e10;
    end
    if X(i,2)<0.5 | X(i,2)>5
        delta=delta+1e10;
    end
    if X(i,3)<0.5 | X(i,3)>5
        delta=delta+1e10;
    end
    if X(i,4)>14 | X(i,4)<8
        delta=delta+1e10;
    end

    if FAs(i,1)<0
        delta=delta-1e12*FAs(i,1);
    end

    if FAs(i,2)<0
        delta=delta-1e14*FAs(i,2);
    end

    if FAs(i,2)>0.005
        delta=delta+1e10;
    end

    %Restricciones de fatiga
    L13=(22-X(i,4))/2;
    if X(i,1)>0 | X(i,2)>0 |X(i,3)>0 |X(i,4)>0
        [g1,g2,g3] = Fatiga(X(i,1),X(i,2),X(i,3),L13,X(i,4),L13) ;

        if g1 <0 | g2< 0 | g3<0
            delta=delta+1e10;
        end
    end

    FO(i,1)=FA(i,1) + sancion*(delta);
end

end
```

## ANEXO M. Redondeo paso

```
function [num]=redondeopaso(num,paso)

elements = size(num,2);
mitad_paso=paso/2;

for i=1:1:elements
    res = mod(num(i),paso(i));
    if res<=mitad_paso
        num(i)=num(i)+(1-res);
    else
        num(i)=num(i)-res;
    end
end
```

## ANEXO N. qt

```
function [Resp] = qt(S_ut,r)
global dataQt;
data= dataQt;
Xq=S_ut;
Yq=r;
longX = size(data,1);
longY = size(data,2);
X=data(1,2:longY);
Y=data(2:longX,1);
V =data(2:longX, 2:longY);

if Xq>max(X)
    Xq=max(X);
end
if Xq<min(X)
    Xq=min(X);
end
if Yq>max(Y)
    Yq=max(Y);
end
if Yq<min(Y)
    Yq=min(Y);
end

Resp = interp2(X,Y,V,Xq,Yq);

end
```

## ANEXO O. qs

```
function [Resp] = qs(S_ut,r)
global dataQs;
data= dataQs;
Xq=S_ut;
Yq=r;
longX = size(data,1);
longY = size(data,2);
X=data(1,2:longY);
Y=data(2:longX,1);
V =data(2:longX, 2:longY);

if Xq>max(X)
    Xq=max(X);
end
if Xq<min(X)
    Xq=min(X);
end
if Yq>max(Y)
    Yq=max(Y);
end
if Yq<min(Y)
    Yq=min(Y);
end

Resp = interp2(X,Y,V,Xq,Yq);

end
```

## ANEXO P. Kts

```
function [Resp] = Kts(D_1,D_2,r)
global dataKts;
data= dataKts;
if D_1 > D_2
    D_may=D_1;
    D_men=D_2;
else
    D_may=D_2;
    D_men=D_1;
end

Xq=D_may/D_men;
Yq=r/D_men;
longX = size(data,1);
longY = size(data,2);
X=data(1,2:longY);
Y=data(2:longX,1);
V =data(2:longX, 2:longY);

if Xq>max(X)
    Xq=max(X);
end
if Xq<min(X)
    Xq=min(X);
end
if Yq>max(Y)
    Yq=max(Y);
end
if Yq<min(Y)
    Yq=min(Y);
end

Resp = interp2(X,Y,V,Xq,Yq);

end
```

## ANEXO Q. Kt

```
function [Resp] = Kt(D_1,D_2,r)
global dataKt;
data= dataKt;
if D_1 > D_2
    D_may=D_1;
    D_men=D_2;
else
    D_may=D_2;
    D_men=D_1;
end
Xq=D_may/D_men;
Yq=r/D_men;

longX = size(data,1);
longY = size(data,2);
X=data(1,2:longY);
Y=data(2:longX,1);
V =data(2:longX, 2:longY);

if Xq>max(X)
    Xq=max(X);
end
if Xq<min(X)
    Xq=min(X);
end
if Yq>max(Y)
    Yq=max(Y);
end
if Yq<min(Y)
    Yq=min(Y);
end

Resp = interp2(X,Y,V,Xq,Yq);

end
```

## ANEXO R. Kfs

```
function [Kf] = K_fs(D_1,D_2,r)
global Sut;
Kt_var = Kts(D_1,D_2,r);
q_var = qs(Sut,r);
Kf=1+q_var*(Kt_var-1);
end
```

## ANEXO S. Kf

```
function [Kf] = K_f(D_1,D_2,r)
global Sut;
Kt_var = Kt(D_1,D_2,r);
q_var = qt(Sut,r);
Kf=1+q_var*(Kt_var-1);
end
```

## ANEXO T. Only FO

```
function [FO] = OnlyFO(f1,f2,w,rho)
val1=1.22; %F0_1 minimo
val2=0.001; %F0_2 minimo

for i=1:size(f1,1)
    for j=1:size(f2,1)
        delta1=(f1(i,j) - val1)/val1;
        delta2=(f2(i,j) - val2)/val2 ;
        lambdas(1)=w(1)*delta1;
        lambdas(2)=w(2)*delta2;
        lambda=max(lambdas);

        FO(i,j)=lambda + rho *(w(1)*delta1+w(2)*delta2) ;
    end
end
end
```

## ANEXO U. Graphs Contour

```
clear; clc;
close all;
%% Definición del Problema
tic
%datos que deben ser agregados por el usuario
%f1=@(x1,x2) x1^2 -x1 + (x2+1)^2; %función a minimizar
%f2=@(x1,x2) x1*x2 - x1^2 +2^x2 -x1^3;
%f2=@(x1,x2,x3) x1^2 +x1 + (x2-1)^2+(x3^2+2);
f1=@(d1,d2,d3,L1,L2,L3) ((0.2818*pi/4)*((d1^2)*L1 + (d2^2)*L2 +
(d3^2)*L3)); %peso del eje
f2=@(Yxy,Yxz) (sqrt((Y_xy)^2+(Y_xz)^2)); %Se Calcula en la
Estática.m %deflexión lateral máxima

a=2;
b=3;

% Array =[ d1 d2 d3 L2]
Min=[a a a 8]; %Limites inferiores de las coordenadas
Max=[b b b 9]; %Limites superiores de las coordenadas
Xpaso=[1/16 1/16 1/16 0.1]; %Paso de cada parametro de entrada

%Datos globales para calculos de Estática , Resistencia y Diseño de
%Máquinas
global dataKt dataKts dataQt dataQs Sut Sfp E;
global Famax_xy Famin_xy Fcmax_xy Fcmin_xy;
global Famax_xz Famin_xz Fcmax_xz Fcmin_xz;
global Ta Tm ;
global redondeo;
redondeo = 0.05;

%Cartas de datos
dataKt=xlsread("ConcEsf","Kt");
dataKts=xlsread("ConcEsf","Kts");
dataQt=xlsread("ConcEsf","qt");
dataQs=xlsread("ConcEsf","qs");

%Propiedades
Sut=85000; %psi
Sfp=43056; %psi
E = 30e6; %psi

%Fuerzas
Famax_xy=162.09; %Libras
Famin_xy=-6.6;
Fcmax_xy=89.65;
Fcmin_xy=3.66;

Famax_xz=412.54;
Fcmax_xz=-315;
Famin_xz=98.6;
Fcmin_xz=-78.75;
```

```

Tmax=3150;           %lb*in
Tmin=787.5;         %lb*in
Ta=(Tmax-Tmin)/2;  %lb*in
Tm=(Tmax+Tmin)/2;  %lb*in

%Estatica(1,2,1.6,0.5,2,0.5)
%% Parámetros del PSO
n=size(Max,2);      %número de coordenadas (variables)
Np=30;              %número de partículas
MaxIter=3;          %número de iteraciones máximas
Cmin=[0.5 0.5 0.5]; %constante de aceleración mínima
Cmax=[2.5 2.5 2.5]; %constante de aceleración máxima
Wmin=0.3;           %factor de inercia mínimo
Wmax=0.9;           %factor de inercia máximo
alpha=1;            %factor para cálculo de estrechamiento

%Parámetros multiobjetivo

%w=[0.01 999]       %Pesos para cada función objetivo
%w=[999 0.01]
w=[0 1]
rho=1                %Factor para garantizar eficiencia del modelo
sancion=1000;

%-----CAMPOS-----

%Contorno 1 - donde X es la función objetivo 1 y Y la función objetivo 2
x1=linspace(0,150);
y1= linspace(0,0.01);
[X1,Y1] = meshgrid(x1,y1);

Z = OnlyFO(X1,Y1,w,rho);
contour(X1,Y1,Z);

%Contorno 2 - donde X es un parámetro de entrada y Y otro parámetro de
%entrada , por ejemplo d1 y L2
% x1=linspace(1,6);
% y1= linspace(1,6);
% const1=4.5;
% const2=12;
% [X1,Y1] = meshgrid(x1,y1);
%
% for i=1:size(X1,1)
%     for j=1:size(X1,2)
%         element=[X1(i,j) const1 Y1(i,j) const2 ] % [ d1 d2 d3 L2]
%         [FOCamp,FOsCamp]=FuncionAdaptacion(f1,f2,element,1,w,rho);

```

```
%          [FOCamp] = Restricciones(element,FOsCamp,FOCamp,sancion, 1);
%          Z(i,j)=FOCamp;
%          end
% end
% contour(X1,Y1,Z);
xlabel("Masa [Lb]")
ylabel("Deflexión [in]")
title("Masa vs Deflexión")
```

**ANEXO V. Parametrización concentrador de esfuerzos Kt.**

Kt:

	A	B	C	D	E	F	G	H	I
	ConcEsf								
	Ddrd	VarName2	VarName3	VarName4	VarName5	VarName6	VarName7	VarName8	VarName9
	Number	Number	Number	Number	Number	Number	Number	Number	Number
1	D/d...	1.0100	1.0200	1.0500	1.1000	1.2000	1.5000	2	3
2	0.0100	1.6400	1.9400	2.4000	2.8000	3.3000	3.8000	0	0
3	0.0200	1.4600	1.6400	2.0400	2.3200	2.6200	2.8000	0	0
4	0.0300	1.3600	1.5400	1.8400	2.0800	2.3200	2.5200	0	0
5	0.0400	1.3200	1.4600	1.7200	1.9400	2.1600	2.3000	2.4200	2.4900
6	0.0500	1.3000	1.4200	1.6400	1.8400	2	2.1400	2.2400	2.4200
7	0.0600	1.2800	1.3800	1.6000	1.7600	1.9000	2.0200	2.1200	2.2800
8	0.0700	1.2600	1.3400	1.5600	1.7000	1.8200	1.9200	2.0200	2.1500
9	0.0800	1.2400	1.3300	1.5200	1.6600	1.7600	1.8400	1.9300	2.0400
10	0.0900	1.2300	1.3200	1.5000	1.6300	1.7000	1.7700	1.8500	1.9600
11	0.1000	1.2000	1.3100	1.4800	1.6000	1.6400	1.7200	1.7800	1.8800
12	0.1100	0	0	0	0	1.6200	1.6800	1.7200	1.8200
13	0.1200	0	0	0	0	1.5800	1.6400	1.6900	1.7600
14	0.1300	0	0	0	0	1.5500	1.6000	1.6400	1.7200

Kts:

	A	B	C	D	E	F	G
	ConcEsfS1						
	VarName1	Ddrd	VarName3	VarName4	VarName5	VarName6	VarName7
	Text	Number	Number	Number	Number	Number	Number
1		D/d...	1.1110	1.2500	1.6660	2	2.5000
2		0.0500	1.4500	1.5700	0	0	0
3		0.1000	1.2850	1.3500	1.3850	0	0
4		0.1300	1.2350	1.2800	1.3150	1.3400	1.3450
5		0.1500	1.2150	1.2550	1.2850	1.3050	1.3150
6		0.1800	1.1850	1.2150	1.2450	1.2600	1.2700
7		0.2000	1.1750	1.2000	1.2200	1.2350	1.2450
8		0.2300	1.1550	1.1750	1.1950	1.2100	1.2150
9		0.2500	1.1400	1.1650	1.1800	1.1900	1.2000
10		0.2800	1.1250	1.1450	1.1600	1.1700	1.1750
11		0.3000	1.1200	1.1350	1.1500	1.1550	1.1600

qt:

	A	B	C
	ConcEsfS2		
	VarName1	VarName2	VarName3
	Number	Number	Number
1		80	100
2	0	0.3500	0.4100
3	0.0050	0.4600	0.5500
4	0.0100	0.5700	0.6250
5	0.0200	0.6480	0.7000
6	0.0300	0.6800	0.7400
7	0.0400	0.7000	0.7550
8	0.0500	0.7350	0.7800
9	0.0600	0.7500	0.7950
10	0.0700	0.7680	0.8050
11	0.0800	0.7850	0.8200
12	0.0900	0.7930	0.8260
13	0.1000	0.8000	0.8370
14	0.1100	0.8050	0.8410
15	0.1200	0.8150	0.8490
16	0.1300	0.8250	0.8550
17	0.1400	0.8380	0.8600
18	0.1500	0.8420	0.8650
19	0.1600	0.8500	0.8730

qs:

	A	B	C
	ConcEsfS3		
	VarName1	VarName2	VarName3
	Number	Number	Number
1		80	120
2	0	0.4100	0.5150
3	0.0050	0.5500	0.6500
4	0.0100	0.6250	0.7400
5	0.0200	0.7000	0.7900
6	0.0300	0.7400	0.8240
7	0.0400	0.7550	0.8430
8	0.0500	0.7800	0.8500
9	0.0600	0.7950	0.8620
10	0.0700	0.8050	0.8750
11	0.0800	0.8200	0.8820
12	0.0900	0.8260	0.8900
13	0.1000	0.8370	0.8980
14	0.1100	0.8410	0.9020
15	0.1200	0.8490	0.9100
16	0.1300	0.8550	0.9180
17	0.1400	0.8600	0.9200
18	0.1500	0.8650	0.9270
19	0.1600	0.8730	0.9350