

ANÁLISIS DEL DISEÑO ÓPTIMO DE UN REDUCTOR DE VELOCIDAD

ANDRES FELIPE CASTILLO TORRES

JHON FERNANDO AVILA URIBE

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS

ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA

Y TELECOMUNICACIONES

BUCARAMANGA

2011

ANÁLISIS DEL DISEÑO ÓPTIMO DE UN REDUCTOR DE VELOCIDAD

ANDRES FELIPE CASTILLO TORRES

JHON FERNANDO AVILA URIBE

Trabajo de Grado para optar al título de INGENIERO ELECTRÓNICO

DR. CARLOS RODRIGO CORREA CELY

DIRECTOR TRABAJO DE GRADO

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS

ESCUELAS DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA

Y TELECOMUNICACIONES

BUCARAMANGA

2011

CONTENIDO

	Pág.
INTRODUCCIÓN	13
1. FUNDAMENTOS TEÓRICOS	14
1.1. <i>REDUCTOR DE GOLINSKI</i>	15
1.2. <i>PARTICLE SWARM OPTIMIZATION</i>	16
1.3. <i>PSO UNIFICADO</i>	17
1.4. <i>FUNCION DE PENALIZACIÓN</i>	18
1.5. <i>MANEJO DE LOS LIMITES</i>	19
2. RESULTADOS Y ANÁLISIS	20
2.1. <i>EJERCICIO 1</i>	20
2.2. <i>EJERCICIO 2</i>	22
2.3. <i>REDUCTOR DE GOLINSKI</i>	23
2.4. <i>EFECTO DE LA PRESICIÓN</i>	25
2.5. <i>EFECTO DEL FACTOR DE UNIFICACIÓN</i>	33
2.6. <i>PSO VS UPSO</i>	34
2.7. <i>AUMENTO DE LAS PARTÍCULAS</i>	36
2.8. <i>ALEATORIEDAD PSO</i>	37
2.9. <i>FACTOR DE RESTRICCIÓN</i>	38

3. OBSERVACIONES Y CONCLUSIONES	43
BIBLIOGRAFÍA	45

FIGURAS

	Pág.
FIGURA 1. REDUCTOR DE VELOCIDAD, FUENTE: AUTORES	15
FIGURA 2. CIRCUNFERENCIA PRIMITIVA, FUENTE: AUTORES	16
FIGURA 3. EFECTO DE LA PRESICIÓN, FUENTE: AUTORES	25
FIGURA 4. EFECTO DE LA PRESICIÓN, FUENTE: AUTORES	29
FIGURA 5. DISTRIBUCIÓN DE LAS PARTICULAS, AUTORES	32
FIGURA 6. PSO VS UPSO , FUENTE: AUTORES	35
FIGURA 7. FUNCIÓN OBJETIVO TAMAÑO DE ENJAMBRE, FUENTE: AUTORES	37
FIGURA 8. NUMERO DE ITERACIONES VS FUNCIÓN OBJETIVO, FUENTE: AUTORES	38
FIGURA 9. EFECTO FACTOR DE RESTRICCIÓN, FUENTE: AUTORES	39

TABLAS

	Pág.
TABLA 1. EFECTO DE LA PRECISIÓN EJERCICIO 1, FUENTE: AUTORES	21
TABLA 2. FACTOR DE UNIFICACIÓN EJERCICIO 1, FUENTE: AUTORES	21
TABLA 3. EFECTO DE LA PRECISIÓN EJERCICIO 2, FUENTE: AUTORES	22
TABLA 4. FACTOR DE UNIFICACIÓN EJERCICIO 2, FUENTE: AUTORES	22
TABLA 5. RESTRICCIONES DE LA FUNCIÓN OBJETIVO, FUENTE: AUTORES	24
TABLA 6. EFECTO DE LA PRECISIÓN, FUENTE: AUTORES	25
TABLA 7. EFECTO DEL FACTOR DE UNIFICACIÓN, FUENTE: AUTORES	33
TABLA 8. PSO VS UPSO, FUENTE: AUTORES	34
TABLA 9. EFECTO DEL NÚMERO DE PARTÍCULAS, FUENTE: AUTORES	36
TABLA 10. ALEATORIEDAD DE UPSO, FUENTE: AUTORES	37
TABLA 11. VARIACIÓN FACTOR DE RESTRICCIÓN, FUENTE: AUTORES	39
TABLA 12. UPSO VS SOFTWARE COMERCIAL, FUENTE: AUTORES	42
TABLA 13. DIMENSIONES ÓPTIMAS, FUENTE: AUTORES	42

RESUMEN

TÍTULO

ANÁLISIS DEL DISEÑO ÓPTIMO DE UN REDUCTOR DE VELOCIDAD *

AUTORES

Andres Felipe Castillo Torres**

Jhon Fernando Avila Uribe**

PALABRAS CLAVES

REDUCTOR DE VELOCIDAD, PSO CON RESTRICCIONES, FUNCIÓN DE PENALIZACIÓN.

DESCRIPCIÓN

La necesidad de obtener resultados óptimos en sistemas utilizados en robótica debido a los inconvenientes que se presentan cuando no se trabaja con un sistema de transmisión apropiado para cada aplicación, conlleva a utilizar técnicas de optimización para el diseño de las dimensiones un reductor de velocidad. Éste problema inicialmente propuesto por J Golinski en 1970, se desarrolló por el método optimización de enjambre de partículas (PSO) y una de sus variantes (UPSO), sujeto a restricciones. Actualmente PSO se utiliza en la cinemática de robots, sistemas de control, procesos de fabricación y procesamiento digital de señales con el objetivo de mejorar la eficiencia de cada proceso o resolver problemas de costos incluso, beneficiando el medio ambiente; por esta razón, los algoritmos evolutivos se posicionan como una de las mejores alternativas en cuanto a optimización se refiere. El algoritmo formulado enfrenta las restricciones planteadas con una función de penalización adaptadas al PSO, estrategia manejada en diferentes métodos de optimización con restricciones, ofreciendo eficiencia, rendimiento en el manejo de las mismas, y reduciendo el tiempo de cómputo. La solución encontrada contrastó muy bien con los resultados encontrados mediante otras técnicas de optimización, confirmando que PSO y su variante UPSO son una herramienta matemática sencilla y robusta para optimización de este problema de ingeniería.

* Trabajo de Grado.

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones.
Director: Carlos Rodrigo Correa Cely.

SUMMARY

TITLE

ANALYSIS OF THE OPTIMAL DESIGN OF A SPEED REDUCER.*

AUTHORS

Andres Felipe Castillo Torres **

Jhon Fernando Avila Uribe **

KEYWORDS

SPEED REDUCER, PSO WITH CONSTRAINS, PENALTY FUNCTION

DESCRIPTION

The necessity of finding optimal solutions in robotic systems due to some objections that are present when is working with the inappropriate transmission system, leads to use optimization techniques for the design of the speed reducer's dimensions. This issue originally proposed by J. Golinski in 1970. It was developed in this paper by particle swarm optimization and one of its variants unified particle swarm optimization subject to constraints. Currently PSO is used in robot kinematics, control systems, manufacturing process, and digital signal processing with the goal of improving the efficiency of each process or solving problems of cost even, helping environment; for this reason evolutionary algorithms are one the best alternatives in optimization. The algorithm formulated for handling constraints uses a penalty function adapted to particle swarm optimization, strategy that has used in different optimization methods with constraints, offering efficiency, performance dealing with constrains and decreasing the computational cost. The solution found contrasts pretty good with many results that have been calculated by others optimization techniques, which has demonstrated that PSO and its variant are a simple and robust mathematical algorithm for solving this engineering problem.

* Degree Work.

** Faculty of Physic mechanical Engineering. School of electric, electronic and telecommunications. Project director: Carlos Rodrigo Correa Cely.

INTRODUCCIÓN

En robótica se necesita realizar movimientos precisos con cierta velocidad lineal y angular y/o el de generar un torque definido. Por esta razón los reductores de velocidad son diseñados adecuadamente para este propósito, ya que además deben cumplir criterios en cuanto a peso y dimensiones.

La búsqueda del reductor de velocidad apropiado para cada robot, implica hacer un diseño específico para cada aplicación, creando así la necesidad de encontrar el reductor óptimo de bajo peso y bajo rozamiento, capaz de girar a una velocidad eficiente con las dimensiones mínimas que se necesiten. Dicho interés se ha incrementado gradualmente en los últimos años, realizando nuevas invenciones para reducir las pérdidas, ya sea en el sistema de control o en el sistema mecánico, surgiendo una tendencia a la optimización inherente a la reducción de costos en la planta o parte de un plan para la sostenibilidad del medio ambiente. Es aquí donde un algoritmo robusto de optimización entra a ser pieza fundamental en la ingeniería, teniendo en cuenta que generalmente los modelos matemáticos están sujetos a restricciones; es fundamental utilizar un método capaz de manejar estas restricciones, generando respuestas que estén dentro del conjunto de soluciones posibles y que respeten las condiciones estipuladas por el diseño.

Este artículo presenta la solución numérica del modelo matemático de un reductor de velocidad sujeto a restricciones así como el método utilizado para minimizarlo [1].

1. FUNDAMENTOS TEÓRICOS

Generalmente en robótica se necesita de un mecanismo capaz de transmitir el movimiento de un actuador a una articulación, comprometiéndose a una buena relación entre movilidad y dimensión, sujeto a un sistema eficiente. Un reductor de velocidad toma gran importancia, debido a que sus características físicas desarrollan efectos sobre el conjunto motor-reductor y en todo el sistema.

Este sistema de transmisión debe ser preciso, además poseer ciertas características dimensionales apropiadas para su ubicación y ensamble dentro del espacio presupuestado por el diseñador; la relación de engranajes no puede ser tan alta puesto que una parte de la potencia del motor se puede estar empleando para mover los engranajes del reductor introduciendo pérdidas en el torque.

La transmisión de velocidad en un reductor se realiza a través de sus engranajes, por ende las dimensiones de un reductor de velocidad se relacionan directamente con la geometría del diente, la tensión en los mismos, el material del piñón, la superficie de contacto, el ángulo de presión, la potencia transmitida y la velocidad angular; igualmente su peso depende de la geometría del engranaje y la geometría de cada uno de sus ejes.

Dependiendo la aplicación se usan diferentes tipos de engranajes los cuales se clasifican en:

Ejes Perpendiculares entre los cuales se encuentran cónicos de dientes rectos, helicoidales cruzados y cónicos de dientes helicoidales.

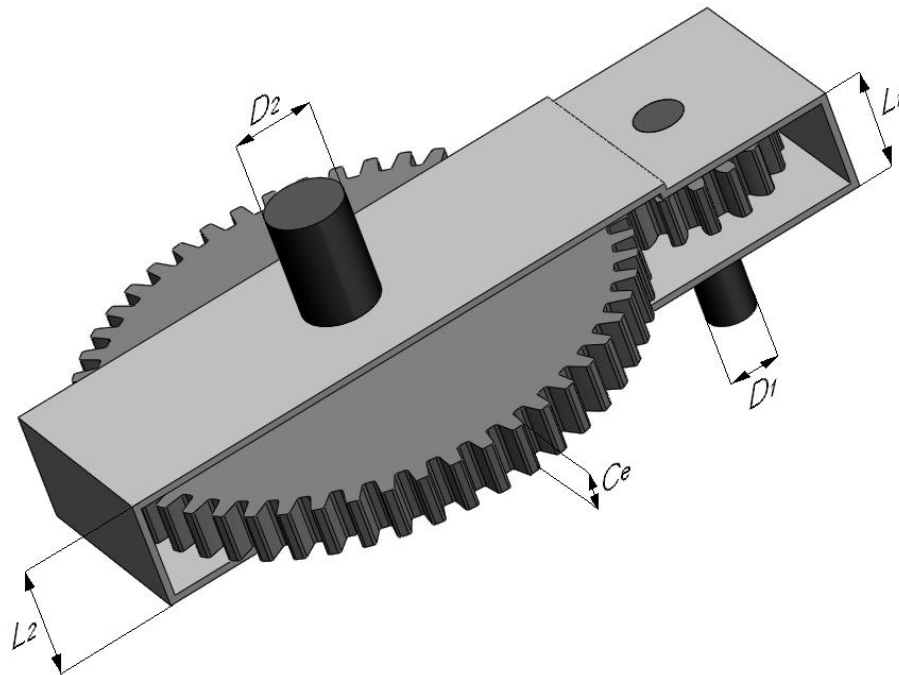
Ejes paralelos en los que se incluyen los engranajes cilíndricos de dientes rectos y cilíndricos de dientes helicoidales.

Se propuso un engranaje cilíndrico recto, ideal para trabajar a velocidades pequeñas y medias en brazos robóticos, cuyos criterios de diseño no son tratados a fondo, debido a que el objetivo de este trabajo fue de abordar un problema de ingeniería a través de una alternativa sencilla y robusta sin utilizar procedimientos matemáticos requeridos por otros algoritmos de optimización.

1.1. REDUCTOR DE GOLINSKI

Inicialmente el modelo matemático fue propuesto por J. Golinski. El objetivo del diseño es minimizar su geometría para que el reductor rote a su más eficiente velocidad, y su peso definido en función de las dimensiones junto con las restricciones que establece el diseño. Ver figura 1.

Figura1. Reductor de velocidad.



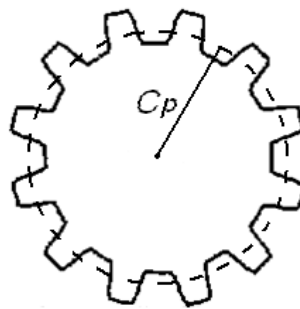
Fuente: Autores.

La función objetivo del reductor se define como [2]:

$$f(x) = \sum_{i=0}^2 [8M_d^2(N_d - 4,4) \frac{\pi C_e}{4} + ((M_d N_d - 6,4M_d)^2 - 5,76D_i^2) \frac{\pi C_e}{12} + (4,76D_i^3) \frac{\pi}{2} + \frac{\pi}{4} D_i^2 L_i] \quad (1)$$

La función objetivo depende también del ángulo de presión ejercido por los engranajes, generalmente es de 20° , además depende del material del cual se fabrica el reductor, por tal motivo, debe soportar las fricciones y tensiones permitidas entre los dientes del engranaje, por otra parte, debe trabajar bajo ciertas condiciones ambientales. Para este caso se utiliza acero al carbono.

Figura 2. Circunferencia Primitiva.



Fuente: Autores.

Cabe resaltar que M_d se define como la relación entre la circunferencia primitiva C_p y el número de dientes para cada piñón N_d como se observa en la figura 2. La definición formal del modelo matemático de un reductor de velocidad se puede consultar en libros de diseño de engranajes [4 ,5].

1.2. PARTICLE SWARM OPTIMIZATION (PSO)

Particle swarm optimization (PSO), fue desarrollado en 1995 por Russell Eberhart y J. Kennedy, influenciados por el comportamiento colectivo de algunas colonias de animales. En el algoritmo se busca encontrar una solución óptima para la función objetivo (*fitness function*). Se inicia seleccionando una población de partículas al azar (posibles soluciones denotadas por un vector posición X_i y un vector velocidad V_i para cada partícula), evaluando cada una de ellas en dicha función. Por cada iteración se tiene en cuenta la experiencia de las partículas vecinas determinando que tan cerca están del óptimo, por lo cual, se modifica la posición y velocidad para la siguiente partícula por medio de la siguiente expresión [6]:

$$V_{i+1} = w * V_i + c_1 * r_1 * (P_l - X_i) + c_2 * r_2 * (P_g - X_i) \quad (2)$$

$$X_{i+1} = X_i + V_{i+1} \quad (3)$$

Trabajando en un espacio vectorial de N dimensiones se tiene:

El vector X_i representa la posición de cada partícula expresado como $X_i = (X_{i1}, X_{i2}, \dots, X_{in})$; el vector V_i representa la velocidad de cada partícula expresada como $V_i = (V_{i1}, V_{i2}, \dots, V_{in})$. Cada posición de la partícula es candidata a la solución por lo cual, en cada iteración se analizan dos valores muy importantes para cada partícula respecto al óptimo de la función, como lo son la mejor posición de cada partícula (P_{best}) representado por $P_i = (P_{i1}, P_{i2}, \dots, P_{in})$; y la mejor posición de toda la población (P_{global}) expresado mediante $P_g = (P_{g1}, P_{g2}, \dots, P_{gn})$. Finalmente se tiene que c_1, c_2 son constantes positivas llamadas “coeficientes de aceleración”, w es la ponderación de inercia; r_1 y r_2 , valores aleatorios que están en un rango de 0 a 1.

1.3. PSO UNIFICADO

Las propiedades de búsqueda del mejor local P_i y el global P_g se modificaron con el objetivo de mejorar el desempeño del algoritmo. Las velocidades del global G_i y el local L_i se redefinen como sigue [7-10]:

$$G_i = \chi \left(V_i + c_1 * r_1 * (P_l - X_i) + c_2 * r_2 * (P_g - X_i) \right) \quad (4)$$

$$L_i = \chi \left(V_i + c_1 * r_3 * (P_l - X_i) + c_2 * r_4 * (P_g - X_i) \right) \quad (5)$$

Estas dos direcciones son unificadas en una sola ecuación.

$$U_i = u * G_i + (1 - u) * r_n * L_i \quad (6)$$

El factor de restricción χ se introduce para limitar la velocidad de las partículas evitando así la explosión del enjambre [7]. El factor de unificación u determina el dominio de las componentes global y local enriqueciendo las capacidades de búsqueda del algoritmo. La ecuación (3) que modela la posición de las partículas se reemplaza entonces por:

$$X_{i+1} = X_i + U_i \quad (7)$$

Se usan diferentes alternativas para manejar las restricciones, pero algunas requieren mucho tiempo de cómputo debido a su estrategia para mantener las partículas en la región factible; por lo tanto, se trabajó con una función de penalización, la cual, modifica la función objetivo por medio de un factor que busca penalizar aquellas partículas que no se encuentren dentro de la región factible.

1.4. FUNCIÓN DE PENALIZACIÓN

Los problemas en ingeniería generalmente están sujetos a restricciones de la forma [11-15]:

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

Para el manejo de estas restricciones se usa una función penalizada $fp(x)$ propuesta en [16] y definida como:

$$fp(x) = f(x) + h * H(x) \quad (8)$$

$$h = k^{\frac{3}{2}}$$

$f(x)$, corresponde a la función objetivo y $H(x)$ es el factor de penalización. Además se introduce un valor de penalización h que varía en función del número de iteraciones k .

$$H(x) = \sum_{i=0}^m (\theta(g_i(x)) * g_i(x)^{r(g_i(x))}) \quad (9)$$

$$g_i(x) = \begin{cases} g_i(x), & g_i(x) > 0 \\ 0, & g_i(x) < 0 \end{cases} \quad (10)$$

Los parámetros de penalización se definen como:

$$\theta(g_i(x)) = \begin{cases} 10, & \text{si } g_i(x) < 0,001 \\ 20, & \text{si } 0,001 \leq g_i(x) \leq 0,1 \\ 100, & \text{si } 0,1 \leq g_i(x) \leq 1,0 \\ 300, & \text{de otro modo} \end{cases} \quad (11)$$

$$r(g_i(x)) = \begin{cases} 1 & \text{si } g_i(x) < 1 \\ 2 & \text{de otro modo} \end{cases} \quad (12)$$

$r(g_i(x))$, hace referencia al poder de la función de penalización, es decir, aplica una sanción en función de la violación de las restricciones. Los valores de los parámetros de penalización se proponen en [17] como resultado de cálculos experimentales.

1.5. MANEJO DE LOS LÍMITES

Todas las restricciones están limitadas a un rango de operación, siendo necesario encontrar un método para que la solución se mantenga dentro de la región factible. Para ello la mejor solución a este problema fue planteada en [18] donde se aplica:

$$x_i = \begin{cases} x_{max} - 0,1 * rand_1 * (x_{max} - x_{min}), & x_i > x_{max} \\ x_{min} + 0,1 * rand_2 * (x_{max} - x_{min}), & x_i < x_{min} \\ x_i, & \text{de otro modo} \end{cases} \quad (13)$$

De esta manera las partículas se verán forzadas a no salirse de la región impuesta por los límites. Para iniciar las partículas igualmente se tiene en cuenta que se encuentren distribuidas aleatoriamente dentro de la región comprendida entre los límites, por lo que se inicializaron siguiendo las siguientes ecuaciones [19]:

$$x_0 = x_{min} + rand_3(x_{max} - x_{min}) \quad (14)$$

$$v_0 = x_{min} + rand_4(x_{max} - x_{min}) \quad (15)$$

2. RESULTADOS Y ANÁLISIS

A continuación se especifican las características del equipo en el cual se realizaron todas las pruebas:

Sistema operativo: Windows 7

Procesador: AMD Athlon II Dual-Core M300 2Ghz

Memoria Ram: 2 Gb

Para comprobar la efectividad del algoritmo se resuelven dos ejercicios propuestos en [12], con los cuales se puede observar la influencia del factor de unificación sobre las partículas solución y el tiempo de cómputo. Además se busca ver el efecto que tiene la precisión sobre las soluciones, y el costo computacional que esto implica.

El error para cada variable se define en el algoritmo como el valor absoluto de la diferencia entre la partícula siguiente y la partícula anterior:

$$error = |x_{i+1} - x_i| \quad (16)$$

Para observar el comportamiento de la función objetivo y el efecto que tiene la precisión sobre ésta, se varía el valor numérico de la precisión definida como parámetro de entrada y criterio de parada del algoritmo una vez que $p < error$. A continuación se muestra los dos ejercicios propuestos y su posterior análisis.

2.1. EJERCICIO 1

Minimizar:

$$f(x) = \frac{1}{3}(x_1 + 1)^3 + x_2$$

Sujeto a:

$$1 - x_1 \leq 0$$

$$-x_2 \leq 0$$

En la tabla 1 se muestra el efecto de la precisión sobre el valor de la función objetivo 1 y el tiempo de cómputo para una población inicial de 100 partículas y con un factor de unificación de $u = 0,5$.

Tabla 1. Efecto de la precisión ejercicio 1, $u = 0,5$, 100 partículas.

Precisión	X_1	X_2	$f(x)$	k^*	t (s)
1×10^{-1}	1,1393	0,5518	3,8155	2	4
1×10^{-2}	1,0001	0,0016	2,6688	20	4
1×10^{-3}	1,0002	0,0006	2,6681	23	4
1×10^{-4}	1,0000	0,0000	2,6667	113	5
1×10^{-5}	1,0000	0,0000	2,6667	119	5
1×10^{-6}	1,0000	0,0000	2,6667	245	5

Fuente: Autores.

La tabla 2, muestra el efecto del factor de unificación sobre el valor de la función objetivo 1 y el tiempo de cómputo para una población inicial de 100 partículas.

Tabla 2. Factor de unificación ejercicio 1, $p = 1 \times 10^{-3}$, 100 partículas.

u	X_1	X_2	$f(x)$	k	t (s)
0	1,0006	0,0000	2,6675	25884	93
0,1	1,0006	0,0000	2,6674	8659	31
0,2	1,0000	0,0001	2,6669	2873	13
0,3	1,0002	0,0005	2,6681	43	6
0,4	1,0009	0,0024	2,6726	24	4
0,5	1,0000	0,0000	2,6667	62	3
0,6	1,0000	0,0000	2,6667	49	4
0,7	1,0000	0,0001	2,6668	40	3
0,8	1,0000	0,0000	2,6667	44	5
0,9	1,0000	0,0000	2,6667	71	3
1	1,0000	0,0000	2,6667	47	4

Fuente: Autores.

* k representa el número de iteraciones y t el tiempo de cómputo para todas las tablas.

2.2. EJERCICIO 2

Minimizar:

$$f(x) = (x_1 - 1)^2 + (x_2 - 5)^2$$

Sujeto a:

$$x_1^2 + x_2^2 \leq 4$$

$$-(x_1 - 2)^2 + x_2 \leq 3$$

La tabla 3, muestra el efecto de la precisión sobre el valor de la función objetivo 2 con 100 partículas como población inicial y factor de unificación de $u = 0,5$.

Tabla 3. Efecto de la precisión ejercicio 2, $u = 0,5$, 100 partículas.

Precisión	x_1	x_2	$f(x)$	k	t (s)
1×10^{-1}	0,7596	4,5363	0,2728	11	6
1×10^{-2}	0,7501	4,5591	0,2568	23	4
1×10^{-3}	0,7495	4,5594	0,2569	22	5
1×10^{-4}	0,7500	4,5625	0,2539	120	4
1×10^{-5}	0,7500	4,5625	0,2539	80	4
1×10^{-6}	0,7500	4,5625	0,2539	103	5

Fuente: Autores.

Tabla 4. Factor de unificación ejercicio 2, $p = 1 \times 10^{-3}$, 100 partículas.

u	X_1	X_2	$f(x)$	k	t (s)
0	0,7506	4,5608	0,2551	10396	48
0,1	0,7500	4,5622	0,2541	14110	54
0,2	0,7501	4,5620	0,2543	1882	14
0,3	0,7500	4,5625	0,2539	634	6
0,4	0,7500	4,5619	0,2545	60	4
0,5	0,7500	4,5625	0,2539	82	4
0,6	0,7495	4,5617	0,2549	31	5
0,7	0,7496	4,5587	0,2574	18	3
0,8	0,7499	4,5622	0,2543	31	3
0,9	0,7500	4,5624	0,2540	50	3
1	0,7500	4,5625	0,2539	142	5

Fuente: Autores.

En la tabla 4, se muestra el efecto del factor de unificación sobre el valor de la función objetivo 2 y el tiempo de cómputo para una población inicial de 100 partículas con una precisión de $p = 1 \times 10^{-3}$.

Como se aprecia en los resultados obtenidos en la tablas 1 y 3, para valores desde $p = 1 \times 10^{-4}$ la solución tiende a un valor constante y se presenta muy poca variación con respecto al valor medio de la función objetivo para cada precisión, por ésta razón, éste valor se selecciona como respuesta analítica.

Un efecto no deseado se advierte al tomar valores del factor de unificación entre $u = 0$ y $u = 0,2$ en vista de que aumenta el número de iteraciones y el tiempo de cómputo empleado para encontrar el óptimo en comparación a valores cercanos a $u = 1$. La mejor opción para el factor de unificación está entre $u = 0,3$ y $u = 0,5$ como consecuencia de sus buenos resultados. Sin embargo $u = 0,5$ ofrece un balance de búsqueda equitativo reflejado en sus resultados numéricos y el tiempo de cómputo empleado. El efecto de la precisión no es muy notable en las tablas anteriores para cada función objetivo, debido a la sencillez de las funciones y la rápida convergencia del algoritmo, pero los resultados confirman la respuesta obtenida por las referencias.

Una vez verificado la efectividad del algoritmo propuesto se resuelve el problema del reductor. Para la selección de los valores numéricos de los parámetros se utilizan los valores propuestos en [10] a razón de resultados obtenidos en pruebas experimentales: $\chi = 0,6$; $c_1 = c_2 = 2,8333$; $u = 0,5$.

2.3. REDUCTOR DE GOLINSKI

La selección de precisión tiene su sentido físico al fabricar el reductor, en vista de no haber una cortadora de metal con la exactitud necesaria para diferenciar las cifras decimales y poder fabricar el reductor. Las tablas y las gráficas son realizadas con una precisión de $p = 1 \times 10^{-4}$.

Función Objetivo:

$$f(\vec{x}) = 0,7854C_eM_d^2(3,3333N_d^3 + 14,9334N_d - 43,0934) - 1,508C_e(D_1^2 + D_2^2) + 7,4777(D_1^3 + D_2^3) + 0,7854(L_1D_1^2 + L_2D_2^2) \quad (17)$$

En la tabla 5 se muestran las restricciones que limitan la función objetivo.

Cada restricción tiene un sentido físico relacionado con la tensión de flexión de los engranajes, la tensión de compresión máxima, límites de longitud y el número de dientes de piñón. Estos valores numéricos son ajustados con los parámetros propuestos en [3].

Tabla 5. Restricciones de la función objetivo.

$g_1(\vec{x}) = \frac{27}{C_eM_d^2N_d} - 1 \leq 0 \quad (18)$ <p>Tensión de doblamiento permitida de dientes del engranaje</p>	$g_2(\vec{x}) = \frac{397,5}{C_eM_d^2N_d} - 1 \leq 0 \quad (19)$ <p>Tensión permisiva superficial compresiva</p>	$g_3(\vec{x}) = \frac{1,93L_1^3}{M_dN_dD_1^4} - 1 \leq 0 \quad (20)$ <p>Deflexión transversal eje 1</p>
$g_4(\vec{x}) = \frac{1,93L_2^3}{M_dN_dD_2^4} - 1 \leq 0 \quad (21)$ <p>Deflexión transversal eje 2</p>	$g_5(\vec{x}) = \frac{1,0}{110D_1^3} \sqrt{\left(\left(\frac{745,0L_1}{M_dN_d}\right)^2 + 16,9 \times 10^6\right)} - 1 \leq 0 \quad (22)$ <p>Momento de tensión y torsión eje 1</p>	
$g_6(\vec{x}) = \frac{1,0}{85D_2^3} \sqrt{\left(\left(\frac{745,0L_2}{M_dN_d}\right)^2 + 157,5 \times 10^6\right)} - 1 \leq 0 \quad (23)$ <p>Momento de tensión y torsión eje 1</p>		$g_7(\vec{x}) = \frac{M_dN_d}{40} - 1 \leq 0 \quad (24)$ <p>Condición de dimensión</p>
$g_8(\vec{x}) = \frac{5M_d}{C_e} - 1 \leq 0 \quad (25)$ <p>Ancho relativo del ancho del engranaje</p>	$g_9(\vec{x}) = \frac{C_e}{12M_d} - 1 \leq 0 \quad (26)$ <p>Ancho relativo del ancho del engranaje</p>	$g_{10}(\vec{x}) = \frac{1,5D_1 + 1,9}{L_1} - 1 \leq 0 \quad (27)$ <p>Restricción de diseño</p>
$g_{11}(\vec{x}) = \frac{1,1D_2 + 1,9}{L_2} - 1 \leq 0 \quad (28)$ <p>Restricción de diseño</p>		

Fuente: Autores

El algoritmo debe ser capaz de encontrar una solución que satisfaga las restricciones. Siete variables se manejan en el diseño del reductor, cada una

con límites inferiores y superiores que representan las condiciones físicas impuestas por el modelo general del peso de un reductor de velocidad:

$$2,6 \leq C_e \leq 3,6 ; 0,7 \leq M_d \leq 0,8 ; 17 \leq N_d \leq 28 ; 7,3 \leq L_1 \leq 8,3 ; 7,8 \leq L_2 \leq 8,3 ; 2,9 \leq D_1 \leq 3,9 ; 5,0 \leq D_2 \leq 5,5.$$

2.4. EFECTO DE LA PRECISIÓN

En la tabla 6, se observa el efecto que produce aumentar la precisión sobre la función objetivo con un factor de unificación $u = 0,5$ y una población de 100 partículas.

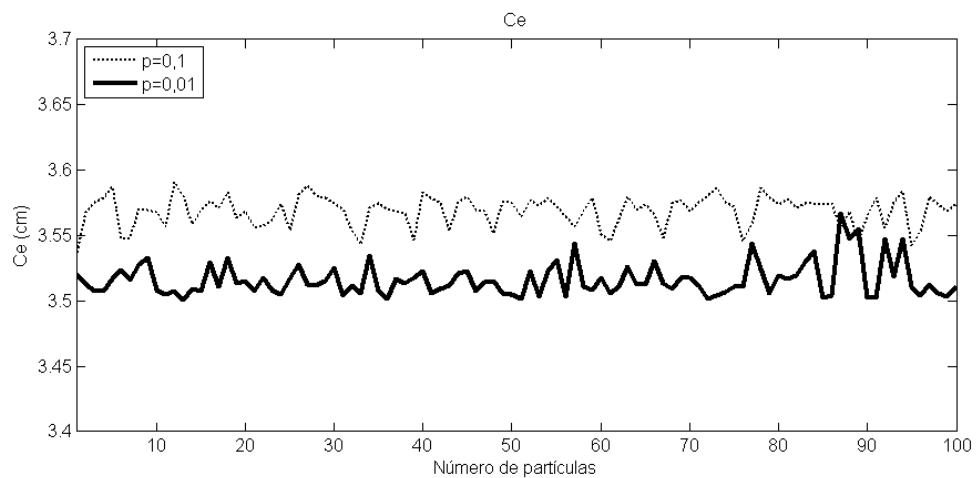
Tabla 6. Efecto de la precisión, $u = 0,5$,100 partículas.

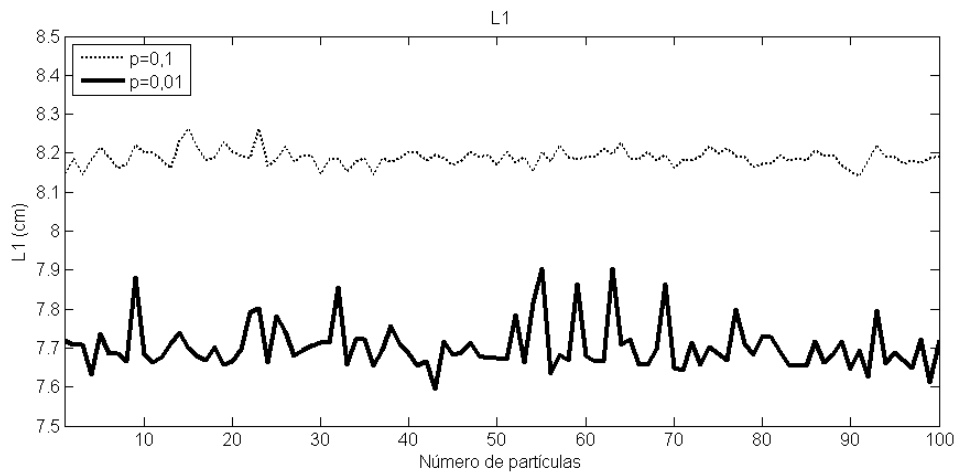
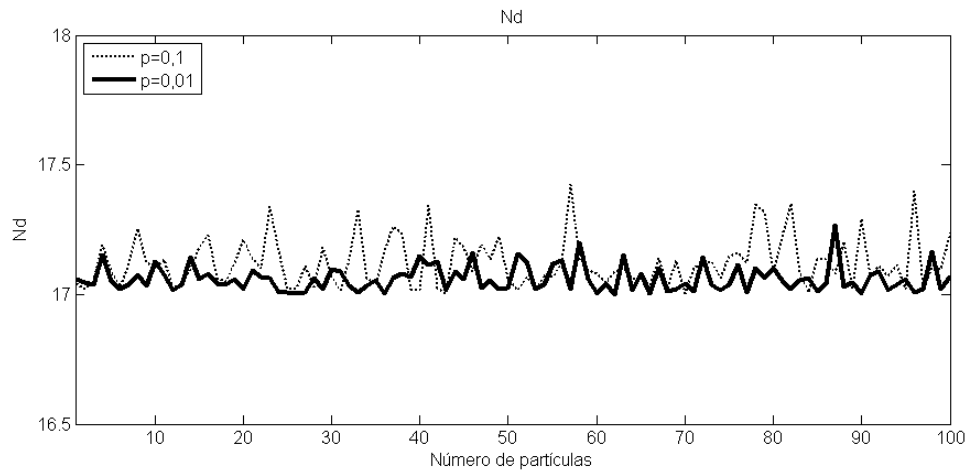
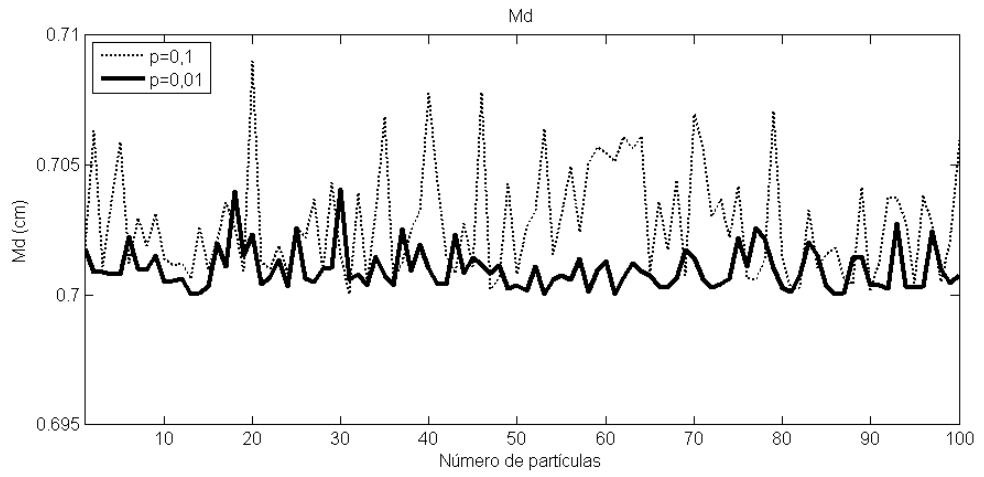
Precisión	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$f(x)$	k	t (s)	Σ
1×10^{-1}	3,5413	0,7016	17,0195	7,8147	7,8126	3,4736	5,3007	3070,1	13	9	26,888
1×10^{-2}	3,5020	0,7004	17,0051	7,6557	8,1416	3,3553	5,2893	3013,3	57	9	16,066
1×10^{-3}	3,5009	0,7000	17,0010	7,3579	8,2057	3,3546	5,2874	3008,0	4556	19	0,687
1×10^{-4}	3,5001	0,7000	17,0001	7,3051	7,8017	3,3503	5,2868	2996,6	7827656	22307	0,087

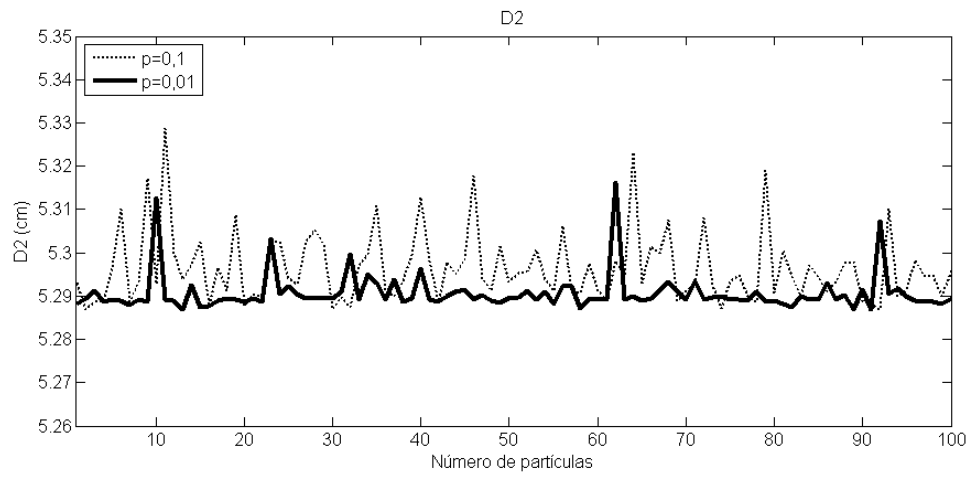
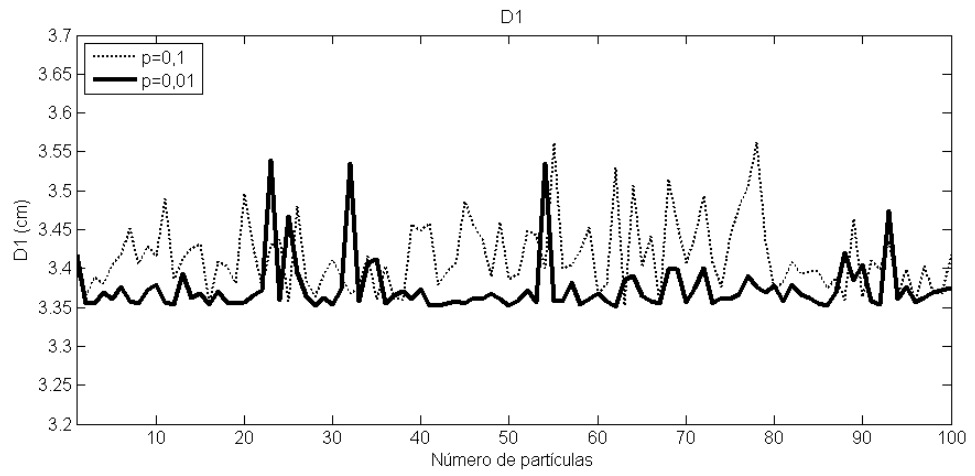
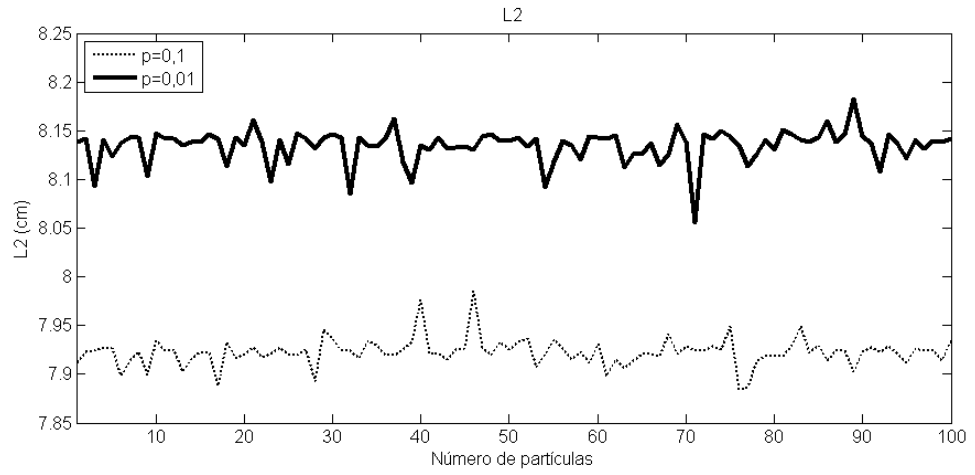
Fuente: Autores.

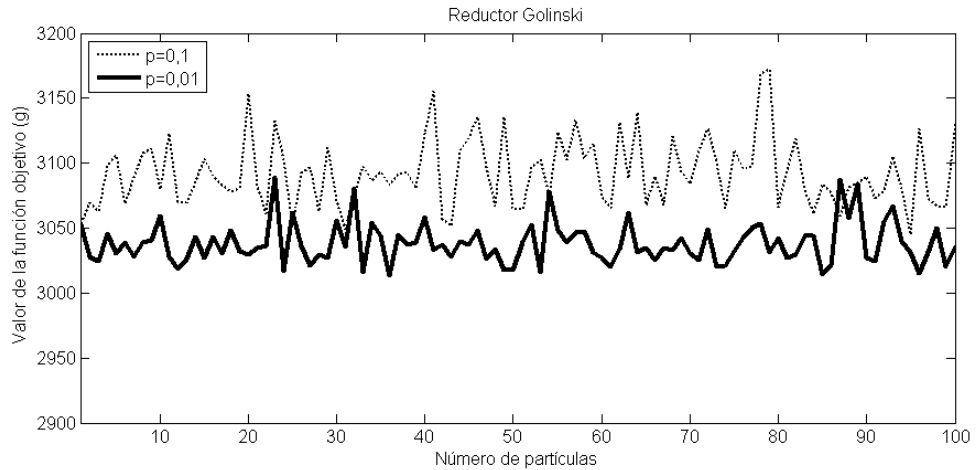
En la siguiente gráfica se observa el efecto de aumentar la precisión sobre la función objetivo y las siete variables de diseño con una población inicial de 100 partículas y un factor de unificación $u = 0,5$.

Figura 3. Efecto de la precisión, $p = 1 \times 10^{-1}$ y $p = 1 \times 10^{-2}$









Fuente: Autores

En la figura 3, se observa el comportamiento de cada variable de diseño en gráficas independientes, en las cuales se presentan dos soluciones en función de cada precisión, estos resultados numéricos se encuentran dentro del rango de soluciones posibles y oscilan dependiendo de la precisión seleccionada. En este caso como consecuencia de trabajar con una baja precisión se presenta gran dispersión de las partículas en cada gráfica respecto al valor medio de cada variable.

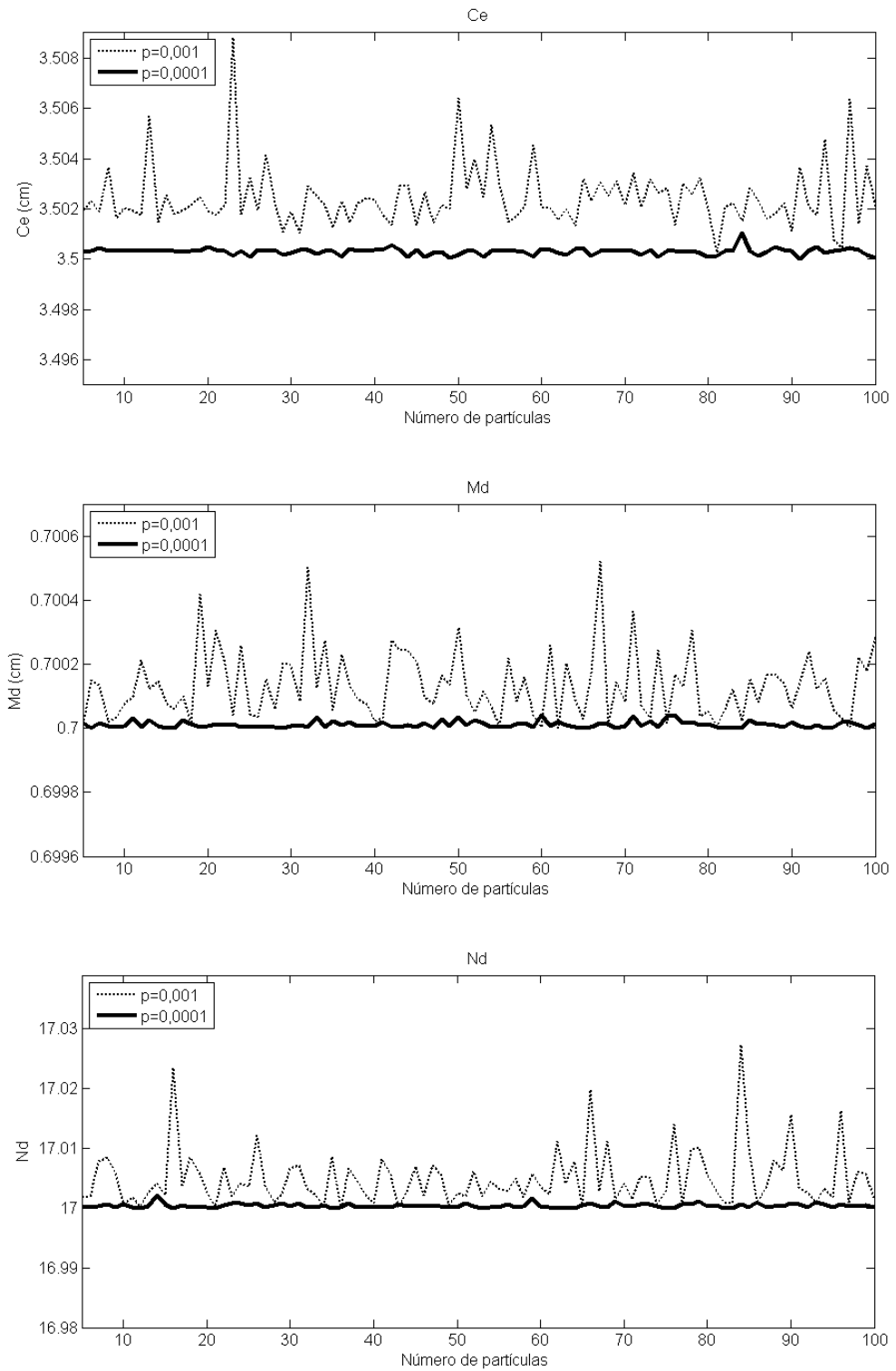
Además los resultados obtenidos muestran que los valores adquiridos por la función objetivo y las siete variables mejoran cada vez en comparación con la precisión anterior. Como consecuencia de ello, aumentar la precisión requiere un mayor costo computacional tal como se observa en los resultados numéricos. La siguiente gráfica corresponde a un aumento de la precisión hasta $p = 1 \times 10^{-4}$ y así poder observar el efecto de ésta sobre el valor numérico de la función objetivo.

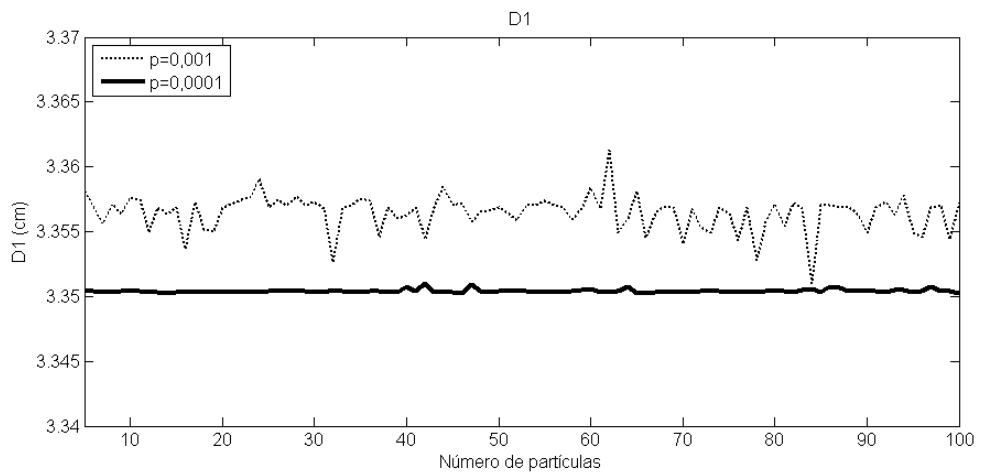
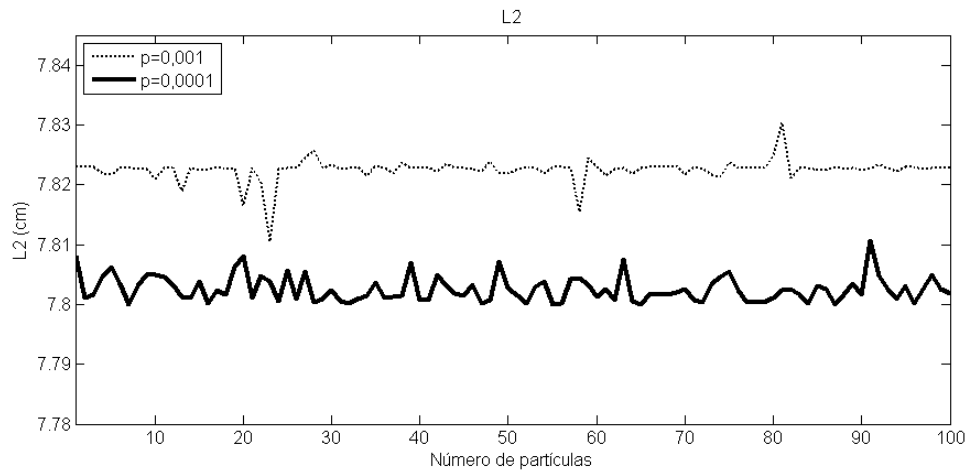
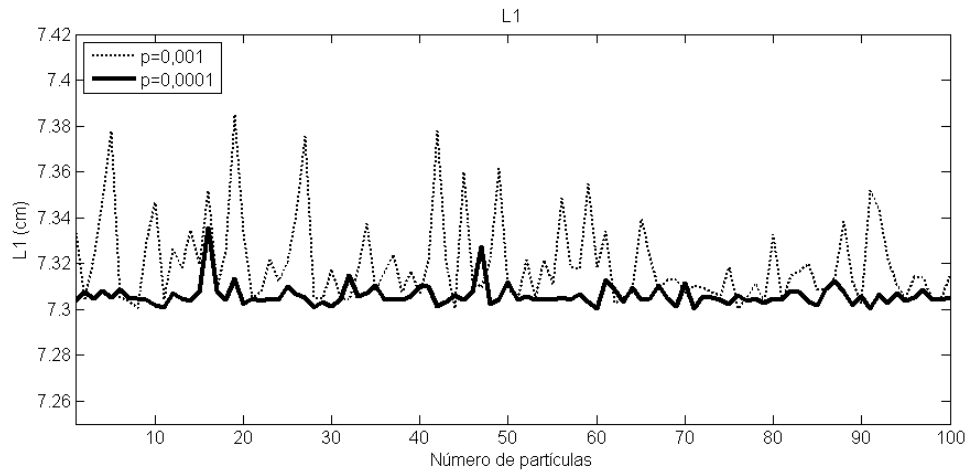
De la figura 4 se observa un cambio positivo en los valores numéricos que adquiere la función objetivo y cada una de las variables de diseño.

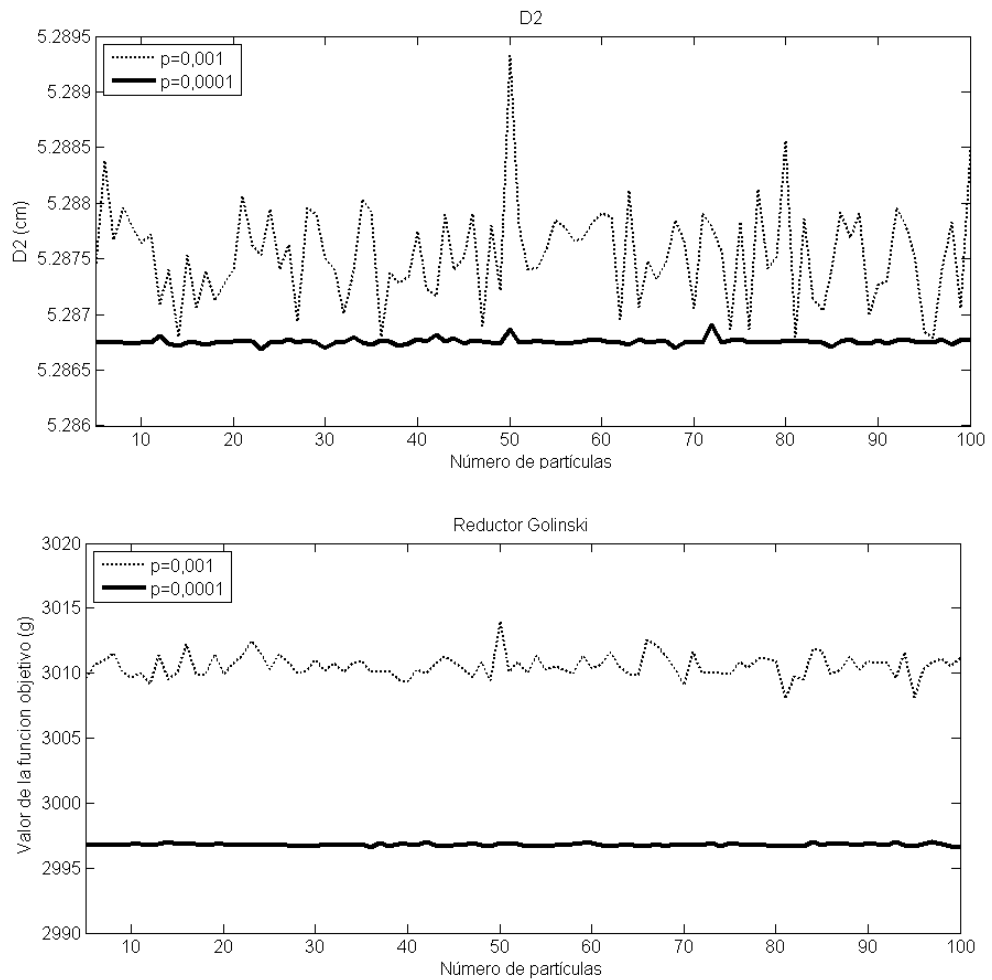
El aumento de precisión se refleja en la oscilación y baja dispersión de las partículas solución. Por tal motivo los valores con $p = 1 \times 10^{-3}$ muestran un comportamiento más aleatorio en comparación con las soluciones encontradas para $p = 1 \times 10^{-4}$ a razón del valor que toma el *error* como criterio de

convergencia, influyendo sobre el número de iteraciones y el tiempo de cómputo como se muestra en los resultados.

Figura 4. Efecto de la precisión, $p = 1 \times 10^{-3}$ y $p = 1 \times 10^{-4}$





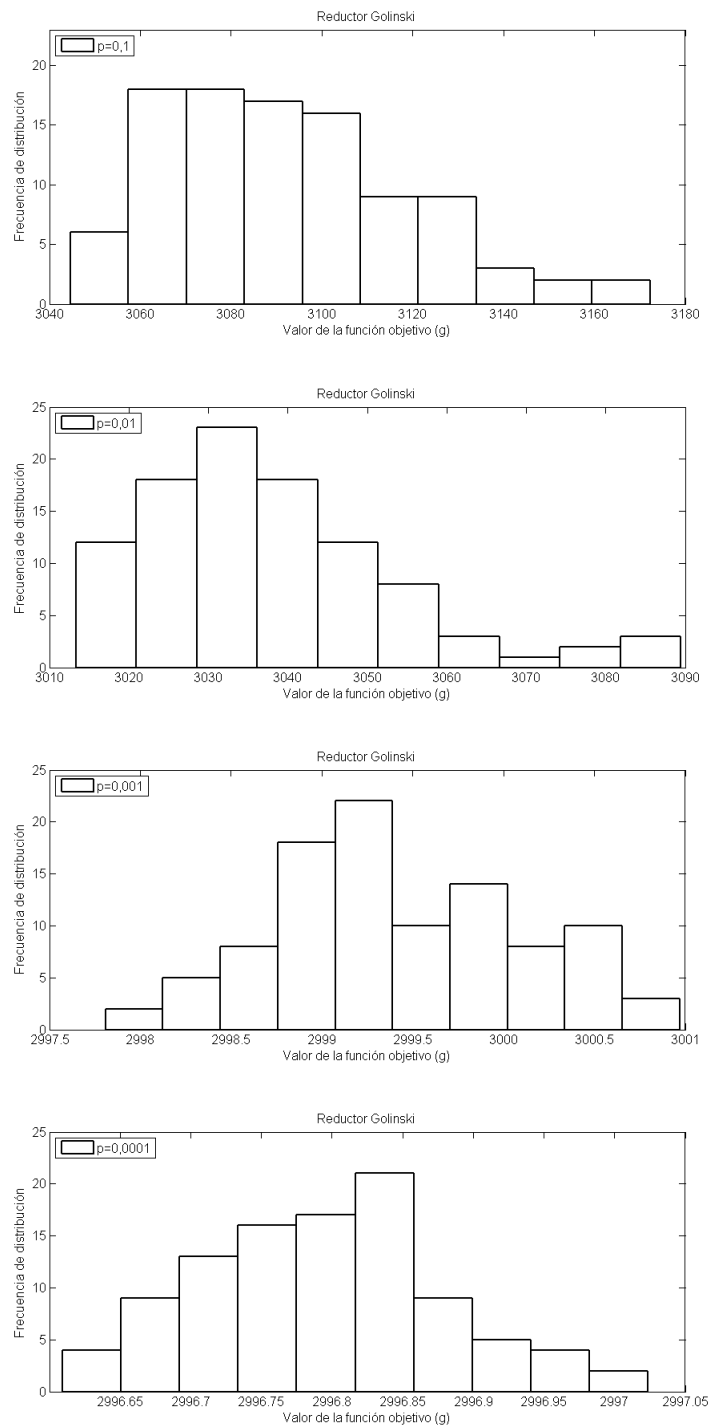


Fuente: Autores.

La figura 5, muestra la distribución de las partículas para cada precisión con $u = 0,5$ y una población inicial de 100 partículas.

De la figura 5, se observa cómo se afecta la distribución de las partículas una vez se cambia la precisión. Además, estas gráficas contrastan con los valores de la desviación estándar para una precisión correspondiente. Por esta razón a medida que se aumenta la precisión se presenta una menor dispersión de partículas respecto al valor medio de la función objetivo, lo cual es un efecto positivo a favor de la precisión.

Figura 5. Distribución de las partículas, $u = 0,5$, 100 partículas.



Fuente: Autores.

La diversidad del enjambre [16] también se afecta por cambios de precisión, en vista de que a menor precisión se muestra una mayor diversidad con gran

dispersión entre partículas y con una mayor precisión se registra una menor diversidad con poca dispersión respecto al valor medio de la función objetivo.

Los valores solución para $p = 1 \times 10^{-4}$ se toman como respuesta, debido al sentido físico que representa la precisión.

2.5. EFECTO DEL FACTOR DE UNIFICACIÓN

Al variar el factor de unificación, se registra un cambio en el tiempo de cómputo y número de iteraciones como se ve en la tabla 7.

Tabla 7. Efecto factor de unificación, $p = 1 \times 10^{-2}$, 100 partículas.

u	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$f(x)$	k	t(s)
0	3,5066	0,7000	17,0007	7,4375	7,8309	3,3517	5,2911	3004,2	1219613	3502
0,1	3,5041	0,7001	17,0317	7,3915	7,9780	3,3547	5,2915	3012,8	30815	98
0,2	3,5044	0,7001	17,0215	7,3043	7,8707	3,3541	5,2896	3006,5	13416	53
0,3	3,5014	0,7001	17,0016	7,7707	7,8079	3,3577	5,2881	3005,0	9481	36
0,4	3,5011	0,7002	17,0002	7,7800	8,0320	3,3574	5,2874	3009,2	975	9
0,5	3,5064	0,7000	17,0010	7,6513	8,0984	3,3569	5,2874	3011,0	160	6
0,6	3,5066	0,7001	17,0044	7,8911	8,1111	3,3594	5,2904	3016,9	224	7
0,7	3,5069	0,7005	17,0052	7,5498	7,8423	3,3619	5,2972	3015,1	45	6
0,8	3,5086	0,7005	17,0003	7,6255	8,0229	3,3551	5,2897	3013,1	178	10
0,9	3,5112	0,7001	17,0196	8,1059	7,8822	3,3536	5,2871	3014,5	136	6
1	3,5034	0,7000	17,0080	8,2401	7,9251	3,3612	5,2892	3014,7	180	6

Fuente: Autores.

El rango para el factor de unificación ideal está entre $u = 0,5$ y $u = 1$, tal como se aprecia en los resultados obtenidos, aunque el mejor valor para este factor es $u = 0,5$, (ofrece incluso un mejor resultado y un menor tiempo de cómputo). Otros valores como $u = 0$ y $u = 0,1$ se acercan también a la respuesta, sin embargo, requieren un mayor gasto de recurso

2.6. PSO VS UPSO

En la tabla 8 se muestra la diferencia entre PSO convencional $w = 0,5$ y UPSO $u = 0,5$. Los dos algoritmos trabajan con una precisión fija $p = 1 \times 10^{-2}$ en vista de que al trabajar con una precisión mayor para PSO aumenta significativamente el tiempo de cómputo.

Tabla 8. PSO vs UPSO, $p = 1 \times 10^{-2}$

	N	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$f(x)$	K	t (s)	Σ
A [†]	20	3,5012	0,7000	17,0036	7,8322	8,1234	3,3514	5,2879	3010,3	337245	215	2,20
B	20	3,5020	0,7003	17,0002	8,0005	7,8060	3,3534	5,2958	3011,3	427	22	4,84
A	40	3,5003	0,7000	17,0050	7,4991	7,8403	3,3538	5,2871	3001,2	829965	$\frac{109}{1}$	1,37
B	40	3,5051	0,7001	17,0010	7,5407	8,1445	3,3777	5,2888	3017,1	281	8	7,35
A	60	3,5054	0,7002	17,0119	7,5184	8,1752	3,3556	5,2872	3013,3	5986	19	4,98
B	60	3,5020	0,7002	17,0014	8,0678	7,9155	3,3596	5,2887	3011,2	204	9	8,04
A	80	3,5016	0,7000	17,0047	7,7038	7,8208	3,3571	5,2876	3004,1	13650	42	5,64
B	80	3,5013	0,7002	17,0027	7,6802	8,2760	3,3579	5,2890	3015,3	203	7	7,57
A	100	3,5004	0,7000	17,0006	7,3643	8,0103	3,3504	5,2871	3002,1	2305416	$\frac{658}{6}$	1,41
B	100	3,5274	0,7013	17,0198	8,2634	7,8925	3,3923	5,2900	3040,1	22	7	34,6

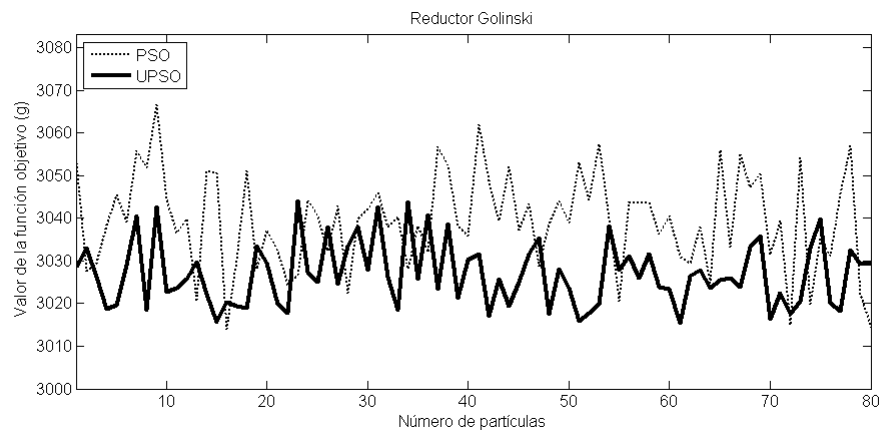
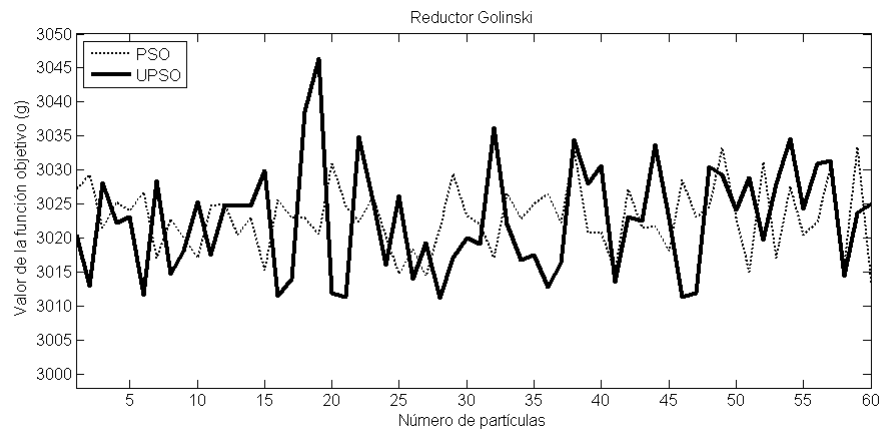
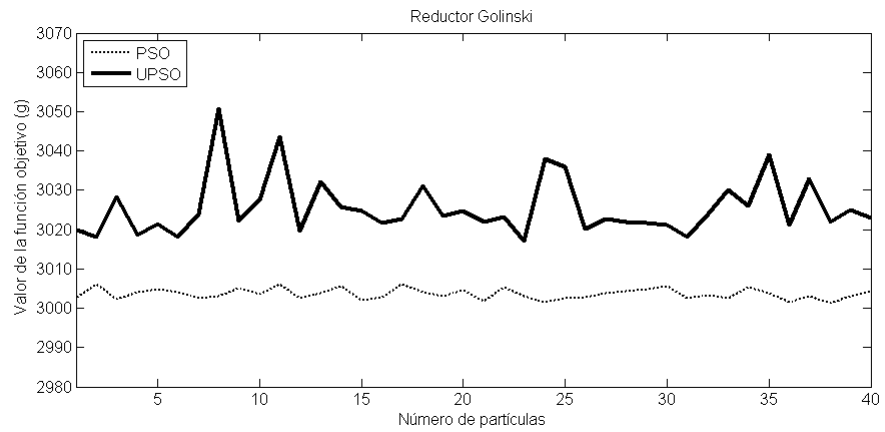
Fuente: Autores.

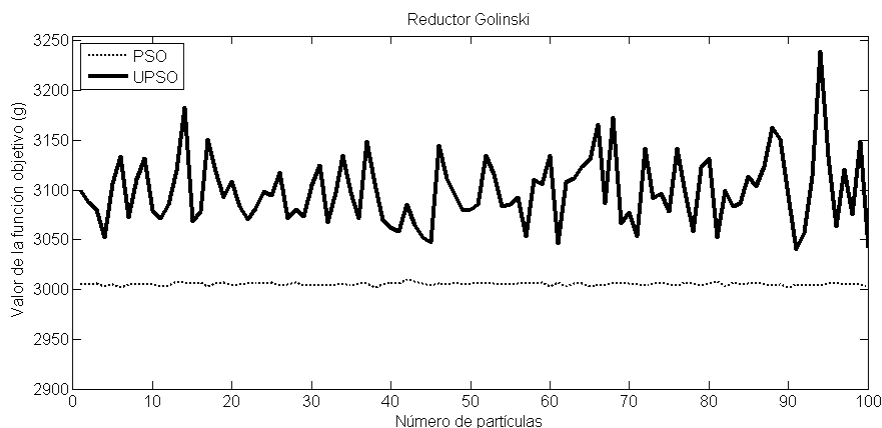
El efecto del factor de unificación sobre PSO se ve claramente en los datos obtenidos. PSO convencional requiere mayor número de iteraciones para acercarse a la solución de UPSO, reflejado en un mayor costo computacional y un incremento en el número de iteraciones.

En la figura 6 se muestra el comportamiento de las funciones objetivo para cada algoritmo; los valores son cercanos pero cabe resaltar que UPSO emplea menos tiempo en llegar al mismo valor en comparación con PSO convencional

[†] A corresponde a PSO y B corresponde a UPSO para la tabla 8.

Figura 6. PSO vs UPSO, $p = 1 \times 10^{-2}$





Fuente: Autores.

2.7. AUMENTO DE PARTÍCULAS

La tabla 9, registra el efecto de aumentar el número de partículas para un valor de precisión constante y factor de unificación $u = 0,5$.

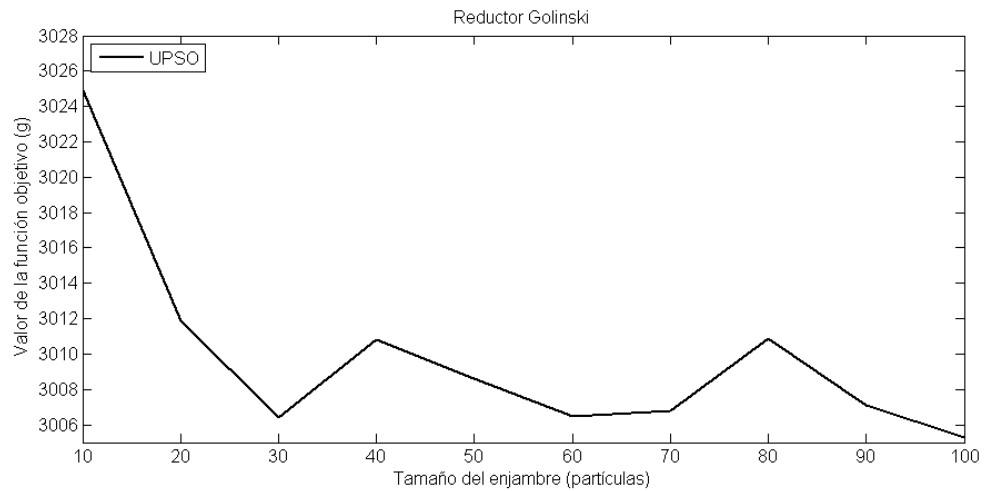
Tabla 9. Efecto del número de partículas, $p = 1 \times 10^{-3}$, $u = 0,5$.

N	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$f(x)$	k	t (s)	Σ
10	3,5068	0,7000	17,0076	8,1444	8,2666	3,3535	5,2958	3024,9	2857	9	1,2332
20	3,5023	0,7000	17,0050	7,3399	8,2504	3,3539	5,2904	3011,9	3376	8	1,8250
30	3,5020	0,7001	17,0002	7,3031	8,1842	3,3505	5,2872	3006,4	77480	90	0,5793
40	3,5019	0,7000	17,0023	8,0888	7,9764	3,3574	5,2877	3010,8	6146	15	1,0228
50	3,5003	0,7000	17,0009	7,8093	8,0788	3,3529	5,2874	3008,6	6241	18	1,1979
60	3,5046	0,7000	17,0008	7,7700	8,0956	3,3528	5,2871	3009,9	4744	13	1,1635
70	3,5001	0,7000	17,0026	7,8889	7,9617	3,3533	5,2872	3006,8	17760	48	0,5717
80	3,5007	0,7000	17,0000	7,9381	8,1335	3,3542	5,2870	3010,9	7061	22	1,1116
90	3,5013	0,7001	17,0016	7,8641	7,9539	3,3526	5,2878	3007,1	9811	70	1,0798
100	3,5007	0,7001	17,0002	7,7007	7,9483	3,3512	5,2889	3005,3	5039	23	0,9783

Fuente: Autores.

Aumentar el número de partículas tiene un efecto positivo sobre el valor numérico que adquiere la función objetivo, marcando así una tendencia hacia el óptimo para un valor de precisión constante. Sin embargo, como se observa en la figura 7, la mejora no es lineal debido a que este tipo de algoritmo presenta procesos estocásticos.

Figura 7. Función objetivo vs Tamaño del enjambre, $p = 1 \times 10^{-3}$, $u = 0,5$



Fuente: Autores.

2.8. ALEATORIEDAD PSO

La tabla 10, muestra la aleatoriedad de UPSO, para un valor constante de partículas, factor de unificación y precisión

Tabla 10. Aleatoriedad de UPSO, $p = 1 \times 10^{-3}$, $u = 0,5$.

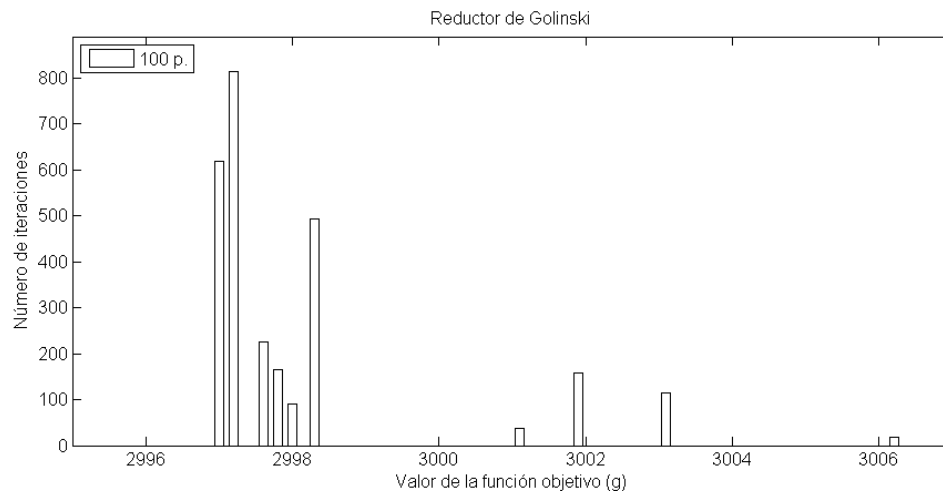
N	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$f(x)$	k	t (s)	D.E. ³
100	3,5005	0,7000	17,0001	7,3039	7,8036	3,3513	5,2868	2997,0	190054	619	55,895
100	3,5007	0,7001	17,0002	7,7788	7,9926	3,3516	5,2874	3006,2	3799	18	142,709
100	3,5005	0,7000	17,0002	7,3130	7,8027	3,3511	5,2869	2997,2	204437	815	57,659
100	3,5013	0,7001	17,0001	7,3552	7,9179	3,3512	5,2874	3001,1	6205	37	150,974
100	3,5009	0,7000	17,0005	7,3311	7,8240	3,3507	5,2870	2998,0	22121	91	79,708
100	3,5009	0,7000	17,0002	7,3152	7,8059	3,3521	5,2868	2997,6	55100	226	216,199
100	3,5016	0,7000	17,0007	7,4372	7,9124	3,3527	5,2874	3001,9	27276	157	127,185
100	3,5012	0,7000	17,0008	7,8374	7,8247	3,3519	5,2871	3003,1	21486	115	158,771
100	3,5024	0,7000	17,0004	7,3172	7,8033	3,3510	5,2875	2998,3	99108	493	214,879
100	3,5007	0,7000	17,0002	7,3202	7,8192	3,3506	5,2872	2997,8	36172	166	98,127

Fuente: Autores.

³ D.E. corresponde a la distancia euclidiana para la tabla 10.

Se observa que los valores solución están relativamente cerca unos de otros para un mismo tamaño del enjambre. Así mismo los valores de la distancia euclidiana, (definida como la menor distancia entre dos puntos del espacio respecto a la media de la función objetivo), oscilan considerablemente entre pruebas debido a la precisión seleccionada. Dichas variaciones se deben a los valores que toman las posiciones y velocidades inicialmente. Por esta razón, el tiempo de cómputo e iteraciones dependen de la posición de las partículas con respecto al óptimo estén ubicadas las partículas para iniciar su proceso de búsqueda. La distancia euclidiana se toma como criterio de convergencia para identificar la proximidad de las partículas en relación con la media, incluso se usa en otros algoritmos de optimización como criterio de parada.

Figura 8. Número de iteraciones vs función objetivo, $p = 1 \times 10^{-3}$, $u = 0,5$, 100 partículas.



Fuente: Autores.

2.9. FACTOR DE RESTRICCIÓN

La tabla 11 muestra el efecto que produce variar el factor de restricción.

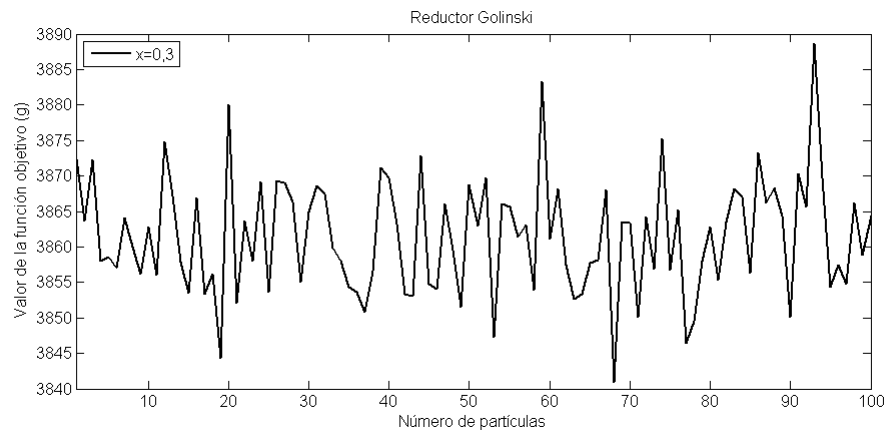
Tabla 11. Variación del factor de restricción, $u = 0,5$, 100 partículas, $p = 1 \times 10^{-2}$

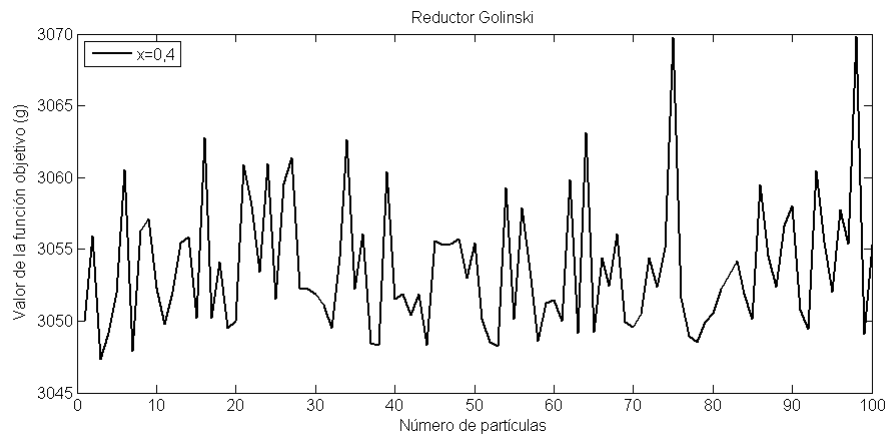
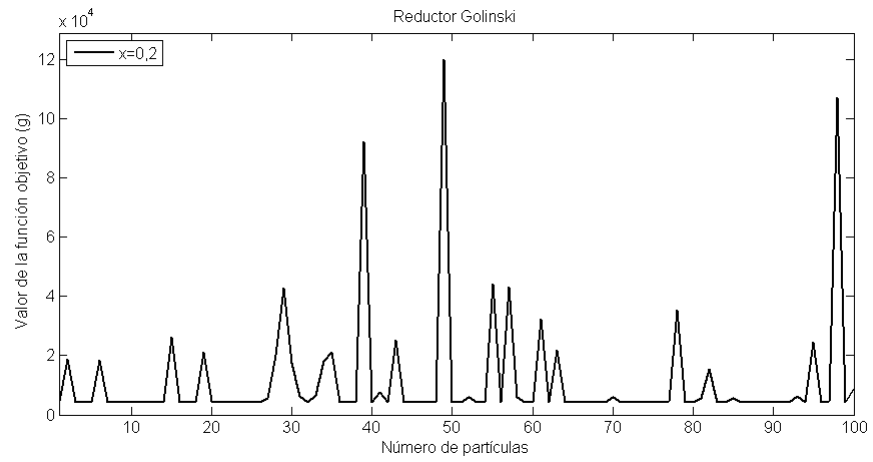
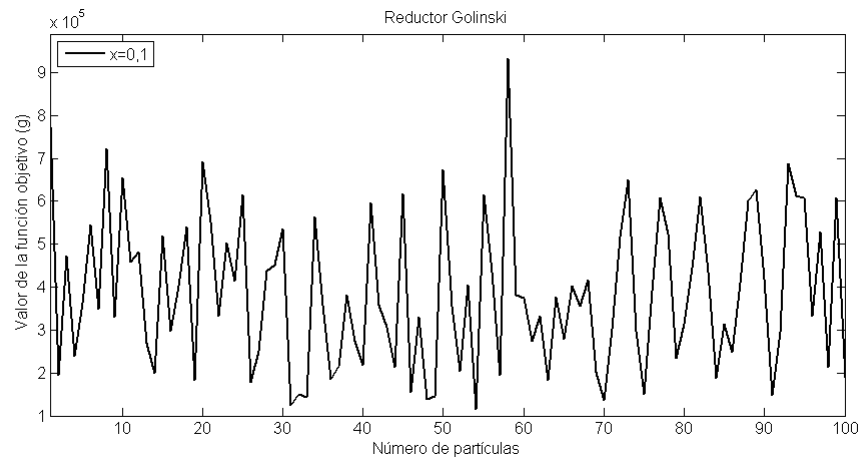
χ	$f(x)$	k	t (s)
0,1	114984,0438	24	5
0,2	4353,3735	24	7
0,3	3840,8745	31	6
0,4	3047,3329	37	10
0,5	3010,6260	120	12
0,6	3011,0318	160	6
0,7	3017,1088	481	8
0,8	3013,5206	2938	16
0,9	3009,3868	4452	20

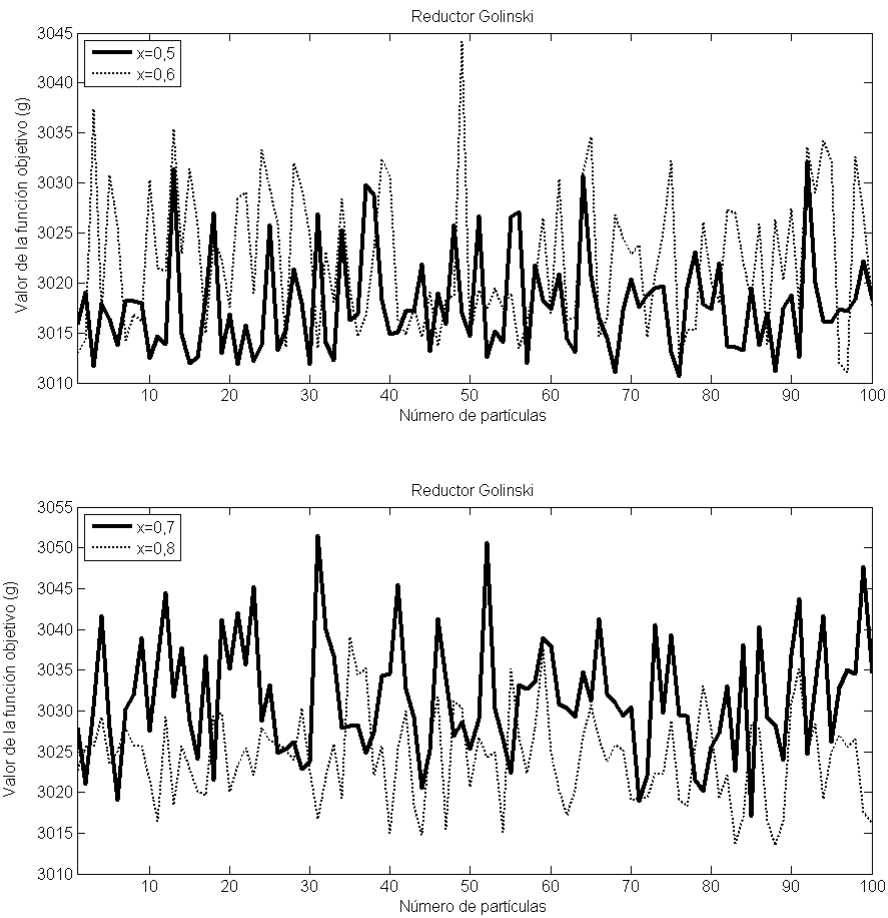
Fuente: Autores.

En la siguiente grafica se observa el comportamiento de las partículas sujeto al cambio del factor de restricción.

Figura 9. Efecto factor de restricción, $u = 0,5$, $p = 1 \times 10^{-2}$.







Fuente: Autores.

El cambio del factor de restricción influye en la dispersión de datos respecto al valor óptimo; entre menor es el factor de restricción más dispersas están las partículas. No obstante, para valores entre $\chi = 0,5$ y $\chi = 0,6$ se presenta un mejor comportamiento aleatorio entregando mejores resultados. El diseño también se resolvió mediante un software comercial para tener un punto de comparación con otra técnica de optimización y sus resultados se muestran en la tabla 9. Igualmente se registran en esta tabla los resultados obtenidos por el método de optimización simple con enjambre de partículas con restricciones (*SiC PSO*) [2].

Tabla 12. UPSO vs software comercial vs SiC PSO.

Método	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$f(x)$
UPSO	3,5001	0,7000	17,0001	7,3051	7,8017	3,3503	5,2868	2996,6093
Software comercial	3,5000	0,7000	17,0000	7,3000	7,8000	3,3502	5,2867	2996,3485
SiC PSO	3,5000	0,7000	17,0000	7,3000	7,8000	3,3502	5,2866	2996,3481

Fuente: Autores.

Estos valores confirman la certeza de los resultados obtenidos en este trabajo con una precisión de $p = 1 \times 10^{-4}$ utilizando UPSO como alternativa de optimización. Las dimensiones óptimas del reductor de Golinski encontradas en este trabajo son:

Tabla 13. Dimensiones óptimas.

C_e	M_d	N_d	L_1	L_2	D_1	D_2	$f(x)$
3,5001	0,7000	17,0001	7,3051	7,8017	3,3503	5,2868	2996,6093

Fuente: Autores.

3. OBSERVACIONES Y CONCLUSIONES

Una vez analizados los resultados de diferentes ejercicios propuestos, a los cuales se les aplicó UPSO, se puede ver que conduce a una solución óptima exacta, resolviendo así el problema planteado. Aumentar la precisión por encima de $p = 1 \times 10^{-4}$ conllevaría a una respuesta más exacta, sin embargo, en el caso del reductor de Golinski no es ganancia debido al sentido físico que representa la precisión.

Basados en los resultados de las iteraciones; el valor ideal para el factor de unificación es $u = 0,5$ a razón de sus propiedades equitativas de búsqueda así como se demuestra en las tablas obtenidas, ofreciendo un buen rendimiento en comparación con los otros factores de unificación seleccionados.

Establecer el factor de unificación entre $u = 0$ y $u = 0,2$ produce un efecto negativo sobre el algoritmo, por tal motivo se seleccionan valores entre $u = 0,5$ y $u = 1$ los cuales ofrecen mejores propiedades de búsqueda.

Aumentar el número de partículas iniciales influye en los resultados obtenidos de manera positiva, sin embargo, es más efectivo aumentar la precisión en vista de que el algoritmo debe establecer que tan cerca esta una partícula de la otra, lo que implica un aumento en el tiempo de ejecución e iteraciones para encontrar el mejor resultado.

La solución numérica del modelo del reductor de Golinski y los ejercicios propuestos se lograron sin necesidad de utilizar herramientas matemáticas tales como Jacobianos, Hessianos u otros para calcular el óptimo de cada función objetivo, sólo se requiere una distribución de partículas al azar dentro del rango de

soluciones posibles y por medio de su modelo evolutivo encuentra la mejor solución.

El óptimo del reductor de Golinski calculado por UPSO con $p = 1 \times 10^{-4}$ contrasta con el resultado obtenido por otros métodos de optimización, demostrando la efectividad de este tipo de algoritmo.

Finalmente, luego de realizar todas las pruebas necesarias variando los diferentes factores que afectan el desempeño del algoritmo elaborado para llegar una solución óptima, se determina que los valores efectivos son: $\chi = 0,6$; $c_1 = c_2 = 2,8333$; $u = 0,5$.

BIBLIOGRAFÍA

- [1] JAN GOLINSKI, Optimal Synthesis Problems Solved by Means of Nonlinear Programming and Random Methods, Journal of Mechanisms Vol 5, pp. 287-309,1970.
- [2] LETICIA C. CAGNINA, SUSANA C. ESQUIVEL AND CARLOS A. COELLO COELLO, "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer", Informática, Vol. 32, 2008, pp. 323–324.
- [3] G.AGRAWAL,S. PARASHAR,K. W. ENGLISH, AND C. L. BLOEBAUM. 'Web-based Visualization Framework for Decision making in Multidisciplinary Design Optimization', Department of Mechanical and Aerospace Engineering University at Buffalo, The State University of New York Buffalo, 2004, pp. 21.
- [4] BUDYNAS, NISBETT, Mechanical Engineering, Shigley's Mechanical Engineering Design, Eighth Edition, McGraw-Hill,2006, pp.348-383.
- [5] KARL-HEINRICH GROTE, Erik K. Antonsson, "Springer Handbook of Mechanical Engineering", Springer, 2008, pp. 334-398.
- [6] TORO OCAMPO ELIANA M, RESTREPO G. YOV STEVEN , GRANADA E. MAURICIO, Adaptación de la técnica de Particle Swarm al Problema de secuenciamiento de tareas Scientia Et Technica, Vol. XII, Núm. 32, 2006, pp. 307-312.
- [7] PARSOPOULOS, K.E., VRAHATIS, M.N., Unified Particle Swarm Optimization in Dynamic Environments, Lecture Notes in Computer Science (LNCS), Vol. 3449, Springer, 2005, pp. 590-599.

- [8] PARSOPOULOS, K.E., VRAHATIS, M.N., Unified Particle Swarm Optimization for Solving Constrained Engineering Optimization Problems, Lecture Notes in Computer Science (LNCS), Vol. 3612, Springer, 2005, pp. 582-591.
- [9] PARSOPOULOS, K.E., VRAHATIS, M.N., Parameter Selection and Adaptation in Unified Particle Swarm Optimization, Mathematical and Computer Modelling, 46 (1-2), Elsevier, 2007, pp. 198-213.
- [10] PARSOPOULOS, K.E., VRAHATIS, M.N., UPSO: A Unified Particle Swarm Optimization Scheme, Lecture Series on Computer and Computational Sciences, Vol. 1, Proceedings of the International Conference of "Computational Methods in Sciences and Engineering" (ICCMSE 2004), VSP International Science Publishers, Zeist, The Netherlands, 2004, pp. 868-873.
- [11] A KURPATI, S AZARM AND J WU, Constraint handling improvements for multiobjective genetic algorithms, Struct. Multidisc Optim. 23, Springer-Verlag, 2002, pp. 204–213.
- [12] S. RAO, "Classical optimization techniques," en *Engineering Optimization: Theory and Practice*, J. Wiley, Ed. 4a ed. New Jersey: J. Wiley & Sons, 2009, pp. 446-483.
- [13] I-LING LING, Particle Swarm Optimization for solving constraint satisfaction problems, B. Sc Simon Fraser University 2005, pp. 36-51.

[14] LETICIA C. CAGNINA AND SUSANA C. ESQUIVE, Solving Engineering Optimization Problems with the Simple Constrained, Universidad Nacional de San Luis, San Luis, Argentina, *Informática* 32, 2008, pp. 319–326.

[15] A. DE FREITAS Y E. PINTO, “Optimization of nonlinear constrained particle swarm,” *Ilumina.*, vol. 12, no. 1, 2006, pp 30-36.

[16] KONSTANTINOS E, PARSOPOULOS & MICHAEL N. VRAHATIS, “Particle swarm optimization and intelligence; advances and applications”, Information science reference, Hershey, New York, IGI global, 2010, pp.142-258.

[17] JINN-MOON YANG, YING-PING CHEN, JORNG-TZONG HORNG, AND CHENG-YAN KAO, Applying Family Competition to Evolution Strategies for Constrained Optimization, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2002, pp 207-208.

[18] GAO F, YIBO QI, QIANG YIN, JIAQING XIAO, A- Novel Particle Swarm Optimization With Special Boundaries Restriction Strategy, Department of Mathematics, School of Science Wuhan University of Technology, Wuhan, 430070, China, 2010, pp. 1-4.

[19] RUBEN E. PEREZ AND KAMRAN BEHDINAN, Particle Swarm Optimization in Structural Design, University of Toronto, Institute for Aerospace Studies, Ryerson University, Department of Aerospace Engineering Canada, 2007, pp. 376-377.