

Diseño óptimo de un disipador de calor con entropía mínima mediante los algoritmos de optimización GSA y DSO

Cristián Jesús Pérez Herrera

Trabajo de Grado para Optar el título de Ingeniero Electrónico

Director

Carlos Rodrigo Correa Cely

Ingeniero Químico.

Estancia postdoctoral/ Universidad of Sttugarth

Codirector

Iván Mauricio Amaya Contreras

Ingeniero Mecatrónico

Investigador postdoctoral en Tecnológico de Monterrey

Universidad Industrial de Santander

Facultad de Ingenierías Físico Mecánicas

Escuela de Ingenierías Eléctrica, Electrónica, y de Telecomunicaciones

Bucaramanga

2018

Dedicado a:

A mis padres

, por ser los portadores de la mejor red telepática de comunicación que supera todo obstáculos y el infinito

A mis tíos Jorge y Elizabeth,

porque juntos son como un gran motor de alta potencia

A mis dos hermosas hermanas, (en neutro)

A mi madrina, y mi tía Ana, por doblar sus rodillas para encender la red

A mi tía Doris Doria, por portar la chispa que enciende todos los motores

A Boltzman, porque lamento su suicidio y la incompreensión de su tiempo,

Y, sobre todo,

Dedicado especialmente a Jesucristo, por estar siempre a mi lado, incluso en esos

momentos que creía caminar sólo.

Debo especiales agradecimientos a:

- *A mi director de proyecto Carlos Rodrigo Correa y codirector de proyecto Iván Mauricio Amaya, por atenderme con toda su disposición y ayudar con la información y conocimiento sobre este tema las muchas veces que fue posible.*
- *A la biblioteca de la Universidad industrial de Santander por su significativo aporte*

Autor.

Contenido

	Pág.
Introducción	19
1. Objetivos	20
1.1 Objetivo General.....	20
1.2 Objetivos Específicos.....	20
2. Fundamentación teórica	21
2.1 Caracterización termodinámica de un disipador de calor	21
2.2 La generación de entropía: criterio de optimización.....	23
2.3 Resistencia térmica en la generación de entropía	25
2.4 Desarrollo del modelo.....	25
2.5 Caracterización de la función objetivo	28
3. Algoritmos de optimización numérica meta- heurísticos: GSA y DSO	32
3.1 Algoritmo de búsqueda gravitacional GSA	33
3.1.1 criterio de parada.....	36
3.1.2 Número de agentes.....	36
3.1.3 Parámetro Kmejores.....	36
3.1.4 diagrama de flujo del algoritmo	36
3.2 Algoritmo de optimización drone squadron (DSO).....	37
3.2.1 Centro de comandos.....	39

3.2.2 Firmware	40
3.2.3 Drones en movimiento.....	42
3.2.4 Actualización del firmware.....	43
3.2.5 Recombinación	43
3.2.6 Pseudocódigo del algoritmo.....	44
3.2.7 Ejemplo del mecanismo de una iteración en DSO.....	46
4. Implementación de los algoritmos con funciones de prueba	49
4.1 Función de prueba Drop-water	49
4.2 Función de prueba Schwefel.....	50
4.3 Función de prueba Rastringin	51
4.4 Función de prueba Rosenbrock.....	52
4.5 Resultados de implementar los algoritmos con las funciones de prueba.....	53
4.5.1 Variación del número de agentes en GSA	53
4.5.2 Variación del número de iteraciones con criterio de parada en GSA	55
4.5.3 Variación con los Kmejores en GSA	56
4.5.4 Variación del número de equipos y drones en DSO.....	56
4.5.5 Variación del número de iteraciones para actualizar el firmware.....	59
5. Implementación de GSA y DSO al modelo del disipador de calor	64
5.1 Modelo matemático restrictivo del disipador de calor.....	64
5.2 Parámetros del modelo.....	65
5.3 Parámetros asignados al algoritmo GSA durante la implementación.....	66
5.4 Parámetros asignados al algoritmo DSO durante la implementación.....	67
5.5 Funciones de implementación.....	68

5.5.1 Función entropía	68
5.5.2 Función límite de los espacios de búsqueda	68
5.5.3 Función rango	68
6. Generación de entropía al variar la geometría del disipador	69
6.1 Resultados con DSO de la entropía al variar los límites de la geometría	72
7. Resultados de optimizar el canal.....	75
7.1 Resultados obtenidos con GSA.....	75
7.2 Resultados obtenidos con DSO.....	76
7.3 Resultados de referencia obtenidos con PSO.....	77
8. Análisis comparativo	78
9. Conclusiones	79
Referencias Bibliográficas	81
Apéndices.....	83

Lista de Tablas

	Pág.
Tabla 1 Ecuaciones que definen la entropía por la transferencia de calor	29
Tabla 2 Ecuaciones que definen la entropía debido a la fricción del fluido	30
Tabla 3 Equipo utilizado.....	49
Tabla 4. Resultados de aplicar 50 iteraciones.....	53
Tabla 5 Resultado de aplicar 100 iteraciones.	54
Tabla 6. Resultados de aplicar 1000 iteraciones.....	55
Tabla 7. Resultados de aplicar criterio de parada	55
Tabla 8. Resultados al variar la selección de Kmejores.....	56
Tabla 9. Resultados de tener 5 equipos y variar el número de drones por equipo.....	57
Tabla 10. Resultados de tener 10 equipos y variar el número de drones por equipo.....	57
Tabla 11. Resultados de tener 50 equipos y variar el número de drones por equipo.....	58
Tabla 12. Resultados de tener 100 equipos y variar el número de drones por equipo.....	58
Tabla 13. Resultados obtenidos sobre la función Rastringin al variar el número de iteraciones .	59
Tabla 14. Parámetros de diseño constantes asumidos para la simulación	65
Tabla 15 Rango de valores de las variables	68
Tabla 16 GE cuando la diferencia de los anchos es mínima.....	70
Tabla 17. GE cuando la diferencia de los anchos es máxima.....	70
Tabla 18 GE cuando el ancho del canal es menor que el de la aleta	71

Tabla 19. Variación de los límites de búsqueda para $K_n=0.001$	72
Tabla 20. Variación de los límites de búsqueda para $K_n=0.1$	73
Tabla 21. Variación de los límites de búsqueda para $K_n=1$	73
Tabla 22. Variación de los límites de búsqueda para $K_n=10$	74
Tabla 23. Resultados con GSA	75
Tabla 24. Resultados con DSO	76
Tabla 25. Resultados con PSO	77

Lista de Figuras

	Pág.
Figura 1. Geometría del Disipador de Calor.....	27
Figura 2. Diagrama de Flujo del Algoritmo GSA.....	37
Figura 3. Función Drop-wave.....	50
Figura 4. Función Schwefel.....	51
Figura 5. Función Rastringin.....	52
Figura 6. Función Rosenbrock.....	53
Figura 7. Al realizarse actualización del firmware cada 5 iteraciones.....	60
Figura 8. Al realizarse actualización del firmware cada 10 iteraciones.....	61
Figura 9. Al realizarse actualización del firmware cada 50 iteraciones.....	62
Figura 10. Al realizarse actualización del firmware cada 100 iteraciones.....	63
Figura 11. Al realizarse actualización del firmware cada 500 iteraciones.....	64

Lista de Apéndices

	Pág.
Apéndice A. Resultados de variar el tamaño de los agentes en GSA.....	83
Apéndice B. Código en matlab de la Función de prueba Drop-water (2D).....	90
Apéndice C. Código en matlab de la Función de Prueba Schwefel.....	92
Apéndice D. Código en matlab de la Función de Prueba Rastringin.....	93
Apéndice E. Código en matlab de la función de prueba Rosenbrock.....	93
Apéndice F. Código en matlab de la función entropía.....	94
Apéndice G. Resultados numéricos de variar el número de drones en DSO.....	96
Apéndice H. Manual de usuario para implementar la función entropía en DSO.....	110

Nomenclatura asociada al disipador

A	Área superficie total [m ²]
Ad	Adimensional
D_h	Diámetro hidráulico [m ²]
f	Factor de fricción
F_d	Fuerza de arrastre
G	Tasa de flujo de volumen [m ³ /s]
H_c	Altura del canal [m]
h_{av}	Coefficiente promedio de transferencia de calor [W/m ² ·K]
k_n	Número de Knudsen
k	Conductividad térmica del solido
k_{ce}	Suma perdidas de entrada y salida
k_{eq}	Relación de conductividades térmicas entre sólido y fluido
k_f	Conductividad térmica del fluido [W/m·K]
L	Longitud del canal [m]
\dot{m}	Tasa de flujo másico total [Kg/s]
η	Eficiencia
N_m	Número total de microcanales
Nu_{Dh}	Número de nusselt basado en el diámetro hidráulico
Pe_{Dh}	Número de Peclet basado en el diámetro hidráulico

Pr	Número de Prandtl
Q	Tasa de transferencia de calor [W]
q	Flujo de calor
R_{th}	Resistencia Térmica

Nomenclatura asociada al algoritmo DSO

μ	Media
σ	Desviación estándar
C	Constante definida por el usuario
CBC	Arreglo 2D que contiene las mejores coordenadas
CBOFV	Matriz con mejores coordenadas de evaluar la F.O
D	Numero de dimensiones de la función objetivo
G	Distribución Gaussiana
GBC	Mejores coordenadas globales encontradas en determinado tiempo
GBOFV	Mejores valores globales de la función objetivo
LB	Matriz con los limites inferiores de la función objetivo
MaxIt	Máximo número de iteraciones
Maxest	Máximo número de iteraciones sin cambio
MVNS	Muestreador normal multivariable
N	Numero de drones en cada equipo
P	Formula de perturbación
Pacc	Probabilidad de aceptar una solución peor que la de CBC

Std-dev	Estándar desviación
t	Número de equipos
TC	Arreglo 2D que contiene las coordenadas prueba de la solución
CalidadEq	Matriz que indica la calidad de los equipos
TmC	Un arreglo 2D que contiene las coordenadas de los equipos
TmOFV	Matriz con los valores de la función objetivo de los equipos
U	La distribución uniforme
UB	Matriz con los límites superiores de la función objetivo

Subíndices

a	Ambiente
av	Promedio
b	Base de la superficie
c	Canal
ce	Expansión y contracción
f	Fluido
fin	Aleta
ff	Fricción del fluido
h	Hidráulico
hs	Disipador de calor
ht	Transferencia de calor
H	Alta
M	Microcanal

L	Baja
th	Térmica
s	Deslizamiento

Resumen

Título: Diseño óptimo de un disipador de calor con entropía mínima mediante los algoritmos óptimos GSA y DSO*.

Autor: Cristian Jesús Pérez Herrera**

Palabras Claves: Algoritmos auto adaptativo, optimización global numérica, micro- canales, coevolución, programación genética y entropía.

Descripción:

Utilizando como criterio la mínima generación de entropía y apoyándose en los algoritmos de optimización metaheurístico GSA (Algoritmo de Búsqueda Gravitacional) y el novedoso hiperheurístico DSO (Optimización por Escuadrón de drones), se obtienen nuevos resultados en el proceso de optimizar los valores geométricos de un disipador de calor tipo micro canal rectangular, el cual se utiliza para refrigerar los circuitos integrados. Estos resultados se hacen posibles al utilizar como modelo matemático específico para el disipador de calor, una función que fue propuesta con anterioridad por varios autores dedicados al tema, y que tiene en cuenta la entropía generada por el dispositivo durante el proceso de transferencia de calor. Siendo esto entonces el criterio a minimizar, se debe escoger adecuadamente unos valores de diseño que se determinan bajo condiciones específicas de funcionamiento; estos resultados se logran gracias a la implementación de esta función objetivo con los algoritmos de optimización modernos, ya que esta función objetivo corresponde a un análisis no lineal y restringido de las variables en estudio, teniendo en cuenta algunos ajustes que se hicieron al modelo inicial para simplicidad del análisis y el proceso de simulación.

Se presenta los resultados novedosos que se consiguieron al utilizar el algoritmo DSO y su respectiva comparación con los resultados obtenidos por GSA y también por otros métodos desarrollados anteriormente a este informe, teniendo en cuenta los aspectos más relevantes al momento de optimizar: tiempo de cómputo, convergencia y resultados satisfactorios. Concluyendo finalmente si las evidencias permiten clasificar al novedoso algoritmos en mejor posición en cuanto a la eficiencia y simplicidad al momento de implementar.

* Trabajo de grado

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

Abstract

Title: Optimal design of a heat sink with minimum entropy using the GSA and DSO optimum algorithm*.

Author: Cristian Jesús Pérez Herrera**

Keywords: Optimization, Self-Adaptive, Metaheuristics,

Description:

Using as a criterion the minimum generation of entropy, and relying on the algorithms of metaheuristic optimization GSA (Algorithm of Gravitational Search) and the novel hyperheuristic DSO (Optimization by Squadron of drones), new results are obtained in the process of optimizing the geometric values of a rectangular type micro channel heat sink, which is used to cool the integrated circuits. These results are possible when using as a specific mathematical model for the heat sink, a function that was previously proposed by several authors dedicated to the subject, and that takes into account the entropy generated by the device during the heat transfer process. Is the entropy then the criterion to be minimized, which implies choosing properly designed values that are determined under specific operating conditions. These results are achieved thanks to the implementation of this objective function with modern optimization algorithms, considering this objective function corresponds to a non-linear and restricted analysis of the variables under study, taking into account some adjustments that were made to the initial model for simplicity of the analysis and the simulation process.

It presents the novel results that were obtained by using the DSO algorithm and its respective comparison with the results obtained by GSA and also by other methods developed earlier in this report, taking into account the most relevant aspects when optimizing: computation time, convergence and satisfactory results. Finally concluding if the evidence allows to classify the novel algorithms in a better position in terms of efficiency and simplicity at the time of implementation.

* Bachelor Thesis

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

Introducción

La continua evolución de los dispositivos semiconductores se ha enfocado en un diseño que busca reducir el tamaño y aumentar la capacidad de procesamiento. Esto conlleva una producción microelectrónica de dispositivos de alta densidad de integración que operan a mayor velocidad, pero con un aumento en la generación de calor en los dispositivos semiconductores debido al efecto calorífico de Joule. Surge entonces la necesidad de buscar mecanismos que aseguren la protección y buen funcionamiento del dispositivo mediante arreglos exteriores para disipar calor. Autores como Culham y Khan han demostrado la posibilidad de obtener una disipación de calor en forma eficiente cuando se adhieren elementos disipadores sobre el sustrato del semiconductor. Para lograr una eficiente refrigeración los disipadores hacen uso de microcanales que están definidos por parámetros y variables físicas. Y es a partir de estas definiciones que se desarrolla un modelo matemático para relacionar las variables y describir el comportamiento térmico del disipador de calor. Este modelo se desarrolla a partir de la entropía generada por el disipador en el proceso de refrigeración y el criterio de minimizar la entropía es de gran utilidad en esta investigación para generar únicamente en este caso los valores óptimos del diseño geométrico del disipador. Este concepto de mínima generación de entropía fue propuesto inicialmente por autores como Bejan (1992), pero luego distintos autores han desarrollado trabajos posteriores.

Para encontrar la solución a este modelo matemático que representa la generación de entropía como la función objetivo de esta optimización, se implementan dos algoritmos de optimización

metaheurística, Gravitational Search Algorithm (GSA) y Drone Squadron Optimization (DSO), demostrando ser buenas opciones para un análisis numérico al ser comparados con los métodos tradicionales como los multiplicadores de lagrange que resuelven problemas restrictivos como el tratado en esta investigación. Luego de la implementación con estos dos nuevos algoritmos, se comparan y analizan los resultados con respecto a otros métodos utilizados anteriormente por Hinojosa y Espinosa con PSO (Particle Swarm Optimization (2010)).

1. Objetivos

1.1 Objetivo General

Soportar una de las líneas de investigación en el grupo CEMOS hacia la optimización relacionada con los algoritmos metaheurísticos GSA(Gravitational Search) y DSO (Drone Squadron optimization) mediante Matlab y resolver la función objetivo planteada para un disipador de calor mediante el diseño de mínima entropía.

1.2 Objetivos Específicos

Implementar los algoritmos de optimización GSA y DSO.

Optimizar el diseño de un disipador de calor mediante el criterio de mínima generación entropía al resolverlo mediante los algoritmos GSA y DSO.

Contrastar los resultados obtenidos frente a los encontrados con el método híbrido PSO-multiplicadores de Lagrange en trabajos previos y reportados en la literatura

2. Fundamentación teórica

En este capítulo se exponen algunos de los conceptos teóricos más importante a tenerse en cuenta durante el desarrollo de la implementación de algoritmos metaheurísticos sobre una función objetivo de carácter termodinámico obtenida a partir del modelamiento de un disipador de calor para uso en microelectrónica. Primero, se describen las características termodinámicas de un disipador de calor general, y consecuentemente una descripción del modelo matemático que se utiliza para definir la generación de entropía, lo cual se utiliza como criterio de optimización para lograr un diseño adecuado del disipador. También se muestra la fundamentación teórica y el funcionamiento de los algoritmos GSA y DSO, con sus características propias de implementación para encontrar la solución a este diseño específico.

2.1 Caracterización termodinámica de un disipador de calor

El disipador de calor es un dispositivo de interés práctico que opera entre un medio de alta temperatura T_H y otro de baja temperatura T_L . El aire es considerado un depósito de energía térmica y un sumidero desde el análisis termodinámico, y se tiene en cuenta como un fluido fundamentalmente importante en la operación de remover el calor. El medio de alta temperatura usualmente es una superficie sólida y el aire constituye el medio de baja temperatura que va a disipar la energía térmica de la superficie sólida.

La energía en tránsito debido a la diferencia de la temperatura y la transferencia de calor en el disipador, se manifiesta en el modo de convección libre, debido a que existe un fluido en movimiento y una superficie sólida, ambas con distintas medidas de temperatura. La convección libre define que el flujo es inducido por fuerzas de empuje que surgen a partir de una diferencia de densidades en el fluido, que se produce por variaciones de la temperatura en el fluido. Es decir, el aire que hace contacto con la superficie sólida experimenta un aumento de temperatura y en consecuencia una reducción en su densidad.

La ley de enfriamiento de Newton establece que el flujo de calor por convección libre será proporcional a la diferencia entre la temperatura de la superficie sólida y la temperatura del fluido en cuestión; $T_s T_\infty$ también es proporcional al denominado coeficiente de transferencia de calor por convección libre h . Éste coeficiente a su vez depende de las variaciones de la velocidad del fluido en la zona de contacto entre el fluido y la superficie sólida, y está influenciado por la geometría de la superficie, la naturaleza del movimiento del fluido (laminar, turbulento o estable) y una variedad de propiedades termodinámicas del fluido de transporte (k, c_p, μ, ρ).

Según los fundamentos termodinámicos, se debe tener muy pendiente el concepto de capa límite de velocidad del flujo, que es de vital importancia para diseños que incluyen transporte de calor por convección libre, y proporciona la base para determinar el coeficiente de fricción local,

un parámetro adimensional a partir del cual se determina la resistencia de rozamiento de la superficie. Esto es posible de entenderse como que la superficie del objeto caliente que se rodea de aire, ocasiona una pérdida de densidad en la capa de aire más próxima a esta, dando lugar a un desplazamiento consecutivo en el tiempo de esta capa de aire a niveles de capas superiores. El hueco que se deja es ocupado por aire de menor temperatura que vuelve a sufrir el mismo efecto, logrando que se genere una corriente de convección que facilita el flujo térmico.

2.2 La generación de entropía: criterio de optimización

La generación de entropía corresponde a la medida cuantitativa de irreversibilidades asociadas a un proceso y se emplea como criterio a minimizar en el diseño geométrico del micro-canal rectangular de un disipador de calor convencional para circuitos integrados (CI) en este trabajo. Desde la mecánica estadística, algunos autores como Atkins (1999) se refieren a la entropía de la siguiente forma:

“Es una variable macroscópica que no puede calcularse a partir de la probabilidad de ocupación de los micro estados equivalentes al sistema, debido a que no existe un valor de la entropía para cada microestado posible. La entropía depende de cómo el sistema se reparte entre los microestados y no del valor que toma una propiedad en cada microestado. Es necesario el uso del concepto de colectivo para calcular el desorden de un colectivo de sistemas en su relación con la entropía, ya que esta depende únicamente de cómo se reparte entre los diferentes microestados posibles (el desorden)”.

La técnica de utilizar la generación de entropía como criterio de optimización es relativamente reciente y poderosa, ya que tiene sus bases en la segunda ley de la termodinámica y es una

técnica propuesta inicialmente por Adrián Bejan (1982) como una metodología multidisciplinaria que incluye áreas de mecánica de fluidos, transferencia de calor y la termodinámica. La mínima generación de entropía MGE se utiliza como una medida de la máxima eficiencia en el proceso de transferencia de calor.

Luego la generación de entropía se utiliza para optimizar el proceso de refrigeración y obtener resultados satisfactorios cada vez mejores en investigaciones posteriores realizadas por autores como Culham y Muzychka (2001) y es en el trabajo realizado por Khan (2009) dónde se utiliza para optimizar el rendimiento de un disipador de calor como el que aquí se propone a través de los parámetros que caracterizan este fenómeno físico-químico.

Este modelo de generación de entropía asociado al disipador de calor, se empieza a producir al establecer como punto de partida a la generación de entropía como el resultado de la suma de una tasa de generación de entropía durante el proceso de transferencia de calor y una tasa generada por la fricción del fluido; de ahí en adelante un desarrollo matemático que incluye todas las variables asociadas al proceso de refrigeración en un disipador. En el proceso de operación del dispositivo adherido al semiconductor estas irreversibilidades van asociadas a aspectos referentes como la caída de presión y la diferencia finita de temperatura en los componentes del disipador. El modelo de la MGE permite encontrar el valor mínimo de cada uno de los componentes del disipador en forma apartada. Se debe tener en cuenta que un proceso se hace muy irreversible al aumentar la generación de entropía, por lo cual la generación de entropía en un proceso real no puede arrojar un valor negativo.

2.3 Resistencia térmica en la generación de entropía

Un aporte significativo para la creación final del modelo matemático que caracteriza la función de generación de entropía en un disipador es obtenido en el trabajo realizado por Knight , Goodling , Hall y Jaeger (1991), al tratar la resistencia térmica de un disipador como un parámetro que está estrechamente relacionado con la función que define la generación de entropía y consecuentemente con el diseño y también la operación de un micro-canal. Este parámetro es quien indica cuanta oposición ofrece el disipador de calor al paso del flujo calorífico y este depende básicamente de las propiedades físicas y térmicas del material en que se fabrican los disipadores.

Un estudio realizado por García González y Rubio Jimenez (2006) refiere que para configuraciones geométricas distintas de canal rectangular y canal triangular, el uso de microcanales rectangulares presenta mayor eficacia para la disipación de calor debido a que para un mismo número de Reynolds se presenta un valor de resistencia térmica menor en un canal rectangular con respecto a un canal triangular.

2.4 Desarrollo del modelo

El diseño general de los disipadores se realiza al colocar un número determinado de aletas paralelas equidistantes de altura H , sobre una tarjeta de largo L y profundidad W , cuando el espesor de la misma es despreciable, tal como aparece en la Figura 1. El objetivo del diseño es transferir calor a través del fluido refrigerante con volumen $H \times L \times W$, que para este caso es aire. Pero para las técnicas enfocadas en consideraciones de estudio microelectrónicos que

utilizan microcanales, se debe tener en cuenta el espesor de cada aleta, para poder analizar el número de canales posibles que espaciados a una distancia fija sobre la tarjeta base, permiten refrigerar satisfactoriamente el dispositivo electrónico y preservar su desempeño en un rango adecuado de temperaturas.

El disipador de calor se modela a partir de los parámetros que intervienen en la generación de entropía bajo las condiciones de operación con máxima eficiencia, buscando encontrar el valor mínimo aportado por cada uno de los componentes en el proceso de transferencia de calor. Está basado en el modelo planteado y desarrollado por Khan, Culham y Yovanovich(2009), y que fue posible después de distintas investigaciones en el tema, como aquella de estos mismos autores que se dedicó a analizar la influencia de las propiedades de los materiales en el rendimiento del proceso de transferencia de calor en disipadores. También este modelo considera como característica principal el hecho de la disminución que presenta la densidad del aire cuando este atraviesa los microcanales; y se relaciona esta disminución con el número de Knudsen (Kn), el cual se obtiene a partir de la relación entre el camino libre promedio (λ) propio del aire, y el diámetro hidráulico efectivo del canal (Dh).

En la producción y construcción final del modelo son varios los aspectos de anteriores investigaciones los que se tienen en cuenta, cabe resaltar:

- El obtenido por Khan (2007), donde se concluye despreciar la transferencia de calor producida por el fenómeno de conducción, debido a que la naturaleza metálica del disipador hecho en aluminio o en bronce se homogeniza con la del dispositivo que se adhiere.

Para que el disipador de calor funcione adecuadamente y para este caso, se debe optimizar entonces la distribución geométrica de los canales, teniendo en cuenta las dimensiones y las

propiedades (físicas y térmicas) propias del material de fabricación, así como también las del fluido refrigerante que lo atraviesa, que en este caso es aire, el fluido refrigerante que pasa a través de los microcanales de forma continua para retirar el calor. El disipador de calor se modela utilizando microcanales rectangulares, y las tres variables que se optimizarán son: H_c , W_c y W_w , que según el modelo de la figura 1, corresponden a la altura del canal, la mitad del ancho del canal y la mitad del espesor del canal respectivamente. Este modelo supone que la pared superior del disipador está aislada y que es la superficie de la base quien se calienta uniformemente debido a la disipación del calor.

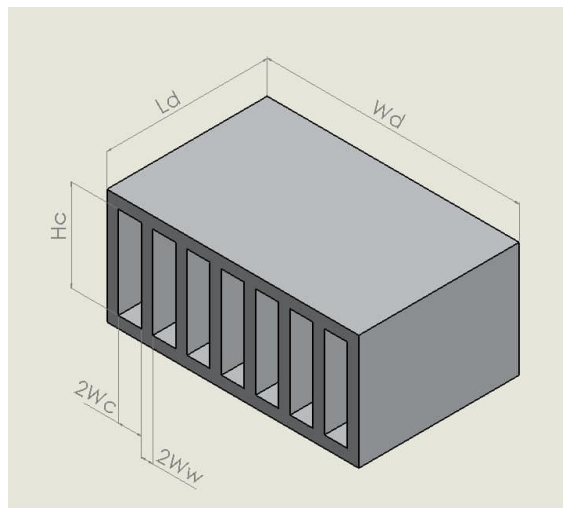


Figura 1. Geometría del Disipador de Calor.

Además de lo expuesto anteriormente, para simplificar el análisis en el desarrollo del modelo, Khan (2009) emplea las siguientes suposiciones:

1. En la superficie interior el flujo de calor es uniforme
2. Las superficies en el canal son lisas
3. La pared superior es adiabática, es decir no intercambia calor.

4. *El material es isotrópico*
5. *El flujo de calor y del fluido están completamente desarrollados*
6. *El flujo de aire puede ser estable o laminar, no turbulento.*
7. *Hay flujo de transición(es decir, $0,1 \leq K_n \leq 10$) o deslizamiento de flujo(es decir $0,001 \leq K_n \leq 0,1$)*
8. *La conducción axial en la aleta es despreciable*
9. *Los cambios en la energía cinética y potencial del aire son despreciables*
10. *Las propiedades primarias y secundarias del fluido son constantes*

2.5 Caracterización de la función objetivo

La generación de entropía asociada al disipador de calor se puede definir mediante la siguiente expresión:

$$S_{gen} = S_{gen,ht} + S_{gen,ff} \quad (1)$$

Donde los términos globales $S_{gen,ht}$ y $S_{gen,ff}$ corresponden a la entropía generada por la transferencia del calor, y por el flujo de fluido, respectivamente. Estas variables se desglosan en las Tablas 1 y 2. De acuerdo al modelo matemático aquí propuesto, la ecuación 10 exhibe una considerable diferencia respecto al modelo planteado en el estudio realizado por Khan.

Tabla 1

Ecuaciones que definen la entropía por la transferencia de calor

Número de ecuación	Fórmula	Unidades
2	$S_{genht} = \frac{Q^2 R_{th}}{T_o T_b}$	$\left[\frac{W}{K}\right]$
3	$Q = qA$	[W]
4	$A = 2N_m L(w_c + H_c \eta_{fin})$	$[m^2]$
5	$N_m = \frac{W - 2w_w}{2(w_c - w_w)}$	[Ad]
6	$\eta_{fin} = \frac{\tanh(mH_c)}{mH_c}$	[Ad]
7	$mH_c = \frac{1}{\alpha_c} \sqrt{2\beta k_{eq} Nu_{D_h} (1 + \alpha_c)}$	[Ad]
8	$R_{th} = \frac{2C_3 \alpha_{hs}}{k_f L C_1 C_2}$	$\left[\frac{K}{W}\right]$
9	$C_3 = \frac{1}{Nu_{D_h}} + \frac{C_1}{Pe_{D_h}}$	[Ad]
10	$Nu_{D_h} = \frac{140}{17(1 + \alpha_c) K_s} = \frac{D_h h_{av}}{k_f}$	[Ad]
11	$K_s = 1 - \frac{6}{17} \left(\frac{U_s}{U_{av}}\right) + \frac{2}{51} \left(\frac{U_s}{U_{av}}\right)^2 + \frac{140}{17} \frac{\zeta_t}{(1 + \alpha_c)}$	[Ad]
12	$\frac{U_s}{U_{av}} = \frac{6\alpha}{1 + 6\alpha}$	[Ad]
13	$\alpha = \frac{2\zeta_u}{1 + \alpha_c}$	[Ad]
14	$\zeta_u = \left(\frac{2 - \sigma}{\sigma}\right) K_n$	[Ad]
16	$Pe_{D_h} = \frac{D_h U_{av}}{\alpha_{th}}$	[Ad]

Número de ecuación	Fórmula	Unidades
17	$D_h = 2(2w_c)$	[m]
18	$U_{av} = \frac{\dot{m}}{N_m \rho (2w_c) H_c}$	$\left[\frac{m}{s}\right]$
19	$\dot{m} = \rho G$	$\left[\frac{Kg}{s}\right]$
20	$\alpha_{hs} = \frac{L}{2w_c}$	[Ad]
21	$C_2 = \frac{(1 + \alpha_c)}{\alpha_c}$	[Ad]
22	$\alpha_c = \frac{2w_c}{H_c}$	[Ad]
23	$C_1 = N_m \alpha_{hs} (2\eta_{fin} + \alpha_c)$	[Ad]
24	$T_b = T_o + QR_{th}$	[K]
25	$m = \sqrt{\frac{2h_{av}}{kw_w}}$	$[m^{-1}]$
26	$S_{genht} = \frac{Q^2}{T_o T_b} \frac{2\alpha_{hs} C_3}{Lk_f C_1 C_2}$	$\left[\frac{W}{K}\right]$

Nota: Khan et.al (2009) en el trabajo titulado “*Optimization of microchannel heat sinks using entropy generation minimization method*” utilizan métodos iterativos como el gradiente y los multiplicadores de LaGrange para encontrar los parámetros óptimos

Tabla 2

Ecuaciones que definen la entropía debido a la fricción del fluido

Número de ecuación	Fórmula	Unidades
(27)	$S_{genff} = \frac{\dot{m} \Delta P}{\rho T_o}$	$\left[\frac{W}{K}\right]$
(28)	$\Delta P = \frac{\rho U_{av}^2}{2} C_4$	[Pa]

Número de ecuación	Fórmula	Unidades
(29)	$C_4 = \left[k_{ce} + \left(f \frac{L}{D_h} \right) \right]$	[Ad]
(30)	$k_{ce} = 1.79 - 2.32 \left(\frac{w_c}{w_c + w_w} \right) + 0.53 \left(\frac{w_c}{w_c + w_w} \right)^2$	[Ad]
(31)	$f = \frac{24}{Re_{D_h}} \left(\frac{1}{1 + 6\alpha} \right) \left(\frac{1}{1 + \alpha_c} \right)$	[Ad]
(32)	$S_{genff} = N_m \frac{\rho w_c H_c U_{av}^3}{T_o} C_4$	$\left[\frac{W}{K} \right]$

Después de obtener la fórmula matemática de la generación de entropía por causa de la transferencia de calor y la producida por la fricción del fluido en las ecuaciones (26) y (32), se logra obtener la función objetivo del problema que corresponde a la generación de entropía total calculada como la suma de ambas; y en donde aparecen los parámetros definidos para el diseño óptimo del micro-canal.

$$S_{gen} = \frac{Q^2}{T_a T_b} \frac{2\alpha_{hs} C_3}{L k_f C_1 C_2} + \frac{N \rho w_c H_c U_{av}^3}{T_a} C_4 \quad (33)$$

Se debe tener en cuenta que en el modelo desarrollado por Khan, se utiliza la velocidad de deslizamiento del flujo y el cambio de temperatura en las condiciones de contorno. El resultado de la generación de entropía en teoría no debería arrojar un valor negativo, ya que esto significaría un proceso imposible de realizar.

3. Algoritmos de optimización numérica meta- heurísticos: GSA y DSO

En esta sección se presenta la fundamentación teórica y el modo de funcionamiento de los dos algoritmos para optimización que se implementan en este trabajo, para encontrar la solución de la función objetivo en término de las restricciones y el criterio de minimización. Hay que destacar que una de las finalidades de este trabajo es comparar las soluciones obtenidas con respecto a la eficiencia y robustez que pueden tener otros métodos meta heurísticos como PSO. La eficiencia es un aspecto importante porque la solución se logra a partir de un procedimiento iterativo. La robustez es vista como la habilidad de un algoritmo para encontrar la solución y es una propiedad deseable ya que las funciones no lineales se comportan de forma impredecible y esto muchas veces no permite avanzar rápidamente en la búsqueda de la solución.

Los nuevos métodos de optimización se basan en teorías inspiradas en la naturaleza y a diferencia de los métodos de busca exhaustiva, estos algoritmos tienen la capacidad de la que carecen otros métodos clásicos de resolución, y es la de explorar y acceder a cualquier región del espacio de búsqueda. Estos algoritmos no garantizan una solución exacta al problema que se aborda, sino una solución de aproximación en donde la calidad de la respuesta depende de los recursos dedicados a la búsqueda. También deben tener un componente aleatorio que dote a los algoritmos de la capacidad para escapar de óptimos locales y explorar distintas zonas del espacio de búsqueda en caso de quedar atascados. A su vez se pueden constituir en un punto intermedio que reúna características de búsqueda aleatoria y de búsqueda dirigida. De forma general los nuevos algoritmos utilizan las técnicas básicas de penalización y reparación, para los problemas

que se plantean con restricciones, cuando hay condiciones que restringen la evaluación de la función objetivo.

3.1 Algoritmo de búsqueda gravitacional GSA

Este algoritmo propuesto por E. Rashedi, Nezamabadi-pour y S. Saryzdi (2009), se fundamenta en la ley de gravitación universal, y considera a los agentes como partículas que atraen y pueden ser atraídas, con una fuerza directamente proporcional al producto de sus masas, e inversamente proporcional al cuadrado de la distancia que las separa.

El rendimiento del algoritmo está ligado al concepto de masa, donde la misma corresponde a la evaluación de la partícula en la función objetivo.

En GSA, de acuerdo a las definiciones teórico físicas, las partículas agentes están especificadas por parámetros y propiedades físicas: masa inercial, M_i , el concepto asociado a la medida de la resistencia de un objeto a cambiar su estado de movimiento cuando una fuerza es aplicada; masa gravitacional activa, M_a , un concepto asociado a la medida de la fuerza de un campo gravitacional generado por un objeto; masa gravitacional pasiva, M_p , considerada como la medida de la fuerza que un objeto experimenta en un campo gravitacional; y posición, X . La solución del problema consiste en localizar la posición de las partículas agentes. En términos generales, el algoritmo GSA se puede definir como:

- *Construir un sistema de n partículas agentes y d dimensiones, donde sus posiciones están descritas de acuerdo a la ecuación (33).*

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \text{ para } i = 1, 2, \dots, n \quad (34)$$

- *Evaluar el sistema construido en la función objetivo. Este proceso es importante porque arroja el cálculo de los mejores y peores valores que se pueden obtener hasta el momento. En este caso, son considerados como mejores los valores mínimos. Dichos valores se nombran como Best y Worst.*
- *Determinar un sistema de masas correspondiente a la función objetivo. La partícula agente con mayor masa representa el resultado con mayor utilidad, principalmente porque esto significa la presencia de mayor fuerza de atracción hacia las otras masas y también una velocidad menor como resultado de una mayor resistencia al movimiento.*
- *Calcular la fuerza de atracción entre las partículas agentes y la aceleración de los agentes. El primero se hace utilizando factores como la distancia euclidiana, la masa, y la constante gravitacional, entre otras. Se expresa como:*

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) \cdot M_{aj}(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad (35)$$

Donde,

G = constante gravitacional

M_{pi} = masa gravitacional pasiva del agente i

M_{aj} = masa gravitacional activa del agente j

$R_{ij}(t)$ = distancia Euclidiana entre los agentes i y j en determinado tiempo

ε = es una constante muy pequeña comparativamente.

El análisis queda limitado cuando se asume que la masa gravitacional activa del agente j y la masa gravitacional pasiva del agente i , son iguales a la masa gravitacional inercial M_i ; serán obvias las simplificaciones que se obtendrán así.

En GSA, la constante gravitacional, G , tendrá un valor inicial, G_0 , que se reduce con el tiempo; esto produce dinamismo en el algoritmo, y a su vez permite controlar la precisión en la búsqueda. Por consiguiente, su valor evoluciona de la siguiente forma,

$$G(t) = G_0 * e^{-\alpha t} \quad (36)$$

De forma general, la distancia Euclidiana entre dos puntos del espacio n dimensional, se deduce utilizando el teorema de Pitágoras.

$$R_{ij}(t) = \|X_i(t), X_j(t)\| \quad (37)$$

Las características estocásticas correspondientes a este algoritmo, se logran cuando se define a la fuerza que actúa sobre el agente i , en una dimensión d , como la suma ponderada aleatoria de las fuerzas ejercidas por los otros agentes.

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t) \quad (38)$$

El cálculo de la aceleración sobre los agentes se obtiene empleando la ley del movimiento. Ésta indica que la variación en la velocidad de un objeto es igual a la fuerza que actúa sobre el sistema dividida por su masa inercial; entonces, la aceleración del agente i en la dimensión d , para un tiempo determinado se escribe así:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (39),$$

Donde M_{ii} representa la masa inercial del agente i .

- *Calcular la velocidad y la posición de los agentes. La velocidad inmediata de un agente se considera como la suma de la aceleración y su velocidad instantánea actual con ajustes aleatorios.*

3.1.1 criterio de parada: el algoritmo está diseñado para ejecutarse por un número específico de iteraciones. Pero debe considerar una parada cuando se detecte que la mayor parte de la población ha convergido a una forma similar, es decir que no presente un cambio significativo en la mejor solución encontrada de acuerdo a un umbral establecido. Este valor se especifica más adelante cuando se implemente el algoritmo sobre la función objetivo del disipador de calor.

3.1.2 Número de agentes: se debe seleccionar un número de agentes determinado que permita reducir la dispersión de los datos al momento de obtener los resultados que arroja el algoritmo. Este valor se especifica más adelante cuando se implemente el algoritmo sobre la función objetivo del disipador de calor.

3.1.3 Parámetro Kmejores: es una función que depende del tiempo y se utiliza para mejorar el rendimiento del algoritmo. Tiene un valor inicial al comienzo, el cual decrece al transcurrir el tiempo. Toma un valor igual a uno, cuando los agentes con mejores valores atraen a los otros, y un valor igual a cero, cuando toda la población ejerce fuerza sobre cada masa.

3.1.4 diagrama de flujo del algoritmo. En la siguiente figura se observa el diagrama del algoritmo GSA.

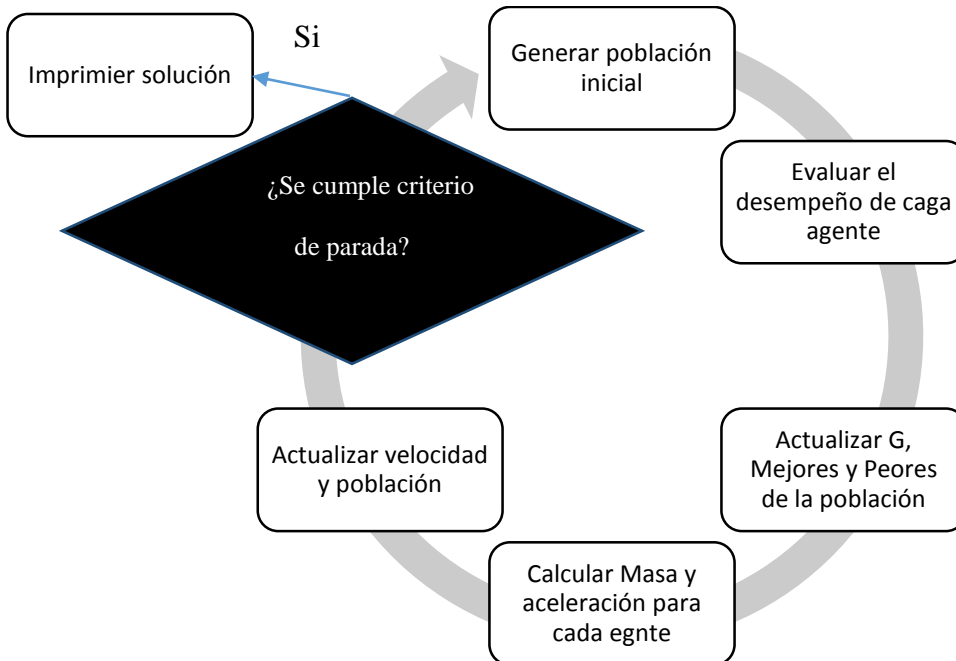


Figura 2. Diagrama de Flujo del Algoritmo GSA

3.2 Algoritmo de optimización drone squadron (DSO)

El algoritmo DSO es una propuesta en optimización numérica global de los profesores Wolfgang Banzhaf y Vinícius Veloso de Melo (2017). A diferencia del anterior algoritmo, éste propone uno inspirado no en la naturaleza, sino en dispositivos creados artificialmente, como los drones. Estos dispositivos navegan de forma autónoma o remota, y pueden actualizarse y mejorarse a través del tiempo, no solo cambiando el hardware, sino también cambiando su software (firmware). La visión general del algoritmo es, según el profesor Veloso de Melo:

DSO funciona como si estuviese compuesto de un escuadrón de drones con diferentes equipos y un centro de comando para todos ellos. Este centro de comando usa información recolectada por y desde cada uno los drones para realizar dos operaciones: mantener el control parcial de

la búsqueda y desarrollar un nuevo firmware (actualización del código) para controlar a los mismos drones

Un dron no representa la solución del problema, sino en cambio se mueve con dirección a una coordenada que corresponderá a la solución. La búsqueda es inmediata y se logra después de varias iteraciones, por lo cual el firmware debe contener los procedimientos y las configuraciones utilizadas por cada uno de los equipos para explorar el paisaje de búsqueda y no quedarse atrapado o atascado en una solución local.

El algoritmo DSO basa su funcionamiento en el movimiento a través del espacio de búsqueda, una característica similar que lo relaciona con cualquier algoritmo de enjambre de partículas. Pero, la diferencia radica en que el enfoque de DSO permite elegir usar recombinación o variación de las posibles soluciones en el espacio de búsqueda con distintos procedimientos. Por esto, DSO puede funcionar de alguna manera como: un algoritmo evolutivo, un algoritmo de enjambre, un algoritmo probabilístico u otro según se vaya desempeñando en el espacio de búsqueda. Esta es la característica principal, que el algoritmo va utilizando un método de optimización que puede ir mejorando con el tiempo. Esta mejora ocurre cuando se detecta alguna característica sobre el espacio de búsqueda que indica que la búsqueda se va realizando sobre una región altamente prometedora que indica solución global. Esto lo convierte en un algoritmo auto-adaptativo con respecto a modificación de su propio código, y no únicamente la modificación de los parámetros de funcionamiento del mismo. Este enfoque híbrido que emplea la co-evolución para mejorar su propio código, es lo que hace a DSO un novedoso algoritmo que puede evolucionar una vez esté corriendo y comportarse de alguna forma como un algoritmo evolutivo o uno de enjambre. DSO no funciona como un prototipo de algoritmo que está eligiendo entre muchos otros algoritmos preestablecidos, el código se va generando una vez se

está implementando. Los drones que se utilizan en la exploración se dividen en equipos del mismo tamaño y cada equipo tiene su propio firmware, el cual controla el movimiento de cada dron de su equipo, y así cada equipo muestrea de forma distinta el espacio de búsqueda del mismo conjunto de posibles soluciones actuales.

El firmware contiene los procedimientos (códigos) y las configuraciones utilizadas por los equipos para explorar el paisaje. El centro de comandos es el lugar central para llevar a cabo las órdenes y supervisar las tareas. Recibe entradas, procesa datos y genera resultados de decisiones internas. El centro de comando actualiza el firmware de los drones siempre que sea posible, decidiendo y adaptando dinámicamente el comportamiento de un equipo al problema en cuestión. El procedimiento de la perturbación se realiza estableciendo un código fuente, como un conjunto de pasos que debe ejecutar cada dron.

A continuación, se presenta una explicación de cada uno de estos componentes y el modo de funcionamiento en detalle del algoritmo.

2.1 Centro de comandos: el centro de comando es un lugar central para llevar a cabo órdenes y separar las actividades a realizar por cada uno de los equipos de búsqueda. Recibe las entradas, procesa los datos y genera nuevas indicaciones de acuerdo a los resultados de las decisiones internas. Se denomina como la parte inteligente del algoritmo DSO, ya que es responsable de producir y dar las nuevas órdenes que ejecutaran los drones en su objetivo de encontrar los resultados que se esperan. Al usar la información recopilada por cada uno de los drones, el centro de comandos modifica el firmware para actualizar los equipos de drones. Esta actualización se consigue a través de un sistema de enfoque hiperheurístico que genera el código del firmware. Este centro de comando es quien otorga al algoritmo DSO características mucho más completas que los algoritmos inspirados en la naturaleza.

3.2.2 Firmware: contiene el mecanismo para generar nuevas coordenadas de prueba (TC) utilizando la perturbación, y este esquema de perturbación se encuentra en el núcleo de cada dron. Se genera un nuevo firmware basado en el esquema de perturbación de recorrido aleatorio.

$$P = \text{Punto de partida} + \text{desplazamiento} \quad (40)$$

$$TC = \text{calcular}(P) \quad (41)$$

Donde *Punto de partida*, es una coordenada de inicio, un punto que hace parte de la solución en el espacio de búsqueda. Y *desplazamiento* es una función que arroja el valor numérico a través de la perturbación y que indicara el movimiento actual que debe hacerse.

Tanto las coordenadas de salida como la función que produce el desplazamiento son modificables, es decir, los diferentes equipos tienen distintas formas de elegir las coordenadas de salida y cómo calcular el desplazamiento. Como ejemplo, se supone que se tienen dos equipos con perturbaciones P_1 y P_2 respectivamente,

$$P_1: \vec{GBC} + (C_1 \times (\vec{GBC} - \vec{CBC}_{dron})) \quad (42)$$

$$P_2: \vec{CBC}_{dron} + (G(0,1) \times (\sqrt{U(0,1)_D} + \vec{CBC}_{dron})) \quad (43)$$

donde:

- C_1 es una constante definida por el usuario
- $G(0,1)$ es un escalar muestreado de una distribución Gaussiana con media cero y desviación estándar en uno.
- $U(0,1)_D$, es una matriz con D números muestreados a partir de una distribución uniforme con mínimo 0 y máximo 1.

El esquema de perturbación usa una representación de estructura de árbol con nodos terminales y no terminales, se inspira en el algoritmo PSO GP, de los autores Poli, Langdon, y Holland (2005)

El centro de comando tiene instrucciones para configurar inicialmente el firmware por lo menos con una perturbación de referencia. Se definen perturbaciones de referencia, es decir, ecuaciones usadas como perturbaciones iniciales para los equipos, pero que pueden ser reemplazadas durante el proceso de optimización.

Las dos perturbaciones de referencia disponibles para este DSO son:

$$1. \vec{CBC}_{r1} + c_1 * (\vec{CBC}_{r2} - \vec{CBC}_{r3}) \quad (44)$$

$$2. MVNS + Step(CBC); \quad (45)$$

Donde r1, r2, y r3 son distintas soluciones aleatorias.

En el algoritmo que utiliza enjambre de partículas a través de programación genética PSO GP, se crea una población inicial utilizando el método de "crecimiento" con un máximo de profundidad de 6 niveles (el nodo raíz está en el nivel 0). En la función de generación de fuerza se considera el uso de una fuerza diferente para cada una de las partículas en el enjambre. Las partículas en un enjambre tienden a seguir un mismo camino similar hacia un objetivo común, pero hay pequeñas diferencias en la forma en que proceden, esta diferencia se puede atribuir a que los individuos poseen personalidad propia, por lo cual, al actualizar por separado a cada individuo, le dan a cada partícula una personalidad. Al principio al inicializar la posición de la partícula, la función generadora de fuerzas de cada partícula también será inicializada. Luego en cada una de las iteraciones, junto con la actualización de la posición de la partícula y la velocidad, también la función generadora de fuerzas se actualiza, usando el cruce y la mutación

del operador de programación genética y logrando mayor diversidad y adaptabilidad con respecto al PSO clásico.

3.2.3 Drones en movimiento. Los drones usan un sistema autónomo para calcular posiciones de destino, moverse hacia ellas, y recopilar información que retorne al centro de Comando. Los mecanismos para calcular la posición de destino en el algoritmo DSO son los mismos que utilizan distintas técnicas de optimización evolutivas y no evolutivas: recombinación y variación. Lo que comúnmente en el algoritmo genético se conoce como mutación y cruzamiento.

El objetivo de este paso es generar las posiciones objetivo de cada dron en cada equipo (TmC). Cuando cada uno de los drones ejecuta el paso de perturbación y genera las coordenadas de prueba (TC_{dron}), se realiza una recombinación con las mejores coordenadas que se encuentran hasta el momento y se genera (TmC), dependiendo, también es posible que no haya una nueva recombinación y suceda que TmC sea igual a TC. La recombinación se realiza después de la perturbación y la elección es aleatoria, por lo que todos los procedimientos de recombinación disponibles para los drones tienen la misma probabilidad de ser seleccionado.

Los drones sólo pueden moverse dentro de un perímetro particular por lo que, si alguna de estas coordenadas generadas se encuentra por fuera del perímetro designado para la búsqueda, se debe hacer una corrección, y esta corrección se designa como una violación en la búsqueda. Las violaciones se acumulan para cada equipo, teniendo en cuenta todos los drones que lo conforman y se calcula de la siguiente manera:

$$violaciones_{equipo} = \sum_{dron=1}^N \sum_{j=1}^D \left\{ \begin{array}{l} |TmC_{equipo,dron,j} - UB_j| \\ |LB_j - TmC_{equipo,dron,j}| \end{array} \right\} +, \quad (46)$$

Donde N es el número de drones por equipo, D es la dimensión del problema, UB es la matriz de límites superior y LB es la matriz de límites inferiores.

Después que los drones se mueven y calculan la función objetivo, se envían sus resultados al centro de comandos que es finalmente quien toma las decisiones de actualización del firmware.

3.2.4 Actualización del firmware. A partir de la información suministrada al centro de comandos, este analiza la calidad del trabajo de cada uno de los equipos teniendo en cuenta el rango de los valores con respecto a la función objetivo y el grado de coordenadas fuera de los límites que generaron cada uno de estos. La calidad de cada uno de los equipos se calcula en cada iteración, puede ser acumulada o promediada después de varias iteraciones para compararse con el umbral.

Una vez el criterio de actualización de firmware es alcanzado, el centro de comandos se encarga de reemplazar el peor firmware por alguna variante del mejor firmware encontrado, no se debe entender esto como una recombinación de códigos ya que la nueva variante debe cumplir que el número de nodos de la estructura de árbol de la nueva perturbación que tendrá el peor equipo debe cumplir con los parámetros iniciales definidos por el usuario. Y la nueva perturbación debe seguir el esquema de la ecuación (40).

3.2.5 Recombinación. Luego que el procedimiento de perturbación genera nuevas coordenadas, se hace el procedimiento de recombinación con las coordenadas actuales, para representar las mejores coordenadas encontradas hasta ahora. Son posibles tres formas de hacer recombinación:

1. No recombinación

2. Recombinación uniforme como en el algoritmo genético / Recombinación binomial
3. Cruce de uno o dos puntos como en el algoritmo genético / Recombinación exponencial

3.2.6 Pseudocódigo del algoritmo. En las líneas siguientes enumeradas paso a paso.

1: **Entradas:** Función objetivo, Límites del problema (\rightarrow, \rightarrow), constantes definidas por el usuario, número de equipos (t), número de drones por equipo (N), número máximo de iteraciones ($MaxIt$), número máximo de iteraciones sin mejora ($MaxEstancamiento$), opciones para el firmware.

2: **Salidas:** mejor solución (GBC) y mejor valor de la función objetivo ($GBOFV$).

3:

4: **Inicializa** el centro de comandos, los drones, y los equipos en TmC , donde TmC especifica las coordenadas de los drones.

5: **Envía** los drones ($t \times N$) a explorar el paisaje en cada coordenada.

6: **Selecciona** las N mejores coordenadas que conformaran la matriz CBC

7:

8: **Mientras** parar si no se cumplen los criterios **hacer**

9: \rightarrow *Procedimiento realizado por cada equipo*

10: **Para cada** *equipo* $\in [1, 2, \dots, t]$ **hacer**

11: **Para cada** *drone* $\in [1, 2, \dots, N]$ **hacer**

12: **Generar** TC usando los esquemas de perturbación actual de los equipos

13: **Escoja** el método de recombinación y aplicarlo para combinar TC con

CBC_{drone} para obtener $TmC_{equipo, drone}$

27: $CBC_{dron} \leftarrow TmC_{drone,mejorID}$

28: **Si** $TmOFV_{drone,mejorID} < GBOFV_{dron}$ **entonces**

29: $mejora \leftarrow verdadero$

30: $GBOF V \leftarrow TmOF V_{dron,mejorID}$

31: $GBC \leftarrow TmC_{drone,mejorID}$

32: **Fin Sí**

33: **Fin Sí**

34: **Fin para**

35: **Sí** $mejora = falso$ **entonces**

36: **Comprobar** y aplicar control de estancamiento

37: **Fin Sí**

38: **Establecer** $CalidadEquipo$ como el promedio de rango más las violaciones de los límites de contorno.

39: **Actualizar** el firmware de acuerdo a $CalidadEquipo$

40: **Fin Mientras.**

3.2.7 Ejemplo del mecanismo de una iteración en DSO En este trabajo sobre DSO, como técnica novedosa para optimización numérica global, se pone de muestra un ejemplo del funcionamiento del algoritmo de la siguiente forma.

Si se quiere minimizar la siguiente función de dos variables, es decir $D=2$

$$f(x) = \sum_{j=1}^D x_j \quad (47)$$

Y si se tienen dos equipos, $t=2$, con dos drones cada uno, $N=2$.

También,

- $\begin{matrix} \rightarrow \\ LB \end{matrix} = [0.0 \ 0.0] ; \begin{matrix} \rightarrow \\ UB \end{matrix} = [5.0 \ 5.0] \quad (48)$

- *El firmware de cada equipo tiene su propia perturbación, es decir,*

- $P1 = CBC - 1; \quad P2 = CBC \times 2 \quad (49)$

- $\begin{matrix} \rightarrow \\ GBC \end{matrix} = [1.0 \ 2.0] ; \quad GBOFV = 3.0; \quad (50)$

- $CBC = \begin{bmatrix} 1.0 & 2.0 \\ 3.0 & 4.0 \end{bmatrix}; \quad \begin{matrix} \rightarrow \\ CBOFV \end{matrix} = [3.0 \ 7.0] \quad (51)$

Luego generamos TC_1 y TC_2 para cada equipo respectivamente de acuerdo a:

- $TC_1 = calcular(P1) = calcular(CBC - 1) = \begin{bmatrix} 0.0 & 1.0 \\ 2.0 & 3.0 \end{bmatrix} \quad (52)$

- $TC_2 = calcular(P2) = calcular(CBC \times 2.0) = \begin{bmatrix} 2.0 & 4.0 \\ 6.0 & 8.0 \end{bmatrix} \quad (53)$

Se recombinan los anteriores valores de coordenadas con los valores de la matriz CBC para obtener la matriz que correspondería a las mejores posiciones encontradas por cada uno de los equipos de la siguiente manera,

$$TmC_{equipo,dron} = [TC_{equipo,dron,1} \ CBC_{dron,2}], \quad dron = 1, \dots, N: \quad (54)$$

Para cada dron del primer equipo tendríamos,

- $TmC_{1,1} = [TC1_{1,1} \ CBC_{1,2}] = [0.0 \ 2.0] \quad (55)$

- $TmC_{1,2} = [TC1_{2,1} \ CBC_{2,2}] = [2.0 \ 4.0] \quad (56)$

Para cada dron del segundo equipo tendríamos,

- $TmC_{1,1} = [TC2_{1,1} \ CBC_{1,2}] = [2.0 \ 2.0] \quad (57)$

- $TmC_{1,2} = [TC2_{2,1} \ CBC_{2,2}] = [6.0 \ 4.0] \quad (58)$

Entonces las coordenadas para cada uno de los equipos quedarían de la siguiente manera,

$$TmC_1 = \begin{bmatrix} 0.0 & 2.0 \\ 2.0 & 4.0 \end{bmatrix}; TmC_2 = \begin{bmatrix} 2.0 & 2.0 \\ 6.0 & 4.0 \end{bmatrix} \quad (59)$$

Se revisa que estas coordenadas no estén por fuera de los límites que se establecieron al principio del problema ($\vec{\rightarrow}_{LB}; \vec{\rightarrow}_{UB}$). También se deben acumular estas violaciones que se utilizaran posteriormente.

$$TmC_1 = \begin{bmatrix} 0.0 & 2.0 \\ 2.0 & 4.0 \end{bmatrix}; TmC_2 = \begin{bmatrix} 2.0 & 2.0 \\ 5.0 & 4.0 \end{bmatrix} \quad (60)$$

$$\vec{\rightarrow}_{violaciones} = [0.0 \ 1.0] \quad (61)$$

Luego se calcula el valor de función objetivo para cada dron de cada uno de los equipos,

$TmOFV_{equipo,drone}$:

$$\vec{\rightarrow}_{TmOFV_1} = [2.0 \ 6.0]; \quad \vec{\rightarrow}_{TmOFV_2} = [4.0 \ 9.0] \quad (62)$$

Se calcula el rango de descendencia para cada dron en cada equipo:

$$Rank_1 = [1.0 \ 1.0]; Rank_2 = [2.0 \ 2.0] \quad (63)$$

Se vuelen a cargar las coordenadas CBC_{dron} y $\vec{\rightarrow}_{GBC}$ con los nuevos mejores valores de cada dron=1,....., N, si es necesario.

- $CBC = \begin{bmatrix} 0.0 & 2.0 \\ 2.0 & 4.0 \end{bmatrix}$ y $\vec{\rightarrow}_{CBOFV} = [2.0 \ 6.0];$ (64)

- $\vec{\rightarrow}_{GBC} = [0.0 \ 2.0]$ y $GBOFV = 2.0$ (65)

- *Debido a que GBOFV ha mejorado, en este caso no es necesario aplicar el control de estancamiento y se procede a calcular la calidad de cada equipo con el rango*

promedio de $\vec{\rightarrow}_{TmOFV}$ más las violaciones:

- $\vec{\rightarrow}_{TeamQuality} = [1.0 \ 2.5]$ (66)

- *Luego se actualiza el firmware del equipo con la calidad más baja, de la siguiente manera:*

$$P2 = CBC - \frac{1.0}{2.0} = CBC - 0.5 \quad (67)$$

4. Implementación de los algoritmos con funciones de prueba

Se usaron algunas funciones de prueba presentes en la literatura para verificar el correcto funcionamiento de los algoritmos GSA y DSO. Cada una de estas funciones tienen características propias como: cantidad de mínimos locales, valles, placas y crestas empinadas. Además, tienen óptimos globales cuya ubicación es conocida. Por tanto, se variaron algunos parámetros del algoritmo, buscando comprobar que, efectivamente, éstos encuentran el óptimo buscado. Estas pruebas se ejecutaron sobre una computadora personal que funciona bajo las características de la tabla 3, y a continuación se describen brevemente las funciones.

Tabla 3

Equipo utilizado

Características de la computadora personal	
CPU	Intel Bay Trail M Quad-Core 3540 up to 2.66GHz
Memoria RAM	4.00 GB

4.1 Función de prueba Drop-wave

Es una función multimodal en dos dimensiones, de gran complejidad, y normalmente se evalúa en un rango de valores de $[-5.12, 5.12]$ para cada una de las dos dimensiones. Muestra un mínimo global igual a -1 para la ubicación $(0,0)$.

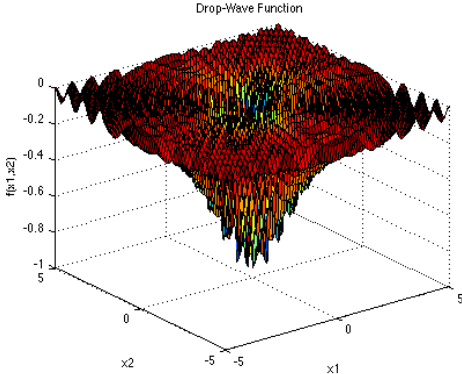
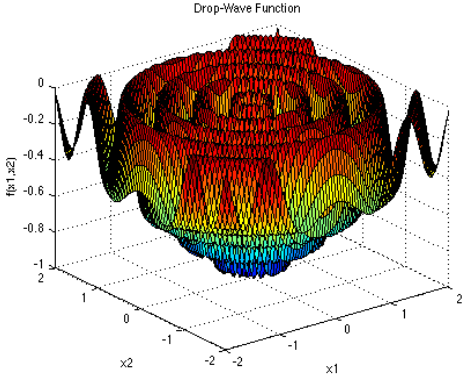
Ecuación	Optimo
$f(x) = -\frac{1 + \cos(12\sqrt{(x_1^2 + x_2^2)})}{0.5(x_1^2 + x_2^2) + 2} \quad (68)$	$f(\vec{0}) = -1 \quad (69)$
<p>Gráfica en dos dimensiones</p> <div style="display: flex; justify-content: space-around;">   </div>	

Figura 3. Función Drop-wave

4.2 Función de prueba Schwefel

Esta es una función compleja con muchos mínimos locales. Está definida para una n-enésima dimensión. Normalmente es evaluada en un rango de valores de $[-500,500]$ para cada una de las dimensiones.

Ecuación	Optimo
$f(x) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{ x_i }) \quad (70)$	$f(\overrightarrow{420.9687}) = 0 \quad (71)$

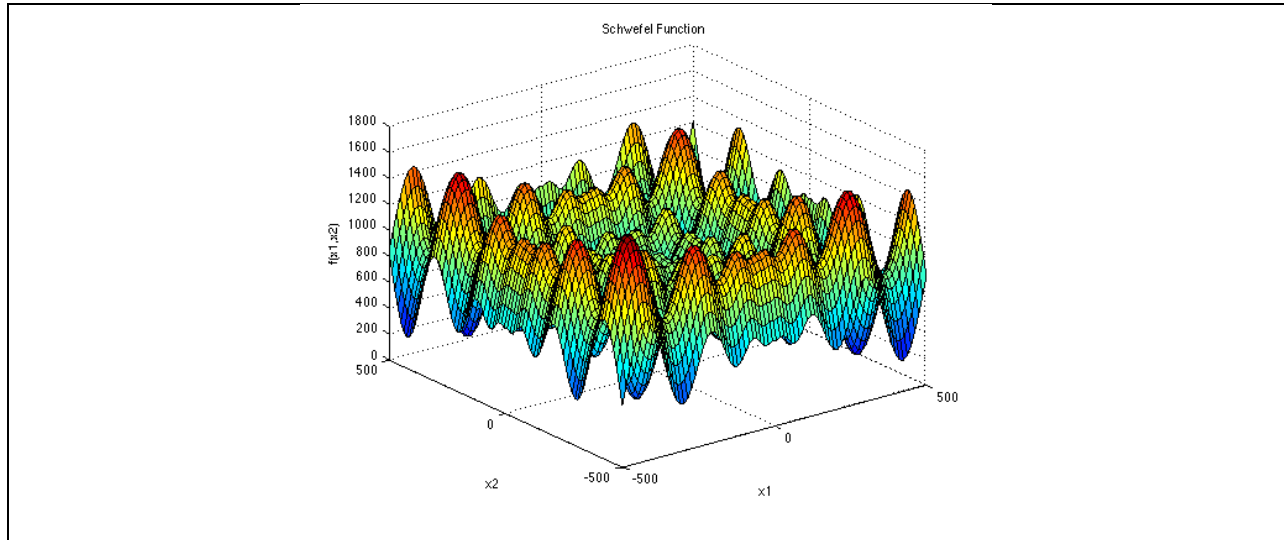


Figura 4. Función Schwefel

4.3 Función de prueba Rastrigin

La función Rastrigin tiene varios mínimos locales. Es multimodal, pero las ubicaciones de los mínimos se distribuyen regularmente. Funciona para una n-enésima dimensión, y cada una de estas se evalúa en el rango de $[-5.12, 5.12]$, con un valor mínimo óptimo de 0.

Ecuación	Optimo
$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$ (72)	$f(\vec{0}) = 0$ (73)
Gráfica en dos dimensiones	

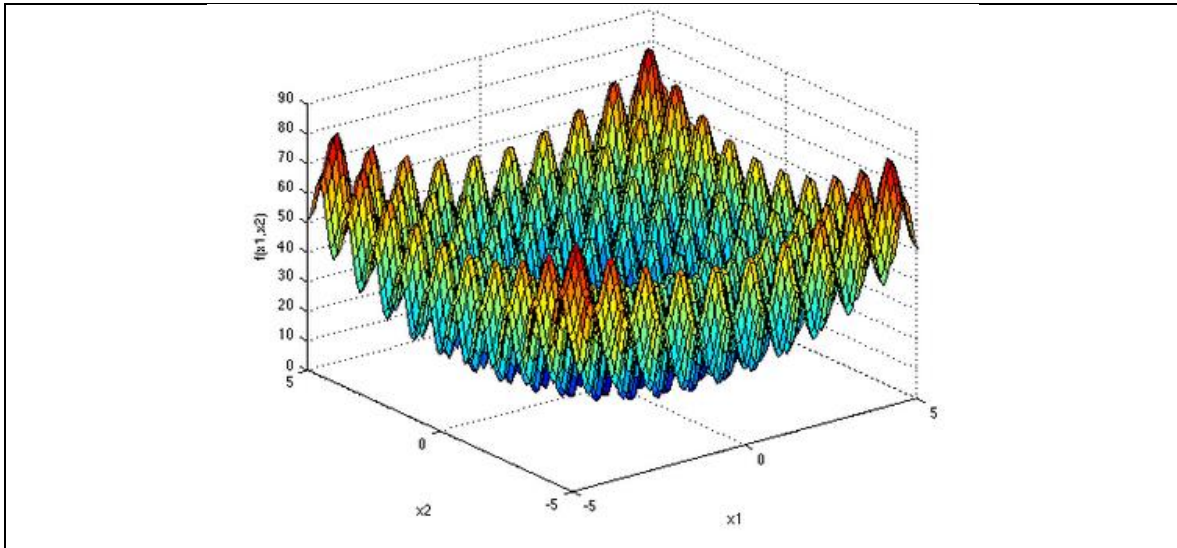


Figura 5. Función Rastrigin

4.4 Función de prueba Rosenbrock

La función Rosenbrock se conoce como la función Valley o Banana. Es una función de prueba popular para los algoritmos de optimización basados en gradiente. La función es unimodal, y el mínimo global se encuentra en un valle estrecho y parabólico. Sin embargo, a pesar de que este valle es fácil de encontrar, la convergencia al mínimo se hace difícil de hallar.

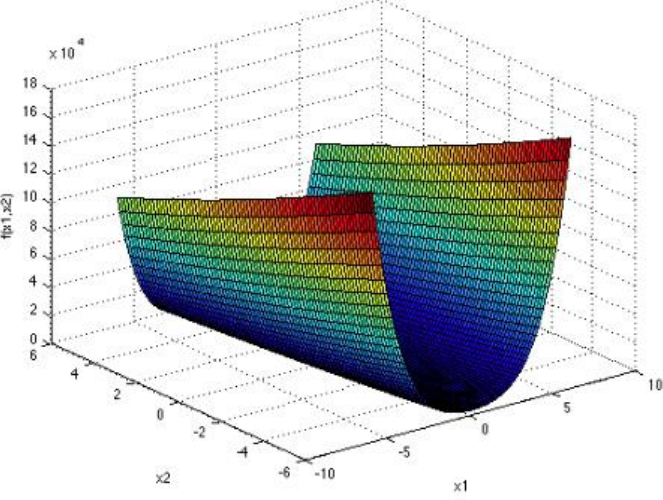
Ecuación	Optimo
$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ <p style="text-align: right;">(74)</p>	$f(\vec{1}) = 0 \quad (75)$
Gráfica en dos dimensiones	
	

Figura 6. Función Rosenbrock

4.5 Resultados de implementar los algoritmos con las funciones de prueba

4.5.1 Variación del número de agentes en GSA. Para estas pruebas, utilizamos tres cantidades de agentes diferentes: 25, 50 y 100. Adicionalmente, analizamos el comportamiento del algoritmo en las funciones de prueba de Rastrigin y Schwefel, con 5 y 10 dimensiones, en tres instantes diferentes (luego de 50, 100, y 1000 iteraciones). Los resultados se presentan en las tablas 4 a 6.

Tabla 4.

Resultados de aplicar 50 iteraciones con GSA

FUNCIÓN- dimensión	Número de agentes					
	25		50		100	
	Mejor	Tiempo [s]	Mejor	Tiempo [s]	Mejor	Tiempo [s]
Drop-(2D)	-0,9854	2,053	-0,9865	3,405	-0,9889	6,679
Schwefel-(10D)	2,7860E3	2,204	2,6406E3	3,594	2,7000E3	8,566
Schwefel-(5D)	1,022E3	3,905	1,158E3	3,143	9,4081E2	7,491
Rastrigin-(10D)	49,5634	2,299	47,8391	3,529	38,3639	8,554
Rastrigin-(5D)	12,8387	2,365	11,7329	3,133	11,4523	7,451

Nota: se muestran los resultados promedios. Con error mínimo de 1.11% y máximo infinito dependiendo de la función.

Tabla 5

Resultado de aplicar 100 iteraciones con GSA

FUNCIÓN- (dimensión)	Número de agentes					
	25		50		100	
	Mejor	Tiempo [s]	Mejor	Tiempo [s]	Mejor	Tiempo [s]
Drop -(2D)	-0,9469	2,707	-0,9359	4,421	-0,9909	12,626
Schwefel-(10D)	2,7292E+3	2,838	2,2154E+3	5,372	1,8738E+3	15,844
Schwefel-(5D)	1,0357E+3	2,762	1,0386E+3	4,924	8,5864E+2	13,923
Rastrigin-(10D)	20,8942	2,613	2,9848	5,684	3,9798	14,577
Rastrigin-(5D)	13,0161	2,530	0,81845	5,081	1,0327	13,230

Nota: se muestran los resultados promedios. Con un error mínimo de 0.5% y máximo infinito dependiendo de la función.

Tabla 6.

Resultados de aplicar 1000 iteraciones con GSA

FUNCIÓN- dimensión	Número de agentes					
	25		50		100	
	Mejor	Tiempo [s]	Mejor	Tiempo [s]	Mejor	Tiempo [s]
Drop-(2D)	-0,9870	11,195	-0,9972	31,330	-0,9996	105,248
Schweffel-(10D)	2,2669E+3	13,563	1,9437E+3	39,643	2,4400E+3	146,624
Schweffel-(5D)	1,0654E+3	12,915	7,7445E+2	36,769	7,2829E+2	128,329
Rastringin-(10D)	6,9647	12,339	1,9899	38,941	0	135,883
Rastringin-(5D)	2,9848	11,994	0,9949	36,527	0	130,190

Nota: se muestran los resultados promedios. Con un error mínimo de 0.4%.y máximo infinito dependiendo de la función.

4.5.2 Variación del número de iteraciones con criterio de parada en GSA. Para lograr una disminución en el tiempo total de ejecución del algoritmo, se plantea un método que detiene la ejecución cuando la solución no presente cambios superiores a un umbral establecido en 0.000001%. Los resultados que se muestran en la siguiente tabla representan el promedio y la desviación estándar, luego de 10 ejecuciones, para pruebas realizadas con 100 agentes y 1000 iteraciones.

Tabla 7

Resultados de aplicar criterio de parada

función	Tiempo [s]	mejor (promedio)	desviación estándar	iteraciones
Drop wave(2d)*	86,357	-0,9768	5,3394E-03	350
Schwefel(10d)*	94,532	2,4400E3	8,51554E+01	621
Rastringin(10d)	95,883	0,0133	3,1587E-05	425

*Nota: se muestran los resultados promedios. * (2d) corresponde a trabajar en dos dimensiones, y (10d) corresponde a diez dimensiones.*

4.5.3 Variación con los Kmejores en GSA: Buscando evitar que GSA quede atrapado en óptimos locales, el algoritmo fue diseñado originalmente bajo una técnica que consiste en calcular la fuerza total sobre cada masa, usando sólo los agentes con mayor desempeño. De esta forma, al inicio todos los agentes participan en el cálculo, mientras que al final sólo un agente atrae a los demás.

Para poder analizar el efecto de este comportamiento, se ejecutaron pruebas con y sin este comportamiento, indicado por la variable Kmejores. Así, Kmejores=0 implica que todos los agentes ejercen fuerza, y Kmejores=1, que sólo las partículas con mayor masa atraen hacia ella a las masas de menor tamaño. Ambos resultados fueron obtenidos para una población de 100 agentes.

Tabla 8

Resultados al variar la selección de Kmejores.

FUNCIÓN	KMEJORES=0			KMEJORES=1		
	Mejor	Desviación estándar	Tiempo [s]	Mejor	Desviación estándar	Tiempo [s]
Drop(2d)	-0,7809	4,8574E-06	11,585	-0,9902	3,9852E-6	12,463
Schewfel(5d)	9,7531E+2	3,5564E-02	14,521	8,5864E+2	3,8755E-2	13,752
Rastringin(5d)	2,0457E-1	3,2854E-04	12,930	1,0327E-1	2,9854E-04	13,120

Nota: se muestran los resultados promedios.

4.5.4 Variación del número de equipos y drones en DSO. En cada prueba se hace variar la cantidad de equipos que exploran el espacio de búsqueda y a su vez también la cantidad de drones por cada equipo, con la intención de probar en qué forma se afectan los resultados

obtenidos con cada una de estas variaciones. El procedimiento se realiza 15 veces, arrojando los siguientes resultados promedios.

Tabla 9

Resultados de tener 5 equipos y variar el número de drones por equipo.

FUNCIÓN- (dimensión)	Número de equipos: 5					
	5 drones		10 drones		100 drones	
	Mejor	Tiempo [s]	Mejor	Tiempo [s]	Mejor	Tiempo [s]
Drop -(2D)	-1	4.421703	-1	2.601488	-1.0000	0.713301
Schweffel-(10D)	1.2737E-04	4.742237	1.2753E-04	2.961756	0.0234	0.870542
Schweffel-(5D)	6.3639E-05	4.901317	6.3637E-05	2.629939	6.3662E-05	0.782424
Rastringin-(10D)	0	0.526316	0	1.083900	0.3496	0.826217
Rosenbrock-(5D)	0.0039	3.895927	0.6975	2.276299	2.4918E-05	1.083440
Rosenbrock (10D)	0.0794	4.752794	0.0029	2.557113	2.5143	0.726003

Nota: se muestran los resultados promedios.

Tabla 10

Resultados de tener 10 equipos y variar el número de drones por equipo.

FUNCIÓN- (dimensión)	Número de equipos: 10					
	5 drones		10 drones		100 drones	
	Mejor	Tiempo [s]	Mejor	Tiempo [s]	Mejor	Tiempo [s]
Drop -(2D)	-1	5.017568	-1	2.849200	-1	0.771426
Schweffel-(10D)	1.2728E-04	3.898015	1.2727E-04	3.086848	0.0026	0.701196
Schweffel-(5D)	6.3637E-05	5.341921	6.3637E-05	2.671267	3.7048E-04	0.805909
Rastringin-(10D)	0	0.906450	0	1.159555	2.1705E-06	0.949289
Rosenbrock-(5D)	0.0997	4.179192	9.8938E-08	3.189090	0.0678	0.893959
Rosenbrock (10D)	3.6756E-04	4.006040	1.4669E-09	2.579800	5.7642	0.885422

Tabla 11.

Resultados de tener 50 equipos y variar el número de drones por equipo.

FUNCIÓN- (dimensión)	Número de equipos: 50					
	5 drones		10 drones		100 drones	
	Mejor	Tiempo [s]	Mejor	Tiempo [s]	Mejor	Tiempo [s]
Drop -(2D)	-1	6.016888	-1	3.236906	-1	0.933948
Schweffel-(10D)	7.9766E-04	5.416927	1.2837E-04	3.727672	0.7121	0.921480
Schweffel-(5D)	7.0491E-05	5.199305	6.3637E-05	2.902838	0.0031	0.814885
Rastringin-(10D)	0	0.921286	0	2.018731	0	0.797286
Rosenbrock-(5D)	1.0029E-12	4.614641	0.1215	2.647327	0.0017	0.885890
Rosenbrock (10D)	3.5114E-15	5.245634	6.0275E-16	3.009112	0.0038	0.932823

Nota: se muestran los resultados promedios.

Tabla 12.

Resultados de tener 100 equipos y variar el número de drones por equipo.

FUNCIÓN- (dimensión)	Número de equipos: 100					
	5 drones		10 drones		100 drones	
	Mejor	Tiempo [s]	Mejor	Tiempo [s]	Mejor	Tiempo [s]
Drop -(2D)	-1	7.176357	-1	3.895603	-1	0.839945
Schweffel-(10D)	1.3710E-04	5.586979	1.4799E-04	3.582716	50.8585	0.929393
Schweffel-(5D)	6.3637E-05	5.082605	6.3641E-05	3.159917	1.0663	0.909146
Rastringin-(10D)	0	1.351555	0	0.794814	1.0789	0.873963
Rosenbrock-(5D)	4.9920E-28	1.103316	1.7928E-05	2.979622	0.3561	0.954294
Rosenbrock (10D)	2.9171E-04	5.044555	1.7506E-05	2.956015	6.1790	0.848926

Nota: se muestran los resultados promedios.

Análisis de las anteriores pruebas: a través de los resultados que arrojan las anteriores funciones de pruebas en las distintas dimensiones, es posible inferir que cuando aumenta el número de equipos y el número de drones por equipo, se obtiene un resultado adecuado en un menor tiempo. Sin embargo, la calidad de la respuesta, es decir el resultado satisfactorio esperado se obtiene cuando el número de drones por equipo no es tan alto, aun cuando esto significa que el tiempo de búsqueda es solo por una pequeña diferencia algo mayor.

4.5.5 Variación del número de iteraciones para actualizar el firmware Uno de los procedimientos más importantes de DSO es la actualización del firmware por parte del centro de comandos luego de un determinado número de iteraciones. De esta forma se mide la calidad de cada uno de los equipos. Se utiliza la función Rastrigin en diez dimensiones para realizar la prueba cuando tanto el número de equipos en DSO como el número de drones por equipo es igual a 10.

Tabla 13.

Resultados obtenidos sobre la función Rastrigin al variar el número de iteraciones

<i>5 iteraciones</i>		<i>10 iteraciones</i>		<i>50 iteraciones</i>		<i>100 iteraciones</i>		<i>500 iteraciones</i>	
mejor	Tiempo [s]	mejor	Tiempo [s]	mejor	Tiempo [s]	mejor	Tiempo [s]	mejor	Tiempo [s]
1.63E-04	1.8205	2.9848	3.173825	0.0056	2.684534	0	1.264319	0	2.136937

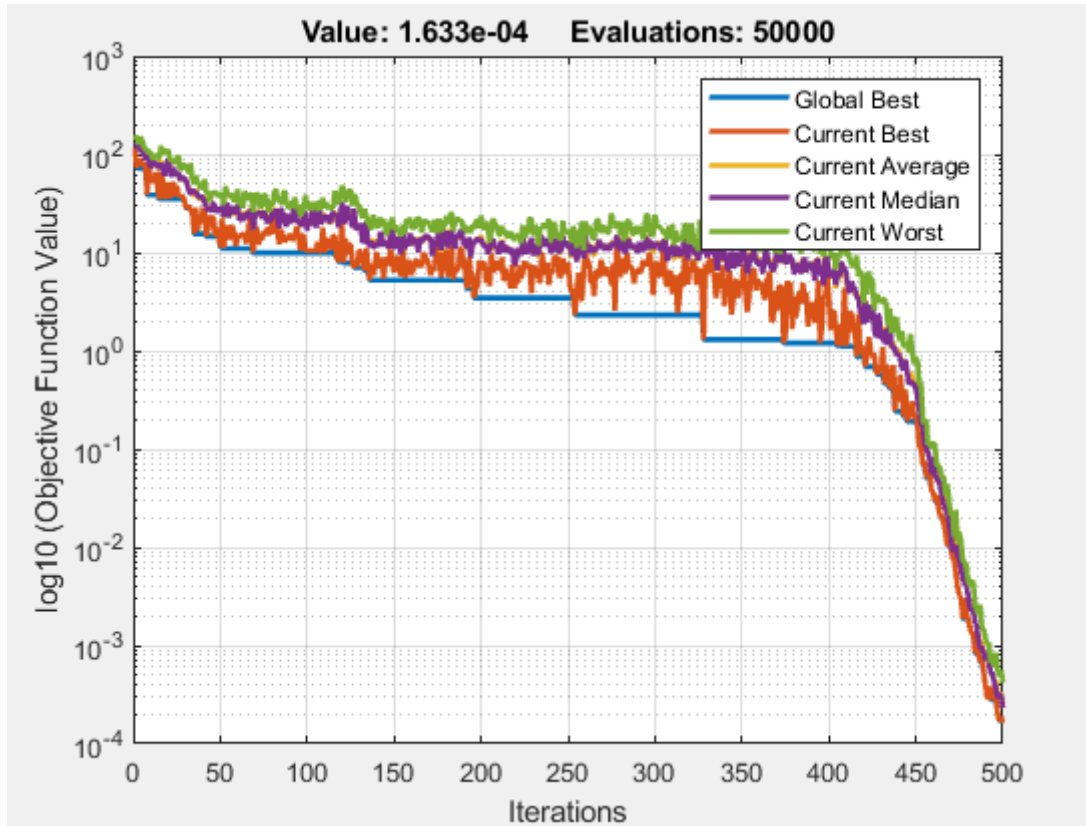


Figura 7. Al realizarse actualización del firmware cada 5 iteraciones.

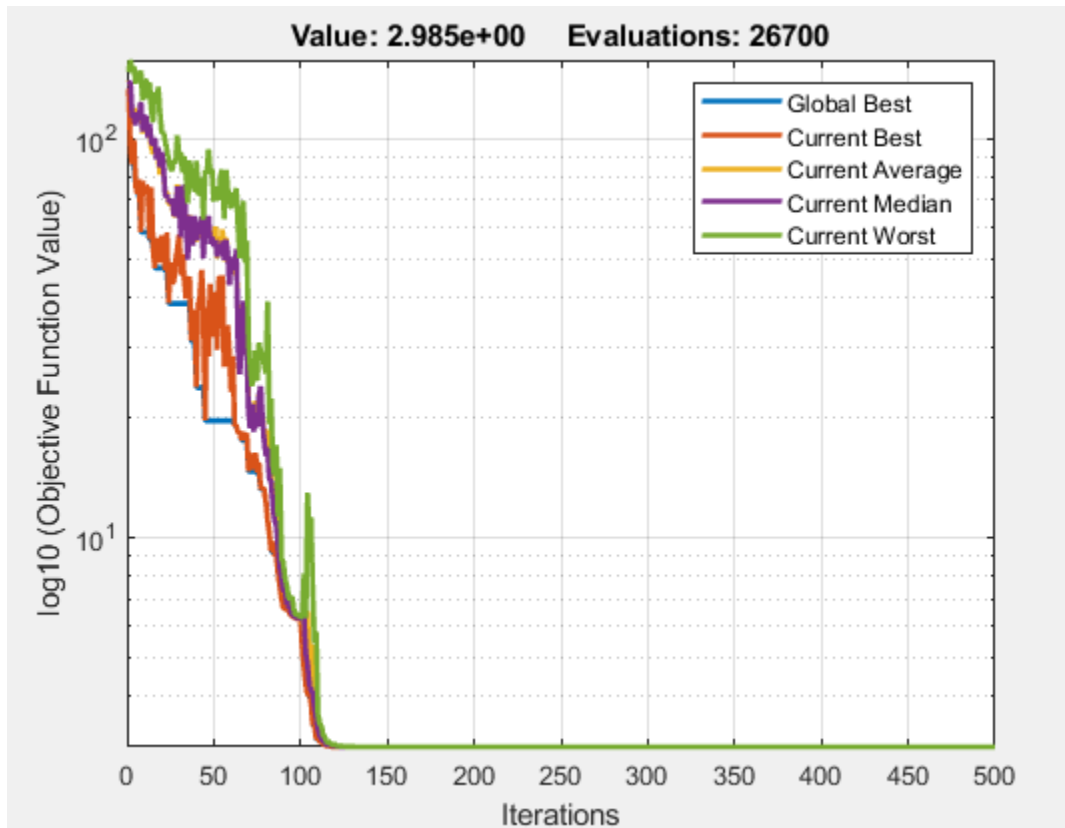


Figura 8. Al realizarse actualización del firmware cada 10 iteraciones.

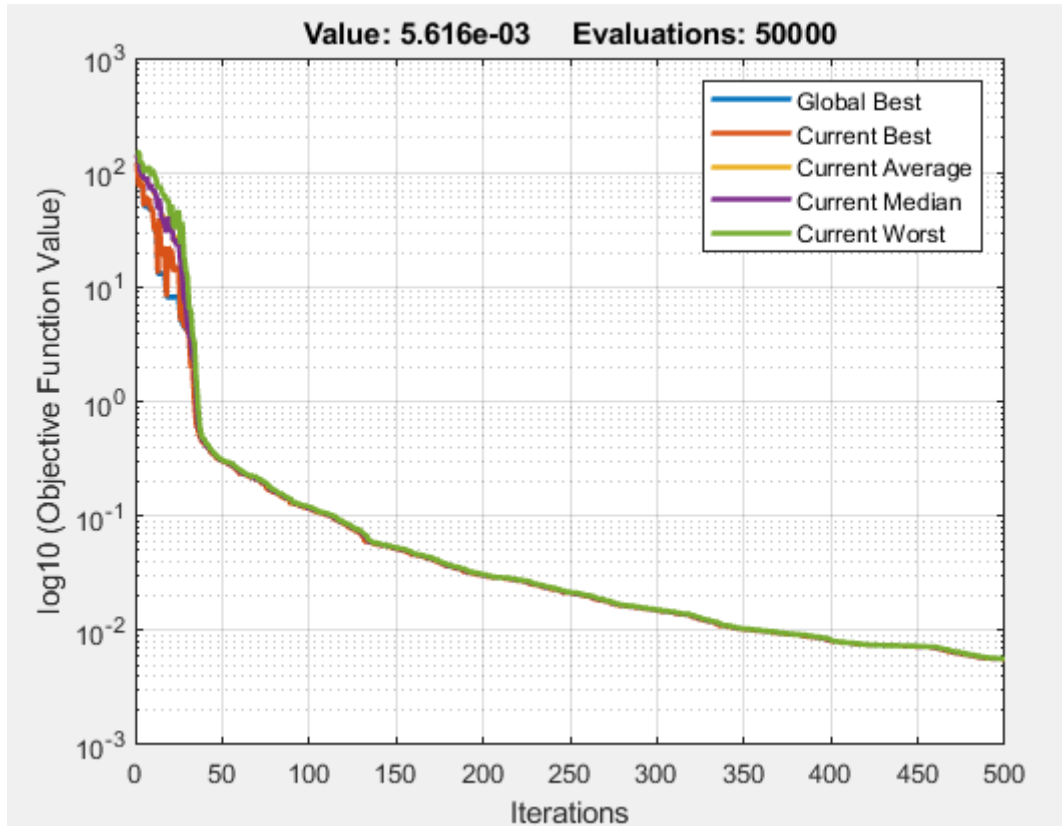


Figura 9. Al realizarse actualización del firmware cada 50 iteraciones.

Al variar el número de iteraciones para conseguir la actualización del firmware en DSO, no se presenta ningún cambio significativo en el tiempo de búsqueda ni en la calidad de la respuesta. En cambio, el número de iteraciones totales y el número de evaluaciones en cada procedimiento si se ve afectado.

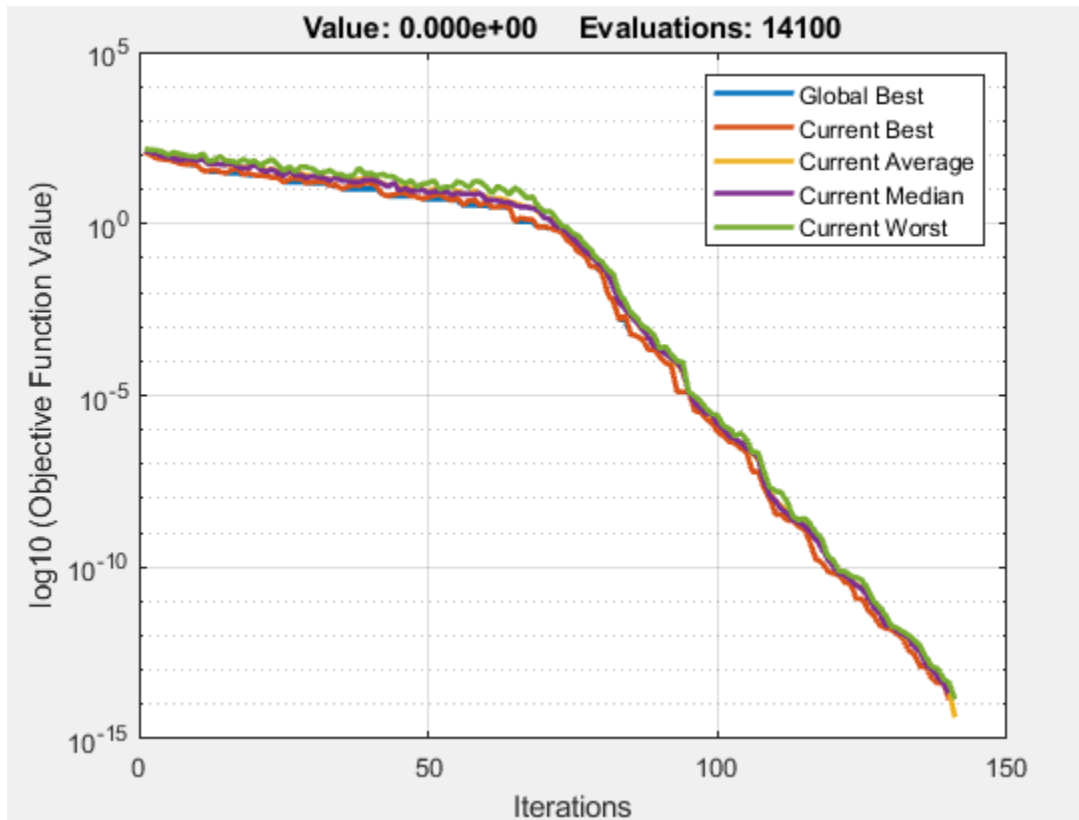


Figura 10. Al realizarse actualización del firmware cada 100 iteraciones.

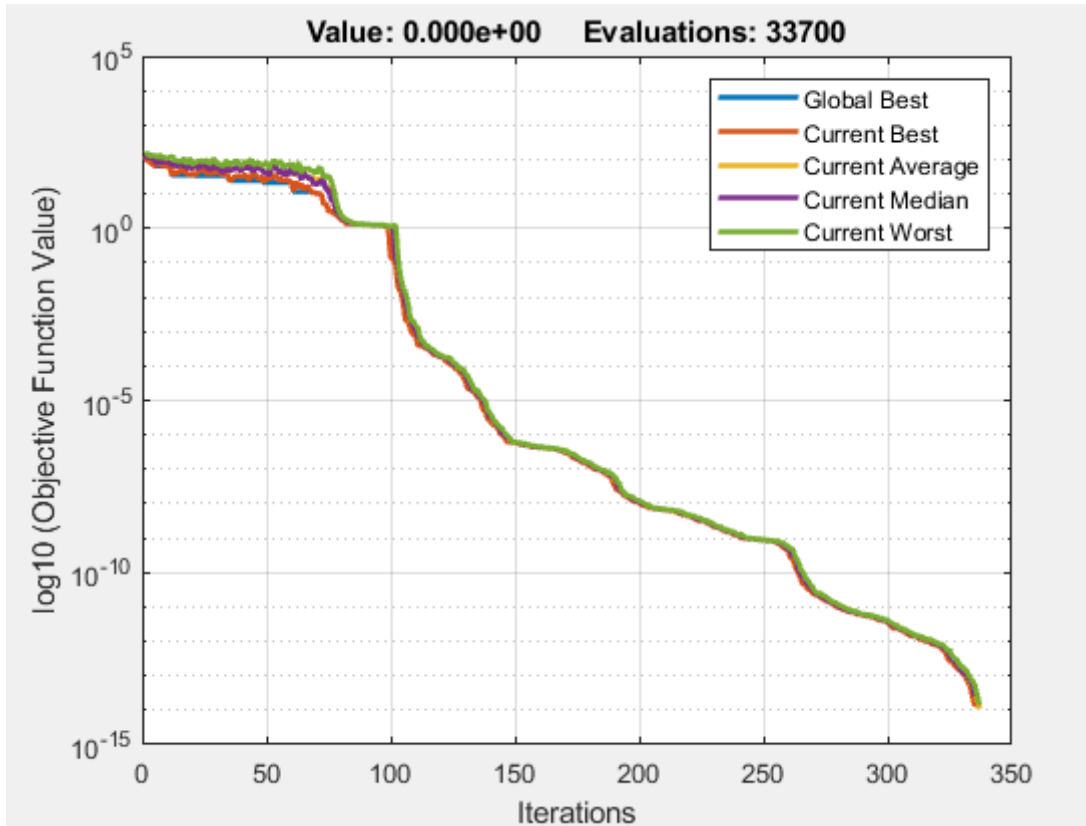


Figura 11. Al realizarse actualización del firmware cada 500 iteraciones.

5. Implementación de GSA y DSO al modelo del disipador de calor

5.1 Modelo matemático restrictivo del disipador de calor

En esta sección se presenta nuevamente el modelo matemático de la tasa de generación de entropía del disipador de calor, sujeto a las restricciones (76) -(79). Se debe tener en cuenta que el número de las restricciones no sean superiores a la dimensión de la función objetivo, ya que si esto sucede se estaría sobre-restringiendo el problema.

$$S_{gen} = \frac{Q^2}{T_a T_b} \frac{2\alpha_{hs} C_3}{L k_f C_1 C_2} + \frac{N \rho w_c H_c U_{av}^3}{T_a} C_4$$

- $2W_c - H_c \leq 1 ; W_c - W_w > 0$ (76)

- $1 * 10^{-6} < W_c < 1 * 10^{-3}$ (77)

- $1 * 10^{-6} < W_w < 1 * 10^{-3}$ (78)

- $100 * 10^{-6} < H_c < 50 * 10^{-3}$ (79)

Estas restricciones son las que le dan sentido a la implementación, específicamente al espacio de búsqueda con que contarán los algoritmos.

5.2 Parámetros del modelo

En total los parámetros definidos en el desarrollo del modelo son 14. La tabla 9 muestra los parámetros asumidos y empleados para la simulación.

Tabla 14.

Parámetros de diseño constantes asumidos para la simulación

Parámetros	Símbolo	Valores asumidos
Longitud del disipador	L (m)	0,051
Ancho del disipador	W (m)	0,051
Conductividad térmica del solido	k (W/m·K)	148
Conductividad térmica del aire	k _f (W/m·k)	0,0261
Densidad del aire	P	1,1614

Parámetros	Símbolo	Valores asumidos
	(kg/m ³)	
Calor específico del aire	C_p (J/kg·K)	1007
Difusividad térmica	α_{th} (m ² /s)	$2,216 \times 10^{-5}$
Viscosidad cinemática(aire)	ν (m ² /s)	$1,58 \times 10^{-5}$
Relación de calores específicos	γ	1,4
Numero de Prandtl (Aire)	Pr	0,71
Flujo de calor	q (W/m ²)	150000
Temperatura ambiente	T_a (K)	298
Coficiente de moento tangencial	Σ	0,85
Coficiente de energía	σ_t	0,85

Nota: estos valores están asignados de acuerdo al modelo inicial desarrollado por Khan (2009) en "Optimization of microchannel heat sinks using entropy generation minimization method"

5.3 Parámetros asignados al algoritmo GSA durante la implementación

El análisis del comportamiento del algoritmo con las diferentes funciones de prueba, cuando se varían los parámetros y los modos de funcionamiento de GSA, permite hacer la selección de

aquellos que permiten una buena implementación sobre la función objetivo en estudio. Por lo cual:

- I. *El algoritmo se detiene cuando no detecta cambios significativos en la búsqueda*
- II. *Solo los agentes con los mejores valores de masa son usados para calcular la fuerza total sobre otras masas*
- III. *El número de masas corresponde a 100 agentes.*
- IV. *El máximo de iteraciones es igual a 100.*
- V. *El valor de alfa es igual 30*
- VI. *El valor de la constante gravitacional inicial es igual a 100*

5.4 Parámetros asignados al algoritmo DSO durante la implementación

- I. Número de equipos:4
- II. Número de drones por equipo:25
- III. El centro de comandos actualiza el firmware cada 10 iteraciones
- IV. El máximo de iteraciones sin mejoras(maxEstancamiento):10
- V. Las constantes del funcionamiento de DSO: $C1=0.8$, $C2=0.012$, $C3=0.72$
- VI. La probabilidad de aceptar una pésima solución cuando la búsqueda se estanca (Pacc): 0.1
- VII. El umbral para reiniciar si hay una población convergente de acuerdo a los valores de la función objetivo(ConvThres): $1e-15$
- VIII. Valor a alcanzar (VTR): $1e-20$
- IX. Reportar cada 10 iteraciones

- X. Criterio de parada(maximaEvaluaciones):50000
- XI. Máximo número de iteraciones: 500, esto no conlleva a un posible reinicio.

5.5 Funciones de implementación

5.5.1 Función entropía: En esta función a cada variable se le asigna una posición dentro de la dimensión del vector posición denominado como X, el cual tiene un tamaño de una fila por un numero de columnas igual al número de variables a analizar (W_c , W_w , H_c , N_m). Se estipulan los parámetros definidos inicialmente para el caso específico del disipador y se calcula finalmente el valor de la generación de entropía como la suma de ambas entropías generadas tal como aparece en la ecuación (32).

5.5.2 Función límite de los espacios de búsqueda: Se comprueban los límites en el espacio de búsqueda de los agentes reiniciando aleatoriamente aquellos agentes que se encuentren fuera del espacio de búsqueda.

5.5.3 Función rango: Esta es la función que da límites y dimensión al espacio de búsqueda para la función objetivo que se trabaja específicamente. El rango para las variables de estudio se encuentra en la siguiente tabla.

Tabla 15

Rango de valores de las variables

Variable	Límite inferior	Límite superior
W_c	10E – 6 metros	1E – 3 metros
W_w	10E – 6 metros	1E – 3 metros

H_c	100E – 6 metros	50E – 3 metros
N_m	1	40

El algoritmo de búsqueda Gravitacional, se inicializa aleatoriamente con la matriz posición [X] de cada uno de los agentes, cuyo tamaño queda definido por un número de filas igual a la dimensión de la función de prueba y un número de columnas igual al tamaño de los agentes. Es decir que, para la implementación de la función de interés en este algoritmo, el tamaño de la matriz es de 4 filas con 100 columnas. También se inicia con ceros la matriz de velocidad [V] la cual tiene un número de filas igual al número de agentes y un número de columnas igual a la dimensión de la función prueba, lo cual quiere decir que la matriz velocidad tiene un tamaño de 100 filas con 6 columnas. El algoritmo de búsqueda gravitacional está diseñado para obtener los mejores valores en casos de maximización o minimización. En la implementación que se realiza para el disipador de calor, el problema es una minimización de la función objetivo, por lo cual la sentencia `min_flag` debe tener un valor a uno, indicando que se debe minimizar.

6. Generación de entropía al variar la geometría del disipador

El comportamiento de la generación de entropía y por consiguiente el análisis de la resistencia térmica en el disipador, se describe en función de cada uno de los parámetros de interés. Se determina la altura óptima del canal cuando se varía el número de Knudsen con $G=0.010$. Adicionalmente, los parámetros W_c , W_w se variaron para tratar de analizar, por medio de los resultados, el comportamiento de la generación de entropía.

En la tabla 11 se pueden apreciar los resultados cuando la diferencia entre el ancho del canal de aire y el ancho de la aleta, es mínima. Es decir, cuando ambos parámetros toman valores muy cercanos. Por un lado, dichos valores tienen una medida del orden de las micras. Por el otro, en milímetros. Naturalmente el número de aletas es una consecuencia física de estos dos parámetros por lo que en este análisis restringido no se considera encontrar un valor. El parámetro Kn se considera una relación de vital importancia para definir y variar las características físicas del fluido aire.

Tabla 16

GE cuando la diferencia de los anchos es mínima.

$W_c = 1.2\mu\text{m}$			$W_c = 1\text{mm}$		
$W_w = 1\mu\text{m}$			$W_w = 0.98\text{mm}$		
Kn	Hc [μm]	S [W/K]	Kn	Hc [μm]	S [W/K]
0.001	50000	280.131	0.001	100	69.306
0.1	49000	295.252	0.1	100	70.201
1	50000	1168.319	1	100	69.903
10	49000	3615.099	10	100	70.683

La tabla 12 contiene los resultados de la prueba cuando la diferencia entre el ancho de ambos parámetros es mayor.

Tabla 17.

GE cuando la diferencia de los anchos es máxima

$W_c = 1\text{mm}$		$W_c = 1\text{mm}$	
$W_w = 1\mu\text{m}$		$W_w = 600\mu\text{m}$	

Kn	Hc [μm]	S [W/K]	Kn	Hc [μm]	S [W/K]
0.001	50000	2.685	0.001	1400	9.238
0.1	11600	3.595	0.1	1300	8.476
1	2300	4.768	1	900	6.938
10	1200	3.486	10	600	5.633

Al analizar los resultados hasta aquí obtenidos, se puede concluir que la altura presenta una disminución cuando aumenta el número de Knudsen, y que en algunos casos esto también significa una reducción o aumento de la generación de entropía. Los resultados que se muestran en la siguiente tabla corresponden a casos cuando $W_c < W_w$. Se aprecia que en los resultados del proceso la generación de entropía resulta negativa.

Tabla 18

GE cuando el ancho del canal es menor que el de la aleta

$W_c = 1\mu\text{m}$			$W_c = 1\mu\text{m}$		
$W_w = 600\mu\text{m}$			$W_w = 1\text{mm}$		
Kn	Hc [μm]	S [W/K]	Kn	Hc [μm]	S [W/K]
0.001	100	-3.194E19	0.001	100	-9.0121E19
0.1	100	-9.527E18	0.1	100	-4.72713E19
1	100	-1.638E18	1	100	-5.3457E18
10	100	-1.226E17	10	100	-6.4070E17

Hasta los resultados aquí mostrados se puede concluir que, en esta prueba, la entropía generada se minimiza cuando el ancho del canal por donde circula aire es mucho mayor que el ancho de la aleta, y ésta tiene la altura máxima posible cuando la relación de Knudsen es mínima.

6.1 Resultados con DSO de la entropía al variar los límites de la geometría

Al aplicar el algoritmo DSO sobre la función objetivo cuando se hace variar el rango de búsqueda de las variables geométricas (W_c y W_w), se obtienen los siguientes resultados fijando la velocidad de flujo volumétrico en un valor bajo (0.002) y uno de mayor valor (0,014), a modo de prueba sólo estos dos valores. Los valores que se muestran en las siguientes tablas corresponden al promedio de haber realizado el procedimiento de búsqueda por 10 veces.

Tabla 19.

Variación de los límites de búsqueda para $Kn=0.001$

$Kn=0.001$					
<i>Límites de búsqueda</i>	G [m ³ /s]	W_c [μm]	W_w [μm]	Hc [μm]	S_{gen} [W/K]
[1μm-1mm]	0,002	1000,000	1,000	619,029	2,204
	0,014	1000,000	1,000	50000,000	2,668
[30μm-1mm]	0,002	1000,000	30,000	551,995	2,289
	0,014	279,549	1000,000	50000,000	2,872
[60μm-1mm]	0,002	1000,000	60,000	530,417	2,322
	0,014	140,743	924,530	12796,147	2,872
[100μm-1mm]	0,002	1000,000	100,000	502,418	2,370
	0,014	354,267	603,667	7356,420	3,037

Análisis de los datos: No registra el menor valor de entropía. En cambio, se encuentra registrado el mayor valor de altura del canal posible hallado en todas las tablas. En la mayor parte de los resultados casi siempre W_c es mayor a W_w y coincide que donde no es así es porque la velocidad de flujo tiene el valor más alto.

Tabla 20.

Variación de los límites de búsqueda para $Kn=0.1$

Kn=0.1					
<i>Límites de búsqueda</i>	G [m ³ /s]	W_c [μm]	W_w [μm]	H_c [μm]	S_{gen} [W/K]
[1μm-1mm]	0,002	1000,000	1,000	557,517	2,153
	0,014	1000,000	1,000	21866,728	3,630
[30μm-1mm]	0,002	1000,000	30,000	515,814	2,219
	0,014	57,426	992,718	41102,021	4.2992
[60μm-1mm]	0,002	1000,000	60,000	496,924	2,255
	0,014	533,814	976,286	27126,208	4.0552
[100μm-1mm]	0,002	1000,000	100,000	472,109	2,305
	0,014	273,116	286,462	11121,373	4,419

Análisis de datos: Cuando el flujo volumétrico es mínimo, la generación de entropía es menor. Al aumentar la velocidad de flujo volumétrico aumenta la generación de entropía y no en todos los casos se cumple que el ancho del canal es mayor al ancho de la aleta.

Tabla 21.

Variación de los límites de búsqueda para $Kn=1$

Kn=1					
-------------	--	--	--	--	--

Limites	G [m ³ /s]	Wc [μm]	Ww [μm]	Hc [μm]	S_{gen} [W/K]
[1μm-1mm]	0,002	1000,000	1,000	404,614	1.982
	0,014	1000,000	1,000	3509,357	5.891
[30μm-1mm]	0,002	1000,000	30,000	390,408	2.023
	0,014	1000,000	30,000	2976,898	6.621
[60μm-1mm]	0,002	1000,000	60,000	378,128	2.063
	0,014	1000,000	60,000	2883,828	6.680
[100μm-1mm]	0,002	1000,000	100,000	361,688	2.120
	0,014	1000,000	100,000	2765,067	6.7520

Análisis de datos de esta tabla: Se encuentra el valor más alto de generación de entropía.

Tabla 22.

Variación de los límites de búsqueda para Kn=10

Kn=10					
Limites	G [m ³ /s]	Wc [μm]	Ww [μm]	Hc [μm]	S_{gen} [W/K]
[1μm-1mm]	0,002	1000,000	1,000	246,100	1,735
	0,014	1000,000	1,000	1762,839	4,351
[30μm-1mm]	0,002	1000,000	30,000	240,295	1,776
	0,014	1000,000	30,000	1713,520	4,432
[60μm-1mm]	0,002	1000,000	60,000	234,234	1.820
	0,014	1000,000	60,000	1670,780	4.497
[100μm-1mm]	0,002	1000,000	100,000	225,932	1.883
	0,014	1000,000	100,000	1423,547	4,125

Análisis de datos: Contiene el menor de valor encontrado de generación de entropía. En todos los resultados W_c es mayor a W_w , y se ajustan a los límites máximos y mínimos posibles respectivamente.

7. Resultados de optimizar el canal

La combinación de los parámetros (H_c , W_c y W_w) que minimiza la generación de entropía para un flujo de aire y un valor de Knudsen (K_n) determinados, corresponderá al micro-canal óptimo.

7.1 Resultados obtenidos con GSA

En la siguiente tabla se plasman los resultados obtenidos mediante el algoritmo GSA, y en donde sólo se considera la generación de entropía como el resultado que produce únicamente la transferencia de calor cuando la temperatura ambiente es 27°C.

Tabla 23.

Resultados con GSA

K_n	G [m ³ / s]	H_c [μm]	W_c [μm]	W_w [μm]	N_m	S_{gen} [W/K] S_{genht}	Tiempo de cómputo total [s]
0,001	0,002	152,425	728,3568	402,9715	19,824	1.1375	6.575
	0,006	442,503	822,5814	482,3832	22,616	1.0912	6.574
	0,010	17574,251	992,8392	79,3161	20,072	1.0238	7.233

K_n	G [m ³ / s]	H _c [μm]	W _c [μm]	W _w [μm]	N _m	S _{gen} [W/K] S _{genht}	Tiempo de cómputo total [s]
0,1	0,014	24892,196	992,8335	117,9032	21,194	0.9607	7.288
	0,002	327,2488	766,4482	349,7304	16,702	1.1747	7.463
	0,006	191,4207	703,1771	412,7161	18,499	1.0537	7.615
	0,010	21745,664	996,4034	63,7442	19,165	1.0304	7.138
1	0,014	19893,9050	759,6831	575,7159	20,955	0.9661	7.093
	0,002	207,4508	903,2078	432,6325	19,700	1.2145	7.160
	0,006	110,0775	566,2470	391,7160	25,889	1.1148	7.090
	0,010	105,5552	318,9976	151,9299	6,6123	1.1318	7.721
10	0,014	46955,4770	834,8655	204,5351	21,272	1.1012	7.333
	0,002	112,0425	933,6750	888,3500	20,787	1.2508	7.577
	0,006	45016,2797	422,5301	38,2643	20,922	1.2439	8.237
	0,010	43805,816	344,7363	23,8978	36,539	1.2229	7.268
	0,014	46019,995	364,0327	41,8851	22,507	1.2021	7.772

7.2 Resultados obtenidos con DSO

En la siguiente tabla se encuentran los resultados obtenidos con DSO.

Tabla 24.

Resultados con DSO

K_n	G [m ³ /s]	H _c [μm]	W _c [μm]	W _w [μm]	N _m	S _{gen} [W/K]	Tiempo de cómputo total [s]
0,001	0,002	502,4187	1000	100	20,9349	2.2041	2.7679
	0,006	2961,7243	952,4618	956,8644	15,8143	3.0579	3.2786
	0,010	45962,9542	158,9091	886,5663	23,2957	2.9796	3.0902

K_n	G [m ³ /s]	H_c [μm]	W_c [μm]	W_w [μm]	N_m	S_{gen} [W/K]	Tiempo de cómputo total [s]
0,1	0,014	43540,5231	474,1546	836,9823	16,5789	2.7702	3.7794
	0,002	473,2484	1000,000	100,000	22,1789	2,4158	3,7741
	0,006	715,1253	914,368	111,025	17,4937	3,7844	3,5478
	0,010	912,5476	851,236	221,569	23,7587	3,5789	3,1498
1	0,014	11121,373	273,116	286,462	18,7453	4,419	3,5479
	0,002	361,688	1000,000	100,000	19,6987	2.120	3,6974
	0,006	569,7259	1000,0000	100,0000	17,369	3,6547	3,8749
	0,010	1258,32879	1000,0000	100,0000	19,5478	4,6578	2,3647
10	0,014	2698,2547	1000,0000	100,0000	21,3698	6.7520	3,2987
	0,002	225,9876	1000,0000	100,0000	22,6987	2,6987	3,9872
	0,006	368,8974	1000,0000	100,0000	17,895	3,6987	3,7843
	0,010	1123,6879	1000,0000	100,0000	18,9654	2,6947	4,6987
	0,014	1521,547	1000,0000	100,0000	21,6984	4,3698	2,9873

7.3 Resultados de referencia obtenidos con PSO

En la investigación realizado por Hinojosa Ferreira y Espinosa Quintero (2011), acerca de este mismo diseño para optimización utilizando el algoritmo PSO, se obtuvieron los resultados que se muestran en la tabla 16.

Tabla 25.

Resultados con PSO

K_n	G [m ³ /s]	H_c [μm]	W_c [μm]	W_w [μm]	N_m	S_{gen} [W/K]	Tiempo de cómputo total [s]
-------	----------------------------	---------------	---------------	---------------	-------	--------------------	-----------------------------------

K_n	G [m ³ /s]	H_c [μm]	W_c [μm]	W_w [μm]	N_m	S_{gen} [W/K]	Tiempo de cómputo total [s]
0,001	0,002	745,4256	372,7128	1000,0000	17,8479	1,0878	150,0242
	0,006	1652,1000	826,0449	1000,0000	13,4170	1,7973	106,6397
	0,010	2000,0000	1000,0000	952,5127	12,5723	2,3228	166,4824
	0,014	2000,0000	1000,0000	1,0000	25,4735	2,5136	172,3132
0,1	0,002	741,5568	370,7784	1000,0000	17,8731	1,1156	173,9672
	0,006	1653,8000	826,9038	1000,0000	13,4107	1,8560	209,3332
	0,010	2000,0000	1000,0000	966,4539	12,4760	2,3867	100,0445
	0,014	2000,0000	1000,0000	1,0000	25,4735	2,8838	344,1369
1	0,002	738,0853	369,0427	999,9999	17,8957	1,1808	239,5358
	0,006	1663,8000	831,8956	1000,0000	13,3741	1,9388	269,1816
	0,010	2000,0000	1000,0000	987,9142	12,3306	2,4690	153,9681
	0,014	2000,0000	1000,0000	525,7840	16,3681	3,0938	304,0764
10	0,002	739,7338	369,8669	1000,0000	17,8849	1,2034	91,6821
	0,006	1668,3000	834,1615	1000,0000	13,3576	1,9579	113,8752
	0,010	2000,0000	1000,0000	993,0413	12,2963	2,4868	118,7487
	0,014	2000,0000	1000,0000	529,7567	16,3230	3,1174	295,8015

8. Análisis comparativo

A pesar de que tanto en GSA como en PSO, la optimización es obtenida por agentes moviéndose en el espacio de búsqueda; la estrategia de movimiento es diferente. Para calcular la dirección de un agente en PSO se necesita conocer únicamente tres posiciones: la posición actual del agente, la mejor posición que ese agente ha encontrado, y la mejor posición que ha encontrado todo el

enjambre. En cambio, para calcular la dirección de un agente en GSA se requiere conocer la fuerza total ejercida por todos los agentes. Teniendo en cuenta que esta fuerza es proporcional a los valores de la función objetivo, se puede analizar que en GSA para el momento de realizar la actualización de los datos se están considerando las posibles soluciones. DSO es un algoritmo diseñado y propuesto para no quedar atrapado en óptimos locales, y debido a la fuente de inspiración de este algoritmo, y a la capacidad auto adaptativa del mismo, es posible conseguir un resultado global o un valor óptimo global que se ajusta al requerimiento del problema. Se puede verificar que DSO es quien requiere mucho menor tiempo de ejecución para encontrar una respuesta, cuando se compara con el algoritmo GSA y PSO.

9. Conclusiones

Este documento mostró la solución a un problema de diseño geométrico en la microelectrónica, implementando dos algoritmos metaheurísticos para optimización: GSA y DSO. Adicionalmente, y con fines de comparación, se tomaron datos reportados por el trabajo de optimización que utilizó el algoritmo PSO. En GSA, y a pesar del análisis restringido que se le da a los agentes respecto a su masa, los datos indican que el algoritmo tiene aplicaciones prácticas para encontrar soluciones no globales en menor tiempo que PSO. Por otra parte, los resultados obtenidos confirman una mejor eficiencia utilizando el algoritmo DSO para obtener optimización global numérica, debido precisamente a la flexible técnica de desarrollo como un algoritmo evolutivo que va adaptando sus parámetros durante el proceso de optimización. En

DSO cada equipo utiliza un desplazamiento y perturbaciones diferentes, lo que les permite generar algo similar a una especie de código propio. Al comparar contra GSA se encuentra que la implementación con DSO produce mejores soluciones para el problema del disipador de calor, en términos de tiempo de respuesta y convergencia a la solución. Al utilizar DSO sí es posible encontrar un valor óptimo para el diseño del disipador considerando ambos aportes de generación de entropía: la que se produce por el proceso de transferencia de calor y la que se tiene en cuenta por la fricción del fluido en cuestión.

Respecto al diseño óptimo del disipador, se puede concluir que la operación de refrigeración que minimiza la generación de entropía se consigue, en la mayoría de las condiciones que se analizaron, cuando el ancho del canal por donde circula el aire es mucho mayor que el ancho de la aleta(diez veces), pero en concordancia con un alto número de aletas de acuerdo al modelo propuesto por Khan, que define a este variable como un resultado también probabilístico y no como resultado de un análisis físico. Es decir, que de forma ideal se obtiene una óptima refrigeración cuando la aleta del disipador se le trata como una superficie y no como un volumen. En teoría este planteamiento necesita mayores pruebas de verificación que pueden ser adelantadas en trabajos posteriores, teniendo en cuenta las técnicas de fabricación disponibles y un estudio riguroso de los materiales que permita asegurar una larga vida útil del dispositivo.

Referencias Bibliográficas

Atkins. (1999). *Química Física (6ª edición)*. Barcelona : Omega.

Bejan, A., (1982). *Entropy Generation through Heat and Fluid Flow*, Wiley, New York

Bejan, A., (1996). *Entropy generation minimization: the new thermodynamics of finite-size devices and finite-time processes*, J. Appi. Phys., voi. 79, pp. 1191-1218

Ferreira, H., & Quintero, E. (2011). “*El método de enjambre de partículas y el criterio de mínima entropía en el diseño óptimo de un dissipador de calor*”. Trabajo de grado EL21976/ para obtener título de ingeniero electrónico UIS.

Khan, Culham, Yovanovich,(2009). *Optimization of microchannels heat sinks using entropy generation minimization method*, IEEE Transactions on components and packaging technologies, vol. 32, no. 2

Khan. (2007). *The influence of material properties and spreading resistance in the Thermal Design of Plate Fin Heat Sinks*.

Poli R, L. W. (s.f.). *Extending particle swarm optimisation via genetic programming*. In: Proceedings of the 8th European conference on genetic programming, EuroGP'05. Springer, Berlin, pp 291-300.

Poli R, L., & Holland, W. O. (2005). *Extending particle swarm optimisation via genetic programming*. Proceedings of the 8th European conference on genetic programming, EuroGP'05. Springer, Berlin, 291-300.

Rashedi, Nezamabadi, Saryazdi, (2009) “*GSA: A Gravitational Search Algorithm*”. Information Sciences, vol. 179,, 2232-2248.

Wolfgang Banzhaf y Vinícius Veloso de Melo, (2017). *Drone Squadron Optimization: a Self-adaptative Algorithm for Global Numerical Optimization*. Neural Comput & Applic, DOI: 10.1007/s00521-017-2881-3

Apéndices

Apendice A. Resultados de variar el tamaño de los agentes en GSA

Función de prueba Drop Wave (2D)**Resultados de aplicar 50 iteraciones**

25 agentes	50 agentes	100 agentes
mejor	mejor	mejor
-0.9367	-0.9741	-0.9997
-0.9871	-0.9974	-0.9974
-0.9972	-0.9784	-0.9976
-0.9783	-0.9147	-0.9874
-0.9945	-0.9971	-0.9874
-0.9754	-0.6469	-0.9871
-0.9654	-0.9741	-0.9987
-0.9478	-0.9815	-0.9945
-0.9872	-0.9713	-0.9974
-0.9874	-0.9125	-0.9971
-0.9754	-0.9974	-0.9745
-0.8745	-0.9635	-0.9657
-0.9871	-0.9874	-0.9874
-0.9771	-0.9971	-0.9657
-0.9874	-0.9945	-0.9871

Resultados de aplicar 100 iteraciones

25 agentes	50 agentes	100 agentes
<i>mejor</i>	<i>mejor</i>	<i>mejor</i>
-0.9645	-0.9354	-0.9874
-0.9487	-0.9635	-0.9987
-0.9654	-0.9354	-0.9745
-0.9745	-0.9354	-0.9987
-0.9621	-0.93547	-0.98214
-0.9871	-0.9635	-0.98714
-0.9642	-0.9647	-0.9785
-0.9647	-0.9214	-0.9802
-0.9871	-0.9715	-0.9981
-0.9247	-0.9715	-0.9914
-0.9612	-0.9524	-0.9324
-0.9418	-0.9354	-0.9987
-0.9715	-0.9871	-0.9871
-0.9615	-0.9871	-0.9587
-0.9625	-0.9547	-0.9216

Resultados de aplicar 1000 iteraciones

25 agentes	50 agentes	100 agentes
<i>mejor</i>	<i>mejor</i>	<i>mejor</i>
-0.9852	-0.9824	-0.9963
-0.9815	-0.9952	-0.9925
-0.9814	-0.9935	-0.9971
-0.9914	-0.8419	-0.9924
-0.9987	-0.9941	-0.9845
-0.9815	-0.9915	-0.9985
-0.9321	-0.9918	-0.9987
-0.9354	-0.9364	-0.9925
-0.9715	-0.9987	-0.9983
-0.9982	-0.9815	-0.9936
-0.9687	-0.9872	-0.9941
-0.9971	-0.9812	-0.9936
-0.9245	-0.9958	-0.9991
-0.9245	-0.9145	-0.9998
-0.9957	-0.9998	-0.9325

Función de prueba Rastringin (5D)*Resultados de aplicar 50 iteraciones*

25 agentes	50 agentes	100 agentes
Mejor	Mejor	Mejor
12,8745	12,3698	12,6587
12,9635	12,2698	11,4789
12,3478	10,3698	11,5874
12,5698	11,2365	11,6587
12,5436	11,2369	11,2547
12,3749	10,3698	11,3287
12,3574	11,5874	11,6587
12,3674	11,9872	11,3641
12,9723	11,3997	11,2587
12,3698	10,3698	11,2398
12,8547	12,8971	11,3614
12,8963	12,3698	11,4529
12,8741	11,5478	11,2369
12,5369	11,2547	11,4789
12,3657	10,8769	11,5812
12,6871	11,5631	11,3698

Resultados de aplicar 100 iteraciones

25 agentes	50 agentes	100 agentes
<i>mejor</i>	<i>mejor</i>	<i>mejor</i>
14,8952	0,6987	0,9941
13,5874	0,8745	2,6598
13,5841	0,2398	0,5698
13,5981	0,2396	0,9951
13,5874	0,5798	0,2369
12,5484	0,9987	1,5987
13,5987	0,9518	0,9851
13,5987	0,5479	0,1254
13,9871	0,2687	0,3687
13,8745	0,9818	0,3689
12,6987	0,9248	0,2356
13,5874	0,8183	3,6871
13,5874	0,8145	0,2687
13,5478	0,8745	1,2387
13,5874	0,5987	2,5876
13,1498	0,1898	1,5984
13,6981	0,8795	0,5681
13,9814	0,4598	2,5984

Resultados de aplicar 50 iteraciones

Resultados de aplicar 1000 iteraciones

25 agentes	50 agentes	100 agentes
<i>mejor</i>	<i>mejor</i>	<i>mejor</i>
2,3654	1,6584	0,0081
2,9984	1,5741	0,0001
3,9819	1,2358	0,0014
1,6659	1,1158	0,0079
2,3698	0,5984	0,0098
2,5874	0,9871	0,0104
3,5874	1,1157	0,0068
2,5984	0,9871	0,0057
1,3685	0,9974	0,0014
2,5874	0,9814	0,0015
2,9981	0,2584	0,3651
2,5874	1,2369	0,0259
3,5447	0,9871	0,0023
2,5874	0,2654	0,1025
2,4792	2,5981	0,3600
2,4795	1,3658	0,0034

25 agentes	50 agentes	100 agentes
<i>mejor</i>	<i>mejor</i>	<i>mejor</i>
48,6987	47,5698	39,4512
49,5213	47,8384	38,4515
48,7842	47,5698	41,2368
47,5896	48,5236	35,1257
47,2368	49,1236	32,1498
48,9634	47,8965	39,4187
47,2541	47,5214	37,5498
48,2369	47,2369	39,1495
47,5412	47,5148	40,2387
49,8725	47,2589	38,1496
41,236	47,1298	35,4719
47,5963	49,4587	36,9871
47,8569	51,2364	36,2145
49,3687	47,5691	37,1294
50,2387	48,2198	39,4587
51,4736	49,2147	41,325
45,1267	52,3647	39,8745

Función de prueba Rastringin (10d)

Apéndice B. Código en Matlab de la Función de Prueba Drop-Wave

Ecuación

$$f(x) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$$

Código en Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% INPUT:
%
% xx = [x1, x2]
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x1 = xx(1);
x2 = xx(2);

frac1 = 1 + cos(12*sqrt(x1^2+x2^2));
frac2 = 0.5*(x1^2+x2^2) + 2;

y = -frac1/frac2;

end
```

Apéndice C. Código en matlab de la función de prueba Schwefel

Ecuación

$$f(x) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|})$$

(40)

Código en Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% INPUT:
%
% xx = [x1, x2, ..., xd]
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

d = length(xx);
sum = 0;
for ii = 1:d
    xi = xx(ii);
    sum = sum + xi*sin(sqrt(abs(xi)));
end

y = 418.9829*d - sum;

end
```

Apéndice D. Código en matlab de la función de prueba Rastrigin

Ecuación

$$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$$

(41)

Código en Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% INPUT:
%
% xx = [x1, x2, ..., xd]
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [y] = rastr(xx)

d = length(xx);
sum = 0;
for ii = 1:d
    xi = xx(ii);
    sum = sum + (xi^2 - 10*cos(2*pi*xi));
end

y = 10*d + sum;

end
```

Apéndice E. Código en matlab de la función de prueba Rosenbrock

Ecuación

$$f(\mathbf{x}) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

(42)

Código en matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%  
% INPUT:  
%  
% xx = [x1, x2, ..., xd]  
%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
d = length(xx);  
sum = 0;  
for ii = 1:(d-1)  
    xi = xx(ii);  
    xnext = xx(ii+1);  
    new = 100*(xnext-xi^2)^2 + (xi-1)^2;  
    sum = sum + new;  
end  
  
y = sum;  
  
end
```

Apéndice F. Código en matlab de la función entropía

```

% Input: Vector de variables
%{
X(1) = Wc;
X(2) = Ww;
X(3) = Hc;
X(4) = Nm;
%}
% Output: The objective function value

function S = entropiaDSO( X )
    %Función de entropía

% Dimensiones de X = 1 x Numero de variables

Wc=X(1);Ww=X(2);Hc=X(3);Nm=X(4);

%G=X(4);Kn=X(5);Nm=X(6);

while ((2*Wc)-Hc)>1 || (Wc<Ww)
    Wc=(1e-3-1e-6)*rand(1)+1e-6;
    Hc=(50e-3-100e-6)*rand(1)+100e-6;
    Ww=(1e-3-1e-6)*rand(1)+1e-6;
end

    %*****Parámetros ya definidos*****
% L: Longitud del disipador
% W: Ancho del disipador
% k: Conductividad térmica del solido
% kf: Conductividad termica del aire
% P: Densidad del aire
% Cp: Calor especifico del aire
% ath=Difusividad térmica
% v: Viscosidad cinemática
% y: Relación de calores específicos: Cp/Cv=>Cv=1007/1.4=719
% Pr: Numero de Prandtl (Aire)
% q: Flujo de calor
% Ta: Temperatura ambiente
% Ct: Coeficiente de momento tangencial
% Ce: Coeficiente de energia

L=0.051;W=0.051;k=148;kf=0.0261;P=1.1614;ALFath=2.216*10^-5;v=1.58*10^-5;y=1.4;
Pr=0.71;q=150000; Cmt=0.85;Ce=0.85;Ta=298;Kn=1;G=0.006;

```

```

%*****Entropia Sgenht*****

Dh=4*Wc;

Nm=(W-(2*Ww))/(2*(Wc-Ww));

ALFAhs=L/(2*Wc);
ALFAC=(2*Wc)/Hc;
C2=(1+ALFAC)/ALFAC;
Beta=Wc/Ww;
keq=kf/k;
mm=P*G;
Uav=mm/(Nm*P*2*Wc*Hc);
PeDh=(Dh*Uav)/ALFAth;
Zu=(2-Cmt)*(Kn/Cmt);
ALFA=(2*Zu)/(1+ALFAC);
Us=(6*ALFA*Uav)/(1+(6*ALFA));
Zt=((2-Ce)/Ce)*((2*y)/(y+1))*(Kn/Pr);
Ks=1-((6*Us)/(17*Uav))+((2/51)*(Us/Uav)^2)+((140*Zt)/(17*(1+ALFAC)));
Nudh=140/(17*(1+ALFAC)*Ks);
hav=(Nudh*kf)/Dh;

m = sqrt((2*hav)/(k*Ww));

m=((1/ALFAC)*sqrt(2*Beta*keq*Nudh*(1+ALFAC)))/Hc;

nfin=(tanh(m*Hc))/(m*Hc);

A=L*W;

A = 2*Nm*L*(Wc+(Hc*nfin));
Q=q*A;
C1=Nm*ALFAhs*((2*nfin)+ALFAC);
C3=(1/Nudh)+(C1/PeDh);
Rth=(2*C3*ALFAhs)/(kf*L*C1*C2);
Tb=Ta+(Q*Rth);
Sgenht=((Q^2)*Rth)/(Ta*Tb);

%***** Entropia Sgenff*****
kce=1.79-((2.32*Wc)/(Wc+Ww))+0.53*((Wc)/(Wc+Ww))^2;
Redh=Dh/(Uav*v); %Dh/Uav*v
f=(24/Redh)*(1/(1+(6*ALFA)))*(1/(1+ALFAC));
C4=kce+(f*L/Dh);
deltaP=(P*C4*(Uav)^2)/2;
Sgenff=(mm*deltaP)/(P*Ta);

%*****Finalmente cálculo de entropía*****

S=Sgenht+Sgenff;
end

```

Apéndice G. Resultados numéricos de variar el número de drones y equipos en DSO con las funciones de prueba

- *Función de prueba Schwefel (5d)*

5 equipos			
5 drones		100 drones	
<i>mejor /solución</i>	<i>tiempo [s]</i>	<i>mejor/ solución</i>	<i>tiempo[s]</i>
6.6343e05/	4.197339	6.6462e-05/	1.097630
[4.209689711374229e+02		[4.209711063143641e+02	
4.209641212583959e+02		4.209708869683957e+02	
4.209687495833802e+02		4.209715879057598e+02	
4.209687523276226e+02		4.209667230388243e+02	
4.209687616583509e+02]		4.209690076705388e+02]	
6.3637e-05/	4.414758	0.0067/	0.683715
[4.209687461810516e+02		[4.211123051906563e+02	
4.209687425251051e+02		4.208039690455525e+02	
4.209687451177753e+02		4.209728155683260e+02	
4.209687389780654e+02		4.209978648012774e+02	
4.209687474493466e+02]		4.209030022099886e+02]	
6.3637e-05/	4.099062	7.9190e-05/	0.653373
[4.209687460986463e+02		[4.209750542995944e+02	
4.209687474293223e+02		4.209604877097082e+02	
4.209687469455539e+02		4.209713456057601e+02	
4.209687478535220e+02		4.209664466123697e+02	
4.209687464882175e+02]		4.209669538959696e+02]	
6.3637e-05/	3.803911	6.3639e-05/	0.627115

[4.209687468176809e+02		[4.209687465735632e+02	
4.209687455008662e+02		4.209687779804240e+02	
4.209687455090572e+02		4.209688166935783e+02	
4.209687456413399e+02		4.209686896382945e+02	
4.209687458328686e+02]		4.209686732471462e+02]	
6.3637e-05/	4.406766	6.3648e-05/	0.763733
[4.209687467595402e+02		[4.209688658025282e+02	
4.209687471289904e+02		4.209687115950354e+02	
4.209687454156854e+02		4.209688102230263e+02	
4.209687462611359e+02		4.209685113677781e+02	
4.209687467745847e+02]		4.209686545905238e+02]	
6.3638e-05/	4.464967	6.3638e-05/	0.705508
[4.209687324885487e+02		[4.209687429239893e+02	
4.209687448428030e+02		4.209687413444640e+02	
4.209686899197890e+02		4.209687924753982e+02	
4.209687417358267e+02		4.209687526950727e+02	
4.209687389445743e+02]		4.209687599722548e+02]	
6.3638e-05/	3.947396	6.3637e-05/	0.783023
[4.209687443042145e+02		[4.209687697025028e+02	
4.209687906992845e+02		4.209687562634612e+02	
4.209687457983932e+02		4.209687488880370e+02	
4.209687485046533e+02		4.209687414142202e+02	
4.209687464461818e+02]		4.209687476821979e+02]	
6.3637e-05/	4.250297	6.3646e-05/	0.739407
[4.209687460613366e+02		[4.209688247026370e+02	
4.209687461224103e+02		4.209688642919307e+02	
4.209687478039323e+02		4.209688503463222e+02	

4.209687509299390e+02		4.209688391524637e+02	
4.209687469360684e+02]		4.209685832232339e+02]	
6.3637e-05/	4.079371	6.4241e-05/	0.715223
[4.209687476374028e+02		[4.209706226951944e+02	
4.209687472628019e+02		4.209682122992405e+02	
4.209687468745984e+02		4.209694873469055e+02	
4.209687467234849e+02		4.209691279622596e+02	
4.209687470788367e+02]		4.209692784990792e+02]	
6.3637e-05/	4.500535	7.2388e-05/	0.770538
[4.209687458269611e+02		[4.209705934000016e+02	
4.209687464434284e+02		4.209630310996746e+02	
4.209687474001318e+02		4.209633042655301e+02	
4.209687458933518e+02		4.209668599875407e+02	
4.209687453835092e+02]		4.209684325055117e+02]	
6.3637e-05/	4.153555	6.3637e-05/	0.872213
[4.209687484356949e+02		[4.209687460082072e+02	
4.209687472014043e+02		4.209687461994449e+02	
4.209687474575822e+02		4.209687456756193e+02	
4.209687468183992e+02		4.209687463526042e+02	
4.209687461737487e+02]		4.209687461398456e+02]	
6.3637e-05/	4.118754	1.2003e-04/	0.733787
[4.209687464100516e+02		[4.209558296258654e+02	
4.209687470063649e+02		4.209608442752248e+02	
4.209687472127805e+02		4.209769202324823e+02	
4.209687461788047e+02		4.209593620866814e+02	
4.209687455874865e+02]		4.209608385607538e+02]	
6.363783131746459e-05/	4.378053	6.663856083832798e-05/	0.835973

[4.209687472585114e+02	[4.209715801392197e+02
4.209687467220446e+02	4.209703483782170e+02
4.209687462918994e+02	4.209680384581383e+02
4.209687457319145e+02	4.209702553007686e+02
4.209687469663360e+02]	4.209655205454882e+02]

50 equipos

5 drones		100 drones	
<i>mejor /solución</i>	<i>tiempo[s]</i>	<i>mejor/ solución</i>	<i>tiempo[s]</i>
6.36395e-05/	5.086987	0.0071/	0.815305
[4.209687224468152e+02		[4.210223985796136e+02	
4.209687493529488e+02		4.208981594720885e+02	
4.209687708434465e+02		4.211470879162912e+02	
4.209687217903947e+02		4.208752217336742e+02	
4.209686523549439e+02]		4.210574458465175e+02]	
6.3642e-05/	5.225742	0.0192/	0.970236
[4.209686889137520e+02		[4.211944598827363e+02	
4.209687491930461e+02		4.208779112321211e+02	
4.209688995130769e+02		4.211145034730390e+02	
4.209686900912258e+02		4.212210043377698e+02	
4.209688355876718e+02]		4.210571476863593e+02]	
1.5159e-04/	5.351997	0.5299/	0.850052
[4.209671712573106e+02		[4.201895210033993e+02	
4.209442644611497e+02		4.198726883457902e+02	
4.209631080034414e+02		4.195744919941366e+02	
4.209679183713720e+02		4.203080639913322e+02	
4.209766680644788e+02]		4.208558552148729e+02]	
6.363783495544340e-05/	5.601025	0.016026635133130/	0.864438

[4.209687426763629e+02		[4.210563731974953e+02	
4.209687428238345e+02		4.209547492748324e+02	
4.209687439204451e+02		4.208975695472606e+02	
4.209687461971274e+02		4.207765086867920e+02	
4.209687458105250e+02]		4.212455227459520e+02]	
6.3637e-05/	5.052235	0.0090/	0.879263
[4.209687471099981e+02		[4.208571220886987e+02	
4.209687467757354e+02		4.210288000072337e+02	
4.209687463063585e+02		4.209917078656796e+02	
4.209687478448857e+02		4.207808243769109e+02	
4.209687457658679e+02]		4.208294085157310e+02]	
6.3637e-05/	7.328617	0.0157/	0.947395
[4.209687474118345e+02		[4.208113631590214e+02	
4.209687464020806e+02		4.209720746837001e+02	
4.209687470767327e+02		4.212507241070279e+02	
4.209687490602069e+02		4.208395496370449e+02	
4.209687466675820e+02]		4.209095762250802e+02]	
6.3637e-05/	4.913181	0.0736/	1.144724
[4.209687486356315e+02		[4.209072054224664e+02	
4.209687441155384e+02		4.210025095926336e+02	
4.209687372609407e+02		4.211660770280834e+02	
4.209687455061048e+02		4.209669365959881e+02	
4.209687406592081e+02]		4.202341189771247e+02]	
6.3519e-05/	6.548729	0.0406/	0.900387
[4.209815449948315e+02		[4.209960685360737e+02	
4.209687547895384e+02		4.213780982476155e+02	
4.209689841699407e+02		4.211547420567209e+02	
4.209496944661048e+02		4.211226458041883e+02	
4.208949464472081e+02]		4.206610208818253e+02]	
6.363784723362187e-05/	5.647129	0.088521090911854/	0.849551

[4.209981544741315e+02		[4.217336909045377e+02	
4.209636547852184e+02		4.206654888886843e+02	
4.2096957841289407e+02		4.208259061728622e+02	
4.2096547816391048e+02		4.210008415700049e+02	
4.2096874154161581e+02		4.210171274296826e+02]	
6.3637e-05/	5.841678	0.0075/	0.983953
[4.209981544741315e+02		[4.210683865317204e+02	
4.209636547852184e+02		4.210612260954570e+02	
4.2096957841289407e+02		4.211100211740733e+02	
4.2096547816391048e+02		4.209959100655966e+02	
4.2096874154161581e+02		4.211093500444756e+02]	
6.3691e-05/	6.045719	0.0312/	1.492893
[4.25191544741315e+02		[4.209866816436175e+02	
4.367636547852184e+02		4.210548038822426e+02	
4.24896957841289407e+02		4.213382616585575e+02	
4.57966547816391048e+02		4.212436439191199e+02	
4.26473874154161581e+02		4.208046972608591e+02]	
6.1947e-05/	7.987154	0.0354/	0.808373
[4.292481544741315e+02		[4.208337222585915e+02	
4.2058776547852184e+02		4.211999425725877e+02	
4.1946957841289407e+02		4.209667482111101e+02	
4.2573547816391048e+02		4.213554181594622e+02	
4.2051974154161581e+02		4.207248186410035e+02]	
6.3521e-05/	6.874269	0.1048/	0.914555
[4.1539981544741315e+02		[4.211157720144075e+02	
4.2578936547852184e+02		4.208506030384866e+02	
4.2826957841289407e+02		4.213666902438861e+02	
4.2649547816391048e+02		4.213972881528054e+02	
4.2924874154161581e+02		4.216414213941644e+02]	

- *Función de prueba Rastrigin (5d)*

5 equipos			
5 drones		100 drones	
<i>mejor/solución</i>	<i>tiempo[s]</i>	<i>mejor/solución</i>	<i>tiempo[s]</i>
0/ [-6.989281573410075e-11 -3.548260902328443e-09 -2.536166745026982e-10 9.582954736852565e-10 -4.059754492169150e-09]	4.296895	0/ [-1.103766074790277e-10 3.663352401925236e-09 - 1.498536590501414e-09 - 4.375023858136616e-09 - 3.379980394910990e-09]	0.698802
0/ [7.820583543352279e-10 -3.373295980101866e-09 8.294846039009483e-11 -1.528798923785010e-09 4.733259090215225e-10]	2.554856	0/ [2.580387940879788e-09 -3.471068861486048e-10 -2.846638259681223e-09 -2.642706556954646e-09 1.316988108210178e-09]	0.746797
5.6843e-14/ [-1.278043873543504e-09 5.438938555018848e-09 1.185773894141153e-08 1.127715475592359e-08 -1.661644336781409e-09]	5.587195	1.2412e-10/ [1.297445615502817e-07 6.803334360719935e-08 -7.463141527687131e-07 2.044683302555054e-07 -7.353123706345731e-08]	0.846326
0/ [0 -2.220446049250313e-16 0 0 0]	0.392891	0/ [-1.411003800931104e-09 2.217163125426024e-09 - 4.359521946085711e-09 4.850116128287095e-10 - 2.378552174826274e-09]	0.717870
0/ [-1.964577753416452e-09 3.821665691913749e-11 -2.120542429366762e-09 -2.579232438696068e-09]	0.675923	0/ [3.205385685434164e-09 2.228500261275501e-11 2.246743833681884e-09 2.905051192482007e-10]	0.737270

2.963699831574774e-09]		-2.759837401904270e-09]	
1.9895e-13/	5.308413	2.7868e-10/	0.981779
[-1.728771851758233e-08		[-8.380759680517589e-07	
-1.802037470164376e-08		-3.021430739713962e-07	
-1.148534923647957e-08		-4.282739434181997e-07	
-4.758133427603901e-10		-5.917724611991358e-07	
-1.761201810624460e-08]		2.783275389446996e-07]	
0/ [3.238506813858900e-10	3.092824	0/ [-1.289999314809847e-09	0.725413
-2.847127837024579e-09		-3.670084807016821e-09	
8.938841963962232e-10		-4.186927643076260e-09	
-1.710298360641434e-09		-1.358505497107321e-09	
-6.897106692853122e-10]		-8.424223909311963e-10]	
0/ [1.640339964498724e-09	1.521940	1.4852e-05/	0.798881
-1.227994523664489e-09		[-4.914408272554779e-05	
3.971181639285992e-09		3.163018658046761e-05	
4.011140171102145e-10		1.719503486325118e-04	
-6.561749066240356e-11]		1.422021815572539e-04	
		1.471754892998199e-04]	
0/ [-3.345019870371454e-10	1.661639	0/ [-5.237143693760664e-10	0.757537
3.979780331355136e-09		-3.493636175469933e-09	
1.024715496722776e-09		-3.362756696693460e-09	
-1.341956308266383e-09		1.401355918431196e-09	
-1.240691773389389e-10]		2.562652727313286e-09]	
2.6574e-12/	4.947741	0.1570/	0.857123
[2.407181377743526e-09		[-0.014094362784291	
1.156362424339845e-07		0.002316502822337 0.005330260305030	
3.650910372632321e-10		0.023601354670829 0.001814368985078]	
2.510327732225528e-09			
2.041660475732743e-13]			
1.4210e-14/	4.525997	0.0027/	0.891452

[-9.283375219866064e-09		[0.002496239492733	
-9.390043632047235e-10		-0.001923638435918	
2.803666875413188e-10		-3.644176825647093e-04	
3.788398252891014e-09		-0.001291091547138	
4.118249737354377e-09]		0.001459047083275]	
0/ [1.500753324780793e-12	0.574533	6.6834e-08/	0.862543
7.468967122298970e-15 0 0		[-1.441885032667163e-05	
1.167214111821857e-14]		-2.310515144252764e-06	
		4.024767126936894e-06	
		9.188604883986800e-06	
		-4.796616049233532e-06]	
0/ [0 0 0 0]	3.793103	2.7935e-09/	0.799479
		[-2.323300366622573e-06	
		9.937803281495857e-07 -	
		2.642623517434044e-06 -	
		6.058880657081619e-07 -	
		5.873727300231736e-07]	

5 drones		100 drones	
<i>mejor/ solución</i>	<i>tiempo[s]</i>	<i>mejor/ solución</i>	<i>tiempo[s]</i>
0/ [1.541248775486218e-10 -2.557273551586266e-10 -1.364548729390077e-12 1.645270825944804e-10 3.047328507633621e-12]	1.648409	0/ [0 0 -1.110223024625157e-16 0 0]	0.751828
0/ [0 -2.801256893114612e-09 -6.462348535570529e-27 6.462348535570529e-27 0]	1.113633	0/ [0 0 0 0]	0.812104
0/ [0 0 0 0]	0.897126	0.0222/ [-0.001613204704604 0.001440166436888 0.009783938667373 -0.003436871799151 - 8.863378128839278e-05]	0.958152
0/ [-1.399072811601263e-10 -2.025724827080106e-10 -4.172679881859971e-10 2.058868144746737e-10 -5.875996290643240e-10]	1.350059	0.2845/ [0.025621630115608 -0.010891410761140 -0.016873843126719 0.012382201849783 -0.014931316015733]	1.025384
0/ [-1.639659830178360e-09 7.062542208166871e-16 3.719381251961350e-09 -2.221973300781166e-15 5.459880984130617e-16]	1.488739	0.005704095071280/ [0.002646010706288 -0.003989911202855 0.001117589013097 8.329660290055177e-04 0.001972072581207]	0.993919
0/ [-1.110223024625157e-16 0 0 - 1.110223024625157e-16 0]	1.068814	0.2230/ [0.023312547838287 -0.016983533420398 0.011505672340215 -0.003469805573010 -0.012213912802602]	0.982214
0/ [0 0 0 0]	0.684570	0/ [1.526628764689895e-14 7.476948224070024e-15]	0.970853

		1.035443352842799e-15	
		5.865759138025932e-14	
		2.058892224498470e-16]	
0/ [5.636598087809709e-15	1.761710	0/ [0 0 0 0]	0.847238
7.916530826659855e-15 -			
1.185400596360973e-09			
1.073718865533849e-15			
2.392415485100668e-16]			
0/ [6.179523757395221e-10 -	1.200329	0.0013/ [7.768450174786729e-04 -	0.997113
1.386133481616188e-10 -		0.002049099066472 7.768450174785754e-	
2.932158936056555e-10 -		04 7.768450174786729e-04	
4.991403079464885e-10		7.768450174786729e-04]	
2.212029179860284e-10]			
0/ [-2.164253574955069e-09	2.185329	0.0089/	0.988161
1.063245754031017e-09		[-0.006562087329895 -	
2.090001335130842e-10 -		4.494523519559257e-04 -	
1.963277579056547e-09		0.001053344057597 -	
1.880164317452861e-10]		8.747411680681792e-04 -	
		4.217825259680841e-06]	
0/ [-4.504962640798446e-09	2.335627	2.4590/	0.966956
1.593925052632303e-09 -		[-0.012968749838399 0.010717928939750	
1.247782703062348e-09		-0.026178874621204 -1.000972591800807	
7.564878934842140e-10		1.031904476782510]	
2.256871496870156e-09]			
0/ [0 0 0 0]	1.144658	1.3559e-06/ [1.274300724177224e-07	1.001566
		8.266986690053102e-05	
		-4.896477835657220e-07	
		-1.270518874439339e-07	
		-1.818857781402117e-12]	
0/ [0 0 0 0]	0.946358	1.159707796664804/ [0.015640045844005	1.097796

-0.055731162946263

-0.030794238503956

-0.032118832473175 0.023539834659354]

- *Función de prueba Rastringin (10d)*

5 equipos

5 drones		100 drones	
<i>mejor/solución</i>	<i>tiempo [s]</i>	<i>mejor/solución</i>	<i>tiempo[s]</i>
0.9949/ [4.834923644608315e-06	5.315748	0/ [-1.103766074790277e-10	0.698802
0.994958594666006 -1.160566985193147e-		3.663352401925236e-09 -	
04 1.931662697852718e-05		1.498536590501414e-09 -	
1.309615586382136e-06 -		4.375023858136616e-09 -	
2.402097895365662e-05		3.379980394910990e-09]	
7.413389350299084e-05 -			
1.005066254025309e-04 -			
5.708237306884243e-11			
5.829443018901190e-05]			
2.9882/	4.685926	4.6613/ [0.038273508720271	2.994899
[-3.595580892128331e-08		0.038273508720271 0.038273508720271	
0.994787412255712		0.038273508720271 0.038273508720270	
9.766429639109312e-08		1.025002597788373 1.025002597788373	
-2.030150191534956e-07		0.038273508720271 0.038273508720271	
-7.165374923759752e-08		0.038273508720271]	
0.995040534774285			
0.999107440301241			
-1.468286234347869e-08			
-6.135818731744776e-06			
-1.986251448706852e-09]			
0/ [5.973253508317460e-09	4.201279	12.5411/ [0.903248

-3.452650784637806e-09		-0.883402047736707	
6.658762998238328e-10		-0.996351249976720 0.987019551080722	
-8.820515466718245e-12		-0.038001634690797 0.896871387422045	
3.034923551032347e-10		0.956031365113139 -0.021453850142389	
-4.560798520451803e-10		0.008801455940856 0.953169176894350	
-8.387445464046207e-10		0.953204479173694]	
6.503057534742080e-10			
-5.424164901174230e-09			
1.355855302516808e-09]			
8.5265e-14/	5.513851	0/ [-3.699235116661637e-09	1.131798
[-7.325831919465833e-09		-3.848840991484069e-09	
-6.222180099303088e-09		-2.115985134210000e-09	
-1.055081550447183e-09		2.599685465730051e-09	
-1.942933814189351e-09		-2.893494850717282e-09	
-1.288792857456108e-09		2.588827254632278e-09	
-5.600096878731215e-09		-5.384327030780876e-10	
-6.195185293893104e-09		3.226417603746985e-09	
-1.017179155116803e-08		1.057988352833453e-09	
1.410266351112562e-08		-4.381912905234111e-09]	
4.410994025918129e-09]			
0/ [-1.946575343639262e-09	1.348977	17.5240/ [1.172252
2.542816614336298e-10		-0.888495371652601 0.987466215581718	
1.700206277066392e-09		0.026793822513587 2.062262307474245	
-3.342534993344144e-09		0.993317159848306 1.062616065259604	
4.959670734529284e-09		0.940943926003466 -0.074000857901561	
-4.941493567956046e-09		1.571437255890595e-04	
2.159364464331294e-09		0.119446728658800]	
4.655251668123566e-09			
-7.315183350163180e-10			
-1.616272801800199e-09]			

5.6843e-14 [5.247480	3.2175e-07 / [2.782548501836142e-06	0.796352
-3.665745077473013e-09		-2.365962381818779e-06	
4.530376702224980e-09		6.106651630444960e-06	
-2.129822320458945e-09		-1.050237287226397e-05	
2.391002971143210e-09		9.261997322996142e-06	
1.145983278481793e-08		2.973630188686094e-05	
1.233867379382055e-08 -		-1.511317113868982e-05	
2.788112479690720e-09 -		-1.083966092742404e-05	
3.980299939876109e-09		1.155181080341118e-05	
5.127166945134875e-09 -		3.393684476138806e-06]	
3.757498657949403e-09]			
0 [4.139032217151783e-10	0.501768	8.3316e-07 / [0.932772
-7.720531200985014e-12		-3.954705881870169e-05	
-1.093201359256206e-10 -		-8.725412157746546e-07	
1.821458450718342e-09		4.134270564713742e-05	
-6.099815317357923e-12		2.739651820546345e-06	
1.075565017465258e-09		-6.669693423826588e-06	
1.211476004627096e-09		2.672203463652625e-05	
8.723615102142843e-12		2.774444690747612e-06	
-6.610435029300010e-10		1.210334167565904e-05	
-7.730785847816313e-11]		2.229022552605435e-06	
		6.335291518767777e-07]	

Apéndice H. Manual de usuario para implementar la función entropía en DSO

Para correr el algoritmo.....

1. Abierto el archivo **DSO-MATLAB master** que contiene a **DSO** y **GPOLS**, nos ubicamos en el archivo **main2.m**
2. En la primera línea del archivo **main2.m** se define la configuración de la función objetivo que vamos a trabajar, es decir, el número de variables ($nDim$) y los límites superiores[UB] e inferiores[LB] de dichas variables.
3. Luego en este mismo archivo se definen el número de equipos ($Nteam$) y la cantidad de drones por equipo (N)
4. En el archivo **DSO.m** también se deben configurar el número de equipos y la cantidad de drones por equipo.

Para configurar los parámetros de DSO y la función objetivo...

5. Se define la configuración de funcionamiento para DSO: constantes de funcionamiento ($C1$, $C2$, $C3$), el número de iteraciones necesarias para realizar actualización de firmware, el número de equipos y la cantidad de drones por equipos.
6. En la última línea de la parte correspondiente a la configuración de DSO, se ingresa después del símbolo @ el nombre de la función objetivo o `costFunction` (aquí se ingresan las funciones de prueba), que en este caso es `entropiaDSO`.

Ejemplo:

```
[Solution, Value, StatsCurveOFV, Evaluations, StopMessage,  
HistoryRanks] = DSO(@entropiaDSO, nDim, LB, UB, setup);  
toc()
```