



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

Departamento de Computación

**Uso de estadística computacional  
para la calibración de parámetros  
en algoritmos evolutivos**

Tesis que presenta:

**Angelina Jane Reyes Medina**

Para obtener el grado de:

**Maestro en Ciencias  
en Computación**

Director de la Tesis:  
Dr. Gregorio Toscano Pulido



© Derechos reservados por  
Angelina Jane Reyes Medina  
2009



Esta investigación fue parcialmente financiada mediante el proyecto No. 51623 del Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas.

This research was partially funded by project number 51623 from "Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas"



La tesis presentada por Angelina Jane Reyes Medina fue aprobada por:

-----

---

Dr. Francisco Rodríguez Henríquez

---

Dr. Guillermo Leguzamón

---

Dr. Gregorio Toscano Pulido, Director

Ciudad de México, Distrito Federal., 18 de Diciembre de 2009



*A mamá y papá por ser mi vida, a mis hermanos por ser mi soporte,  
a Xibalba por ser mi luz, a Karol por ser mi lluvia  
y a ti por ser viento.*



# Agradecimientos

- Un agradecimiento especial al Dr. Gregorio y su esposa Gaby, por su amistad y apoyo en esta etapa de mi vida.
- Un agradecimiento especial a mis profesores Ulises y Jaime, por insistirme en continuar.
- Un agradecimiento a Brisbane, Lil, Raymundo y Victor, por su incondicional amistad.
- Un agradecimiento a Ismael por creer en mi y a mis amigos de la universidad por tolerar mi ausencia en sus reuniones.
- Un agradecimiento a Sofi nuestra amada secretaria, por sus maternales consejos.
- Un agradecimiento a Fabiola e Irasema por sus platicas vespertinas de todos los días.
- Un agradecimiento a todos mis amigos del CINVESTAV por apoyarme y acompañarme.
- Un agradecimiento a los alumnos del LTI por su calurosa amistad.
- Un agradecimiento a mis profesores del CINVESTAV y el LTI por sus conocimientos que me brindaron.
- Agradezco el apoyo recibido a través del proyecto CONACyT 90548, titulado “Uso de técnicas híbridas para resolver problemas de optimización multiobjetivo”, cuyo Investigador responsable es el Dr. Gregorio Toscano Pulido.



# Índice General

<b>Índice General</b>	<b>I</b>
<b>Índice de Figuras</b>	<b>v</b>
<b>Índice de Tablas</b>	<b>vii</b>
<b>Índice de Algoritmos</b>	<b>ix</b>
<b>Publicaciones</b>	<b>xi</b>
<b>Resumen</b>	<b>xiii</b>
<b>Abstract</b>	<b>xv</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Motivación . . . . .	3
1.3. Objetivo general . . . . .	4
1.3.1. Objetivos específicos . . . . .	4
1.4. Contribuciones esperadas . . . . .	5
1.5. Organización del documento . . . . .	5
<b>2. Algoritmos evolutivos</b>	<b>7</b>
2.1. Introducción . . . . .	7
2.2. Conceptos básicos de computación evolutiva . . . . .	8
2.3. Algoritmos Evolutivos . . . . .	9
2.3.1. Programación Evolutiva . . . . .	12
2.3.2. Estrategias Evolutivas . . . . .	12
2.3.3. Algoritmos Genéticos . . . . .	13
2.3.4. Programación Genética . . . . .	14
2.4. Inteligencia de cúmulos . . . . .	15
2.5. Optimización mediante cúmulos de partículas . . . . .	16
2.5.0.1. PSO versión <i>gbest</i> . . . . .	18
2.5.0.2. PSO versión <i>lbest</i> . . . . .	19
2.6. Topologías adicionales de interconexión . . . . .	20
2.6.1. Topología en anillo . . . . .	22
2.6.2. Topología completamente conectada . . . . .	23
2.6.3. Topología en estrella . . . . .	23
2.6.4. Topología en malla . . . . .	24
2.6.5. Topología en árbol . . . . .	24

2.6.6.	Topología toroidal . . . . .	25
2.6.7.	Algoritmos de agrupamiento . . . . .	25
2.6.7.1.	Algoritmo de agrupamiento <i>k-means</i> . . . . .	26
2.6.7.2.	Algoritmo de agrupamiento jerárquico . . . . .	26
2.7.	Variaciones de PSO libres de parámetros . . . . .	27
2.7.1.	PSO barebones . . . . .	27
2.7.2.	PSO barebones y evolución diferencial . . . . .	28
<b>3.</b>	<b>Configuración de parámetros en Algoritmos Evolutivos</b>	<b>33</b>
3.1.	Introducción . . . . .	33
3.2.	Configuración de parámetros . . . . .	34
3.3.	Enfoques para la calibración de parámetros . . . . .	36
3.3.1.	Meta-algoritmos evolutivos . . . . .	37
3.3.2.	Meta-estimación de algoritmos de distribución . . . . .	38
3.3.3.	Estadística computacional . . . . .	39
3.4.	Estadística computacional . . . . .	40
3.4.1.	Métodos estadísticos computacionales clásicos . . . . .	41
3.4.1.1.	Diseño de experimentos . . . . .	41
3.4.1.2.	Análisis de Varianza . . . . .	45
3.4.1.3.	Modelos de regresión . . . . .	47
3.4.1.4.	Trabajos relacionados . . . . .	49
3.4.2.	Métodos estadísticos computacionales modernos . . . . .	49
3.4.2.1.	Análisis y Diseño de Experimentos de Computadora . . . . .	50
3.4.2.2.	Árboles de regresión . . . . .	51
3.4.2.3.	Métodos de muestreo en DACE . . . . .	51
3.4.2.4.	Trabajos relacionados . . . . .	52
<b>4.</b>	<b>Calibración de parámetros usando estadística computacional</b>	<b>55</b>
4.1.	Introducción . . . . .	55
4.2.	Diseños Factoriales . . . . .	56
4.2.1.	Diseños Factoriales $2^k$ . . . . .	58
4.3.	Metodología de Superficie de Respuesta . . . . .	59
4.3.1.	Funcionamiento de RSM . . . . .	63
4.3.1.1.	Variables . . . . .	63
4.3.1.2.	Hipótesis: aproximación lineal . . . . .	63
4.3.1.3.	Diseños factoriales y modelo lineal de primer orden . . . . .	64
4.3.1.4.	Prueba 1: de falta de ajuste del modelo de primer orden . . . . .	65
4.3.1.5.	Hipótesis: aproximación cuadrática . . . . .	65
4.3.1.6.	Diseños CCD y $2^k$ y modelo lineal de segundo orden . . . . .	65
4.3.1.7.	Prueba 2: de falta de ajuste del modelo de segundo orden . . . . .	65
4.3.1.8.	Análisis canónico . . . . .	66
4.3.2.	Método de ascenso empinado . . . . .	66
4.3.3.	Análisis de superficie de respuesta de segundo orden . . . . .	68

4.4.	Estimación de los parámetros en modelos de regresión lineal . . . . .	70
4.4.1.	Estimación de la varianza $\sigma^2$ . . . . .	75
4.4.2.	Propiedades de los estimadores . . . . .	76
4.5.	Prueba de hipótesis en modelos de regresión múltiple . . . . .	76
4.5.1.	Prueba para la significancia de regresión . . . . .	77
4.6.	Diseños para ajustar la superficie de respuesta . . . . .	79
4.6.1.	Diseños para ajustar el modelo de primer orden . . . . .	80
4.6.2.	Diseños para ajustar el modelo de segundo orden . . . . .	81
<b>5.</b>	<b>Metodología</b>	<b>83</b>
5.1.	Introducción . . . . .	83
5.2.	Experimento 1 . . . . .	84
5.2.1.	Resultados . . . . .	85
5.2.2.	Conclusiones . . . . .	89
5.2.3.	Análisis estadístico entre topologías . . . . .	90
5.2.4.	Prueba estadística $t$ . . . . .	93
5.2.5.	Resultados del análisis estadístico . . . . .	95
5.2.6.	Conclusiones del análisis estadístico . . . . .	97
5.3.	Experimento 2 . . . . .	103
5.3.1.	Resultados . . . . .	104
5.3.2.	Interacción entre variables . . . . .	109
5.3.3.	Conclusiones . . . . .	112
5.4.	Experimento 3 . . . . .	116
5.4.1.	Metodología de superficie de respuesta . . . . .	116
5.4.2.	Resultados . . . . .	118
5.4.3.	Conclusiones . . . . .	119
5.5.	Metodología propuesta para la calibración de parámetros en AEs . . . . .	123
5.5.1.	Experimento 4a . . . . .	125
5.5.1.1.	Resultados . . . . .	126
5.5.1.2.	Conclusiones . . . . .	128
5.5.2.	Experimento 4b . . . . .	129
5.5.2.1.	Resultados . . . . .	130
5.5.2.2.	Conclusiones . . . . .	131
5.5.3.	Experimento 4c . . . . .	131
5.5.3.1.	Resultados . . . . .	132
5.5.3.2.	Conclusiones . . . . .	133
<b>6.</b>	<b>Conclusiones y trabajo futuro</b>	<b>135</b>
6.1.	Conclusiones . . . . .	135
6.2.	Contribuciones obtenidas . . . . .	138
6.3.	Trabajo futuro . . . . .	139
<b>A.</b>	<b>Diseños factoriales <math>2^2</math></b>	<b>141</b>

<b>B. Ejemplo de la metodología de superficie de respuesta</b>	<b>147</b>
<b>C. Funciones de prueba</b>	<b>155</b>
<b>D. Gráficas de caja</b>	<b>157</b>
<b>Bibliografía</b>	<b>159</b>

# Índice de Figuras

2.1. Topología completamente conectada. . . . .	19
2.2. Topología en anillo. . . . .	19
2.3. Topologías de interconexión usadas en este estudio. . . . .	22
3.1. Modelo general de un proceso o sistema. . . . .	41
3.2. Interacción entre variables. . . . .	44
4.1. Un diseño factorial de dos factores. . . . .	56
4.2. Un diseño factorial de dos factores con interacción. . . . .	57
4.3. Aproximación del modelo lineal estimado en un diseño factorial. . . . .	60
4.4. Superficie de respuesta $\eta$ , para un proceso con dos variables, $x_1$ y $x_2$ . . . . .	61
4.5. Gráfica de contornos de una superficie de respuesta con dos variables. . . . .	61
4.6. Diagrama de flujo de la metodología de superficie de respuesta. . . . .	64
4.7. Superficie de respuesta de primer orden y ruta de ascenso empinado. . . . .	67
4.8. Posibles respuestas de un punto estacionario. . . . .	69
4.9. Forma canónica del modelo de segundo orden. . . . .	71
5.1. Comparación entre topologías de interconexión implementadas en: 1) PSO, 2) PSOB, 3) PSOBEXP, 4)PSOBDE. . . . .	100
5.2. Comparación entre topologías de interconexión implementadas en: 1) PSO, 2) PSOB, 3) PSOBEXP, 4)PSOBDE. . . . .	101
5.3. Comparación entre topologías de interconexión implementadas en: 1) PSO, 2) PSOB, 3) PSOBEXP, 4)PSOBDE. . . . .	102
5.4. Gráficas del diseño factorial completo en todas las funciones de prueba con la topología toroidal. . . . .	110
5.5. Gráficas del diseño factorial completo en todas las funciones de prueba con la topología toroidal. . . . .	111
5.6. Análisis de varianza (ANOVA) en cada una de las funciones de prueba. . . . .	113
5.7. Análisis de varianza (ANOVA) en cada una de las funciones de prueba. . . . .	114
5.8. Análisis de varianza (ANOVA) en cada una de las funciones de prueba. . . . .	115
A.1. Combinación de tratamientos en el diseño $2^2$ . . . . .	142
D.1. Gráfica de cajas. . . . .	158



# Índice de Tablas

3.1. Datos para un experimento con un único factor. . . . .	45
4.1. Datos para la regresión lineal múltiple. . . . .	71
5.1. Prueba de los promedios de distribuciones normales con varianza desconocida, pero supuesta igual. . . . .	92
5.2. Prueba de los promedios de distribuciones normales con varianza desconocida, pero supuesta diferente. . . . .	93
5.3. Análisis estadístico de topologías para PSO. . . . .	98
5.4. Análisis estadístico de topologías para PSOBB. . . . .	98
5.5. Análisis estadístico de topologías para PSOBBEXP. . . . .	99
5.6. Análisis estadístico de topologías para PSOBBDE. . . . .	99
5.7. Rangos y niveles para cada factor en el experimento factorial completo. . . . .	104
5.8. Diseño factorial completo para las topologías: 1) Anillo, 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol y; algoritmos de agrupamiento: 7) <i>K-means</i> , 8) Jerárquico. . . . .	105
5.9. Diseño factorial completo para las topologías: 1) Anillo, 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol y; algoritmos de agrupamiento: 7) <i>K-means</i> , 8) Jerárquico. . . . .	106
5.10. Diseño factorial completo para las topologías: 1) Anillo, 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol y; algoritmos de agrupamiento: 7) <i>K-means</i> , 8) Jerárquico. . . . .	107
5.11. Valores de los parámetros de PSO obtenidos por RSM. Las topologías son: 1) Anillo , 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol, 7) <i>K-means</i> y 8) Jerárquico. . . . .	121
5.12. Valores de los parámetros de PSO obtenidos por RSM. Las topologías son: 1) Anillo , 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol, 7) <i>K-means</i> y 8) Jerárquico. . . . .	122
5.13. Valores de los parámetros de PSO obtenidos por RSM. Las topologías son: 1) Anillo , 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol, 7) <i>K-means</i> y 8) Jerárquico. . . . .	123
5.14. Diseño factorial completo para el experimento 4a. f01) <i>Sphere</i> , f02) <i>Schwefel</i> , f03) <i>Step</i> , f04) <i>Rosenbrock</i> , f05) <i>Rotated hyper-ellipsoid</i> , f06) <i>Schwefel 2.26</i> , f07) <i>Rastrigin</i> , f08) <i>Ackley</i> y f09) <i>Griewank</i> . . . . .	126
5.15. Resultados de la prueba estadística <i>t</i> para el experimento 4a. f01) <i>Sphere</i> , f02) <i>Schwefel</i> , f03) <i>Step</i> , f04) <i>Rosenbrock</i> , f05) <i>Rotated hyper-ellipsoid</i> , f06) <i>Schwefel 2.26</i> , f07) <i>Rastrigin</i> , f08) <i>Ackley</i> y f09) <i>Griewank</i> . . . . .	127
5.16. Valores de las funciones objetivo para cada una de las topologías, obtenidas con la metodología propuesta. . . . .	128

5.17. Valores de las funciones objetivo para cada una de las topologías, obtenidas con la metodología propuesta. . . . .	128
5.18. Rango de valores para los parámetros de PSO derivados del experimento 4a. f01) <i>Sphere</i> , f02) <i>Schwefel</i> , f03) <i>Step</i> , f04) <i>Rosenbrock</i> , f05) <i>Rotated hyper-ellipsoid</i> , f06) <i>Schwefel 2.26</i> , f07) <i>Rastrigin</i> , f08) <i>Ackley</i> y f09) <i>Griewank</i> . . . . .	129
5.19. Resultados de la prueba estadística <i>t</i> y RMS para el experimento 4b. 1) Anillo , 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol, 7) K-means y 8) Jerárquico. . . . .	130
5.20. Valores de las funciones objetivo despues de aplicar RSM en el experimento 4b. f01) <i>Sphere</i> , f02) <i>Schwefel</i> , f03) <i>Step</i> , f04) <i>Rosenbrock</i> , f05) <i>Rotated hyper-ellipsoid</i> , f06) <i>Schwefel 2.26</i> , f07) <i>Rastrigin</i> , f08) <i>Ackley</i> y f09) <i>Griewank</i> . . . . .	130
5.21. Resultados de la prueba estadística <i>t</i> y RMS para el experimento 4c. . . . .	132
5.22. Valores de las funciones objetivo despues de aplicar RSM en el experimento 4c. f01) <i>Sphere</i> , f02) <i>Schwefel</i> , f03) <i>Step</i> , f04) <i>Rosenbrock</i> , f05) <i>Rotated hyper-ellipsoid</i> , f06) <i>Schwefel 2.26</i> , f07) <i>Rastrigin</i> , f08) <i>Ackley</i> y f09) <i>Griewank</i> . . . . .	133
A.1. Datos del diseño factorial $2^2$ . . . . .	141
B.1. Datos del proceso para ajustar el modelo de primer orden . . . . .	148
B.2. Datos del proceso para ajustar el modelo de primer orden . . . . .	150
B.3. Datos para el segundo modelo de primer orden. . . . .	151
B.4. Datos para el diseño de composición central. . . . .	152

# Índice de Algoritmos

1.	Esquema general de un algoritmo evolutivo. . . . .	12
2.	Algoritmo básico de la programación evolutiva. . . . .	13
3.	Algoritmo general de la estrategia evolutiva. . . . .	14
4.	Algoritmo genético simple. . . . .	14
5.	Algoritmo general de la programación genética. . . . .	15
6.	Algoritmo de PSO versión <i>gbest</i> . . . . .	21
7.	Metodología propuesta. . . . .	125



# Publicaciones

Angelina J Reyes Medina, Gregorio Toscano Pulido and José Gabriel Ramírez Torres. *A Comparative Study of Neighborhood Topologies for Particle Swarm Optimizers*. ICEC 09, Madeira Portugal.



## **Uso de estadística computacional para la calibración de parámetros en algoritmos evolutivos**

por

**Angelina Jane Reyes Medina**

Maestro en Ciencias del Departamento de Computación

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2009

Dr. Gregorio Toscano Pulido, Director

El comportamiento de un algoritmo evolutivo (AE) para un problema de optimización dado está usualmente dirigido por sus operadores (*p.ej.*, cruza, mutación, selección, etc.) y por los valores seleccionados para los parámetros del algoritmo. Calibrar los parámetros de este tipo de enfoques es un problema difícil debido a que pueden tomar diferentes valores y por lo tanto, el número de posibles combinaciones podría ser muy elevado.

Por lo tanto, podemos decir que cuando usamos un AE, existe un problema adicional de optimización que necesita ser resuelto: la calibración de parámetros.

Hasta recientemente dicha calibración se había hecho con base en la experiencia del usuario, por referencias bibliográficas o bien aplicando el método experimental de prueba y error. Sin embargo, recientemente los investigadores de la comunidad de computación evolutiva han comenzado a utilizar métodos clásicos y estadísticos del diseño de experimentos como una alternativa viable para calibrar adecuadamente los parámetros de sus enfoques. Esta nueva tendencia, busca obtener conclusiones científicas a través de una experimentación sistemática.

Esta tesis propone usar el diseño de experimentos y métodos de superficie de respuesta para crear una metodología viable para la calibración de parámetros en algoritmos evolutivos. De manera particular, en este trabajo se calibrarán los parámetros intrínsecos del algoritmo “optimización mediante cúmulos de partículas” con el objetivo de incrementar su desempeño promedio.



## **On the use of computational statistics for parameter tuning on evolutionary algorithms**

by

**Angelina Jane Reyes Medina**

Master of Science from the Computer Science Department  
Research Center for Advanced Study from the National Polytechnic Institute, 2009  
Dr. Gregorio Toscano Pulido, Advisor

The behavior of an evolutionary algorithm (EA) for a given optimization problem is directed by both, the operators (crossover, mutation, etc), and the values selected for the parameters of the algorithm. The issue of setting the parameters and operators (which can be seen as parameters also) plays a key role on the performance of such approach. Tuning well these parameters is a hard problem, since they can usually take several values, and therefore, the number of possible combinations is usually very high.

Therefore, when we use EAs we can say that there is an additional optimization problem to be solved: parameter tuning.

Until recently, the parameters had been tuned using the common sense, taking values from the specialized literature or by trial and error. However, nowadays the evolutionary computation researchers have started using design of experiments approaches in order to tune the parameters of their approaches.

This thesis uses the design of experiments and response surface methods to develop a viable methodology in order to solve the parameter tuning problem. Specifically, in this work the particle swarm optimization's parameters will be tuning in order to increase the average performance of PSO.



# 1

## Introducción

### 1.1 Introducción

El comportamiento de un algoritmo evolutivo (AE) usualmente está dirigido tanto por sus operadores de variación (*p.ej.*, cruza, mutación y selección), como por los valores asociados a estos operadores. Calibrar adecuadamente los valores de los parámetros de un AE para producir un incremento en su desempeño es un problema difícil, en gran medida por la cantidad de parámetros y amplio rango de valores (hace prohibitivo un análisis combinatorio completo). Consecuentemente, la dimensionalidad de este problema provoca que los investigadores en el área no utilicen una metodología científica que calibre los parámetros; y por lo tanto, la configuración inicial de los parámetros se ha venido realizando con base en la experiencia, por referencias bibliográficas, o bien, mediante prueba y error.

Recientemente, los investigadores de la comunidad de computación evolutiva han comenzado a utilizar métodos clásicos y estadísticos de diseño de experimentos (DOE, por sus siglas en Inglés) [4, 6] como una alternativa viable para sintonizar adecuadamente los parámetros de sus enfoques, provocando el desarrollo de enfoques cuyos resultados mantienen una calidad competitiva, pero

utilizando menos evaluaciones de la función objetivo. Paralelamente, para evitar la sintonización de parámetros por parte del usuario, recientemente los investigadores del área han hecho algunos intentos por desarrollar esquemas adaptativos que se empotran en el AE. Sin embargo, aún se necesitan pruebas sistemáticas que produzcan salidas cuantificables (que permitan medir la variabilidad), para generar conocimiento del comportamiento de los operadores y como influyen estos en el AE.

La **optimización mediante cúmulos de partículas** (PSO, por sus siglas en inglés) es un AE relativamente nuevo que ha sido usado para resolver exitosamente una amplia variedad de tareas de optimización. Su alta velocidad de convergencia y su relativa simplicidad hacen de PSO un candidato viable para ser utilizado como método de resolución de diversos problemas de optimización.

El enfoque PSO puede ser visto como un algoritmo de comportamiento distribuido que desempeña, en su versión más general, una búsqueda multidimensional. PSO Simula el vuelo de las aves, donde la posición de cada individuo (en el contexto de PSO se conoce como partícula) es afectada por la posición de su mejor vecino (*v.g.*, componente social) y por su propia experiencia obtenida hasta el momento (*v.g.*, componente cognitivo). Diversos enfoques de PSO han sido desarrollados con la finalidad de mejorar su desempeño; tales enfoques han demostrado que el comportamiento convergente en PSO está fuertemente ligado a los valores de: peso de la inercia, coeficiente cognitivo y coeficiente social. Debido a ello, algunos trabajos se han enfocado en eliminar la dependencia de dichos parámetros (mediante el uso de auto-ajuste) con la finalidad de evitar su calibración.

No obstante, el auto-ajuste requiere que los parámetros a ser calibrados hayan sido previamente caracterizados, lo cual nos regresa al problema inicial.

Por otra parte, investigaciones dentro del paradigma han encontrado que la topología de interconexión entre partículas interactúa directamente con la función a ser optimizada. Estos estudios han mostrado, teóricamente, que la topología de vecindad afecta significativamente el desempeño del algoritmo. De tal manera, algunos tipos de topologías de interconexión pueden trabajar bien con algunas funciones mientras que con otras presentarán problemas. Por lo tanto, la topología de interconexión juega un papel importante en el desempeño de PSO.

En este trabajo se presenta una comparación de cuatro versiones de PSO, tres de ellas libres de parámetros. Para llevar a cabo esta comparación, se implementaron seis topologías y dos algoritmos

de agrupamiento, con la finalidad de determinar si la forma de comunicación entre partículas que las diferentes versiones de PSO implementan, afectan o mejoran la convergencia del algoritmo. De acuerdo a los resultados obtenidos por los experimentos, se concluye que PSO es altamente dependiente de la topología que implementa.

Posteriormente, este trabajo propone usar los métodos estadísticos clásicos, tal como el diseño de experimentos y métodos de superficie de respuesta, como una alternativa viable para la sintonización de parámetros en algoritmos evolutivos, nuestro objetivo es verificar si este tipo de métodos permiten la calibración de parámetros de una forma más robusta y eficaz. Para probar la eficacia de estos métodos, se calibraron los parámetros de PSO con el objetivo de obtener un conjunto de valores para sus parámetros que mejore su desempeño promedio.

## 1.2 Motivación

Existen diferentes AEs que han probado ser efectivos en numerosas aplicaciones de optimización. Sin embargo la convergencia de los mismos esta fuertemente ligada a los valores de sus parámetros. Dichos valores pueden alterar significativamente el comportamiento del AE en cuestión.

La configuración adecuada de los parámetros de un AE es fundamental para su buen desempeño. Sin embargo, es un problema difícil, puesto que existe un gran número de opciones posibles y únicamente se tiene un pequeño conocimiento del efecto de los parámetros en el desempeño del AE. Además, dependiendo del problema a manejar, los valores de los parámetros del AE podrían cambiar, de tal manera que diversos tipos de problemas podrían necesitar una configuración diferente en sus parámetros (v.g., no existe una configuración general válida para todo tipo de problema).

La configuración de parámetros en AEs se divide comúnmente en dos clases: **control de parámetros** y **calibración de parámetros** [18]. La primera clase hace referencia los valores de los parámetros que cambian durante la ejecución del AE, en este caso se necesitan parámetros iniciales y estrategias de control adecuadas que permitan al algoritmo adaptarse automáticamente ante cualquier cambio producido en el problema. Por otro lado, la segunda clase es más simple, puesto que los valores de los parámetros no cambian durante la ejecución del AE, por lo tanto sólo se necesitan conocer los

valores iniciales de los parámetros antes de ejecutar el AE.

Sin embargo, tanto el control de parámetros como la calibración de parámetros demandan un costo computacional elevado, pero a diferencia de la primera, la calibración de parámetros sólo se lleva a cabo una vez (*v.g.*, antes de la ejecución del algoritmo).

Por otro lado, para establecer mecanismos que controlen los parámetros dinámicamente durante la ejecución de un AE (*v.g.*, control de parámetros), se necesita realizar previamente una caracterización de los mismos, procedimiento que usualmente se realiza mediante una previa calibración de los parámetros. Es por ello que la calibración de parámetros ha sido vista como una alternativa favorable al momento de configurar los parámetros de un AE.

Recientemente han aparecido diferentes investigaciones referentes a la calibración de parámetros por medio de métodos estadísticos. Utilizar métodos estadísticos para calibrar los parámetros de AEs, significa implementar métodos estadísticos computacionalmente exhaustivos que permitan obtener, a través de pruebas rigurosas, un conjunto de valores adecuados para estos parámetros.

A pesar de la importancia de tener parámetros adecuadamente calibrados y de la gran cantidad de enfoques evolutivos que han aparecido recientemente para solucionar dicho problema, típicamente no se sigue ninguna metodología para realizar dicha calibración,

## 1.3 Objetivo general

- Usar métodos estadísticos tradicionales para calibrar los parámetros de un algoritmo evolutivo.

### 1.3.1 Objetivos específicos

- Determinar si la topología de interconexión entre partículas permite acelerar la convergencia de PSO.
- Determinar si mediante la variación de  $w$ ,  $c_1$  y  $c_2$  podemos provocar un incremento en el desempeño de PSO para una función y una topología determinada.
- Determinar si existe un conjunto de valores para los parámetros de PSO ( $w$ ,  $c_1$  y  $c_2$ ) que

presenten un buen desempeño cuando se utiliza una topología de interconexión específica en cualquiera de las funciones de prueba utilizadas en esta tesis.

- Determinar si existe un conjunto de valores para los parámetros de PSO ( $w$ ,  $c1$  y  $c2$ ) que presenten un buen desempeño cuando se trata de optimizar una función de prueba en específico usando cualquiera de las topologías utilizadas en esta tesis.
- Determinar si existe un conjunto de valores para los parámetros de PSO ( $w$ ,  $c1$  y  $c2$ ) que presenten un buen desempeño independientemente de la topología o de la función de prueba (de las utilizadas en esta tesis).

## 1.4 Contribuciones esperadas

Al finalizar este trabajo se espera obtener:

- Un análisis del desempeño de cuatro versiones de PSO.
- Un análisis del desempeño de las cuatro versiones de PSO cuando utilizan diferentes formas de comunicación entre partículas.
- Una metodología basada en el diseño de experimentos y superficie de respuesta para calibrar los parámetros de un AE.
- Un estudio sobre la calibración de los parámetros de PSO cuando se utiliza la metodología propuesta.
- Un conjunto de parámetros que mejoren el desempeño promedio de PSO cuando sean usados en diferentes funciones de prueba o topologías de interconexión.

## 1.5 Organización del documento

El resto de este documento está organizado de la siguiente manera. En el capítulo 2 se introducen algunos conceptos básicos de la computación evolutiva utilizados a lo largo del documento.

Posteriormente, en el mismo capítulo, se describen brevemente los paradigmas que conforman el área de la computación evolutiva con la finalidad de introducir algunos conceptos necesarios por el algoritmo de optimización utilizado a lo largo de este documento (PSO). Debido a que este trabajo recae fuertemente en el tema de calibración de parámetros en AEs, en el capítulo 3 se revisa el estado de arte de dicho enfoque. Posteriormente, el capítulo 4 presenta el enfoque propuesto para calibrar los parámetros en AEs, (particularmente PSO). Las características de implementación de la metodología propuesta, así como los resultados obtenidos por los experimentos se muestran en el capítulo 5. Finalmente, el capítulo 6 presenta las conclusiones y trabajo futuro sobre este trabajo de investigación.

# 2

## Algoritmos evolutivos

### 2.1 Introducción

La **computación evolutiva** (CE) (o **algoritmos evolutivos**, AEs), está conformada por paradigmas inspirados en el Neo-Darwinismo<sup>1</sup>. Actualmente la CE proporciona diferentes algoritmos de optimización capaces de simular comportamientos inteligentes. A pesar que recientemente han aparecido una gran cantidad de métodos en esta área, la mayoría comparte los siguientes componentes principales: un proceso de inicialización, un mecanismo de selección de padres, operadores de variación y un mecanismo de selección de supervivientes.

El objetivo de este capítulo es describir brevemente cada uno de los componentes de los AEs y su papel correspondiente en el proceso de optimización. Así mismo, se describen en términos generales los paradigmas que existen hasta el momento.

---

<sup>1</sup>La teoría del Neo-Darwinismo combina el seleccionismo de August Weismann [49], las leyes de la herencia de Gregor Mendel [34] y la teoría de la evolución por selección natural de Charles Darwin [17].

## 2.2 Conceptos básicos de computación evolutiva

Por su inspiración biológica, la CE adoptó conceptos biológicos en su propia terminología. Algunos conceptos básicos utilizados en este trabajo son los siguientes:

- Denominamos **cromosoma** a una estructura de datos que contiene una cadena de parámetros de diseño o genes. Esta estructura de datos puede ser almacenada, por ejemplo, como una cadena de bits o un arreglo de enteros.
- Se llama **gene** a una subsección de un cromosoma que usualmente codifica el valor de un parámetro.
- Se denomina **genotipo** a la codificación (*p.ej.*, binaria) de los parámetros que representan una solución del problema a resolver.
- Se denomina **fenotipo** a la decodificación del cromosoma. Es decir, a los valores obtenidos al pasar de la representación (del genotipo) a la necesitada por la función objetivo.
- Se denomina **individuo** a un miembro de la población de soluciones potenciales a un problema. Cada individuo contiene un cromosoma que representa una solución posible del problema a resolver.
- Se denomina **aptitud** al valor que se asigna a cada individuo y que indica qué tan bueno es con respecto a los demás para la solución de un problema. Por ejemplo, si  $f(x) = x^2$ , entonces  $f(10102) = 100$  (donde  $f(x)$  es la aptitud de un individuo).
- Se llama **paisaje de aptitud** (*fitness landscape*) a la hipersuperficie que se obtiene al aplicar la función de aptitud a cada punto del espacio de búsqueda.
- Llamamos **generación** a una iteración de la medida de aptitud y a la creación de una nueva población por medio de los operadores de reproducción.

- Se llama **operador de reproducción** a todo aquel mecanismo que influencia la forma en que se pasa la información genética de padres a hijos. Los operadores de reproducción caen en tres amplias categorías:
  - Cruza
  - Mutación
  - Reordenamiento
- La **cruza** es un operador que forma un nuevo cromosoma combinando partes de cada uno de los cromosomas padres.
- Se denomina **mutación** a un operador que forma un nuevo cromosoma a través de alteraciones (usualmente pequeñas) de los valores de los genes de un solo cromosoma padre.
- Un **operador de reordenamiento** es aquél que cambia el orden de los genes de un cromosoma, con la esperanza de juntar los genes que se encuentren relacionados, facilitando así la producción de bloques constructores.
- Se llama **bloque constructor** a un grupo pequeño y compacto de genes que han co-evolucionado de tal forma que su introducción en cualquier cromosoma tiene una alta probabilidad de incrementar la aptitud de dicho cromosoma.

## 2.3 Algoritmos Evolutivos

En términos generales, la teoría Neo-Darwiniana de la evolución natural postula que la diversidad de las especies sobre el planeta puede ser explicada a través de cuatro procesos [24].

- Reproducción: este proceso asegura que el material genético será heredado de una generación a otra.
- Mutación: se presenta cuando existe un error de copiado de material genético de una generación a la siguiente.

- **Competición:** es un proceso natural, en el cual los organismos tienden a enfrentarse continuamente para sobrevivir. Aquellos que sobrevivan podrán heredar su material genético a la siguiente generación.
- **Selección:** en un ambiente en el que sólo puede existir un número limitado de individuos, los organismos que compitan mejor por los recursos, podrán sobrevivir y reproducirse.

La evolución natural puede ser vista como un problema de optimización, donde el objetivo es adaptar las especies a su medio ambiente. Debido a ello, no es raro que se haya tomado inspiración de la teoría Neo-darwiniana para desarrollar técnicas que resuelvan problemas, en un principio de optimización.

Particularmente, los AEs fueron diseñados a partir del mismo principio (debido al hecho de que en la naturaleza, las poblaciones evolucionan a través de generaciones usando mecanismos de selección natural y supervivencia), siendo capaces de generar soluciones a problemas difíciles del mundo real. Sin embargo, el correcto funcionamiento de los AEs depende de la implementación adecuada de los siguientes puntos:

- La codificación de las estructuras de datos usadas para representar las soluciones del problema. Cada solución es conocida como **individuo** y el conjunto de individuos es llamado **población**.
- Los operadores que modifiquen los rasgos de los individuos (también llamados operadores de variación).
- Una función de aptitud que represente el ambiente.
- La supervivencia del más apto, se lleva a cabo por medio del proceso de selección, el cual juega el papel de presión del ambiente.

Hoy en día, los AEs son muy populares en muchas disciplinas, principalmente por presentar diversas ventajas en optimización con respecto a técnicas tradicionales. Algunas de las ventajas que presentan los AEs pueden ser:

- Pueden incorporar fácilmente conocimiento *a priori* del problema a resolver, aunque no necesitan poseerlo.
- Son conceptualmente muy simples.
- Tienen un amplio rango de aplicación.
- Pueden explotar fácilmente arquitecturas paralelas.
- Pueden adaptar sus propios parámetros.
- Son menos susceptibles a quedar atrapados en mínimos o máximos locales<sup>2</sup>.
- Pueden cooperar con otras técnicas de optimización o búsqueda.

La CE se compone principalmente por paradigmas de optimización de aprendizaje máquina y paradigmas inspirados en los mecanismos de la evolución. Sin embargo, por razones sobre todo históricas, suele hablarse de cuatro paradigmas principales:

- Programación Evolutiva
- Estrategias Evolutivas
- Algoritmos Genéticos
- Programación Genética

En las siguientes secciones se describen brevemente cada uno de estos paradigmas pero básicamente, la idea principal detrás de todos ellos es la misma: dada una población de individuos, la presión del ambiente origina la selección natural (*“el supervivencia del más apto”*), lo que origina un aumento en la aptitud de la población. Dada una función de calidad a ser minimizada, se puede generar aleatoriamente un conjunto de soluciones (*v.g.*, individuos o elementos del dominio de la función), y aplicar la función de calidad como una medida de aptitud abstracta (el individuo con aptitud más alta será el mejor). Basados en esta aptitud, algunos de los mejores individuos son

---

<sup>2</sup>Debido a que son técnicas poblacionales.

elegidos como semillas para la siguiente generación, a los cuales se les aplicarán los operadores de variación (recombinación o mutación) para obtener un nuevo conjunto de individuos (descendientes) que podrán competir contra sus padres por un lugar en la siguiente generación. Este proceso se repite hasta que un individuo con suficiente calidad sea encontrado o hasta que otra condición de paro sea satisfecha.

El esquema general de un algoritmo evolutivo se muestra en el Algoritmo 1.

---

**Algoritmo 1** Esquema general de un algoritmo evolutivo.

---

- 1: Inicializar una población con soluciones generadas aleatoriamente
  - 2: Evaluar cada solución
  - 3: **while** no se satisfaga la condición de paro **do**
  - 4:   Seleccionar los padres que se reproducirán
  - 5:   Aplicar los operadores de variación a los padres seleccionados
  - 6:   Evaluar los nuevos candidatos
  - 7:   Seleccionar los mejores individuos (entre todos) para la nueva generación
  - 8: **end while**
- 

### 2.3.1 Programación Evolutiva

A mediados de los 1960s, Lawrence J. Fogel [20] concibió la inteligencia como un comportamiento adaptativo, enfatizando los nexos de comportamiento entre padres e hijos, en vez de buscar emular operadores genéticos específicos. Esta técnica fue llamada **programación evolutiva** (PE).

La PE no requiere de un operador de cruce, ya que es una abstracción del proceso evolutivo a nivel de especies (*p.ej.*, dos especies diferentes no pueden ser combinadas). El mecanismo de selección implementado por la PE es probabilístico (usualmente se usa selección por torneo estocástico). Sin embargo, hoy en día existen diversas variaciones de esta técnica.

El algoritmo básico de la PE se muestra en el Algoritmo 2.

### 2.3.2 Estrategias Evolutivas

Las **estrategias evolutivas** (EE) fueron desarrolladas a mediados de los 1960s en Alemania por Ingo Rechenberg, Peter Bienert y Hans-Paul Schwefel [2]. El método consiste en realizar un ajuste

---

**Algoritmo 2** Algoritmo básico de la programación evolutiva.

---

- 1: Inicializar una población con soluciones generadas aleatoriamente
  - 2: **repeat**
  - 3:   Aplicar el operador de mutación
  - 4:   Evaluar los nuevos candidatos
  - 5:   Seleccionar aquellos individuos que se mantendrán
  - 6:   Generar la nueva población
  - 7: **until** se satisfaga la condición de paro
- 

discreto aleatorio en el “*gene*” de cada individuo mediante un operador inspirado en el mecanismo de mutación existente en la naturaleza.

La versión original fue llamada  $(1 + 1) - EE$  y consiste en usar un padre para generar un único descendiente. Este descendiente pasará a la siguiente generación sólo si es mejor que el padre (este tipo de selección se considera *extintiva*, porque los peores individuos obtienen una probabilidad de ser seleccionados igual a *cero*).

En la  $(1 + 1) - EE$ , un nuevo individuo es generado usando:

$$\bar{x}^{t+1} = \bar{x}^t + \mathcal{N}(0, \sigma)$$

donde  $t$  se refiere a la generación actual y  $\mathcal{N}(0, \sigma)$  es un vector de números Gaussianos independientes con media cero y desviación estándar  $\sigma$ .

El proceso de optimización utilizando una EE no involucra únicamente las variables a optimizar, sino también los parámetros propios de la EE (v.g., la desviación estándar del operador de mutación). Para ello las EE utilizan un proceso conocido como **auto-adaptación**. Por otro lado, la recombinación es posible pero sólo como un operador secundario y el mecanismo de selección implementado usualmente es determinista. El algoritmo general de la EE se muestra en el Algoritmo 3.

### 2.3.3 Algoritmos Genéticos

Los **algoritmos genéticos** (AG) (denominados originalmente “planes reproductivos genéticos”) fueron desarrollados por John H. Holland [25] a principios de 1960s. El interés principal de Holland

**Algoritmo 3** Algoritmo general de la estrategia evolutiva.

- 
- 1: Inicializar la población  $G$  con soluciones generadas aleatoriamente
  - 2:  $t := 0$
  - 3: Evaluar  $G(t)$
  - 4: **repeat**
  - 5:    Seleccionar  $G_{temp}$  de  $G(t)$
  - 6:    Reproducir  $G_{temp}$  para generar  $G(t + 1)$
  - 7:    Aplicar mutación a  $G(t + 1)$
  - 8:    Evaluar  $G(t + 1)$
  - 9:    Seleccionar supervivientes
  - 10:    $t := t + 1$
  - 11: **until** se satisfaga la condición de paro
- 

era estudiar la adaptación natural para resolver problemas de aprendizaje máquina. Hoy en día los AG, son los AEs más populares. El algoritmo general de un AG simple [13] se muestra en el Algoritmo 4.

**Algoritmo 4** Algoritmo genético simple.

- 
- 1: Inicializar la población  $G$  con soluciones generadas aleatoriamente
  - 2:  $t := 0$
  - 3: **repeat**
  - 4:    Evaluar  $G(t)$
  - 5:    Seleccionar  $G_1(t)$  de  $G(t)$
  - 6:    Aplicar cruce y mutación a  $G_1(t)$  para generar  $G(t + 1)$
  - 7:     $t := t + 1$
  - 8: **until** se satisfaga la condición de paro
- 

### 2.3.4 Programación Genética

Nichal Lynn Cramer [15] y después John R. Koza [33] propusieron (de forma independiente) una representación de árbol en algoritmos evolutivos, en la cual se implementaba un operador de cruce para intercambiar subárboles entre diferentes programas de computadora. La diferencia principal entre ambas propuestas, fue que Cramer usaba una función de aptitud interactiva (v.g., el usuario manualmente suministraba el valor de aptitud de cada árbol de la población), mientras Koza lo hacía automáticamente. Hoy en día, la propuesta de Koza es la que prevalece y es conocida como **programación genética** (PG).

Es importante señalar que la programación genética a veces se considera un subcampo de los algoritmos genéticos. El algoritmo general de la PG [3] se muestra en el Algoritmo 5.

---

**Algoritmo 5** Algoritmo general de la programación genética.

---

- 1: Inicializar una población  $G$  con soluciones generadas aleatoriamente
  - 2:  $t := 0$
  - 3: **repeat**
  - 4:   Evaluar los programas de la población  $G(t)$
  - 5:   Seleccionar  $G_1(t)$  de  $G(t)$
  - 6:   Aplicar los operadores genéticos a  $G_1(t)$  para generar  $G(t + 1)$
  - 7:    $t := t + 1$
  - 8: **until** se satisfaga la condición de paro
- 

## 2.4 Inteligencia de cúmulos

Por otro lado, existen sistemas de optimización inspirados en el comportamiento social de las colonias de insectos y otras sociedades colectivas. Estos sistemas son comúnmente llamados **inteligencia de cúmulos (IC)**.

Los sistemas de IC a menudo se consideran dentro de la familia de los algoritmos evolutivos, debido a que utilizan una población de individuos simples (llamados “agentes” en este contexto) que evolucionan a través del tiempo para encontrar una solución aceptable para el problema.

La **optimización mediante cúmulos de partículas** y la **optimización mediante colonia de hormigas** son dos formas de IC. La característica principal de estos sistemas es que en lugar de usar operadores genéticos para adaptarse al ambiente, cada agente evoluciona por sí mismo de acuerdo a su experiencia pasada y su interacción local con otros agentes. A la larga, la interacción local entre agentes causa un comportamiento global inteligente que desemboca en la localización de una solución aceptable.

Debido a que esta tesis recae fuertemente en la optimización mediante cúmulos de partículas las secciones 2.5 y 2.7 proporcionan una descripción más detallada de este enfoque.

## 2.5 Optimización mediante cúmulos de partículas

La **optimización mediante cúmulos de partículas** (PSO, por sus siglas en Inglés) es un algoritmo de optimización estocástico basado en una población de individuos, propuesto originalmente por Eberhart y Kennedy en 1995 [29]. PSO simula el comportamiento social del vuelo de las aves o banco de peces para encontrar rápidamente una solución al problema de optimización ya que aprovecha el comportamiento colectivo que las aves presentan al volar para encontrar comida.

PSO es considerado un AE aunque existen claras diferencias entre ambos enfoques. La principal diferencia es que PSO no implementa un mecanismo de selección de aptitud, en su lugar los individuos persisten a lo largo del tiempo adaptándose a los cambios del ambiente. Aquellos individuos mejor adaptados liderarán al resto de la población a través del tiempo.

En PSO, cada individuo es llamado **partícula** y se representa mediante un vector  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ij})^T$  para  $i = 1, \dots, s$  y  $j = 1, \dots, n$ , donde  $s$  es el número de partículas y  $n$  el número de dimensiones (v.g., el número de parámetros de la función a optimizar). Cada partícula representa una solución candidata al problema de optimización y el conjunto de todas ellas se conoce como **cúmulo**.

Siguiendo la analogía con el vuelo de las aves, en PSO, un cúmulo de partículas vuela a través del espacio de búsqueda hiper-dimensional. Debido a esto, PSO se considera un algoritmo de comportamiento distribuido que desempeña una búsqueda multi-dimensional. La posición de cada partícula está influenciada por la mejor posición visitada por sí misma (v.g., su propia experiencia) y la mejor posición visitada por sus vecinos (v.g., la experiencia de sus partículas vecinas). De acuerdo con esto, la posición de cada partícula,  $\mathbf{x}_i$ , se ajusta en cada iteración<sup>3</sup>, ( $t$ ), del algoritmo usando en la ecuación (2.1),

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (2.1)$$

donde,  $\mathbf{v}_i$  es un vector de la forma  $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{ij})^T$  y representa la velocidad de movimiento,  $i$  representa a la partícula actual y  $j$  representa la dimensión del problema. Por lo tanto,  $\mathbf{v}_i$  está dada

---

<sup>3</sup>A diferencia de los AE clásicos en los sistemas de IC, las generaciones son llamadas iteraciones o pasos.

por la ecuación (2.2).

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1\mathbf{r}_1(t)(\mathbf{y}_i(t) - \mathbf{x}_i(t)) + c_2\mathbf{r}_2(t)(\hat{\mathbf{y}}_i(t) - \mathbf{x}_i(t)) \quad (2.2)$$

A partir de la ecuación (2.2):

- $w$  es la velocidad de inercia [42].
- $c_1$  y  $c_2$  son dos constantes, llamados coeficientes de aceleración.
- $\mathbf{r}_1$  y  $\mathbf{r}_2$  son dos vectores aleatorios de la forma  $\mathbf{r}_1 = (r_{11}, r_{12}, \dots, r_{1j})$  y  $\mathbf{r}_2 = (r_{21}, r_{22}, \dots, r_{2j})$ . Su valor está dado por una distribución uniforme con límite inferior 0 y límite superior 1,  $r_{ij}, r_{2j} \sim \mathcal{U}(0, 1)$ .
- $\mathbf{y}_i(t)$  es un vector de la forma  $\mathbf{y}_i(t) = (y_{i1}, y_{i2}, \dots, y_{ij})(t)$ . Representa la mejor posición de la partícula  $x_i$  encontrada por sí misma en el paso  $t$ .
- $\hat{\mathbf{y}}_i(t)$  es un vector de la forma  $\hat{\mathbf{y}}_i(t) = (\hat{y}_{i1}, \hat{y}_{i2}, \dots, \hat{y}_{ij})(t)$ . Representa la mejor posición obtenida por los vecinos de la partícula  $x_i$  en el paso  $t$ .

Es importante destacar que la velocidad de cada partícula  $x_i$  se determina a partir de tres factores importantes:

- La **velocidad de inercia**, expresada en la ecuación (2.2) como  $\mathbf{v}_i(t)$ . Actúa como un plazo de impulso para prevenir oscilaciones excesivas en la dirección de búsqueda.
- El **componente cognitivo**, expresado en la ecuación (2.2) como  $c_1\mathbf{r}_1(t)(\mathbf{y}_i(t) - \mathbf{x}_i(t))$ . Representa la mejor posición que una partícula ha encontrado por sí misma donde  $c_1$  representa la fuerza de atracción de esta posición. Análogamente con el vuelo de las aves, el componente cognitivo representa la tendencia natural de los individuos a regresar al ambiente donde experimentaron su mejor desempeño.
- El **componente social**, expresado en la ecuación (2.2) como  $c_2\mathbf{r}_2(t)(\hat{\mathbf{y}}_i(t) - \mathbf{x}_i(t))$ . Representa la mejor posición que todo el vecindario ha logrado obtener. Esta posición se obtiene,

comparando la posición de sus vecinos con la propia donde  $c_2$  indica la fuerza de atracción de la mejor posición alcanzada por el cúmulo. Análogamente con el vuelo de las aves, este componente representa la tendencia de los individuos a seguir al individuo más apto.

Para establecer el componente social  $\hat{y}_i$  de cada partícula es necesario definir la topología de interconexión del vecindario (v.g., aquellos vecinos con los que se intercambiará información directamente). No es raro que a lo largo de la historia de PSO se hayan realizado estudios referentes a la forma de comunicar las partículas dentro del cúmulo [27, 30, 26], para determinar si la forma de transmitir la información de vuelo entre partículas afecta la convergencia del algoritmo. Estos estudios han mostrado que la topología de interconexión implementada afecta significativamente el desempeño del cúmulo y que además su efecto depende de la función a optimizar. Por lo tanto, algunas topologías de interconexión trabajarán relativamente bien en algunas funciones mientras que en otras presentarán problemas [27].

Históricamente, PSO ha sido estudiado en dos formas de superposición de vecinos<sup>4</sup> llamadas *lbest* y *gbest*, las cuales se describen a continuación.

### 2.5.0.1. PSO versión *gbest*

Si todas las partículas del cúmulo se comunican usando una topología completamente conectada (mostrada en la figura 2.1), entonces el algoritmo resultante de PSO se denomina como **mejor global** (*gbest*). En esta topología, cada partícula compara su desempeño contra todos los miembros de la población.

En esta versión de PSO, el componente social  $\hat{y}_i$  se refiere a la partícula con la mejor posición en todo el cúmulo. Por lo tanto, todos los individuos son atraídos hacia la mejor solución encontrada por cualquier miembro de la población. Asumiendo un problema de minimización,  $\hat{y}_i$  está dado por la ecuación (2.3):

$$\hat{y}_i(t) \in \{\mathbf{y}_0(t), \mathbf{y}_1(t), \dots, \mathbf{y}_s(t)\} = \min\{f(\mathbf{y}_0(t)), f(\mathbf{y}_1(t)), \dots, f(\mathbf{y}_s(t))\} \quad (2.3)$$

<sup>4</sup>También llamada interconexión o comunicación de partículas.

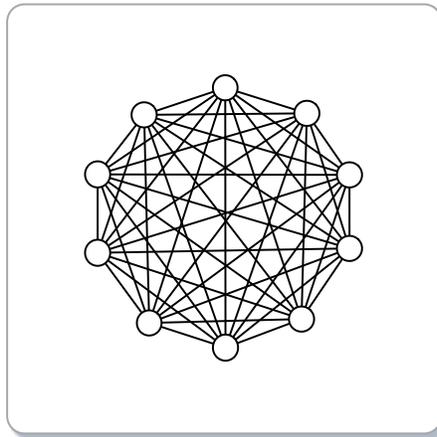


Figura 2.1: Topología completamente conectada.

### 2.5.0.2. PSO versión *lbest*

Si las partículas en el cúmulo se comunican en forma de anillo (mostrada en la figura 2.2), entonces el algoritmo resultante se conoce como **mejor local** (*lbest*). En esta versión de PSO, el cúmulo completo se divide en superposiciones de vecinos, v.g., dos partículas preguntan al mismo vecino por su mejor posición. La tradición sugiere que cada partícula posea  $k$  partículas vecinas adyacentes, comúnmente con  $k = 2$ .

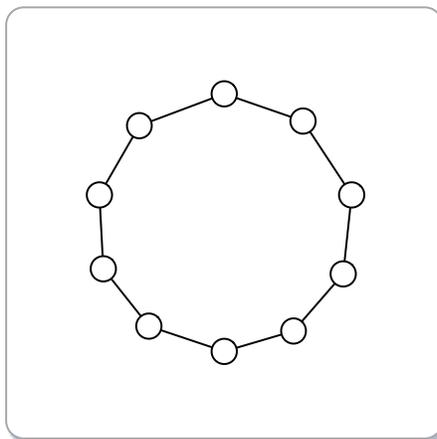


Figura 2.2: Topología en anillo.

La diferencia principal entre ambas versiones del algoritmo, es que la versión *lbest* promueve la **exploración** del espacio de búsqueda, mientras que la versión *gbest* favorece la **explotación**. Se

denomina exploración, al proceso de visitar completamente nuevas regiones del espacio de búsqueda, para ver si se puede encontrar una zona prometedora o bien para evitar quedar atrapado en óptimos locales (v.g., involucra grandes saltos hacia lo desconocido). Mientras que la explotación normalmente involucra movimientos finos, de tal manera, es buena para encontrar óptimos locales. De lo anterior se desprende que, la versión *gbest* de PSO tiende a converger más rápido que la versión *lbest*, pero es más susceptible a converger a óptimos locales.

El Algoritmo 6 muestra el pseudocódigo de PSO en su versión *gbest*. Básicamente, las partículas se inicializan aleatoriamente en todo el espacio de búsqueda (definido por los límites inferior y superior de la función). Al iniciar el algoritmo  $y_i$ , la mejor posición de la partícula  $i$  encontrada por sí misma es igual a la posición inicial y se elige la partícula con mayor aptitud en todo el cúmulo asignando su valor a  $\hat{y}_i$  (la partícula con mejor posición en todo el cúmulo). Posteriormente, el cúmulo vuela a través del espacio de búsqueda hasta que cierto criterio de terminación sea satisfecho<sup>5</sup>. La velocidad de vuelo de cada partícula se actualiza usando la ecuación (2.2) y su posición usando la ecuación (2.1). Al evaluar la nueva posición de cada partícula se determina si la nueva posición es mejor que la almacenada hasta el momento, de ser así, se reemplaza  $y_i$  con la nueva posición. De igual forma, se obtiene nuevamente la partícula con mejor posición en todo el cúmulo y se compara con la almacenada en  $\hat{y}_i$ , si la nueva posición es mejor, entonces se reemplaza el valor original.

## 2.6 Topologías adicionales de interconexión

En la literatura especializada se pueden encontrar diferentes trabajos que estudian los efectos que las diferentes topologías de comunicación tienen sobre el algoritmo [10, 30, 27, 26]. La mayoría de estos estudios son propuestas teóricas e implementaciones de otras pocas formas de comunicar las partículas. Los resultados de estos estudios, muestran que existen topologías que se desempeñan mejor que otras, dependiendo del problema a resolver, aunque sólo un pequeño conjunto de problemas de prueba y de topologías de interconexión han sido explorados.

---

<sup>5</sup>El criterio de terminación, se refiere al criterio adoptado para concluir la ejecución del algoritmo. Usualmente, se asocia al número total de iteraciones que el algoritmo realizará.

---

**Algoritmo 6** Algoritmo de PSO versión *gbest*.

---

```
1: for cada partícula  $i$  do
2:   Inicializar  $\mathbf{x}_i$  y  $\mathbf{v}_i$  aleatoriamente
3:   Evaluar  $\mathbf{x}_i$  en la función de aptitud
4:   Inicializar  $\mathbf{y}_i = \mathbf{x}_i$ 
5: end for
6: Seleccionar  $\hat{\mathbf{y}}_i$ 
7: repeat
8:   for cada partícula  $i$  do
9:     for cada dimension  $j$  do
10:      Calcular la velocidad de  $v_{ij}$  usando:
          
$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) + c_2r_{2j}(t)(\hat{y}_{ij}(t) - x_{ij}(t))$$

11:      Actualizar la posición de  $x_{ij}$  usando:
          
$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$$

12:     end for
13:     Evaluar la nueva posición de  $\mathbf{x}_i$ 
14:     if  $\mathbf{x}_i \leq \mathbf{y}_i$  then
15:        $\mathbf{y}_i = \mathbf{x}_i$ 
16:     end if
17:     if  $\mathbf{x}_i \leq \hat{\mathbf{y}}_i$  then
18:        $\hat{\mathbf{y}}_i = \mathbf{x}_i$ 
19:     end if
20:   end for
21: until alcanzar el criterio de terminación
```

---

Adicionalmente, se ha observado que después de unas cuantas iteraciones del algoritmo, las partículas dentro del cúmulo se agrupan en una o varias regiones del espacio de búsqueda. De tal manera se desprende la posibilidad de que exista una topología de comunicación, o que dividan el cúmulo de partículas en dos o más grupos para permitir que las partículas converjan a diversas regiones del espacio de búsqueda desde el inicio del algoritmo.

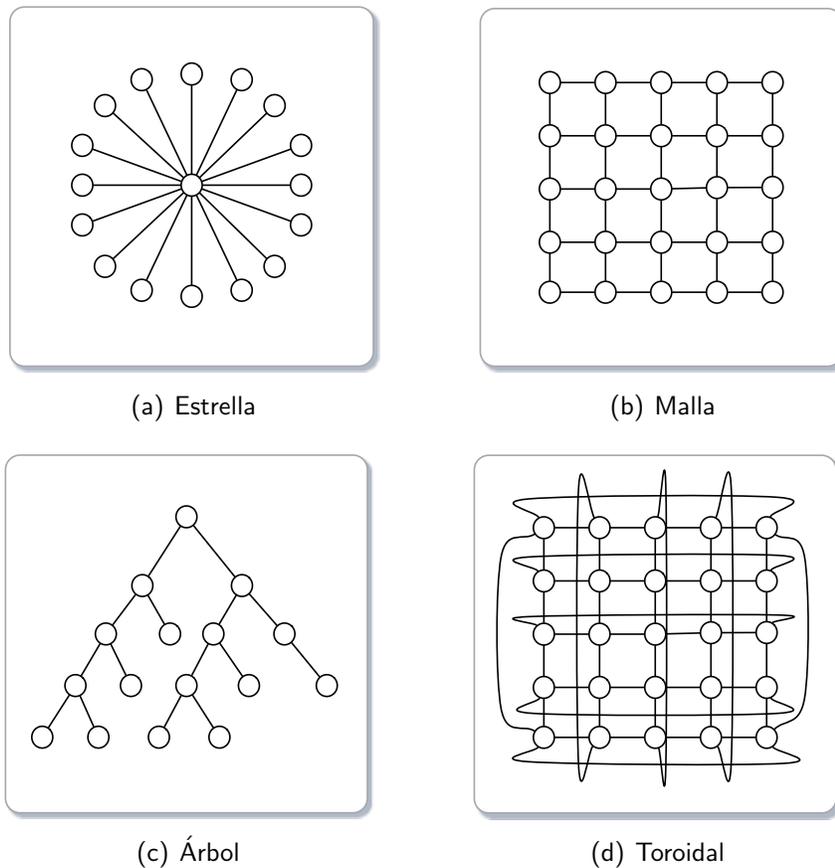


Figura 2.3: Topologías de interconexión usadas en este estudio.

### 2.6.1 Topología en anillo

La **topología en anillo** también es conocida como la versión *lbest* en PSO (mostrada en la figura 2.2). En esta topología cada partícula es afectada por el mejor desempeño de sus  $k$  vecinos inmediatos en la población.

En la topología en anillo, las partículas están conectadas estrechamente y por lo tanto pueden reaccionar cuando una partícula tiene un incremento en su aptitud. Esta reacción se diluye proporcionalmente con respecto a la distancia. Entonces, es posible que un segmento de la población converja a un óptimo local, mientras otros segmentos converjan a diferentes puntos del espacio de búsqueda. Aunque eventualmente, una buena solución atraerá al cúmulo hacia su dirección.

### 2.6.2 Topología completamente conectada

En la **topología completamente conectada**, también conocida como *gbest* (mostrada en la figura 2.1), todas las partículas están conectadas directamente unas con otras. Por lo tanto, todas las partículas en el cúmulo redireccionan su vuelo hacia la posición de aquella partícula que presente la mejor aptitud.

Kennedy *et al*en [31, 30], comentan que la topología *gbest* tiende a converger más rápido que la topología *lbest*. No obstante, también es más susceptible a converger a óptimos locales. Aún así, *gbest* ha sido la topología más usada por la comunidad especializada.

### 2.6.3 Topología en estrella

En la **topología en estrella** (mostrada en la figura 2.3(a)) la información de vuelo de todas las partículas pasa a través de un sólo individuo.

En esta topología, una partícula (el nodo central) influye en el vuelo de todas las partículas y su vuelo es influenciado por la mejor partícula del cúmulo (excluyéndose así misma). Al iniciar el algoritmo, la partícula central es elegida aleatoriamente de entre todo el cúmulo, posteriormente en cada iteración,  $t$ , la partícula central cambiará a aquella que presente mejor aptitud.

La topología en estrella aísla los individuos unos de otros, ya que la información de vuelo sólo puede ser transmitida por la partícula central. El papel de partícula central es comparar el desempeño de cada individuo de la población y ajustar su propia trayectoria hacia aquel que presente el mejor desempeño. De esta manera, el individuo central actúa como memoria a corto plazo o filtro, alentando la velocidad de transmisión de información de aquellas soluciones buenas hacia el resto de la población.

Esta característica previene la convergencia prematura aunque se puede presuponer que también destruye la habilidad de colaboración entre la población.

#### 2.6.4 Topología en malla

En la **topología en malla** (mostrada en la figura 2.3(b)), cada nodo posee un número diferente de conexiones. Los nodos en las cuatro esquinas de la malla se conectan tan solo con sus dos vecinos adyacentes, los vecinos en los límites de la malla están conectados a sus tres vecinos adyacentes y los nodos en el centro de la malla tienen cuatro conexiones (correspondientes a sus vecinos inmediatos, localizados al norte, sur, este y oeste de la partícula).

Puede observarse que existe una superposición de conexiones vecinas entre partículas (v.g., dos partículas podrían preguntar por la mejor posición vecina a una misma partícula), permitiendo la redundancia en el proceso de búsqueda. En la implementación de esta topología, las partículas son asignadas a los nodos de la malla de izquierda a derecha y de arriba a bajo. La malla se construye al inicio del algoritmo y permanece igual a lo largo de la ejecución del algoritmo.

#### 2.6.5 Topología en árbol

La **topología en árbol** también es conocida como **topología jerárquica** (mostrada en la figura 2.3(c)). Esta topología posee un nodo raíz (el nivel más alto de la jerarquía), el cual se conecta con uno o más individuos que están un nivel abajo de la jerarquía (v.g., el segundo nivel), mientras que cada uno de los nodos del segundo nivel podrían ser a su vez, nodos raíz de uno o más nodos del siguiente nivel (v.g., el tercer nivel) y así sucesivamente hasta que todos los nodos hayan tomado su lugar en la jerarquía. Al incorporarse todos los nodos en la estructura de conexión, la jerarquía del árbol es simétrica.

Para fines de este trabajo, la topología en árbol se construyó como un árbol binario (usando el índice de las partículas como nodos), el nodo raíz es seleccionado aleatoriamente de entre todo el cúmulo y el resto de partículas son distribuidas en las ramas del árbol, tratando de mantener el balance entre ellas. El nodo raíz (principal), busca por la mejor posición vecina entre sus hijos

inmediatos (v.g., el segundo nivel) y él mismo, para redireccionar su vuelo hacia esa posición. El segundo nivel, así como todos los que presenten hijos en sus ramas, buscan la mejor posición vecina en sus nodos hijos y su nodo padre. El mismo procedimiento se sigue hasta el último nivel, donde los nodos hoja sólo preguntarán por la mejor posición a su nodo padre.

### 2.6.6 Topología toroidal

Topológicamente, un toroide es una superficie cerrada definida como el producto de dos círculos (mostrado en la figura 2.3(d)), comúnmente llamada *torus* [21].

La **topología toroidal** es muy similar a la topología en malla, excepto porque todas las partículas en el cúmulo poseen cuatro vecinos adyacentes. A diferencia de la topología en malla, las esquinas de la topología toroidal se conectan con su partícula simétrica en el otro extremo de la malla (v.g., como si se tratase de un círculo cerrado). Geométricamente, esto mismo ocurre con los límites de un toroide.

En la topología toroidal, la asignación de las partículas a los nodos es similar a la topología en malla.

### 2.6.7 Algoritmos de agrupamiento

El agrupamiento está definido por el número promedio de vecinos que dos nodos conectados, cualesquiera, tienen en común [27]. Por naturaleza, en PSO, las partículas se agrupan en más de una región del espacio de búsqueda después de algunas iteraciones del algoritmo. Este agrupamiento, usualmente, indica la presencia de óptimos locales. Por lo tanto, es una idea razonable investigar si la información acerca de la distribución de las partículas podría ser aprovechada para mejorar la trayectoria de las partículas y por ende, la búsqueda [31].

En este trabajo se han implementado dos tipos de algoritmos de agrupamiento: *k-means* y *jerárquico*, con la finalidad de estudiar el comportamiento de las trayectorias que las partículas pueden tomar cuando la información espacial acerca de ellas se toma en cuenta para ajustar el vuelo.

### 2.6.7.1. Algoritmo de agrupamiento *k-means*

La idea principal del algoritmo de **agrupamiento *k-means*** es definir  $k$  centroides *a priori* de la ejecución del algoritmo. Los centroides, uno por cada grupo, deben ser colocados de manera estratégica en todo el espacio de búsqueda, debido a que diferentes ubicaciones causan diferentes resultados. El siguiente paso es asociar cada partícula de la población al centroide más cercano, el proceso se repite hasta que todas las partículas de la población pertenecen a un grupo previamente definido por el centroide. Se puede notar que cada centroide actúa como un punto atractor de la población, debido a que capta las partículas más cercanas de la población. Para asociar cada partícula de la población al centroide más cercano se utiliza la distancia euclidiana como medida de similitud.

En nuestra implementación se han fijado *a priori* cuatro centroides, que inicialmente son asignados a las posiciones de cuatro partículas escogidas al azar de entre todo el cúmulo. Entonces todas las partículas restantes son asociadas al centroide más cercano mediante su distancia euclidiana. Es importante señalar que en cada iteración del algoritmo el centroide de cada grupo cambia (se toma la partícula que presente mejor aptitud en cada grupo) y el resto de las partículas son asociadas nuevamente a los centroides elegidos.

### 2.6.7.2. Algoritmo de agrupamiento jerárquico

En el algoritmo de **agrupamiento jerárquico** los grupos son construidos mediante medidas de similaridad entre grupos. En la primera etapa del algoritmo, se considera que existe un número de grupos igual al número de partículas en la población (v.g., cada partícula es considerada un grupo), entonces en cada etapa del algoritmo, dos grupos se unirán si la distancia entre ellos es la más corta (v.g., los más similares) de entre todos los grupos. El algoritmo continua hasta obtener el número de grupos deseados. Al final la distancia entre un grupo y otro será la más grande posible (es posible que al final un grupo posea un sólo individuo).

En la implementación del algoritmo de agrupamiento jerárquico, al igual que con el algoritmo *k-means*, se buscan cuatro grupos al inicio del algoritmo los cuales cambian en cada iteración.

## 2.7 Variaciones de PSO libres de parámetros

En algunos enfoques evolutivos, tal como los algoritmos genéticos, la información pasada acerca del proceso de búsqueda tiende a perderse al efectuarse una generación del algoritmo, normalmente la población actual es la única “memoria” usada por los individuos. Debido a la naturaleza de PSO, cada partícula en el cúmulo implementa una memoria,  $y_i$ , lo que permite beneficiarse de su experiencia pasada en las siguientes iteraciones del proceso. Esta característica, además de su notable velocidad de convergencia y facilidad de implementación han promovido su estudio usando problemas de optimización mono-objetivo, así como multi-objetivo.

Sin embargo, estudios empíricos y teóricos han demostrado que el comportamiento de convergencia de PSO está fuertemente ligado a los valores del peso de la inercia,  $w$ , y los valores de los coeficientes de aceleración,  $c_1$  y  $c_2$  [48, 14]. De tal manera que, seleccionar valores incorrectos para estos parámetros podría ocasionar divergencia en el algoritmo o ciclar la trayectoria de las partículas.

Existen diversas recomendaciones para los valores de dichos parámetros [48, 14, 47], siendo los más utilizados:

$$w = 0.72$$
$$c_1 = c_2 = 1.49$$

Teóricamente se ha demostrado que estos valores son convergentes en gran parte de los problemas de prueba mono-objetivo, aunque en la práctica no son universalmente aplicables a todo tipo de problema, ya sean mono o multi-objetivo [47].

Un gran número de variaciones de PSO han sido desarrollados principalmente para mejorar la exactitud de las soluciones, acelerar la convergencia del algoritmo o eliminar la dependencia de dichos parámetros [31]. A continuación se describe algunas variaciones de PSO.

### 2.7.1 PSO barebones

van den Bergh y Engelbrecht [48], y Clerc y Kennedy [14] probaron teóricamente que cada partícula converge a un peso promedio, dependiendo de la mejor posición encontrada por sí misma

y la mejor posición encontrada por sus vecinos, esto puede expresarse en la ecuación (2.4).

$$\lim_{t \rightarrow +\infty} x_{ij}(t) = \frac{c_1 y_{ij} + c_2 \hat{y}_{ij}}{c_1 + c_2} \quad (2.4)$$

Este comportamiento provee las bases para la versión de PSO **barebones** (llamada en este trabajo **PSOBB**). PSOBB fue propuesta por Kennedy en el 2003 [28]. El algoritmo de PSOBB reemplaza las ecuaciones (2.1) y (2.2) por la ecuación (2.5).

$$x_{ij}(t+1) = \mathcal{N}\left(\frac{y_{ij}(t) + \hat{y}_j(t)}{2}, |y_{ij}(t) - \hat{y}_j(t)|\right) \quad (2.5)$$

En PSOBB, las posiciones de las partículas son seleccionadas aleatoriamente a partir de una distribución Gaussiana con media igual al promedio de su mejor posición personal y la mejor posición encontrada por sus vecinos (*v.g.*, el atractor del cúmulo); y desviación estándar igual a la diferencia entre la mejor posición encontrada por sí misma y la mejor posición encontrada por sus vecinos, la cual se aproxima a cero conforme  $t$  se incrementa. En el límite, todas las partículas convergerán al punto atractor. Esta versión de PSO facilita la exploración del espacio de búsqueda.

Kennedy también propuso una versión alternativa a PSOBB (referida en este trabajo como **PSOBBEXP**). La cual reemplaza las ecuaciones (2.1) y (2.2) con la ecuación (2.6).

$$x_{ij}(t+1) = \begin{cases} \mathcal{N}\left(\frac{y_{ij}(t) + \hat{y}_j(t)}{2}, |y_{ij}(t) - \hat{y}_j(t)|\right) & \text{Si } \mathcal{U}(0, 1) > 0.5 \\ y_{ij}(t) & \text{en otro caso} \end{cases} \quad (2.6)$$

Basado en la ecuación anterior, existe el 50% de probabilidad que la  $j$ -ésima dimensión de la partícula cambie a la mejor posición personal. Esta versión de PSO favorece la explotación de la mejor posición personal encontrada por cada partícula.

### 2.7.2 PSO barebones y evolución diferencial

La **evolución diferencial** (ED) es un algoritmo evolutivo, propuesto por Storn y Price [45]. Aunque, la ED comparte similitudes con algunos AEs, difiere significativamente de ellos en el

sentido que la información de distancia y dirección de la población actual se usan para guiar el proceso de búsqueda.

La ED usa las diferencias entre vectores seleccionados aleatoriamente (individuos), como la fuente de variación aleatoria para un tercer vector (individuo), referido como el vector “objetivo”. Entonces, agregando vectores de diferencia ponderada al vector objetivo se generan soluciones de prueba. Este proceso es el operador de mutación ya que el vector objetivo es mutado. Entonces aplicando recombinación se produce un descendiente el cual se acepta para la siguiente generación sólo si supera la aptitud del padre.

El algoritmo ED se describe con más detalle siguiendo los tres operadores de evolución: mutación, cruza y selección.

*Mutación:* para cada padre,  $x_i(t)$ , de la generación  $t$ , se crea un vector de prueba,  $v_i(t)$ , mutando un vector objetivo. El vector objetivo,  $x_{i_3}(t)$ , se selecciona aleatoriamente, con  $i \neq i_3$ . Entonces, dos individuos  $x_{i_1}(t)$  y  $x_{i_2}(t)$  son elegidos aleatoriamente con  $i_1 \neq i_2 \neq i_3 \neq i$  y se calcula el vector de diferencia,  $x_{i_1} - x_{i_2}$ . Posteriormente, se calcula el vector de prueba de acuerdo a la ecuación (2.7):

$$\mathbf{v}_i(t) = \mathbf{x}_{i_3}(t) + F(\mathbf{x}_{i_1}(t) - \mathbf{x}_{i_2}(t)) \quad (2.7)$$

donde el último término representa el tamaño de paso de mutación.  $F$  es un factor de escala, usado para controlar la amplificación de la variación diferencial. Notemos que  $F \in (0, \infty)$

*Cruza:* la ED sigue un enfoque de recombinación discreta, donde elementos del vector padre,  $x_i(t)$ , son combinados con elementos del vector de prueba,  $\mathbf{v}_i(t)$  para producir un descendiente,  $\mu_i(t)$ , usando cruza binomial:

$$\mu_{ij}(t) = \begin{cases} v_{ij}(t) & \text{Si } \mathcal{U}(0, 1) < p_r \text{ o } j = r \\ x_{ij}(t) & \text{en otro caso} \end{cases}$$

donde  $j = 1, \dots, N_d$  se refiere a una dimensión específica,  $N_d$  es el número de genes (parámetros) de un sólo cromosoma,  $r \sim \mathcal{U}(1, \dots, N_d)$  y  $p_r$  es la probabilidad de reproducción (con  $p_r \in [0, 1]$ ).

Entonces, cada descendiente es una combinación lineal estocástica de tres individuos elegidos

aleatoriamente cuando  $\mathcal{U}(0, 1) < p_r$ , en otro caso el descendiente hereda directamente del padre. Incluso cuando  $p_r = 0$  al menos uno de los parámetros del descendiente diferirá de los del padre (forzado por la condición  $j = r$ ).

*Selección:* la ED implementa un procedimiento sencillo, el descendiente generado,  $\boldsymbol{\mu}_i(t)$ , reemplaza al padre,  $\mathbf{x}_i(t)$ , sólo si la aptitud del descendiente es mejor que la del padre.

Aunque estudios empíricos han mostrado que la convergencia de la ED es relativamente insensible a los valores de los parámetros de control, su desempeño puede ser mejorado en gran medida si el valor de sus parámetros se optimiza.

Abbass en [1] propone un enfoque auto-adaptativo basado en Pareto para la ED enfocado a problemas de optimización multi-objetivo llamado *Self-adaptive Pareto DE* (SPDE, por sus siglas en Inglés). En SPDE, para cada variable se genera el parámetro  $F$  a partir de una distribución normal,  $\mathcal{N}(0, 1)$ . Cada individuo,  $i$ , tienen su propia probabilidad de reproducción,  $p_i$ . El parámetro  $p_i$  es inicializado primero, para cada individuo en la población a partir de una distribución uniforme entre 0 y 1, entonces  $p_i$  se adapta como:

$$p_i(t) = p_{i_1}(t) + \mathcal{N}(0, 1) \times (p_{i_2}(t) - p_{i_3}(t)) \quad (2.8)$$

donde  $i_1, i_2, i_3 \sim \mathcal{U}(1, \dots, s)$ , y  $i_1 \neq i_2 \neq i_3$ . De acuerdo con Abbass [1] el enfoque propuesto se desempeña bien, comparado con otros enfoques evolutivos de optimización multi-objetivo.

Por otro lado, Omran *et al*[40] propone una estrategia auto-adaptativa para ED, la cual hace uso del enfoque denotado en la ecuación (2.8) para adaptar dinámicamente el factor de escala. Entonces, el operador de mutación cambia a la forma mostrada en la ecuación (2.9).

$$\mathbf{v}_i(t) = \mathbf{x}_{i_3}(t) + F_i(t)(\mathbf{x}_{i_1}(t) - \mathbf{x}_{i_2}(t)) \quad (2.9)$$

donde

$$F_i(t) = F_{i_4}(t) + \mathcal{N}(0, 0.5) \times (F_{i_5}(t) - F_{i_6}(t)) \quad (2.10)$$

con  $i_4 \neq i_5 \neq i_6$  y  $i_4, i_5, i_6 \sim \mathcal{U}(1, \dots, s)$ .

La probabilidad de cruce puede ser muestreada a partir de una distribución Gaussiana como en la ecuación (2.10) o adaptada de acuerdo a la ecuación (2.8).

PSO tiene la ventaja de poseer pruebas formales para mostrar que las partículas convergen a un punto atractor, mientras que PSOBBD utiliza esta información muestreando soluciones candidatas, distribuidas normalmente al rededor del punto atractor. La ED tiene la ventaja que no favorece a ninguna distribución definida previamente para muestrear tamaños de pasos de la mutación, además que su operador de selección sigue un proceso de “escalando la colina”. Los tamaños de paso de la mutación son determinados como diferencias entre individuos en la población actual, además implementa estrategias auto-adaptativas de ED para evitar la optimización de control de parámetros.

La versión de PSO *barebones* y evolución diferencial, utiliza las bondades de PSO *barebones* y las estrategias ED auto-adaptativas para formar un algoritmo de optimización híbrido, llamado en esta tesis **PSOBBDDE**.

Propuesta por [39], en PSOBBDDE la posición de cada partícula se actualiza usando la ecuación (2.11).

$$x_{ij}(t) = \begin{cases} p_{ij}(t) + r_{2j} \times (x_{i_{1j}}(t) - x_{i_{2j}}(t)) & \text{Si } \mathcal{U}(0, 1) > p_r \\ y_{i_{3j}}(t) & \text{en otro caso} \end{cases} \quad (2.11)$$

donde

$$p_{ij}(t) = r_{1j}(t)y_{ij}(t) + (1 - r_{1j}(t))\hat{y}_{ij}(t) \quad (2.12)$$

con  $i_1, i_2, i_3 \sim \mathcal{U}(1, \dots, s)$ ,  $i_1 \neq i_2 \neq i$ ,  $r_{1j}, r_{2j} \sim \mathcal{U}(0, 1)$  y  $p_r$  es la probabilidad de recombinación.

A partir de la ecuación (2.12),  $\mathbf{p}_i(t)$  es el atractor de partículas, resultado de un promedio ponderado estocástico de la mejor posición personal y la mejor posición global, tomados del PSOBBD.

A partir de la ecuación (2.11) el operador de mutación de la ED se usa para para explorar al rededor del atractor actual,  $\mathbf{p}_i(t)$ , lo cual se consigue agregando un vector de diferencia al atractor.

El mecanismo de cruce se consigue seleccionando aleatoriamente una mejor posición personal,  $\mathbf{y}_{i_3}$ , ya que estas (mejores) posiciones personales representan una memoria de las mejores soluciones encontradas por cada individuo desde que inicia el proceso de búsqueda.

Notemos también que el factor de escala es una variable aleatoria.

La actualización de posición en la ecuación (2.12) se consigue de la siguiente manera: para una proporción  $(1 - p_r)$  de las actualizaciones, se usa información de una mejor posición personal,  $\mathbf{y}_{i_3}$ , seleccionada aleatoriamente (facilitando la explotación); mientras que para una proporción de  $p_r$  de las actualizaciones del tamaño de paso, son mutaciones del punto atractor,  $\mathbf{p}_i$  (facilitando la exploración). Los tamaños de paso de mutación están basados en el vector de diferencia entre dos partículas seleccionadas aleatoriamente  $\mathbf{x}_{i_1}$  y  $\mathbf{x}_{i_2}$ .

A partir de lo anterior, la versión de PSOBBDDE mantiene un buen balance entre exploración y explotación. De tal manera, debe ser notado que la explotación de las mejores posiciones personales no se enfoca en una posición específica y que la mejor posición personal,  $\mathbf{y}_{i_3}$ , se selecciona aleatoriamente en cada actualización de posición.

# 3

## Configuración de parámetros en Algoritmos Evolutivos

### 3.1 Introducción

La configuración de parámetros de un AE es fundamental para su buen desempeño. Sin embargo, es un problema difícil, puesto que existe un gran número de opciones posibles y únicamente se tiene un pequeño conocimiento del efecto de los parámetros en el desempeño del AE. Además, dependiendo del problema a manejar, los valores de los parámetros del AE podrían cambiar, de tal manera que diversos tipos de problemas podrían necesitar una configuración diferente en sus parámetros (*v.g.*, no existe una configuración general válida para todo tipo de problema).

El propósito de este capítulo es presentar una descripción general de los enfoques principales para configurar los parámetros de un AE, identificar las corrientes actuales de investigación y argumentar otros enfoques alternativos. Debido a que este trabajo recae fuertemente en la calibración de parámetros, se pone especial interés en este enfoque, el cual ha venido tomando fuerza al ser manejado mediante estadística computacional.

## 3.2 Configuración de parámetros

La configuración de parámetros en AEs comúnmente se divide en dos clases: **control de parámetros** y **calibración de parámetros** [18]. La primera clase hace referencia los valores de los parámetros que cambian durante la ejecución del AE, en este caso se necesitan parámetros iniciales y estrategias de control adecuadas que permitan al algoritmo adaptarse automáticamente ante cualquier cambio producido en el problema. Por otro lado, la segunda clase es más simple, puesto que los valores de los parámetros no cambian durante la ejecución del AE, por lo tanto sólo se necesitan conocer los valores iniciales de los parámetros antes de ejecutar el AE.

La ejecución de un AE es un proceso adaptativo intrínsecamente dinámico, debido a los diversos factores que influyen en él, *p.ej.*, la semilla de números aleatorios o los valores de sus parámetros que utiliza (diferentes valores podrían ser óptimos en diferentes etapas del proceso evolutivo), por lo tanto el uso de parámetros rígidos que no cambien su valor (durante la ejecución del AE) es opuesto a su naturaleza estocástica. El control de parámetros ha sido visto como una forma de mantener la propiedad dinámica del AE, conforme se realizan cambios a los valores de sus parámetros.

Es común clasificar las estrategias para el control de parámetros en tres enfoques: **deterministas**, **adaptativas** y **auto-adaptativas**.

- El control de parámetros determinístico es aquel donde los valores de un parámetro es alterado por alguna regla determinista. Esta regla modifica el parámetro de estrategia de forma predeterminada (*v.g.*, especificada por el usuario) sin usar alguna reacción de la búsqueda. Usualmente, la regla se activa cuando un número predeterminado de generaciones se ha alcanzado.
- El control de parámetros adaptativo ocurre cuando alguna reacción de la búsqueda sirve de entrada al mecanismo usado para determinar la dirección (o magnitud) de cambio del parámetro de estrategia. Aunque, la subsecuente acción del AE podría decidir si los nuevos valores persisten y se propagan a través de la población, el punto importante es que el mecanismo de actualización usado para controlar los valores de los parámetros se suministra externamente, posiblemente siendo parte del ciclo evolutivo estándar.

- El control de parámetros auto-adaptativo se refiere a usar la idea de la evolución para auto-adaptar los parámetros. Esta técnica es similar a un algoritmo genético, donde los parámetros a ser adaptados se codifican en cromosomas y son sometidos a los operadores de variación (cruza y mutación), los cromosomas con mejor valor son los mejores individuos y por lo tanto serán los que se reproduzcan para generar descendientes propagando así los mejores valores de los parámetros. La principal diferencia entre el control de parámetros adaptativo y el control de parámetros auto-adaptativo es que en este último, la asignación y actualización de los diferentes parámetros de estrategia es enteramente implícito, *v.g.*, ellos son los operadores de variación y selección del ciclo evolutivo.

Por otro lado, la calibración de parámetros, es el enfoque más usado en el diseño de algoritmos. La técnica consiste en probar diferentes valores para los parámetros en los problemas de prueba y seleccionar aquellos que arrojen mejores resultados.

Las desventajas de usar calibración de parámetros basados en experimentación se pueden resumir en los siguientes puntos:

- Los parámetros de un AE no son independientes, intentar probar todas las posibles combinaciones es prácticamente imposible.
- La calibración de parámetros consume mucho tiempo de cómputo, a pesar de que los parámetros sean optimizados uno por uno dejando de lado las interacciones.
- Los valores de los parámetros seleccionados no son necesariamente óptimos a pesar del esfuerzo computacional invertido.

Sin embargo, tanto el control de parámetros como la calibración de parámetros demandan un costo computacional elevado, pero a diferencia de la primera, la calibración de parámetros sólo se lleva a cabo una vez (*v.g.*, antes de la ejecución del algoritmo).

Por otro lado, para establecer mecanismos que controlen los parámetros dinámicamente durante la ejecución de un AE (*v.g.*, control de parámetros), se necesita realizar previamente una caracterización de los mismos, procedimiento que usualmente se realiza mediante una previa calibración de los

parámetros. Es por ello que la calibración de parámetros ha sido vista como una alternativa favorable al momento de configurar los parámetros de un AE.

En los últimos años, ha habido un creciente interés por parte de la comunidad científica en buscar nuevos enfoques para calibrar los parámetros de algoritmos evolutivos. Estos enfoques están basados en diferentes vertientes de investigación, tales como: meta-AEs [23], meta estimación de algoritmos de distribución [38] y estadística computacional [8, 16], dichos enfoques se describen a continuación.

### 3.3 Enfoques para la calibración de parámetros

Cuando se trabaja con AEs es común encontrar dos clases de parámetros, aquellos parámetros que tienen su dominio de valores en un subconjunto de  $\mathbb{R}$  y aquellos que tienen un dominio finito de posibilidades. Por ejemplo, un operador de cruce puede encontrarse en ambas clases, si consideramos que existen un número finito de operadores de cruce (*p.ej.*, de un punto, uniforme, promedio) o que existe algún valor que determina su comportamiento, como el porcentaje de cruce ( $p_c$ ). Sobra decir que ambas clases de parámetros son esenciales para determinar el buen desempeño del AE.

Smit y Eiben [43] denominan **parámetros simbólicos** a la primera clase y **parámetros numéricos** a la segunda. Esta diferencia es esencial al momento de buscar una configuración adecuada para estos parámetros, ya que ello determinará el método de búsqueda empleado. En caso de los parámetros con un dominio en  $\mathbb{R}$ , pueden emplearse heurísticas de búsqueda y métodos de optimización para encontrar valores aceptables. Por otro lado, el dominio de los parámetros simbólicos no tiene una estructura de búsqueda y sólo puede ser tratado por enumeración o métodos de búsqueda de malla.

Es importante resaltar que el número de parámetros de un AE no está especificado, ya que el diseño<sup>1</sup> del AE determinará el número de parámetros obtenidos. Siguiendo a Smit y Eiben [43], el problema de calibrar los parámetros en un AE puede formalizarse como sigue: denotemos a los parámetros simbólicos como,  $q_1, q_2, \dots, q_m$  y su dominio como,  $Q_1, Q_2, \dots, Q_m$ ; de igual manera denotemos los parámetros numéricos como,  $r_1, r_2, \dots, r_n$  y su dominio como,  $R_1, R_2, \dots, R_n$ . Por

---

<sup>1</sup>Los operadores a utilizar, así como los rangos permitidos estarán dados por el problema a resolver.

lo tanto, el problema de calibración de parámetros puede ser visto como un problema de búsqueda en el espacio de parámetros  $S$  tal que,

$$S = Q_1 \times Q_2 \cdots \times Q_m \times R_1 \times R_2 \cdots \times R_n \quad (3.1)$$

Las soluciones pueden ser vistas como vectores de parámetros con máxima **utilidad**, donde la utilidad de un vector de parámetros  $\bar{p} \in S$  es el desempeño del AE usando los valores de  $\bar{p}$ . Entonces, el **paisaje de desempeño** de los parámetros o **paisaje de utilidad** puede ser definido como un paisaje abstracto donde los vectores de parámetros determinan la forma de  $S$  y la altura de algún lugar en  $S$  estará dada por la utilidad de  $\bar{p}$ . La noción de utilidad, aunque está basada en el desempeño del AE puede ser definida conforme a la calidad de las soluciones o velocidad del algoritmo, *v.g.*, la definición de utilidad depende de la definición de aptitud.

Entonces, un problema de calibración de parámetros posee un espacio de búsqueda con vectores de parámetros como posibles soluciones, una medida de calidad (utilidad), un método para evaluar la calidad de un punto en el espacio de búsqueda (pruebas) y finalmente un método de búsqueda (un procedimiento de calibración) que buscará un punto con máxima altura.

Sin embargo, es común encontrar en la literatura investigaciones enfocadas en la calibración de parámetros numéricos, debido a que estos representan un problema de optimización mucho más difícil que los parámetros simbólicos. El problema recae en encontrar un conjunto de valores  $\in R$  que permitan al AE desempeñarse correctamente, obteniendo soluciones aceptables. Los métodos descritos a continuación se enfocan en la configuración de parámetros numéricos.

### 3.3.1 Meta-algoritmos evolutivos

Encontrar un conjunto adecuado de valores para los parámetros de un AE es una tarea de optimización compleja, ya que comprende una función objetivo que implícitamente es no lineal, variables interactuando entre sí, múltiples óptimos locales y ruido<sup>2</sup> [43]. Irónicamente, este es el tipo de problemas donde los AEs se desempeñan competitivamente como heurísticas de búsqueda. Por lo

---

<sup>2</sup>Debido a la naturaleza estocástica de los AEs.

tanto, no sería raro implementar un AE para optimizar los parámetros de otros.

Grefenstette [22] fue el primero en conducir correctamente los experimentos de un **meta-Algoritmo Evolutivo** (meta-AE). Siguiendo este enfoque, Eiben *et al*[18] usaron un meta algoritmo genético para optimizar algoritmos genéticos. En un meta-AE, los individuos están representados por vectores numéricos, cada uno de estos vectores pertenece a un parámetro del meta-AE base a ser configurado. Para evaluar el desempeño de cada vector, el meta-AE base se ejecuta varias veces usando los valores del vector. Básicamente, cualquier algoritmo evolutivo puede ser usado como un meta-AE, si éste puede manejar vectores de valores reales como individuos.

Actualmente, la **Estrategia Evolutiva de Adaptación de la Matriz de Covarianza** (CMA-ES, por sus siglas en Inglés) es el estado del arte de las estrategias evolutivas estándares [23] por ser casi libre de parámetros [23]. CMA-ES es un método estocástico para la optimización de parámetros reales de funciones no lineales y no convexas. CMA-ES puede ser visto como un meta-AE, ya que estima la distribución de los parámetros a partir de un conjunto de pasos seleccionados, a diferencia del método de **Estimación del Algoritmo de Distribución** (EDA, por sus siglas en Inglés) que estima la distribución de los parámetros a partir de un conjunto de puntos seleccionados.

La principal desventaja de los meta-AEs es que consumen demasiado tiempo de cómputo, además de que algunos requieren la configuración de sus propios parámetros, lo que nos lleva al problema inicial.

### 3.3.2 Meta-estimación de algoritmos de distribución

El método para la **Estimación de Relevancia y Calibración de Valores** de los parámetros de un AE (REVAC, por sus siglas en Inglés) [38] es un EDA que estima la relevancia de los parámetros por medio de la normalización de la entropía de *Shannon*<sup>3</sup>. REVAC mide cómo la entropía de *Shannon* se distribuye sobre la totalidad de los parámetros, resultando en una medida sencilla de relevancia de parámetros.

REVAC es un método de búsqueda estocástico basado en población. De igual manera que EDA,

---

<sup>3</sup>La entropía de *Shannon* es una medida de relevancia de parámetros.

REVAC trata de encontrar un vector de parámetros óptimos estimando la distribución de los mejores valores sobre el dominio de cada parámetro (v.g., la función de densidad). Empezando con una distribución uniforme sobre el espacio de búsqueda inicial de los parámetros, REVAC asigna la probabilidad más alta a las regiones que incrementen el desempeño esperado. Es decir, dado un vector de parámetros, la utilidad de este vector es el desempeño del AE ejecutado con esos valores para los parámetros. Dado entonces, un conjunto de vectores de parámetros con utilidad conocida REVAC puede generar nuevos vectores que tengan la más alta utilidad esperada. Los mejores vectores de la población son seleccionados para ser los padres de los nuevos vectores hijos, los cuales reemplazan al vector más viejo de la población. En este enfoque, sólo un vector es reemplazado en cada generación. Entonces, los nuevos valores de los parámetros son generados con una distribución uniforme. Al terminar el algoritmo, la distribución estimada por parámetro representa un modelo del paisaje de utilidad, que permite obtener una visión de la sensibilidad y relevancia de los parámetros, así como el costo y beneficio de calibrar cada parámetro.

Aunque, REVAC es susceptible a converger a un máximo local, suavizando consistentemente la distribución y maximizando la entropía, lo forzaría a converger a un máximo. Además de esta característica, la entropía de *Shannon* permite calcular la fracción de información invertida en un parámetro en particular. Sin embargo, el suavizado se consigue escogiendo un buen valor para el número de vecinos cuando se define el intervalo de mutación. Este es un aspecto importante en REVAC, pues un valor muy grande puede conducir al algoritmo a converger a un punto de estancamiento, por otro lado, un valor muy pequeño puede producir resultados poco fiables.

### 3.3.3 Estadística computacional

Utilizar **estadística computacional** para calibrar los parámetros de AEs, significa implementar métodos estadísticos computacionalmente exhaustivos que permitan obtener, a través de pruebas rigurosas, un conjunto de valores adecuados para estos parámetros. Los métodos estadísticos están conformados por una serie de pasos perfectamente planeados, para permitir recolectar datos de tal forma que puedan ser analizados por métodos estadísticos dando por resultado conclusiones válidas

y objetivas.

Recientemente han aparecido diferentes investigaciones referentes a la calibración de parámetros por medio de métodos estadísticos. Particularmente, Bartz-Beielstein [5] propone usar métodos estadísticos para proponer teorías con base en la experimentación y no basadas en observaciones.

Existen diversos métodos estadísticos que pueden ser utilizados en la calibración de parámetros, tales como: el diseño de experimentos, diseños factoriales, análisis de varianza, análisis de regresión, entre otros. En las siguientes secciones se describen algunos de los métodos utilizados en estadística computacional para calibrar los parámetros en AEs, así como de las recientes investigaciones que hacen uso de ellos.

## 3.4 Estadística computacional

El nuevo experimentalismo [5] es una tendencia actual en computación evolutiva para llevar a cabo experimentos a través de una serie de pasos bien definidos que permitan al experimentador obtener resultados para validar o refutar hipótesis. El nuevo experimentalismo propone implementar métodos estadísticos que permitan al experimentador aprender no sólo de los resultados, si no también de los errores.

Bartz-Beielstein [5] divide los métodos estadísticos para experimentos de computadora en dos enfoques principales: **estadística computacional clásica** y **estadística computacional moderna**. La principal diferencia entre ambos enfoques es que los segundos suponen que la varianza de la muestra es desconocida, mientras que los métodos estadísticos clásicos requieren que las observaciones (o errores) sean variables aleatorias independientemente distribuidas.

Básicamente, ambos enfoques hacen uso del análisis de varianza para analizar los datos y resultados de los experimentos. Por lo tanto, ambos enfoques requieren que los experimentos se repitan un gran número de veces con la finalidad de obtener una conclusión robusta. Sin embargo, el costo computacional que los métodos estadísticos consumen es relativamente alto y a veces sólo se consigue determinar un conjunto de parámetros que mejora muy poco el desempeño del algoritmo. No obstante, debido a la forma semi-automática de los métodos, el experimentador puede aprender del

experimento , si así lo desea.

### 3.4.1 Métodos estadísticos computacionales clásicos

Dentro de los métodos estadísticos computacionales clásicos se encuentran: el diseño de experimentos, análisis de varianza y los modelos de regresión lineal. En las siguientes secciones se describen en términos generales estos métodos.

#### 3.4.1.1. Diseño de experimentos

Un **experimento** es una prueba o serie de pruebas<sup>4</sup> de un **proceso** o **sistema**, donde las variables de entrada se modifican intencionalmente con la finalidad de estudiar su comportamiento así como los efectos en la respuesta. Por otro lado, un **proceso** puede ser visto como una combinación de máquinas, métodos, personas y otras fuentes que transforman alguna entrada en una salida que tiene una o más respuestas observables (ver figura 3.1).

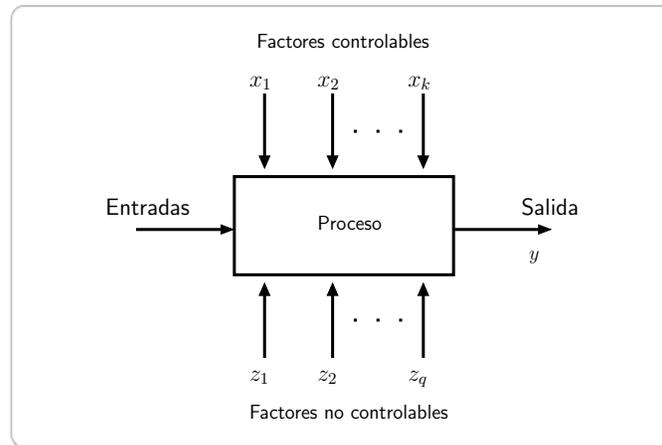


Figura 3.1: Modelo general de un proceso o sistema.

Un experimento se lleva a cabo para estudiar el desempeño de un proceso, algunos de los objetivos al momento de realizar un experimento podrían incluir los siguientes puntos:

1. Determinar que variables influyen significativamente en la respuesta  $y$ .

<sup>4</sup>También llamadas ejecuciones individuales.

2. Determinar los valores de las variables que minimicen la variabilidad de la respuesta  $y$ .
3. Determinar los valores de las variables que minimicen los efectos de las variables no controlables  $z$ 's.

El **diseño estadístico de experimentos** (DOE, por sus siglas en Inglés) se refiere al proceso de planear el experimento de tal forma que los datos recolectados puedan ser analizados por métodos estadísticos, dando por resultado conclusiones válidas y objetivas [35]. DOE tiene una larga tradición en estadística clásica debido a la confiabilidad que sus soluciones proporcionan, por lo tanto, no es raro que haya sido desarrollado para diferentes áreas de aplicación tales como: agricultura [19], optimización industrial [11], simulación por computadora [32], entre muchas otras [44].

El diseño de experimentos posee tres principios básicos **replicación**, **aleatoriedad** y **bloqueo**. **Replicación** se refiere a repetir el experimento tantas veces como sea necesario con el objetivo de obtener un error estimado experimental y determinar los efectos de un factor en el experimento. El **error estimado** es una unidad básica de medición para determinar si las diferencias observadas en los datos son estadísticamente diferentes; además, permite obtener un estimado más preciso de los efectos de un factor sobre la respuesta.

**Aleatoriedad** es la asignación del material experimental y el orden en el cual las pruebas del experimento son llevadas a cabo. La aleatoriedad permite promediar los efectos de factores externos que podrían presentarse durante el experimento. Generalmente, la aleatoriedad en un experimento, está dada por un generador de números aleatorios.

**Bloqueo** es una técnica de diseño usada para mejorar la precisión de las comparaciones entre factores de interés. A menudo, el bloqueo se usa para reducir o eliminar la variabilidad (conocida también como ruido), es decir: factores que podrían influir en la respuesta experimental pero no estamos directamente interesados en ellos. Generalmente, un bloque es un conjunto de condiciones experimentales relativamente homogéneas.

El enfoque estadístico del diseño de experimentos es necesario si deseamos obtener conclusiones significativas de los datos y más aún cuando los datos están sujetos a un error experimental. Por lo tanto, existen dos aspectos importantes en cualquier problema experimental: el **diseño de experi-**

mentos y el **análisis estadístico** de los datos. Ambos aspectos están estrechamente relacionados, ya que el método de análisis estadístico dependerá del diseño empleado.

En DOE, los **parámetros de entrada** que serán variados durante el experimento son llamados factores. Otros nombres frecuentemente utilizados son: variables de diseño, variables predictoras, variables de entrada, regresores o variables independientes. Algunas de las variables del proceso,  $x_1, x_2, \dots, x_k$ , son controlables, mientras que otras variables,  $z_1, z_2, \dots, z_q$ , no lo son. Las **variables controlables** son aquellos factores de entrada variados intencionalmente durante la prueba, aunque también algunas se pueden mantener constantes para asegurar su efecto nulo en la respuesta. Por otro lado, las **variables no controlables** son aquellos factores de los cuales no se tiene control y se espera que su efecto sobre la respuesta sea mínimo.

En DOE, el vector de variables de diseño está representado por  $x = (x_1, \dots, x_k)^T$ . Cada variable en un experimento posee una configuración llamada **nivel** o **tratamiento**, comúnmente escalados en el rango de  $[-1, +1]$ . La relación entre los niveles de una variable determina si la variable es **cualitativa** (*p.ej.*, esquema de selección) o **cuantitativa** (*p.ej.*, tamaño de población). Podemos observar que las variables cualitativas y cuantitativas son análogas con los parámetros simbólicos y numéricos definidos por Smit y Eiben.

Se dice que existe interacción entre variables, cuando el efecto de una variable en la respuesta depende en el nivel de otra variable. El manejo de interacción entre variables es una característica de los métodos de DOE y una debilidad del método **una-variable-a-la-vez** (OVAT, por sus siglas en Inglés). Mientras que DOE considera todas las posibles combinaciones de los niveles de las variables, cuantificándolas de tal forma que puedan ser usadas para entender mejor la respuesta, el método de OVAT ignora la interacción entre ellas, el éxito de DOE proviene de esta consideración.

Consideremos dos variables  $A$  y  $B$ , ambas pueden ser dos conjuntos o dos estados indicados por  $-1$  y  $+1$ . Las figuras 3.2(a) y 3.2(b) muestran como las respuestas  $y_1$  y  $y_2$  dependen de  $A$  y  $B$ . En la figura 3.2(a), las líneas que conectan los dos niveles de  $B$  son paralelas, lo cual indica que los niveles de  $B$  no influyen en cómo la variable  $A$  afecta la respuesta, por lo tanto es probable que no exista interacción alguna entre  $A$  y  $B$ . Por otro lado, en la figura 3.2(b), las líneas que conectan los niveles de  $B$  divergen, lo cual indica que la elección del nivel de  $B$  determina cómo la variable  $A$

afecta la respuesta, por lo tanto es probable que exista interacción entre  $A$  y  $B$ .

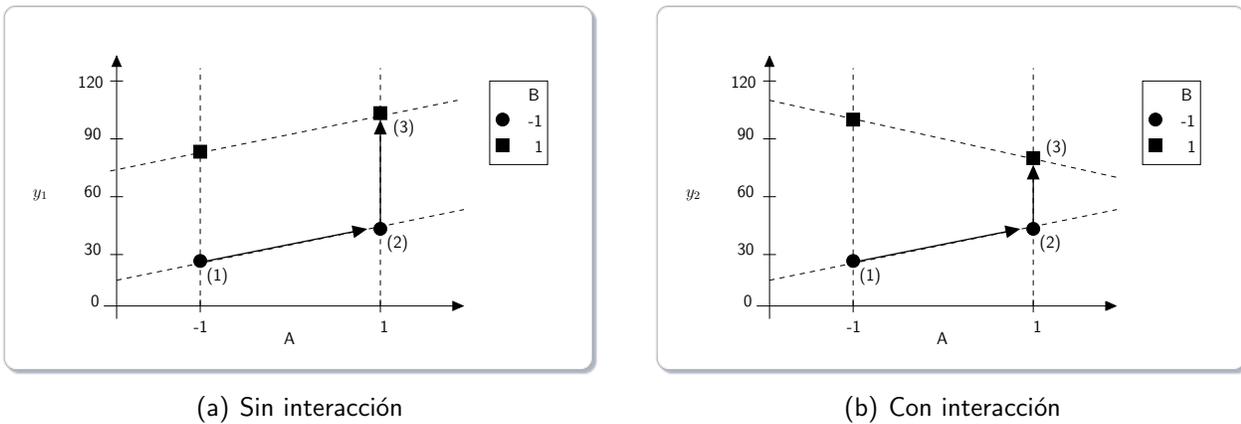


Figura 3.2: Interacción entre variables.

Dos de las principales consideraciones que distinguen al diseño de experimentos son el número de variables de diseño que incluyen y la complejidad del modelo que ellas proveen. Para un número reducido de variables de diseño existen diseños de experimentos específicos, pero los diseños que abarcan un gran número de variables de diseño son los llamados **experimentos de proyección** y **experimentos de superficie de respuesta**.

Los experimentos de proyección son usados para estudiar un gran número de variables de diseño con el propósito de identificar las más importantes. Algunos experimentos de proyección pueden evaluar muchas variables con muy pocas ejecuciones experimentales, pero sólo pueden usar dos niveles por cada variable de diseño y no pueden resolver interacciones entre pares de variables. Esta característica hace riesgosos a los experimentos de proyección.

Por otra parte los experimentos de **superficie de respuesta**, son más complejos y difíciles de administrar que los experimentos de proyección, así que generalmente involucran de dos a cinco variables. Cada variable en este modelo, debe ser cuantitativa y poseer tres o más niveles. El beneficio de usar muchos niveles, es que el modelo de superficie de respuesta provee modelos muy complejos que incluyen hasta el más mínimo efecto de las variables en la respuesta, además de la interacción entre ellas y términos para medir la curvatura inducida en la respuesta por cada variable de diseño. Se hablará más adelante acerca de esta metodología.

## 3.4.1.2. Análisis de Varianza

Un **modelo** es una descripción matemática de un proceso que define el comportamiento de la respuesta en función de las variables de entrada. Un buen modelo es aquel que explica el comportamiento sistemático de los datos originales de manera concisa, la forma del modelo depende del tipo de variables de diseño usadas en el experimento. Si un experimento contiene un conjunto de variables de diseño cuantitativas con diversos niveles, entonces el modelo consiste en la medición del tratamiento. Si el experimento contiene una sola variable cuantitativa que cubre un rango de valores, entonces el modelo consiste en una ecuación que relaciona la respuesta con los predictores cuantitativos.

Los experimentos que involucran predictores cualitativos usualmente son analizados con el modelo de **análisis de varianza** (ANOVA, por sus siglas en Inglés). Los experimentos que involucran predictores cuantitativos usualmente son analizados con **modelos de regresión**. Aquellos experimentos que combinan variables cualitativas y cuantitativas son analizados usando modelos de regresión especiales, llamados **modelos lineales generales**. En esta sección se describe, en términos generales, el primer modelo mientras que en la sección 3.4.1.3 se describen los últimos dos.

Siguiendo a Montgomery [35], ANOVA es uno de los principios más usados en estadística inferencial. Supongamos que se tienen  $m$  tratamientos de un factor. Los datos podrían aparecer como se muestran en la tabla 3.1, donde una entrada (*p.ej.*,  $y_{ij}$ ) representa la  $j$ -ésima observación tomada bajo el tratamiento  $i$ . En general, existen  $n$  observaciones bajo el  $i$ -ésimo tratamiento.

Tratamiento	Observación				Total	Promedio
1	$y_{11}$	$y_{12}$	$\dots$	$y_{1n}$	$y_{1.}$	$\bar{y}_1.$
2	$y_{21}$	$y_{22}$	$\dots$	$y_{2n}$	$y_{2.}$	$\bar{y}_2.$
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$	$\vdots$
$m$	$y_{m1}$	$y_{m2}$	$\dots$	$y_{mn}$	$y_{m.}$	$\bar{y}_m.$
					$y_{..}$	$\bar{y}_{..}$

Tabla 3.1: Datos para un experimento con un único factor.

De la tabla 3.1,  $y_{i.}$  representa la suma total de las observaciones bajo el  $i$ -ésimo tratamiento

y  $\bar{y}_i$  representa el promedio de dichas observaciones bajo el  $i$ -ésimo tratamiento. Similarmente,  $y_{..}$  representa la suma total de todas las observaciones y  $\bar{y}_{..}$  representa el promedio total de dichas observaciones, expresado de otra forma:

$$y_{i.} = \sum_{j=1}^n y_{ij}, \quad \bar{y}_{i.} = y_{i.}/n \quad i = 1, 2, \dots, m \quad (3.2)$$

$$y_{..} = \sum_{i=1}^m \sum_{j=1}^n y_{ij}, \quad \bar{y}_{..} = y_{..}/N, \quad (3.3)$$

donde  $N = nm$  es el número total de observaciones. La **suma total de cuadrados** corregida (mostrada en la ecuación (3.4)) mide la variabilidad total en los datos.

$$SS_T = \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \bar{y}_{..})^2 \quad (3.4)$$

$$\sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \bar{y}_{..})^2 = n \sum_{i=1}^m (\bar{y}_{i.} - \bar{y}_{..})^2 + \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \bar{y}_{i.})^2 \quad (3.5)$$

La suma total de cuadrados puede dividirse en una **suma de cuadrados debido los tratamientos** ( $SS_{TREAT}$ ) y una **suma de cuadrados debido al error** ( $SS_E$ ).  $SS_{TREAT}$  es la diferencia entre los promedios de los tratamientos y el promedio total.  $SS_E$  es la diferencia de las observaciones y el promedio del tratamiento. Este principio fundamental en ANOVA puede reescribirse como se muestra en la ecuación (3.6).

$$SS_T = SS_{TREAT} + SS_E \quad (3.6)$$

### 3.4.1.3. Modelos de regresión

Los **modelos de regresión** son elementos centrales en el diseño de experimentos, pues permiten analizar el efecto de las variables sobre la respuesta y su relación entre ellas.

En general, supongamos que existe una sola **variable dependiente**  $y$  que depende de  $k$  variables regresoras, por ejemplo  $x_1, x_2, \dots, x_k$ . La relación entre esas variables puede expresarse por medio de un modelo matemático llamado modelo de regresión, el cual se ajusta para un conjunto de datos tomados como muestra. Una **muestra** o **punto de diseño** es aquella instancia específica del vector de variables de diseño, ejecutada con una determinada configuración de los parámetros que da por resultado una respuesta aleatoria.

En algunos casos, el experimentador conoce la forma exacta de la relación funcional entre  $y$  y  $x_1, x_2, \dots, x_k$ , digamos  $y = \phi(x_1, x_2, \dots, x_k)$ . Aunque, en muchos casos, la verdadera relación es desconocida y el experimentador debe elegir una función apropiada para aproximar  $\phi$ . Es común utilizar polinomios de orden menor como funciones de aproximación.

En el diseño de experimentos es común utilizar modelos de regresión lineal, ya que describe la relación entre variables. Supongamos dos variables de diseño  $x_1$  y  $x_2$ , el modelo que describe su relación se muestra en la ecuación (3.7).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon \quad (3.7)$$

Este modelo se conoce como **modelo de regresión lineal múltiple** con dos variables independientes, ya que la ecuación (3.7) es una función lineal de parámetros desconocidos  $\beta_0$ ,  $\beta_1$  y  $\beta_2$ . El modelo describe un plano en el espacio de dos dimensiones  $x_1$ ,  $x_2$ , acotado por los límites inferior y superior de cada variable. El parámetro  $\beta_0$  define la intercepción en el plano,  $\beta_1$  y  $\beta_2$  son llamados coeficientes de regresión parcial, porque  $\beta_1$  mide el cambio esperado en  $y$  por unidad de cambio en  $x_1$  cuando  $x_2$  permanece constante, y  $\beta_2$  mide el cambio esperado en  $y$  por unidad de cambio en  $x_2$  mientras  $x_1$  permanece constante.

En general, la variable de respuesta  $y$  podría estar relacionada con  $k$  variables regresoras. Enton-

ces, el modelo mostrado en la ecuación (3.8)

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon \quad (3.8)$$

es llamado modelo de regresión lineal múltiple con  $k$  variables regresoras, donde los parámetros  $\beta_j, j = 0, 1, \dots, k$  son llamados coeficientes de regresión. El modelo describe un hiperplano en el espacio  $k$ -dimensional de las variables regresoras  $\{x_j\}$  y el parámetro  $\beta_j$  representa el cambio esperado en la respuesta  $y$  por unidad de cambio en  $x_j$  cuando el resto de las variables independientes  $x_i$  ( $i \neq j$ ) permanecen constantes.

Modelos más complejos, en apariencia con la ecuación (3.7), pueden ser analizados con técnicas de regresión lineal múltiple. Por ejemplo, consideramos un término de interacción de dos variables en el modelo de la ecuación (3.7), el modelo de regresión resultante sería como el mostrado en la ecuación (3.9).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \epsilon \quad (3.9)$$

Si  $x_3 = x_1 x_2$  y  $\beta_3 = \beta_{12}$ , la ecuación (3.9) podría reescribirse como:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon \quad (3.10)$$

El cual es llamado modelo de regresión lineal múltiple con tres regresores.

Otro ejemplo sería un modelo de superficie de respuesta de segundo orden con dos variables, como el mostrado en la ecuación (3.11).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \epsilon \quad (3.11)$$

Si  $x_3 = x_1^2$ ,  $x_4 = x_2^2$ ,  $x_5 = x_1 x_2$ ,  $\beta_3 = \beta_{11}$ ,  $\beta_4 = \beta_{22}$  y  $\beta_5 = \beta_{12}$  entonces la ecuación (3.11) podría expresarse como

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \epsilon \quad (3.12)$$

el cual corresponde a un modelo de regresión lineal. En general, cualquier modelo de regresión que es lineal en los parámetros (los valores de  $\beta$ 's) es un modelo de regresión lineal, a pesar de la forma de la superficie de respuesta que se genere.

#### 3.4.1.4. Trabajos relacionados

Existen diversos trabajos para calibrar los parámetros de un AE que hacen uso de métodos estadísticos clásicos. Bartz-Beielstein usa modelos de regresión para analizar el desempeño de un algoritmo evolutivo en [4]. En dicho trabajo se utilizan los métodos de DOE, métodos de superficie de respuesta y métodos de regresión basados en árboles, para analizar y mejorar el desempeño de un algoritmo de optimización.

En [16], Czarn *et al* usan una metodología estadística para el estudio exploratorio en algoritmos genéticos. En este trabajo, mediante ANOVA examinan la relación entre los operadores de cruce y mutación para diversas modalidades de un problema. Usando un diseño de bloques completos aleatorios, controlan la variación observada en las ejecuciones del algoritmo genético debido a la semilla para generar números aleatorios. Esto permite reducir la variación en las respuestas y poder realizar comparaciones con ANOVA para ajustar el modelo de regresión lineal utilizan el método de expansión de Taylor.

#### 3.4.2 Métodos estadísticos computacionales modernos

La **estadística computacional moderna**, combina técnicas clásicas del diseño de experimentos, clasificación y árboles de regresión (CART, por sus siglas en Inglés), diseño moderno y análisis de experimentos de computadora (DACE, por sus siglas en Inglés), diferentes técnicas de muestreo, entre otras. Es importante señalar que, los modelos de DACE pueden ser vistos como una extensión del enfoque clásico del análisis de regresión.

A continuación se describen, en términos generales, los métodos mencionados anteriormente.

### 3.4.2.1. Análisis y Diseño de Experimentos de Computadora

Como hemos visto, los modelos de regresión lineal usados en estadística clásica están basados en la ecuación (3.7), donde el error  $\epsilon$  y la respuesta  $y$  deben tener la misma varianza, lo que significa que las muestras aleatorias pueden ser modeladas con una distribución normal,  $\mathcal{N}(0, 1)$ , e implica replicar las ejecuciones para reducir la varianza y obtener un estimado del error aleatorio. Debido a ello, la adecuación de este tipo de modelos se determina usando una prueba de hipótesis que dependa de la igualdad de las varianzas.

En estadística clásica se considera que la varianza es siempre constante, contrario a los métodos estadísticos modernos los cuales suponen que la varianza en los modelos de regresión es no homogénea o que varía en cada muestra.

El **Análisis y Diseño de Experimentos de Computadora** (DACE, por sus siglas en Inglés) es un modelo de procesos estocásticos que expresa la respuesta determinista como resultado de un modelo de regresión y un proceso aleatorio [36]. DACE considera cada ejecución de un algoritmo como un proceso estocástico y provee métodos para predecir valores desconocidos en dicho proceso.

Los **modelos de procesos estocásticos** en DACE son vistos como una alternativa a los modelos de regresión en estadística clásica. Supongamos un conjunto de  $m$  puntos de diseño  $x = (x_1, \dots, x_m)^T$  con  $y_i \in \mathbb{R}$ , en DACE una función determinista evalúa los  $m$  puntos de diseño  $x$ . El vector de  $m$  respuestas resultante es denotado como  $y = (y_1, \dots, y_m)^T$  con  $y_i \in \mathbb{R}$ , donde cada respuesta  $y(x_i)$ , para una entrada  $x_i$   $d$ -dimensional, es producto de un modelo de regresión  $\mathcal{F}$  y un proceso estocástico  $Z$  como el mostrado en la ecuación (3.13).

$$Y(x) = \mathcal{F}(\beta, x) + Z(x) \quad (3.13)$$

Supongamos que existen  $q$  funciones  $f_j : \mathbb{R}^d \rightarrow \mathbb{R}$ , el modelo de regresión usado por un proceso estocástico se define en la ecuación (3.14).

$$F(\beta, x) = \sum_{j=1}^q \beta_j f_j(x) = f(x)^T \beta \quad (3.14)$$

Es posible usar polinomios de orden 0, 1 y 2 en este tipo de modelos. Por ejemplo, el modelo de regresión constante con  $q = 1$  es  $f_1(x) = 1$ ; el modelo lineal con  $q = d + 1$  es  $f_1(x) = 1$ ,  $f_2(x) = x_1, \dots, f_{d+1}(x) = x_d$ ; y el modelo cuadrático es:  $f_1(x) = 1$ ,  $f_2(x) = x_1, \dots, f_{d+1}(x) = x_d$ ,  $f_{d+2}(x) = x_1x_1, \dots, f_{2d+1}(x) = x_1x_d, \dots, f_q(x) = x_dx_d$ .

Por otra lado, el proceso aleatorio  $Z(\cdot)$  tiene una media cero y covarianza  $V(w, x) = \sigma^2 \mathcal{R}(\theta, w, x)$ , con varianza del proceso  $\sigma^2$  y modelo de correlación  $\mathcal{R}(\theta, w, x)$ . La función de correlación exponencial *EXP* y la función de correlación Gaussiana *GAUSS* son usadas comúnmente en DACE [36]

### 3.4.2.2. Árboles de regresión

La **Clasificación y árboles de regresión** (CART) fue introducido por Breiman *et al*[12]. Se refiere a una herramienta no paramétrica<sup>5</sup> para el análisis y estructura de los datos. CART es usada para proyectar variables y verificar la adecuación del modelo de regresión, puede ser vista como un tipo de selección variable, similar a las técnicas de regresión incremental, en el análisis de regresión clásico.

Consideremos un conjunto de variables de diseño  $X = x_1, \dots, x_d^T$  y una variable de respuesta cuantitativa  $Y$ . Las variables de diseño son llamadas variables predictoras en el contexto de CART. Un árbol de regresión es una colección de reglas tales como “si  $x_1 \leq 5$  y  $x_4 \in A, C$ , entonces el valor predicho de  $Y$  es 14.2”, las cuales son ordenadas en forma de un árbol binario.

Comparado con los modelos lineales, los modelos basados en árboles son fáciles de interpretar cuando aparecen predictores cualitativos y cuantitativos en el modelo. Los métodos de regresión basados en árboles no pueden reemplazar a los métodos estadísticos clásicos, pero pueden complementarlos.

### 3.4.2.3. Métodos de muestreo en DACE

Los métodos de muestreo modernos han sido desarrollados para experimentos de computadora deterministas que no tienen un error aleatorio. En DACE se asume que pueden encontrarse carac-

<sup>5</sup>No paramétrica significa que utiliza métodos libres de distribución los cuales no recaen en la hipótesis de que los datos poseen una distribución de probabilidad.

terísticas interesantes del modelo verdadero en todo el espacio muestral, por ello utiliza diseños exploratorios o métodos para tratar espacios llenos, los cuales colocan un conjunto de muestras en el interior del espacio de diseño.

Uno de estos métodos es el **muestreo de Monte Carlo** (MC), el cual coloca muestras aleatoriamente en el espacio de diseño. Por lo tanto, es posible que regiones relativamente grandes permanezcan sin explorar. Por otro lado, existen diferentes enfoques de MC para tratar con este problema, algunos de ellos dividen el espacio de diseño en subintervalos con igual probabilidad para colocar las muestras, lo que supone requerir únicamente  $2^d$  muestras.

El muestreo por **hipercubos latinos** (LHS, por sus siglas en Inglés) es una alternativa al muestreo de Monte Carlo. LHS es superior bajo ciertas condiciones al muestreo de MC y provee gran flexibilidad al escoger el número de muestras. Sin embargo, LHS podría resultar en una organización mal diseñada de puntos de muestra.

#### 3.4.2.4. Trabajos relacionados

En [9], Bartz-Beielstein *et al* emplean DOE, CART y DACE para el análisis experimental de algoritmos de optimización. El objetivo es investigar la interacción entre problemas de optimización, algoritmos y ambientes. Por medio de esta técnica se parametrizan y se comparan ambos algoritmos: uno basado en población y el otro en búsqueda directa.

En [41], Pham usa un experimento factorial completo en ANOVA para calibrar los parámetros de un algoritmo evolutivo para resolver problemas de optimización de control dinámico en ingeniería. También usa reducción de pasos progresivo para mejorar la eficiencia del algoritmo para estas aplicaciones.

La Optimización de Secuencial Parámetros (SPO, por sus siglas en Inglés) es quizás el trabajo más representativo que utiliza métodos estadísticos modernos. SPO fue introducido por Bartz-Beielstein [6] y puede ser interpretado como una heurística de búsqueda, específicamente diseñada para la calibración de parámetros y su análisis. SPO puede ser implementado algorítmicamente porque emplea técnicas estadísticas de DACE. SPO empieza con un conjunto de puntos de diseño, los cuales son elegidos usando LHS. Cada punto de diseño es evaluado varias veces para determinar su desempeño.

La respuesta es tratada con un modelo de regresión con polinomios de segundo orden y una función de correlación Gaussiana para representar la relación entre vectores y resultados. Los nuevos vectores son generados y probados usando este modelo. El mejor punto de diseño es incluido en el siguiente conjunto y reevaluado, posiblemente doblando el número de ejecuciones (para decrementar el error estimado). SPO conserva el costo computacional para determinar un conjunto de parámetros que mejoren el desempeño del algoritmo, aunque esta mejora es relativamente baja. Una desventaja de este enfoque es que SPO requiere la especificación de algunos parámetros:

- Específicos del algoritmo: el tipo y rango de los parámetros, así como el criterio de optimización.
- Específicos de SPO: tamaño y número de muestras por punto de diseño; número de candidatos para LHS, número de puntos (viejos y nuevos) de diseño a seleccionar y; criterio de terminación.

En [8, 7], Bartz-Beielstein *et al* usan una combinación de métodos estadísticos clásicos, DACE y CART, para configurar los parámetros del algoritmo de optimización mediante cúmulos de partículas en un sistema de control de grupo supervisor de elevador. La técnica utiliza árboles de regresión para proyectar las variables y verificar la adecuación del modelo, además de considerar un diseño factorial fraccionado para mejorar el árbol.



# 4

## Calibración de parámetros usando estadística computacional

### 4.1 Introducción

Los métodos estadísticos pueden analizar la configuración experimental de algoritmos en problemas de prueba específicos con la meta de mejorar su desempeño. Sin embargo, estos métodos implican cómputo intensivo.

El objetivo de este capítulo es describir el funcionamiento de los métodos estadísticos computacionales implementados para calibrar los parámetros de un AE. Empezando con un diseño factorial, el resultado de este experimento permite obtener un rango acotado de valores sobre los cuales se aplica la metodología de superficie de respuesta. Modelos lineales de primer y segundo orden son utilizados para conocer la relación entre variables y respuesta.

## 4.2 Diseños Factoriales

Cuando se necesitan conocer los efectos de los factores en la respuesta de un proceso, los **diseños factoriales** son una buena alternativa en este tipo de experimentos, pues investigan todas las posibles combinaciones entre los niveles de los factores. Por ejemplo, supóngase que existen  $a$  niveles de un factor  $A$  y  $b$  niveles de un factor  $B$ . Cada réplica del experimento contiene todas las  $ab$  combinaciones de niveles.

El **efecto de un factor** está definido como el cambio en la respuesta producido por un cambio en el nivel del factor, comúnmente denominado **efecto principal** por referirse a los factores primarios en el experimento. Por ejemplo, en un diseño factorial con dos factores y dos niveles cada uno (mostrado en la figura 4.1) es posible referirnos a sus niveles como “alto” y “bajo” o “-” y “+” respectivamente.

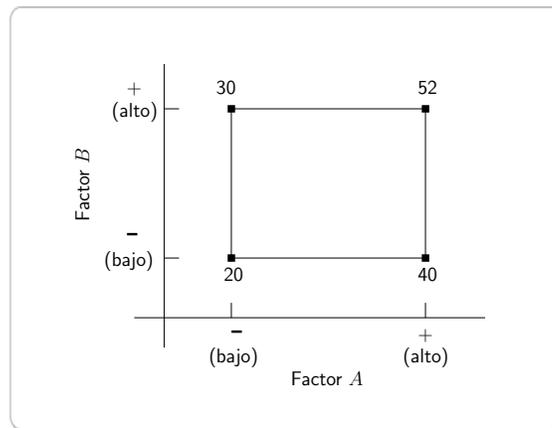


Figura 4.1: Un diseño factorial de dos factores.

El efecto principal del factor  $A$  puede ser visto como la diferencia entre la respuesta promedio en el nivel bajo de  $A$  y el promedio de la respuesta en el nivel alto de  $A$ , es decir:

$$A = \frac{40 + 52}{2} - \frac{20 + 30}{2} = 21$$

Esto significa que incrementando el factor  $A$  desde su nivel bajo a su nivel alto, causará un

incremento de la respuesta promedio en 21 unidades. Algo similar ocurre con el efecto principal del factor  $B$ .

$$B = \frac{30 + 52}{2} - \frac{20 + 40}{2} = 11$$

Si los factores  $A$  y  $B$  tuvieran más de dos niveles, el procedimiento cambiaría puesto que existen otras formas de definir los efectos de un factor, de las cuales se hablará más adelante.

Por otra parte, en algunos experimentos es posible encontrar diferencias en la respuesta de los niveles de un factor con los niveles de los demás factores, esto ocurre cuando el nivel de un factor, digamos  $A$ , depende del nivel elegido en el factor  $B$ , entonces se dice que existe **interacción** entre factores. Por ejemplo, consideremos el diseño factorial de dos factores mostrado en la figura 4.2.

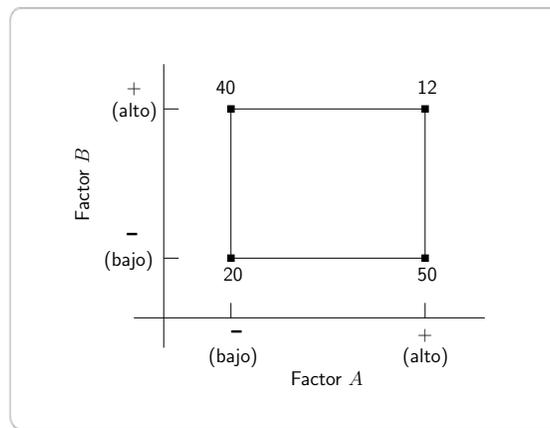


Figura 4.2: Un diseño factorial de dos factores con interacción.

En el nivel bajo del factor  $B$ , el efecto de  $A$  es

$$A = 50 - 20 = 30$$

y en el nivel alto del factor  $B$  el efecto de  $A$  es

$$A = 12 - 40 = -28$$

Debido a que el efecto de  $A$  depende del nivel elegido en  $B$  existe interacción entre  $A$  y  $B$ .

La **magnitud del efecto de interacción** es la diferencia promedio de los dos efectos de  $A$ , v.g.,  $AB = (-28 - 30)/2 = -29$ , por lo tanto se puede concluir que existe una gran interacción entre factores (otro ejemplo similar sobre interacción entre factores fue mostrado en la el capítulo 3).

### 4.2.1 Diseños Factoriales $2^k$

El diseño factorial  $2^k$  considera  $k$  factores con dos niveles cada uno (puede manejar niveles cuantitativos o cualitativos), lo que significa un reducido número de ejecuciones con las cuales los  $k$  factores pueden estudiarse en un diseño factorial completo. Consecuentemente, este tipo de diseños son ampliamente utilizados en **experimentos de proyección de factores**.

Ya que sólo existen dos niveles por cada factor, se asume que la respuesta es aproximadamente lineal sobre el rango de niveles escogidos. Una objeción importante en los diseños factoriales  $2^k$ , es que los efectos no lineales permanecen ocultos, por lo cual sólo son usados para conseguir una descripción general de los efectos de cada factor y su interacción entre ellos y no para obtener valores exactos. Un ejemplo de un diseño factorial  $2^2$  se muestra en el apéndice A.

Sin embargo, existen otras técnicas que pueden emplearse para determinar la interacción entre factores como son los modelos de regresión y los métodos de superficie de respuesta. En caso de que se utilice un modelo de regresión, es necesario expresar la interacción entre factores en forma de un polinomio, por ejemplo el diseño factorial de dos factores (mostrado en la figura 4.1) podría reescribirse de la siguiente forma:

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_{12}x_1x_2 + \epsilon$$

donde  $y$  es la respuesta, los  $\beta$ 's son los parámetros que serán determinados,  $x_1$  es una variable que representa el factor  $A$ ,  $x_2$  es una variable que representa el factor  $B$  y  $\epsilon$  es un término de error aleatorio. Las variables  $x_1$  y  $x_2$  están definidas en una escala de  $-1$  a  $+1$  (los niveles alto y bajo de  $A$  y  $B$ ), y  $x_1x_2$  representa la interacción entre  $x_1$  y  $x_2$ .

En general, los diseños factoriales se usan para obtener una proyección del efecto de los factores en la respuesta de un proceso. Los diseños factoriales son bastante utilizados en DOE por ser más

eficientes que los experimentos de un factor a la vez (OVAT). Sin embargo, los diseños factoriales sólo exploran los límites del espacio de diseño mientras que el interior permanece inexplorado.

Siguiendo a Bartz-Beielstein en [5], el modelo clásico en DOE es

$$\tilde{y} = y + \epsilon \quad (4.1)$$

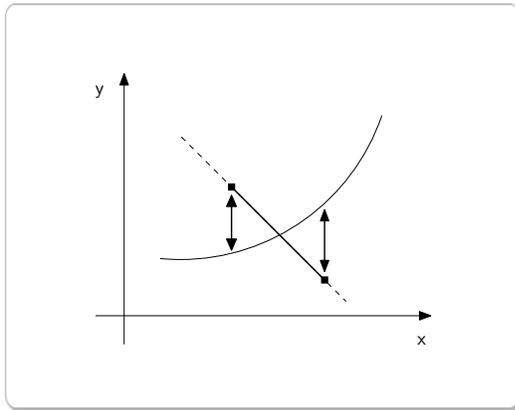
donde  $\tilde{y}$  es la respuesta medida,  $y$  es el valor verdadero y  $\epsilon$  es un término de error. Se asume que los errores son independientes e idénticamente distribuidos. La ecuación (4.1) se usa para modelar la hipótesis de que  $\epsilon$  está siempre presente. Por lo tanto, la meta del DOE es colocar las muestras en el espacio de diseño de tal forma que se minimice la influencia de  $\epsilon$ , para ello emplea un modelo de aproximación como se muestra en la ecuación (4.2),

$$\hat{y} = f(x, \tilde{y}(x)) \quad (4.2)$$

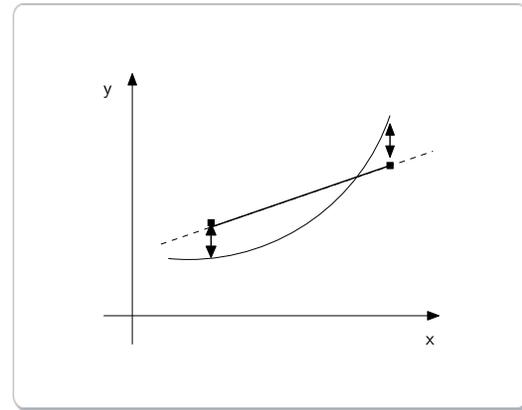
donde  $f$  es un polinomio de orden menor y  $x$  denota un punto muestra. Los diseños factoriales colocan los puntos de diseño en los límites del espacio de diseño, para permitir al modelo lineal estimado aproximarse mejor al modelo real. Podemos observar en la figura 4.3(a) que el error estimado es mayor, si las muestras son colocadas en el interior del espacio de diseño, por lo tanto el modelo lineal estimado tiene una aproximación pobre al modelo real. Sin embargo, si las muestras son colocadas en el límite del espacio de diseño (como se muestra en la figura 4.3(b)), el error estimado es menor y el modelo lineal estimado se aproxima mejor al modelo real.

### 4.3 Metodología de Superficie de Respuesta

La **metodología de superficie de respuesta** (RSM, por sus siglas en Inglés), es una colección de técnicas matemáticas y estadísticas usadas para modelar y analizar problemas en los cuales la respuesta de interés está influenciada por diversas variables [35]. El objetivo de RSM es determinar las condiciones de operación óptimas del sistema o detectar una región donde los requerimientos de operación sean satisfechos.



(a) Muestras colocadas en el interior del espacio de diseño.



(b) Muestras colocadas en los límites del espacio de diseño.

Figura 4.3: Aproximación del modelo lineal estimado en un diseño factorial.

Supongamos que existen  $x_k$  factores de algún proceso con  $m$  niveles cada uno, la respuesta del proceso estará dada por alguna combinación de niveles de los  $k$  factores. Esto significa, que existe alguna función de  $x_1, x_2, \dots, x_k$  tal que:

$$y = f(x_1, \dots, x_k) + \epsilon \quad (4.3)$$

donde  $\epsilon$  representa el ruido o error observado en la respuesta. Si denotamos la respuesta esperada como:

$$E(y) = f(x_1, \dots, x_k) = \eta$$

entonces la relación  $\eta = f(x_1, \dots, x_k)$  existente entre  $\eta$  y los niveles de los  $k$  factores puede representarse a través de una hipersuperficie (subconjunto de un espacio euclídeo  $(k+1)$ -dimensional) llamada **superficie de respuesta**.

La superficie de respuesta  $\eta$  puede visualizarse fácilmente en una gráfica si consideramos únicamente un proceso con dos variables, digamos  $x_1$  y  $x_2$ , como se muestra en la figura 4.4.

Una técnica utilizada para ayudar a visualizar la forma que puede tener una superficie de respuesta tridimensional consiste en representar la **gráfica de contornos** de la superficie (ver figura 4.5), en la que se trazan las llamadas **líneas de contorno**, las cuales son curvas correspondientes a valores

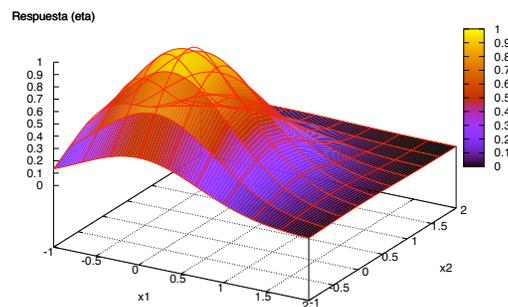


Figura 4.4: Superficie de respuesta  $\eta$ , para un proceso con dos variables,  $x_1$  y  $x_2$

constantes<sup>1</sup> de la respuesta sobre el plano  $x_1x_2$  (plano cuyos ejes coordenados vienen dados por los niveles  $x_1$  y  $x_2$  de los factores). Geométricamente, cada línea de contorno es una proyección sobre el plano  $x_1x_2$  de una sección de la superficie al ser intersectada por un plano paralelo a los ejes  $x_1x_2$ . La gráfica de contornos es útil para estudiar los niveles de los factores en los que se da un cambio en la forma o altura de la superficie de respuesta.

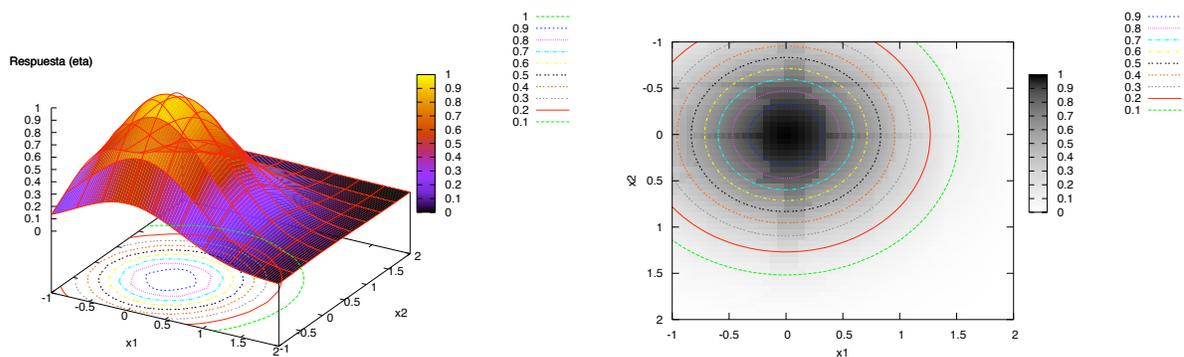


Figura 4.5: Gráfica de contornos de una superficie de respuesta con dos variables.

La existencia de gráficas de contorno no está limitada a tres dimensiones. Sin embargo, involucrar más de tres factores dificulta su representación geométrica.

Por otro lado, en la mayoría de los problemas se desconoce la verdadera relación entre variables

<sup>1</sup>Una altura particular de la superficie de respuesta.

independientes y respuesta, *v.g.*, la verdadera forma de  $\eta$ . Para aproximar esta relación, RSM utiliza un **modelo de primer orden** (ecuación (4.4)).

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \cdots + \beta_kx_k + \epsilon \quad (4.4)$$

Se llama modelo de primer orden ya que utiliza un polinomio de primer grado para expresar la relación funcional entre variables y respuesta. Sin embargo, si el modelo de primer orden no expresa una aproximación adecuada de la superficie verdadera, entonces, un polinomio de orden mayor debería ser utilizado.

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{i < j} \beta_{ij} x_i x_j + \epsilon \quad (4.5)$$

El modelo mostrado en la ecuación (4.5) es llamado **modelo de segundo orden** porque utiliza un polinomio de grado dos para expresar la relación variables-respuesta, la cual probablemente presenta alguna curvatura en la superficie de respuesta. La mayoría de los problemas que aplican RSM [8] utilizan uno o ambos modelos, pues resulta complicado que un modelo polinomial sea una aproximación exacta a la verdadera relación entre variables y respuesta, pero usualmente trabajará bien en una región relativamente pequeña.

Por otra parte, el método de **mínimos cuadrados** se usa para estimar los parámetros en los polinomios de aproximación, ya que el análisis de la superficie de respuesta se realiza usando la superficie ajustada. Si la superficie ajustada es una aproximación adecuada de la función de respuesta verdadera, entonces el análisis de la superficie ajustada podría ser aproximadamente igual al análisis del sistema actual<sup>2</sup>. Si se usan diseños experimentales para recolectar los datos (*p.ej.*, diseños factoriales), entonces los parámetros pueden ser estimados más efectivamente. Los diseños que ajustan la superficie de respuesta son llamados **diseños de superficie de respuesta** de los cuales se hablará en la sección 4.6.

---

<sup>2</sup>El sistema actual hace referencia a aquella región en la superficie verdadera en la que nos encontramos actualmente.

### 4.3.1 Funcionamiento de RSM

Un procedimiento es **secuencial** si cualquier etapa del experimento depende de los resultados obtenidos previamente por él, si es así, el procedimiento puede ser tratado como un algoritmo. RSM se considera un procedimiento secuencial ya que utiliza métodos que necesitan de los resultados obtenidos previamente para ejecutar los siguientes, además estos resultados permiten incrementar el grado de complejidad de los modelos que utiliza.

El diagrama de flujo utilizado por RSM se muestra en la figura 4.6 mientras que su funcionamiento, métodos y diseños que utiliza se describen en las siguientes secciones. Posteriormente, en el apéndice B se puede observar un ejemplo numérico de dicha metodología.

#### 4.3.1.1. Variables

Consideremos un experimento en el cual se desea maximizar la respuesta,  $y$ , de un proceso donde intervienen dos variables,  $x_1$  y  $x_2$ . El experimentador determina la región de exploración de cada variable. Para simplificar las operaciones, las variables independientes son codificadas en un intervalo de  $[-1, 1]$ . Entonces,  $\xi_1$  denota la variable natural  $x_1$  y  $\xi_2$  denota la variable natural  $x_2$ , mientras que las variables codificadas estarán dadas por  $x_1$  y  $x_2$ .

$$x_1 = \frac{\xi_1 - center\_point_1}{step\_size_1} \quad \text{y} \quad x_2 = \frac{\xi_2 - center\_point_2}{step\_size_2}$$

donde  $center\_point_1$  y  $center\_point_2$  son los puntos centrales de diseño y  $step\_size_1$  y  $step\_size_2$  son los tamaños de paso para cada variable del proceso. Los puntos centrales así como los tamaños de paso son elegidos por el experimentador.

#### 4.3.1.2. Hipótesis: aproximación lineal

Suponiendo que el punto óptimo se encuentra en una región con curvatura en la superficie. Al iniciar el procedimiento, RSM considera que nos encontramos en una región plana y lo suficientemente alejada del óptimo, por lo tanto un modelo polinomial de primer orden (mostrado en la ecuación

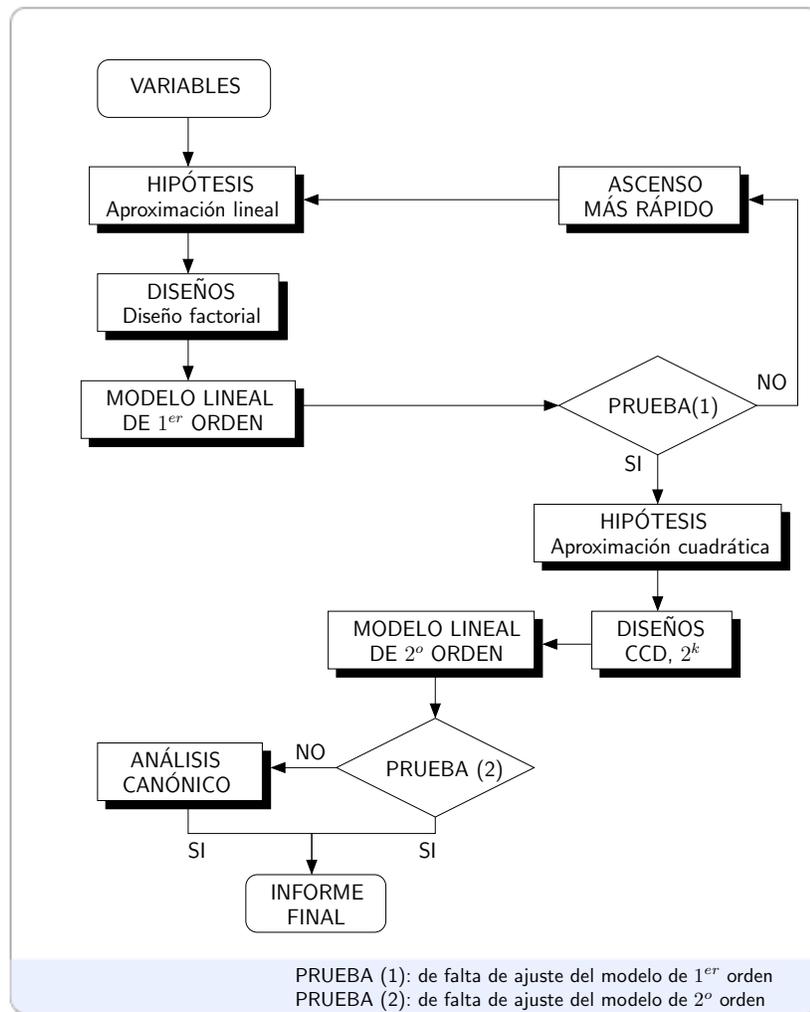


Figura 4.6: Diagrama de flujo de la metodología de superficie de respuesta.

(4.4)) puede ser apropiado para aproximar la verdadera superficie de respuesta en esta etapa del experimento.

#### 4.3.1.3. Diseños factoriales y modelo lineal de primer orden

Haciendo uso del diseño factorial  $2^k$  y agregando réplicas al centro del diseño, se recolectan las muestras para ajustar el modelo de primer orden por medio del método de mínimos cuadrados.

#### 4.3.1.4. Prueba 1: de falta de ajuste del modelo de primer orden

Para verificar la adecuación del modelo de primer orden se hace uso de la prueba estadística  $F$ . Si el modelo de primer orden ha sido ajustado correctamente, podemos movernos desde el centro de diseño hacia alguna dirección de la superficie con el método de ascenso empinado.

El experimentador ejecutará puntos a lo largo de esta ruta hasta observar un decremento en la respuesta. Entonces, otro modelo de primer orden será ajustado cerca del último punto encontrado en el método de ascenso empinado. Se obtendrán las nuevas regiones de exploración para cada variable del proceso, así como las nuevas variables codificadas. Nuevamente, se usará un diseño factorial  $2^k$  con puntos centrales agregados.

#### 4.3.1.5. Hipótesis: aproximación cuadrática

Si encontramos que el modelo de primer orden no es una aproximación adecuada (indicaría que nos encontramos en una superficie curva), un modelo de segundo orden deberá ser implementado; para su aplicación se necesitan recolectar más puntos de diseño para ajustar este modelo.

#### 4.3.1.6. Diseños CCD y $2^k$ y modelo lineal de segundo orden

Los diseños utilizados para ajustar el modelo de segundo orden son los diseños de composición central y diseños factoriales  $2^k$ , que además de ser ortogonales facilitan la rotabilidad de los puntos de diseño.

#### 4.3.1.7. Prueba 2: de falta de ajuste del modelo de segundo orden

Basado en la prueba estadística  $F$ , decidimos si el modelo de segundo orden describe una relación verdadera entre respuesta y variables. En caso que haya sido ajustado correctamente, se necesita caracterizar la superficie de respuesta, v.g., determinar si el punto estacionario encontrado es un punto de máxima respuesta, un punto de respuesta mínima o un punto silla.

#### 4.3.1.8. Análisis canónico

El análisis canónico expresa el modelo ajustado en su forma canónica, los eigenvalores  $\lambda_k$  de dicho modelo determinan que tipo de punto estacionario ha sido encontrado.

### 4.3.2 Método de ascenso empinado

Frecuentemente, las condiciones de operación iniciales de un sistema podrían estar alejadas del óptimo. En tales circunstancias es necesario algún método que permita desplazarnos hacia la vecindad del óptimo rápida y eficientemente. El método de **ascenso empinado** es un procedimiento secuencial que permite ascender por la ruta que presente mayor incremento en la respuesta. En caso de un problema de minimización, la técnica es llamada **método de descenso más rápido**.

Considérese el modelo de primer orden ajustado:

$$\hat{y} = \hat{\beta}_0 + \sum_{i=1}^k \hat{\beta}_i x_i \quad (4.6)$$

la superficie de respuesta de este modelo  $\hat{y}$  son una serie de líneas paralelas como se muestra en la figura 4.7.

La dirección de la ruta de ascenso empinado es la dirección en la cual  $\hat{y}$  se incrementa más rápido. Es común tomar la ruta paralela a la normal de la superficie de respuesta ajustada, ya que además de atravesar la región de interés, los pasos a lo largo de la ruta son proporcionales a los coeficientes de regresión  $\{\hat{\beta}_j\}$ . Sin embargo, el tamaño de paso *step\_size* es determinado por el experimentador basado en el conocimiento previo del proceso o alguna otra consideración.

El algoritmo para determinar las coordenadas de una ruta (punto o dirección) a seguir es el siguiente.

Supongamos que el punto  $x_1 = x_2 = \dots = x_k = 0$  es el punto central (también llamado punto de diseño o punto origen), entonces:

1. Debemos elegir un tamaño de paso en una de las variables del proceso, digamos  $\Delta x_j$ . Usualmente, se elige la variable que conocemos más, o aquella variable que tenga el coeficiente de

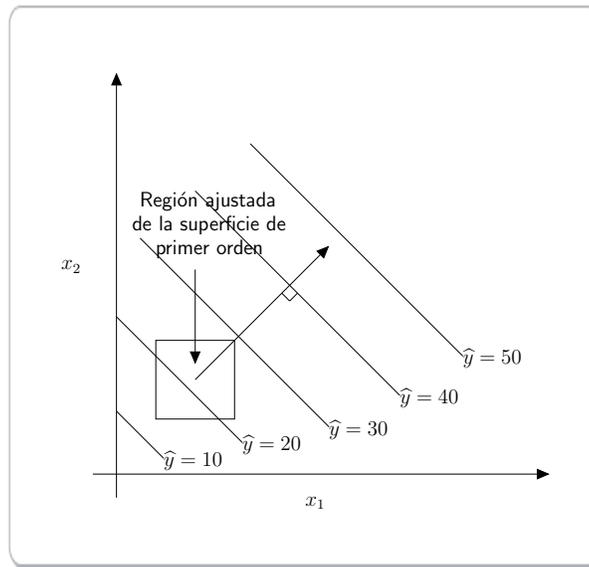


Figura 4.7: Superficie de respuesta de primer orden y ruta de ascenso empinado.

regresión absoluto más grande  $|\hat{\beta}_j|$ .

2. El tamaño de paso de las variables restantes estará dado por:

$$\Delta x_i = \frac{\hat{\beta}_i}{\hat{\beta}_j / \Delta x_j} \quad i = 1, 2, \dots, k \quad i \neq j$$

3. Aunque las variables codificadas son más fáciles de manipular matemáticamente, las variables naturales son usadas por las ejecuciones del proceso. Por lo tanto, se deberá convertir el  $\Delta x_i$  de variable codificada a una variable natural, usando:

$$\Delta x_1 = \frac{\Delta \xi_1}{step\_size} \quad y \quad \Delta x_2 = \frac{\Delta \xi_2}{step\_size}$$

Los experimentos continúan a lo largo de la ruta mientras no se observe un decremento en la respuesta, en este caso, un nuevo modelo de primer orden puede ser ajustado, una nueva ruta de ascenso empinado determinada y de esta manera continuar con el proceso. En caso contrario, el proceso se detiene pues significa que se ha detectado una falta de ajuste en el modelo y por lo tanto se ha llegado a una región con curvatura en la superficie. Por lo tanto, otros métodos pueden ser

implementados para obtener un mejor estimado del punto óptimo.

### 4.3.3 Análisis de superficie de respuesta de segundo orden

Cuando nos encontramos en un punto relativamente cercano al punto óptimo, un modelo que incorpore curvatura será requerido para aproximar la superficie, el modelo comúnmente utilizado es un modelo de segundo orden (mostrado en la ecuación (4.5)).

Con este modelo deseamos encontrar los niveles de  $x_1, x_2, \dots, x_k$  que optimicen la respuesta del proceso. Si dicho punto existe podría estar en el conjunto de  $x_1, x_2, \dots, x_k$ , cuyas derivadas parciales sean  $\partial\hat{y}/\partial x_1 = \partial\hat{y}/\partial x_2 = \dots = \partial\hat{y}/\partial x_k = 0$ , este punto se conoce como **punto estacionario**. El punto estacionario puede representar un punto de **respuesta máxima**, un punto de **respuesta mínima** o un **punto silla**, mostrados en las figuras 4.8(a), 4.8(b) y 4.8(c), respectivamente.

Para localizar el punto estacionario se necesita reescribir el modelo de segundo orden en forma de matriz,

$$\hat{y} = \hat{\beta}_0 + \mathbf{x}'\mathbf{b} + \mathbf{x}'\mathbf{B}\mathbf{x} \quad (4.7)$$

donde

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_k \end{bmatrix} \quad \text{y} \quad \mathbf{B} = \begin{bmatrix} \hat{\beta}_{11} & \hat{\beta}_{12}/2 & \dots & \hat{\beta}_{1k}/2 \\ & \hat{\beta}_{22} & \dots & \hat{\beta}_{2k}/2 \\ & & \ddots & \\ \text{sym} & & & \hat{\beta}_{kk} \end{bmatrix}$$

$\mathbf{b}$  es el vector ( $k \times 1$ ) de los coeficientes de regresión de primer orden y  $\mathbf{B}$  es una matriz simétrica ( $k \times k$ ) donde los elementos de la diagonal principal son los coeficientes cuadráticos puros ( $\hat{\beta}_{ii}$ ) y los elementos fuera de la diagonal son la mitad de los coeficientes cuadráticos mixtos ( $\hat{\beta}_{ij}, i \neq j$ ). La derivada de  $\hat{y}$  con respecto a los elementos del vector  $\mathbf{x}$  es igual a cero, es decir:

$$\frac{\partial\hat{y}}{\partial\mathbf{x}} = \mathbf{b} + 2\mathbf{B}\mathbf{x} = 0 \quad (4.8)$$

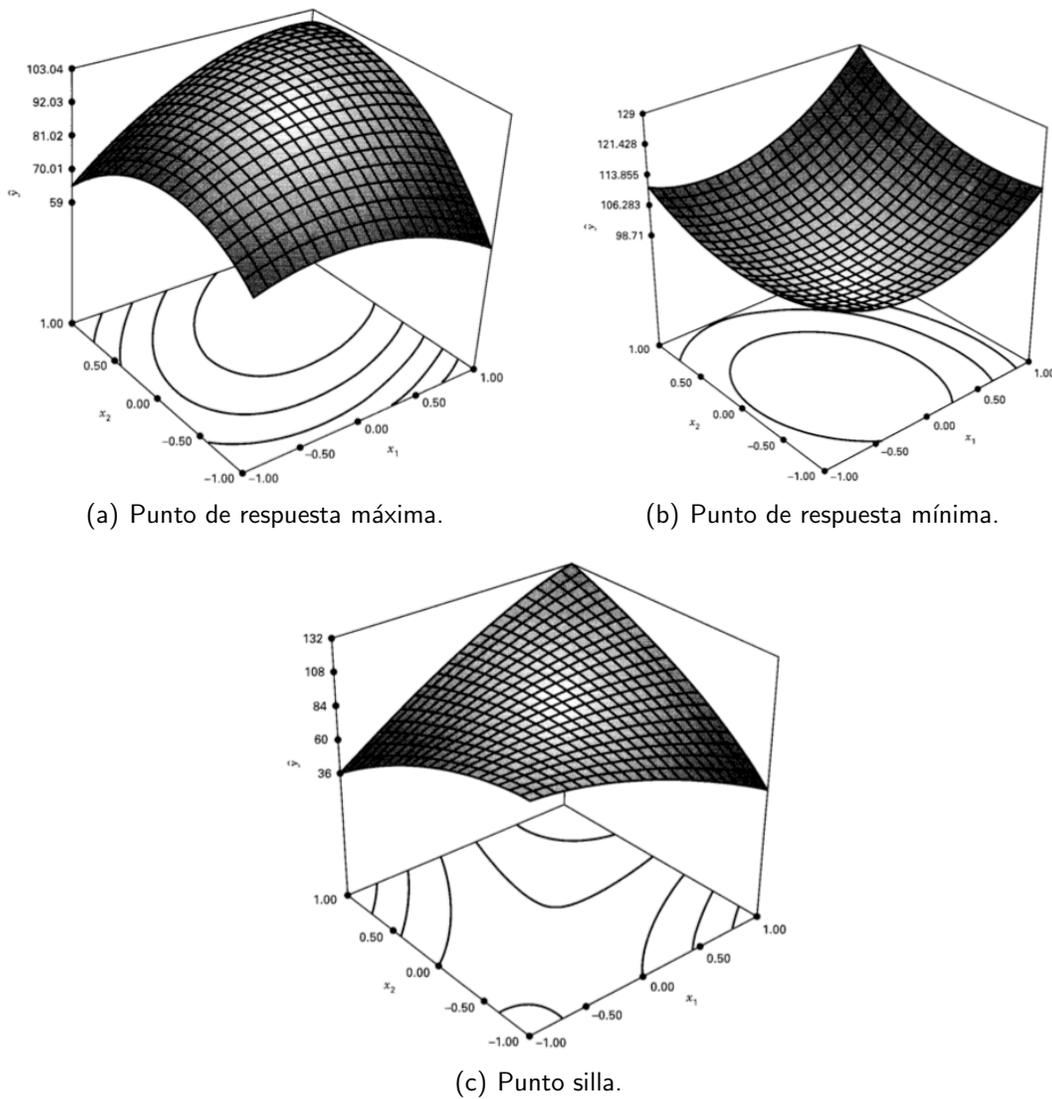


Figura 4.8: Posibles respuestas de un punto estacionario.

El punto estacionario es la solución a la ecuación (4.8),

$$\mathbf{x}_s = -\frac{1}{2}\mathbf{B}^{-1}\mathbf{b} \quad (4.9)$$

o sustituyendo la ecuación (4.9) en la ecuación (4.7) se puede encontrar la respuesta predicha en el punto estacionario como

$$\hat{y}_s = \hat{\beta}_0 + \frac{1}{2}\mathbf{x}'_s\mathbf{b} \quad (4.10)$$

Una vez que se ha encontrado el punto estacionario, se necesita **caracterizar la superficie de respuesta** en las inmediaciones del punto. **Caracterizar** en este caso, significa determinar si el punto estacionario corresponde a un punto máximo, un punto mínimo o un punto silla.

Cuando existen dos o tres variables la forma más simple de hacerlo es examinando la gráfica de contornos del modelo ajustado, pero cuando existen más variables se usa el **análisis canónico**.

Para llevar a acabo el análisis canónico se debe transformar el modelo a un nuevo sistema de coordenadas, donde el punto central es el punto estacionario  $\mathbf{x}_s$ , entonces los ejes de este sistema son rotados hasta quedar paralelos a los ejes principales de la superficie ajustada (ver figura 4.9), lo que dará como resultado la forma canónica del modelo (ecuación (4.11)).

$$\hat{y} = \hat{y}_k + \lambda_1 w_1^2 + \lambda_2 w_2^2 + \dots + \lambda_k w_k^2 \quad (4.11)$$

donde los  $\{w_i\}$  son las variables independientes transformadas y los  $\{\lambda_i\}$  son constantes, representando los *eigen-valores* de la matriz  $\mathbf{B}$ .

La naturaleza de la superficie de respuesta se puede determinar a partir de los signos y magnitudes de los  $\{\lambda_i\}$ , en caso que estos sean todos positivos,  $\mathbf{x}_s$  es un punto de respuesta mínimo; si son negativos,  $\mathbf{x}_s$  es un punto de respuesta máximo; si tienen signos diferentes,  $\mathbf{x}_s$  es un punto silla. Además, la superficie tiene un ascenso más rápido en la  $w_i$  dirección para la cual  $|\lambda_i|$  es el más grande.

## 4.4 Estimación de los parámetros en modelos de regresión lineal

El método de **mínimos cuadrados** es la técnica más utilizada para estimar los coeficientes de regresión en un modelo de regresión lineal múltiple. Supongamos que existen  $n > k$  observaciones en la variable de respuesta denotadas por  $y_1, y_2, \dots, y_n$ . Cada respuesta  $y_i$  posee una observación de todas y cada una de las variables de regresión denotadas por  $x_{ij}$ , donde  $i$  significa la  $i$ -ésima observación o nivel de la variable  $x_j$ . Los datos en la tabla 4.1 expresan esta relación. Se asume

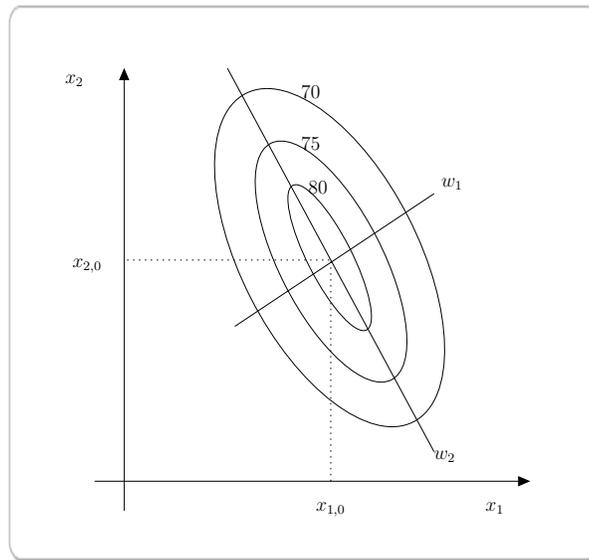


Figura 4.9: Forma canónica del modelo de segundo orden.

que el término de error  $\epsilon$  en el modelo tienen  $E(\epsilon) = 0$  y  $V(\epsilon) = \sigma^2$  y que los  $\{\epsilon_i\}$  son variables aleatorias correlacionadas.

$y$	$x_1$	$x_2$	$\dots$	$x_k$
$y_1$	$x_{11}$	$x_{12}$	$\dots$	$x_{1k}$
$y_2$	$x_{21}$	$x_{22}$	$\dots$	$x_{2k}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$y_n$	$x_{n1}$	$x_{n2}$	$\dots$	$x_{nk}$

Tabla 4.1: Datos para la regresión lineal múltiple.

Se puede reescribir el modelo de regresión lineal múltiple (ecuación (4.4)) en términos de las observaciones mostradas en la tabla 4.1, es decir:

$$\begin{aligned}
 y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + \epsilon_i \\
 &= \beta_0 + \sum_{j=1}^k \beta_j x_{ij} + \epsilon_i, \quad i = 1, 2, \dots, n
 \end{aligned} \tag{4.12}$$

El método de mínimos cuadrados elige los  $\beta$ 's en la ecuación (4.12) de tal forma que la suma de

cuadrados de los errores  $\epsilon_i$  sea minimizada, es decir:

$$\begin{aligned} L &= \sum_{i=1}^n \epsilon_i^2 \\ &= \sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^k \beta_j x_{ij} \right)^2 \end{aligned} \quad (4.13)$$

Por lo tanto, la función  $L$  es minimizada con respecto a  $\beta_0, \beta_1, \dots, \beta_k$ . Los estimadores de mínimos cuadrados denotados como  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_k$ , deberían satisfacer la ecuación (4.14):

$$\left. \frac{\partial L}{\partial \beta_0} \right|_{\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_k} = -2 \sum_{i=1}^n (y_i - \hat{\beta}_0 - \sum_{j=1}^k \hat{\beta}_j x_{ij}) = 0 \quad (4.14)$$

y

$$\left. \frac{\partial L}{\partial \beta_j} \right|_{\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_k} = -2 \sum_{i=1}^n (y_i - \hat{\beta}_0 - \sum_{j=1}^k \hat{\beta}_j x_{ij}) x_{ij} = 0, \quad j = 1, 2, \dots, k \quad (4.15)$$

Simplificando la ecuación (4.15) obtenemos:

$$\begin{aligned} n\hat{\beta}_0 &+ \hat{\beta}_1 \sum_{i=1}^n x_{i1} + \hat{\beta}_2 \sum_{i=1}^n x_{i2} + \dots + \hat{\beta}_k \sum_{i=1}^n x_{ik} &= \sum_{i=1}^n y_i \\ \hat{\beta}_0 \sum_{i=1}^n x_{i1} &+ \hat{\beta}_1 \sum_{i=1}^n x_{i1}^2 + \hat{\beta}_2 \sum_{i=1}^n x_{i1}x_{i2} + \dots + \hat{\beta}_k \sum_{i=1}^n x_{i1}x_{ik} &= \sum_{i=1}^n x_{i1}y_i \\ \vdots &\vdots &\vdots \\ \hat{\beta}_0 \sum_{i=1}^n x_{ik} &+ \hat{\beta}_1 \sum_{i=1}^n x_{ik}x_{i1} + \hat{\beta}_2 \sum_{i=1}^n x_{ik}x_{i2} + \dots + \hat{\beta}_k \sum_{i=1}^n x_{ik}^2 &= \sum_{i=1}^n x_{ik}y_i \end{aligned} \quad (4.16)$$

Notemos que son  $p = k + 1$  ecuaciones, llamadas **ecuaciones normales de mínimos cuadrados**, una para cada coeficiente de regresión desconocido. Sus soluciones, serán los **estimadores de mínimos cuadrados** de los coeficientes de regresión denotados por  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_k$ .

Es más cómodo resolver las ecuaciones normales si se expresan en forma de matriz, por lo tanto el modelo de la ecuación (4.12) puede reescribirse como:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

donde

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}, \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$\mathbf{y}$  es el vector de observaciones de tamaño  $(n \times 1)$ ,  $\mathbf{X}$  es la matriz de niveles de las variables independientes de tamaño  $(n \times p)$ ,  $\boldsymbol{\beta}$  es el vector de coeficientes de regresión de tamaño  $(p \times 1)$  y  $\boldsymbol{\epsilon}$  es el vector de errores aleatorios de tamaño  $(n \times 1)$ .

Por lo tanto, el problema es encontrar el vector de estimadores de mínimos cuadrados  $\hat{\boldsymbol{\beta}}$  que minimice:

$$L = \sum_{i=1}^n \epsilon_i^2 = \boldsymbol{\epsilon}'\boldsymbol{\epsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

donde  $L$  puede expresarse como:

$$\begin{aligned} L &= \mathbf{y}'\mathbf{y} - \boldsymbol{\beta}'\mathbf{X}'\mathbf{y} - \mathbf{y}'\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \\ &= \mathbf{y}'\mathbf{y} - 2\boldsymbol{\beta}'\mathbf{X}'\mathbf{y} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \end{aligned} \quad (4.17)$$

por que  $\boldsymbol{\beta}'\mathbf{X}'\mathbf{y}$  es una matriz de tamaño  $(1 \times 1)$ , o escalar y su transpuesta  $(\boldsymbol{\beta}'\mathbf{X}'\mathbf{y})' = \mathbf{y}'\mathbf{X}\boldsymbol{\beta}$  también lo es. Los estimadores de mínimos cuadrados necesitan satisfacer:

$$\left. \frac{\partial L}{\partial \boldsymbol{\beta}} \right|_{\hat{\boldsymbol{\beta}}} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = 0$$

lo que se simplifica en:

$$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{y} \quad (4.18)$$

la ecuación (4.18) representa las ecuaciones normales de mínimos cuadrados en forma de matriz, la cual es idéntica a la ecuación (4.16). Para resolver el sistema de ecuaciones, multiplicamos ambos lados de la ecuación (4.18) por el inverso de  $\mathbf{X}'\mathbf{X}$ , entonces el estimador de mínimos cuadrados de  $\beta$  es:

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \quad (4.19)$$

Escribiendo la ecuación (4.18) en detalle obtenemos:

$$\begin{bmatrix} n & \sum_{i=1}^n x_{i1} & \sum_{i=1}^n x_{i2} & \cdots & \sum_{i=1}^n x_{ik} \\ \sum_{i=1}^n x_{i1} & \sum_{i=1}^n x_{i1}^2 & \sum_{i=1}^n x_{i1}x_{i2} & \cdots & \sum_{i=1}^n x_{i1}x_{ik} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum_{i=1}^n x_{ik} & \sum_{i=1}^n x_{ik}x_{i1} & \sum_{i=1}^n x_{ik}x_{i2} & \cdots & \sum_{i=1}^n x_{ik}^2 \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_{i1}y_i \\ \vdots \\ \sum_{i=1}^n x_{ik}y_i \end{bmatrix}$$

De esta manera se puede observar que  $\mathbf{X}'\mathbf{X}$  es una matriz simétrica de tamaño  $(p \times p)$  y  $\mathbf{X}'\mathbf{y}$  es un vector columna de tamaño  $(p \times 1)$ . Los elementos de la diagonal de  $\mathbf{X}'\mathbf{X}$  son la suma de cuadrados de los elementos en las columnas de  $\mathbf{X}$  y los elementos fuera de la diagonal son la suma de productos cruzados de los elementos en las columnas de  $\mathbf{X}$ . Además, notemos que los elementos de  $\mathbf{X}'\mathbf{y}$  son la suma de productos cruzados de las columnas de  $\mathbf{X}$  y las observaciones  $\{y_i\}$ .

Por lo tanto, el modelo de regresión ajustado es:

$$\hat{y} = \mathbf{X}\hat{\beta} \quad (4.20)$$

En notación escalar, el modelo ajustado es:

$$\hat{y}_i = \hat{\beta}_0 + \sum_{j=1}^k \hat{\beta}_j x_{ij}, \quad i = 1, 2, \dots, n$$

La diferencia entre la observación actual y su correspondiente valor ajustado  $e_i = y_i - \hat{y}_i$  se llama **residuo**. El vector de residuos de tamaño  $(n \times 1)$  se denota como:

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} \quad (4.21)$$

#### 4.4.1 Estimación de la varianza $\sigma^2$

En los modelos de regresión se necesita **estimar**  $\sigma^2$ . Para desarrollar un estimador de este parámetro, consideremos la suma de cuadrados de los residuos, como:

$$\begin{aligned}
 SS_E &= \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\
 &= \sum_{i=1}^n e_i^2 \\
 &= \mathbf{e}'\mathbf{e}
 \end{aligned}
 \tag{4.22}$$

Sustituyendo  $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$ , tenemos:

$$\begin{aligned}
 SS_E &= (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \\
 &= \mathbf{y}'\mathbf{y} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} - \mathbf{y}'\mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} \\
 &= \mathbf{y}'\mathbf{y} - 2\hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} + \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}}
 \end{aligned}
 \tag{4.23}$$

Y como  $\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{y}$ , entonces:

$$SS_E = \mathbf{y}'\mathbf{y} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} \tag{4.24}$$

La ecuación (4.24) se conoce como **error** o **suma residual de cuadrados**, con  $n - p$  grados de libertad. Entonces,

$$E(SS_E) = \sigma^2(n - p)$$

Un estimador imparcial de  $\sigma^2$  está dado por

$$\hat{\sigma}^2 = \frac{SS_E}{n - p} \tag{4.25}$$

### 4.4.2 Propiedades de los estimadores

Como hemos visto, el método de mínimos cuadrados produce un estimador imparcial de los parámetros  $\beta$  en el modelo de regresión lineal, lo cual se puede demostrar tomando el valor esperado de  $\hat{\beta}$  como sigue:

$$\begin{aligned}
 E(\hat{\beta}) &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}] \\
 &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\beta + \epsilon)] \\
 &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\epsilon] \\
 &= \beta
 \end{aligned} \tag{4.26}$$

(4.27)

ya que  $E(\epsilon) = 0$  y  $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} = \mathbf{I}$ , entonces  $\hat{\beta}$  es un estimador imparcial de  $\beta$ .

La propiedad de varianza de  $\hat{\beta}$  se expresa en la **matriz de covarianza** como:

$$Cov(\hat{\beta}) \equiv E\{[\hat{\beta} - E(\hat{\beta})][\hat{\beta} - E(\hat{\beta})]'\} \tag{4.28}$$

la cual es sólo una matriz simétrica, donde los  $i$ -ésimos elementos de la diagonal principal representan la varianza de los coeficientes de regresión individual  $\hat{\beta}_i$  y los  $(ij)$ -ésimos elementos representan la covarianza entre  $\hat{\beta}_i$  y  $\hat{\beta}_j$ , por lo tanto la matriz de covarianza de  $\hat{\beta}$  es:

$$Cov(\hat{\beta}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \tag{4.29}$$

## 4.5 Prueba de hipótesis en modelos de regresión múltiple

Cuando se utilizan los modelos de regresión lineal múltiple se requieren **pruebas de hipótesis** para medir su utilidad. Estos procedimientos requieren que los errores  $\epsilon_i$  en el modelo estén normal e independientemente distribuidos con media cero y varianza  $\sigma^2$ , v.g.,  $\epsilon \sim NID(0, \sigma^2)$ . Como resultado

de esta hipótesis, las observaciones  $y_i$  estarán normal e independientemente distribuidas con media  $\beta_0 + \sum_{j=1}^k \beta_j x_{ij}$  y varianza  $\sigma^2$ .

### 4.5.1 Prueba para la significancia de regresión

La prueba para la **significancia de regresión** es una prueba para determinar si existe una relación lineal entre la variable de respuesta  $y$  y las variables de regresión  $x_1, x_2, \dots, x_k$ . Para iniciar el procedimiento es necesario suponer dos hipótesis  $H_0$  y  $H_1$ , de la forma:

$$\begin{aligned} H_0 &: \beta_1 = \beta_2 = \dots = \beta_k = 0 \\ H_1 &: \beta_j \neq 0 \quad \text{para al menos un } j \end{aligned} \quad (4.30)$$

El rechazo de la hipótesis  $H_0$  en la ecuación (4.30) implica que al menos una variable de regresión  $x_1, x_2, \dots, x_k$  contribuya significativamente al modelo. La prueba involucra un análisis de varianza, dividiendo la suma total de cuadrados  $SS_T$  en una suma de cuadrados debido al modelo (o debido a la regresión) y una suma de cuadrados debido al residuo (o error), digamos:

$$SS_T = SS_R + SS_E \quad (4.31)$$

Sin embargo, si la hipótesis nula  $H_0 : \beta_1 = \beta_2 = \dots = \beta_k = 0$  es verdadera, entonces la distribución de  $SS_R/\sigma^2$  tiene la forma de  $\chi_k^2$  (chi-cuadrada) donde el número de grados de libertad para  $\chi^2$  es igual al número de variables de regresión en el modelo. Además,  $SS_E/\sigma^2$  se distribuye como  $\chi_{n-k-1}^2$  y  $SS_E$  y  $SS_R$  son independientes. El procedimiento de prueba para  $H_0 : \beta_1 = \beta_2 = \dots = \beta_k = 0$  es:

$$F_0 = \frac{SS_R/k}{SS_E/(n-k-1)} = \frac{MS_R}{MS_E} \quad (4.32)$$

se rechaza  $H_0$  si  $F_0$  excede a  $F_{\alpha, k, n-k-1}$ . Alternativamente, se puede utilizar el enfoque  $P$ -value para probar la hipótesis y, entonces, rechazar  $H_0$  si el  $P$ -value para el estadístico  $F_0$  es menor que  $\alpha$ . Existen diversos valores para  $\alpha$ , los cuales dependen del número de variables independientes en el

modelo  $k$ ,  $n > k$  el número de observaciones en la variable de respuesta y el valor de  $\alpha$ , es decir:

$$F_0 = f(\alpha, k, n - k - 1)$$

Es importante señalar que el valor de  $n$  depende del diseño utilizado para ajustar la superficie de respuesta, del cual hablaremos en la sección 4.6. Por otra parte, el valor de  $F_0$  puede obtenerse de las tablas de puntos porcentuales de la distribución  $F$  obtenidas de [35].

Una formula para  $SS_R$  puede derivarse a partir de la suma de errores de cuadrados  $SS_E$ , es decir:

$$SS_E = \mathbf{y}'\mathbf{y} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} \quad (4.33)$$

Y además,

$$SS_T = \sum_{i=1}^n y_i^2 - \frac{(\sum_{i=1}^n y_i)^2}{n} = \mathbf{y}'\mathbf{y} - \frac{(\sum_{i=1}^n y_i)^2}{n}$$

$SS_E$  se puede reescribir como:

$$SS_E = \mathbf{y}'\mathbf{y} - \frac{(\sum_{i=1}^n y_i)^2}{n} - \left[ \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} - \frac{(\sum_{i=1}^n y_i)^2}{n} \right]$$

o

$$SS_E = SS_T - SS_R$$

Por lo tanto, la suma de regresión de cuadrados es:

$$SS_R = \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} - \frac{(\sum_{i=1}^n y_i)^2}{n} \quad (4.34)$$

entonces, la suma total de cuadrados es:

$$SS_T = \mathbf{y}'\mathbf{y} - \frac{(\sum_{i=1}^n y_i)^2}{n} \quad (4.35)$$

Por otro lado, es posible determinar si los términos **no significantes** han sido incluidos en el

modelo con el coeficiente de determinación múltiple  $R^2$ , donde:

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T} \quad (4.36)$$

$R^2$  es una medida de la cantidad de reducción en la variabilidad de  $y$  obtenida por usar variables regresoras  $x_1, x_2, \dots, x_k$  en el modelo. Aunque, un valor grande de  $R^2$  no necesariamente implica que el modelo de regresión sea el correcto, ya que agregando términos al modelo  $R^2$  se incrementa. Sin embargo, existe una variante de  $R^2$  llamada **estadístico  $R^2$  ajustado** definida como:

$$R_{adj}^2 = 1 - \frac{SS_E/(n-p)}{SS_T/(n-1)} = 1 - \left(\frac{n-1}{n-p}\right)(1 - R^2) \quad (4.37)$$

A diferencia de  $R^2$ ,  $R_{adj}^2$  no siempre se incrementa conforme se agregan variables al modelo, de hecho si los términos innecesarios son agregados,  $R_{adj}^2$  tiende a decrecer. Si  $R^2$  y  $R_{adj}^2$  difieren drásticamente, entonces se concluye que términos no significantes han sido incluidos en el modelo.

## 4.6 Diseños para ajustar la superficie de respuesta

Ajustar y analizar la superficie de respuesta puede resultar cómodo si se eligen correctamente los diseños experimentales para dicho proceso. Algunas características deseables al momento de escoger un diseño podrían ser las siguientes:

1. Proporcionar una distribución razonable de los puntos de diseño en la región de interés.
2. Permitir la adecuación del modelo, incluyendo la falta de ajuste.
3. Permitir la construcción secuencial de diseños de orden mayor.
4. Proveer un estimando del error.
5. Proveer un estimado preciso de los coeficientes del modelo.
6. Permitir la predicción de la varianza a través de la región experimental.

7. Robustez razonable en contra de valores externos o valores perdidos.
8. No requerir un gran número de ejecuciones.
9. No solicitar un gran número de niveles en las variables independientes.
10. Asegurar la simplicidad al momento de calcular los parámetros del modelo.

Aunque a veces estas características entran en conflicto [37].

#### 4.6.1 Diseños para ajustar el modelo de primer orden

Recordemos el modelo de primer orden con  $k$  variables

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \epsilon \quad (4.38)$$

Los **diseños ortogonales** son la única clase de diseños que minimizan la varianza de los coeficientes de regresión  $\{\hat{\beta}_i\}$ . Un diseño de primer orden es ortogonal si los elementos fuera de la diagonal de la matriz  $\mathbf{X}'\mathbf{X}$  son todos cero. Esto implica que los productos cruzados de las columnas de la matriz  $\mathbf{X}$  suman cero.

La clase de diseños ortogonales incluye a los diseños factoriales  $2^k$  y a los diseños factoriales fraccionarios, en los cuales los efectos principales no están ligados. Por lo tanto los niveles de cada factor pueden codificarse en alto y bajo (como se vio anteriormente).

Sin embargo, los diseños  $2^k$  no permiten un estimado del error experimental a menos que se repliquen las ejecuciones. Por lo tanto, es común agregar observaciones al **centro de diseño** (v.g., el punto  $x_i = 0, i = 1, 2, \dots, k$ ). Las ventajas de agregar dichas réplicas es que permiten calcular un estimado del error  $\hat{\sigma}^2$ , no afecta los  $\{\hat{\beta}_i\}$  para  $i \leq 1$  y no altera la propiedad ortogonal del diseño, pero por otro lado el estimado de  $\beta_0$  se convierte en el promedio total de todas las observaciones.

### 4.6.2 Diseños para ajustar el modelo de segundo orden

El **diseño de composición central** (CCD, por sus siglas en Inglés) es el diseño más utilizado para ajustar los modelos de segundo orden. Básicamente, consiste de un diseño factorial  $2^k$  con  $n_F$  ejecuciones,  $2k$  ejecuciones axiales o (con forma de estrella) y  $n_C$  ejecuciones en el centro.

En la práctica el CCD surge de un experimento secuencial. Es decir, si un diseño factorial  $2^k$  ha sido usado para ajustar el modelo de primer orden y el modelo presenta una falta de ajuste, entonces se agregan ejecuciones axiales para permitir a los términos cuadráticos incorporarse en el modelo. Por lo tanto, el CCD es más eficiente cuando se requieren ajustar modelos de segundo orden.

Sin embargo, existen dos parámetros en CCD que deben ser especificados: la distancia  $\alpha$  desde el centro de diseño hacia los puntos axiales y el número de puntos centrales  $n_C$ .

Es deseable que el modelo de segundo orden proporcione consistencia y varianza estable de la respuesta predicha en los puntos de interés  $\mathbf{x}$  es decir:

$$V[\hat{y}(\mathbf{x})] = \sigma^2 \mathbf{x}' (\mathbf{X}'\mathbf{X})^{-1} \mathbf{x} \quad (4.39)$$

Los diseños de superficie de respuesta de segundo orden pueden ser **rotables**, *v.g.*, que la  $V[\hat{y}(\mathbf{x})]$  es la misma para todos los puntos  $\mathbf{x}$  que están a la misma distancia del centro de diseño (la varianza de la respuesta predicha es constante en esfera). La rotabilidad es una característica deseable en un diseño de superficie de respuesta, ya que el propósito de RSM es encontrar el punto óptimo o adecuado donde las condiciones de operación sean satisfechas, por lo tanto es deseable un diseño que provea igual precisión de estimación en todas las direcciones. Es importante señalar que los diseños de primer orden ortogonales también son rotables.

Un diseño de composición central es rotable si el valor de  $\alpha$  se elige correctamente. El valor de  $\alpha$  depende del número de puntos en la porción factorial del diseño. Si  $\alpha = (n_F)^{1/4}$  produce un diseño de composición central rotatable, donde  $n_F$  es el número de puntos usados en la porción factorial del diseño.

Por otro lado, la elección de  $\alpha$  en el CCD es determinado por la región de interés. Cuando esta

región es una esfera, el diseño puede incluir ejecuciones centrales para proveer una varianza estable de la respuesta predicha. Generalmente, se recomienda agregar de tres a cinco ejecuciones.

# 5

## Metodología

### 5.1 Introducción

Existen diferentes algoritmos evolutivos que han probado ser efectivos en numerosas aplicaciones de optimización. Sin embargo, la efectividad de los mismos esta fuertemente ligada a los valores de sus parámetros. Consecuentemente, es importante calibrar sus parámetros adecuadamente.

En este capítulo se propone una metodología basada en métodos estadísticos computacionales para calibrar los parámetros de un AE, tales como el diseño de experimentos (DOE), análisis de varianza (ANOVA), modelos de regresión y métodos de superficie de respuesta (RSM). Debido al uso extenso y buen desempeño que PSO ha mostrado en la práctica, se probará la metodología propuesta para determinar si existe una topología de interconexión entre partículas, un conjunto de valores para el peso de inercia ( $w$ ) y los coeficientes de aceleración ( $c_1$  y  $c_2$ ), que provoquen un incremento en el desempeño promedio del algoritmo.

A demás de los valores de  $w$ ,  $c_1$ ,  $c_2$  y la topología de interconexión para comunicar las partículas (ver sección 2.5), existen otros parámetros que pueden afectar significativamente el desempeño de PSO, v.g., el tamaño de la población, el número de iteraciones, la semilla de números aleatorios para

generar la población inicial, entre otros. Sin embargo, este trabajo será enfocado en evaluar únicamente los efectos de sus parámetros intrínsecos. Por lo tanto, sólo se estudiará el comportamiento que induce  $w$ ,  $c_1$ ,  $c_2$  y la topología de interconexión.

A continuación se describen los experimentos realizados para validar la propuesta metodológica. Estos experimentos fueron desarrollados en el lenguaje C y R.

## 5.2 Experimento 1

Debido a los buenos resultados que PSO ha presentado en la práctica, decidimos analizar las diferentes versiones de PSO: PSO, PSOB, PSOBEXP y PSOBDE (descritas en la sección 2.5) con la finalidad de obtener aquella versión que mejor resuelva el conjunto de pruebas propuesto manteniendo un número bajo de evaluaciones de la función objetivo. También queremos determinar si la topología de interconexión utilizada para comunicar a las partículas, afecta el desempeño de los algoritmos propuestos, e identificar qué topología mejora el desempeño promedio de la versión básica de PSO.

Para que estos objetivos sean cumplidos, se implementarán seis topologías de interconexión: anillo, completamente conectada, malla, toroidal y árbol (mostradas en la figura 2.3); y dos algoritmos de agrupamiento: *k-means* y jerárquico (descritos en la sección 2.6), en las cuatro versiones de PSO. Algunos aspectos sobre su implementación serán:

- Las conexiones entre partículas serán sin dirección, sin peso y no varían en el curso de la prueba.
- Las topologías de interconexión serán construidas basadas en el índice de cada partícula, por lo tanto cada partícula posee un identificador único en la población entera.
- En los algoritmos de agrupamiento se usará la distancia euclidiana para medir la distancia entre partículas.
- Para los algoritmos de agrupamiento, las conexiones se actualizarán dinámicamente en cada iteración del algoritmo.

Adicionalmente, para validar el experimento se seleccionaron nueve funciones de prueba de la literatura especializada. La definición matemática de estas funciones se presentan en el apéndice C, tal conjunto de funciones provee un buen balance entre funciones unimodales y multimodales. Para cada una de estas funciones la meta es encontrar el mínimo global.

Los parámetros de implementación para cada una de las versiones de PSO son:

- Para el algoritmo básico de PSO, los valores de sus parámetros serán:  $w = 0.72$  y  $c_1 = c_2 = 1.49$ .
- Para el algoritmo de PSOBBDDE se usará una probabilidad de recombinación,  $p_r = 0.9$  sugerida por Storn y Price en [46].
- Para todos los algoritmos usados en esta sección se usará un tamaño de cúmulo  $s = 50$  con 200 iteraciones cada uno.

Los valores de estos parámetros serán predeterminados para todos los experimentos, los cuales usarán un control de parámetros estático. La población inicial será generada a partir de una distribución uniforme entre los límites especificados en cada función. Cada una de estas configuraciones se ejecutará 30 veces en las nueve funciones de prueba con 30 variables (dimensiones) cada una.

La distribución entre partículas para las topologías en malla y toroidal, será de  $10 \times 5$ . Para los algoritmos de agrupamiento *k-means* y jerárquico, se establecerán 4 grupos.

### 5.2.1 Resultados

Los resultados del primer experimento se muestran en las figuras 5.1 a la 5.3. Se ha decidido utilizar gráficas de cajas debido a las ventajas que presentan cuando se requiere comparar dos grupos de muestras (una descripción más detallada sobre graficas de cajas se proporciona en el apéndice D).

Basados en estos resultados, se describe a continuación el análisis realizado para cada una de las topologías implementadas.

### ■ Topología en anillo.

Cuando se utilizó esta topología, la versión PSOBBDE fue la que mostró el mejor desempeño en siete de las nueve funciones de prueba, siendo únicamente superado por la versión básica de PSO en la función *rotated hyper-ellipsoid* (ver figura 5.2(b)) y el problema generalizado de *Schwefel 2.26* (ver figura 5.2(c)).

La versión básica de PSO presentó un desempeño similar al de PSOBBDE en la función de *Rosenbrock* (ver figura 5.2(a)), en la cual ambos obtuvieron los mejores resultados.

Por otro lado, la versión de PSOBBEXP fue la que presentó el peor desempeño en todas las funciones de prueba, comparado con las demás versiones de PSO.

La versión de PSOBB nunca obtuvo el mejor desempeño en alguna de las funciones de prueba, aunque tampoco mostró un comportamiento desfavorable.

### ■ Topología en malla

Al igual que la topología en anillo, la versión de PSOBBDE fue la que mostró el mejor comportamiento en seis funciones de prueba (ver figuras 5.1(a), 5.1(b), 5.1(c), 5.3(a), 5.3(b) y 5.3(c)). Presentó un comportamiento similar a las versiones de PSO y PSOBB en la función de *Rosenbrock* (ver figura 5.2(a)), en la cual las tres obtuvieron los mejores resultados.

Para la función *rotated hyper-ellipsoid* (ver figura 5.2(b)) y el problema generalizado de *Schwefel 2.26* (ver figura 5.2(c)) el mejor desempeño lo obtuvo la versión básica de PSO.

De igual manera que la topología en anillo, la versión de PSOBBEXP presentó el peor comportamiento en todas las funciones de prueba, exceptuando quizás el problema generalizado de *Schwefel 2.26* (ver figura 5.2(c)), donde fue ligeramente mejor que PSOBBDE.

### ■ Topología toroidal

En la topología toroidal, la versión que de PSOBBDE fue la técnica que se desempeño mejor en seis funciones de prueba (ver figuras 5.1(a), 5.1(b), 5.1(c), 5.3(a), 5.3(b) y 5.3(c)). Fue

superado por la versión básica de PSO en la función *rotated hyper-ellipsoid* (ver figura 5.2(b)) y en el problema generalizado de *Schwefel 2.26* (ver figura 5.2(c)).

PSO y PSOBBD presentan el mismo comportamiento que PSOBBDDE en la función de *Rosenbrock* (ver figura 5.2(a)), donde las tres heurísticas consiguieron el mejor comportamiento.

Nuevamente, la versión PSOBBDDE fue la que presentó el peor comportamiento en todas las funciones de prueba.

#### ■ Topología completamente conectada

PSOBBDDE presentó nuevamente un buen comportamiento en la mayoría de las funciones de prueba, manteniendo su mal desempeño en el problema generalizado de *Schwefel 2.26* (ver figura 5.2(c)), donde en esta ocasión la topología PSOBBD supero la versión de PSO y obtuvo el mejor comportamiento.

Sorpresivamente la versión de PSOBBDDE mejoró su comportamiento en cuatro de las funciones (mostradas en las figuras 5.1(a), 5.1(b), 5.1(c) y 5.2(a)), aunque mantuvo su mal comportamiento en la la función *rotated hyper-ellipsoid* (ver figura 5.2(b)).

Otro cambio notable lo presentó versión básica de PSO, que si bien no había obteniendo los mejores resultados en las topologías anteriores, mostraba un comportamiento favorable. PSO obtuvo el peor comportamiento en cinco de nueve funciones de prueba, y dejó de obtener los mejores resultados para el problema generalizado de *Schwefel 2.26* (ver figura 5.2(c)) al implementar esta topología.

#### ■ Topología en estrella

Al igual que en la topología completamente conectada, la versión de PSO obtuvo el peor comportamiento en cinco funciones, sólo mantuvo el mejor comportamiento en la función *rotated hyper-ellipsoid* (ver figura 5.2(b)).

PSOBBDDE obtuvo el peor comportamiento en dos funciones de prueba (ver figuras 5.2(b) y 5.2(b))

Por otro lado, las cuatro versiones: PSO, PSOBB, PSOBBEXP y PSOBBDE presentaron los mejores resultados en la función de *Rosenbrock* (ver figura 5.2(a)).

#### ■ Topología en árbol

PSOBBEXP volvió a presentar los peores resultados en ocho funciones de prueba. Por otro lado, PSO volvió a obtener los mejores resultados en el problema generalizado de *Schwefel* 2.26 (ver figura 5.2(c)) y en la función *rotated hyper-ellipsoid* (ver figura 5.2(b)).

PSOBBDE obtuvo los mejores resultados en siete funciones, manteniendo los peores resultados en el problema generalizado de *Schwefel* 2.26 (ver figura 5.2(c)).

#### ■ Algoritmo de agrupamiento k-means

PSOBBDE y PSOBB presentaron los mejores comportamientos en cuatro funciones de prueba (mostradas en las figuras 5.1(a), 5.1(b), 5.1(c) y 5.2(a)).

PSOBBEXP fue el enfoque que peor se comportó en todas las funciones de prueba.

Y por otro lado, la versión básica de PSO mantuvo su mejor comportamiento en la función *rotated hyper-ellipsoid* (ver figura 5.2(b)) y en el problema generalizado de *Schwefel* 2.26 (ver figura 5.2(c)).

#### ■ Algoritmo de agrupamiento jerárquico

Al igual que el algoritmo de *k-means*, PSOBBDE y PSOBB fueron los que mejor se desempeñaron en cuatro funciones de prueba (mostradas en las figuras 5.1(a), 5.1(b), 5.1(c) y 5.2(a)). Al contrario de PSO que obtuvo el peor comportamiento en estas mismas funciones.

Y aunque resulta repetitivo, PSOBBDE obtuvo los mejores resultados en siete funciones de prueba.

Por otro lado, PSOBBEXP mejoró su comportamiento con esta topología a diferencia de las anteriores formas de interconexión.

En la función de *Rosenbrock* (ver figura 5.2(a)) las versiones PSO, PSOBB y PSOBBDE mostraron el mejor comportamiento.

PSO mantuvo el mejor comportamiento en la función *rotated hyper-ellipsoid* (ver figura 5.2(b)). Y la versión PSOBBD fue la que mejor se comportó en el problema generalizado de *Schwefel 2.26* (ver figura 5.2(c)).

### 5.2.2 Conclusiones

Se puede concluir que la versión PSOBBD se desempeñó mejor que cualquier otro enfoque, presentando un comportamiento robusto en todas las topologías de interconexión.

Aunque su desempeño en la función *rotated hyper-ellipsoid* (ver la figura 5.2(b)) y el problema generalizado de *Schwefel* (ver la figura 5.2(c)) fue inferior con respecto a la versión básica de PSO. De igual forma podemos decir que el comportamiento de la versión PSOBBD fue altamente independiente de la topología de interconexión que se utilizó.

Contrario a PSOBBD, la versión PSOBBD EXP es altamente dependiente de la topología de interconexión, ya que sólo presenta un buen comportamiento cuando se implementan las topologías: completamente conectada y en estrella.

PSOBBD en general presenta un buen desempeño (después de PSOBBD). Las topologías con las que se comporta mejor fueron: completamente conectada y en estrella (véase las figuras 5.1(b) y 5.1(c) por ejemplo); precedidas por las topologías, en malla y toroidal. Es importante notar que la versión PSOBBD al igual que la versión PSOBBD EXP presentaron buenos resultados únicamente cuando se utilizan las topologías estrella y completamente conectada.

Por otro lado, aunque la versión básica de PSO no es la mejor de entre las cuatro versiones analizadas, se pueden obtener observaciones muy importantes a partir de este estudio:

1. Contrario a la tradición de PSO, las topologías que mostraron el peor comportamiento (cuando la versión básica de PSO fue utilizada) fueron las topologías: completamente conectada y en estrella.
2. Adicionalmente, las topologías completamente conectada y en estrella, mostraron un decremento en su desempeño con referencia a las demás topologías probadas.

3. Las topologías en malla y toroidal presentaran los mejores resultados. Aunque la topología en anillo también mostró resultados favorables.

En general, podemos decir que el desempeño de PSO esta fuertemente ligado a la topología de interconexión que utilice.

### 5.2.3 Análisis estadístico entre topologías

Aunque las gráficas de caja nos permiten visualizar gráficamente el efecto de las topologías en el desempeño de los algoritmos de PSO y por medio de ellas concluir visualmente qué topología-algoritmo se desempeño mejor, es importante llevar a cabo una prueba estadística que nos permita realizar conclusiones más objetivas de los resultados.

De tal manera, e posible llevar a cabo un análisis estadístico de los resultados por medio de la prueba estadística  $t$  que permita determinar si: *la diferencia promedio entre dos topologías es significativa o no*. **La prueba estadística  $t$**  se utiliza cuando se desconoce la varianza en las muestras pero sus promedios son distribuciones normales.

Para llevar a cabo cualquier prueba estadística, es necesario en primera instancia plantear una hipótesis estadística. Una **hipótesis estadística** es una suposición acerca de la probabilidad de distribución de los parámetros, la cual refleja una conjetura acerca de la situación del problema.

En este trabajo se desea conocer si: *¿la topología de interconexión afecta el desempeño del algoritmo de PSO?*, de acuerdo a los resultados obtenidos anteriormente, es evidente que existe una diferencia en el desempeño de PSO cuando se utilizan topologías de interconexión diferentes. Sin embargo deseamos conocer si esta diferencia es estadísticamente significativa. Para tal efecto se formula una hipótesis nula, como:

*No existe diferencia significativa en la respuesta promedio al resolver funciones utilizando una topología,  $t_i$  y funciones que se resuelven con otra topología,  $t_j$ .*

Y una hipótesis alternativa como:

*Existe diferencia significativa en la respuesta promedio.*

Las cuales son llamadas hipótesis  $H_0$  e hipótesis  $H_1$ , respectivamente. Formalmente estas hipótesis pueden reescribirse como:

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_2 \neq \mu_1$$

donde  $\mu_1$  y  $\mu_2$  son la respuesta promedio de cada una de las topologías  $t_1$  y  $t_2$ , respectivamente. Por lo tanto se rechazará  $H_0$  si  $H_1$  es verdadera, aunque a veces es posible rechazar  $H_0$  cuando ésta es verdadera, lo cual no es alarmante pues estos errores podrían aparecer cuando se utilizan este tipo de pruebas estadísticas.

Por lo tanto, si la hipótesis nula se rechaza cuando ésta es verdadera se dice que ha ocurrido un error de *tipo I*, por otro lado cuando la hipótesis nula no se rechaza cuando ésta es falsa, se dice que ha ocurrido un error de *tipo II*. La probabilidad de que ocurra alguno de estos dos errores está dada por:

$$\alpha = P(\text{error de tipo I}) = P(\text{rechaza } H_0 | H_0 \text{ es verdadera})$$

$$\beta = P(\text{error de tipo II}) = P(\text{no rechaza } H_0 | H_0 \text{ es falsa})$$

El procedimiento general en una prueba de hipótesis, es especificar un valor de  $\alpha$  también llamado **nivel de significancia de la prueba**, de tal manera que se minimice la probabilidad de error de *tipo II* llamado  $\beta$ .

Teniendo en cuenta nuestra hipótesis nula y alternativa el siguiente paso es efectuar una prueba estadística que nos permita rechazar o aceptar la hipótesis. La prueba t para dos muestras, nos permitirá efectuar dicho examen, la cual se obtiene como sigue.

Si se desconoce la varianza de las muestras pero se suponen idénticas, la prueba apropiada para comparar ambos promedios en un diseño completamente aleatorio sería:

$$t_0 = \frac{\bar{y}_1 - \bar{y}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \quad (5.1)$$

donde  $\bar{y}_1$  y  $\bar{y}_2$  son el promedio de las muestras,  $n_1$  y  $n_2$  corresponden al tamaño de cada muestra,  $S_p^2$  es un término, que estima la varianza de las muestras  $\sigma_1^2 = \sigma_2^2 = \sigma^2$ , obtenida de la siguiente forma:

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2} \quad (5.2)$$

donde  $S_1^2$  y  $S_2^2$  son la varianza de las muestras individuales. Para determinar el rechazo o aceptación de  $H_0$  se puede comparar  $t_0$  con la distribución  $t$  con  $n_1 + n_2 - 2$  grados de libertad. Si  $|t_0| > t_{(\alpha/2, n_1+n_2-2)}$ , donde  $t_{(\alpha/2, n_1+n_2-2)}$  es el porcentaje superior  $\alpha/2$  de la distribución  $t$  con  $n_1 + n_2 - 2$  grados de libertad, se podría rechazar  $H_0$  y concluir que el promedio del desempeño de las topologías difiere significativamente. A esta prueba se le conoce usualmente como **prueba  $t$  de dos muestras** resumida en la tabla 5.1. Los valores para la distribución  $t$  (con determinados grados de libertad), se pueden obtener de las tablas del libro de Montgomery [35].

También se puede obtener la prueba  $t$  si se supone que las varianzas de las muestras son diferentes así como el criterio de rechazo de la hipótesis, para ello se utilizan las fórmulas mostradas en la tabla 5.2.

Si $\sigma_1^2 = \sigma_2^2$		
Hipótesis	Prueba estadística	Criterio de rechazo
$H_0 : \mu_1 = \mu_2$	$t_0 = \frac{\bar{y}_1 - \bar{y}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$	$ t_0  > t_{\alpha/2, v}$
$H_1 : \mu_1 \neq \mu_2$	$v = n_1 + n_2 - 2$	

Tabla 5.1: Prueba de los promedios de distribuciones normales con varianza desconocida, pero supuesta igual.

Alternativamente se puede usar el enfoque del valor  $P$  para probar la hipótesis, en el cual se asigna una probabilidad a  $\alpha$ , digamos  $\alpha = 0.05$ , el cual sirve como un nivel de significancia para aceptar o rechazar la hipótesis nula. Entonces el valor de  $P$  es el nivel más pequeño de importancia que puede dar lugar al rechazo de la hipótesis nula,  $H_0$ .

Aunque no es fácil computar el nivel exacto de  $P^1$ , es costumbre decir que los datos son signifi-

<sup>1</sup>Algunos programas de computadora lanzan un valor por defecto cuando no es posible calcular el valor de  $P$ .

Si $\sigma_1^2 \neq \sigma_2^2$		
Hipótesis	Prueba estadística	Criterio de rechazo
$H_0 : \mu_1 = \mu_2$	$t_0 = \frac{\bar{y}_1 - \bar{y}_2}{S_p \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$	$t_0 < -t_{\alpha, v}$
$H_1 : \mu_1 < \mu_2$		
$H_0 : \mu_1 = \mu_2$		
$H_1 : \mu_1 > \mu_2$	$v = \frac{\left(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}\right)^2}{\frac{(S_1^2/n_1)^2}{n_1-1} + \frac{(S_2^2/n_2)^2}{n_2-1}}$	$t_0 > t_{\alpha, v}$

Tabla 5.2: Prueba de los promedios de distribuciones normales con varianza desconocida, pero su-  
puesta diferente.

cantes cuando la hipótesis  $H_0$  ha sido rechazada por la prueba  $t$ . Aún asó, suele pensarse en el valor de  $P$  como el valor más pequeño de  $\alpha$  en el cual los datos aún son significantes, entonces se puede determinar la significancia de los datos sin imponer un valor de significancia preseleccionado como lo hace la prueba  $t$ . Entonces, es posible rechazar la hipótesis nula si se cumple  $\alpha \geq P$ . Sin embargo, se considera que utilizar el valor de  $\alpha$  como nivel de significancia de los datos es un tanto subjetivo, una discusión detallada puede encontrarse en [4].

### 5.2.4 Prueba estadística $t$

Siguiendo con la hipótesis nula  $H_0$  y la hipótesis alternativa  $H_1$  planteadas anteriormente, se utilizará la prueba estadística  $t$  para aceptar o rechazar la hipótesis nula: *No existe diferencia significativa en la respuesta promedio al resolver funciones utilizando una topología,  $t_i$  y funciones que se resuelven con otra topología,  $t_j$ .*

Además de obtener la significancia de las topologías cuando se comparan unas con otras, esta prueba también nos permitirá detectar la topología o las topologías que mejoren el desempeño promedio en cada una de las versiones del algoritmo de PSO presentadas en el experimento 1 de la sección 5.2.

Suponiendo que se desconoce la varianza en las muestras, para llevar a cabo la prueba estadística  $t$  se necesita:

- **Determinar si las varianzas de las muestras son iguales o diferentes**, para ello se emplea la prueba estadística  $F$ .

La prueba estadística  $F$  verifica la hipótesis de que las desviaciones estándar de dos poblaciones normalmente distribuidas son iguales. El valor para la distribución  $F$  utilizado para probar la hipótesis con un valor de  $\alpha = 0.05$  y  $u = 278$  y  $v = 278$  grados de libertad es  $F = 1.0$ .

Si el valor estadístico  $F_0$  obtenido al comparar dos muestras, es mayor o igual al valor reportado en las tablas de distribución para el estadístico  $F = 1.0$ , entonces se dice que las varianzas son diferentes y se rechaza la hipótesis nula. Como una forma de comprobar el resultado arrojado por la prueba  $F$  se procede a comprobar el valor obtenido de  $P$  con el de  $\alpha$ , si  $\alpha > P$  se confirma que las varianzas en las muestras son diferentes rechazando definitivamente  $H_0$ , en caso contrario se dice que las varianzas son iguales.

- **Calcular el error estándar de la diferencia entre dos medias**, también denominado **margen de error** de la prueba  $t_0$  y aplicar la prueba estadística  $t$  dependiendo de la igualdad o desigualdad de las varianzas. En este paso se compara el valor obtenido de  $t_0$  con los valores extraídos de las tablas de valores para la distribución  $t$  con más de 120 grados de libertad y con un valor de  $\alpha = 0.05$ .
  - Si las varianzas en las muestras son diferentes el valor del estadístico  $t = 1.645$ .
  - Si las varianzas en las muestras son iguales el valor del estadístico  $t = 2.807$  con  $\alpha/2 = 0.0025$ .

Utilizando los criterios de rechazo para los valores  $t_0$  (mostrados en la tabla 5.2) se determina si se acepta o se rechaza la hipótesis nula.

Alternativamente, se puede utilizar el valor de  $P$  obtenido de la prueba  $t$ , con un intervalo de confianza del 95 % y  $\alpha = 0.05$ . Si el valor de  $\alpha > P$  se rechaza la hipótesis  $H_0$ , sin embargo no se ha utilizado en esta prueba.

Los valores para la prueba estadística  $F$  y la prueba estadística  $t$  se obtuvieron de las tablas de distribución proporcionadas en [35].

### 5.2.5 Resultados del análisis estadístico

Las tablas 5.3 a 5.6 muestran los resultados de la prueba estadística  $t$  aplicada en los algoritmos: PSO, PSobb, PSobbExp y PSobbDe, para determinar si la diferencia de los promedios entre dos topologías,  $(\mu_i - \mu_j)$  es estadísticamente significativa o no

En este experimento se dice que una **muestra**,  $x_i$ , es el valor obtenido al ejecutar una sola vez un algoritmo de PSO con una topología en específico en una de las nueve funciones de prueba. El **valor promedio del desempeño de una topología** es el promedio de 279 muestras tomadas de las nueve funciones de prueba (v.g., 31 ejecuciones en cada una de las funciones), con la topología en turno.

En las tablas el valor numérico que aparece de bajo de cada etiqueta **topología**, es el valor promedio del desempeño de la topología indicada en la etiqueta.

Sin embargo, es importante mencionar que para llevar a cabo este experimento fue necesario normalizar cada una de las muestras  $x_i$  de la siguiente forma:

$$x^* = \frac{x - \min_j}{\max_j - \min_j}$$

donde  $\max_i$  y  $\min_j$  con  $j = 1, \dots, 9$  representan los valores máximos y mínimos obtenidos al ejecutar cada una de las funciones objetivo  $j$ , en las ocho topologías (cada una de las combinaciones entre topología y función  $j$  fue ejecutada 31 veces).

Siguiendo con las tablas de resultados, si comparamos una topología tomada de las filas,  $t_i$ , digamos **completa**, con una topología tomada de las columnas,  $t_j$ , digamos **estrella**, podemos observar que existe una clara diferencia en sus promedios:  $\bar{t}_i = 0.3325060$  y  $\bar{t}_j = 0.3703755$  y a partir de las gráficas de cajas mostradas en las figuras 5.1 a la 5.3 se puede determinar la topología que mejora el desempeño de PSO (tomando como base de comparación la topología que logre minimizar más rápidamente la función objetivo). Sin embargo es una forma un tanto subjetiva de afirmar la topología que mejor se desempeña.

Las tablas de resultados de la prueba  $t$  nos permiten concluir de manera objetiva, basada en ex-

perimentación y no simplemente en observaciones, la topología o topologías que mejoran desempeño de PSO. Tomemos como ejemplo la tabla 5.3 la cual corresponde al desempeño alcanzado por el algoritmo básico de PSO al ser ejecutado con las ocho topologías.

Siguiendo con la metodología de la prueba estadística  $t$ , primero se determino la igualdad o diferencia en la varianza de ambas muestras con la prueba estadística  $F$ , el resultado arrojo igualdad en las varianzas (denotado como  $VI$  en las tablas). Posteriormente, se aplico la prueba  $t$  utilizando el criterio de rechazo de la hipótesis nula cuando las varianzas son iguales  $|t_0| > t_{\alpha/2,v}$ , entonces si  $t_0 = -2.5921$  su valor absoluto es  $|t_0| = 2.5921$  y si el valor de  $t = 2.807$ , la prueba  $|t_0| > t_{\alpha/2,v}$  es incorrecta ya que  $2.5921 < 2.807$  lo que significa que la hipótesis  $H_0$  se acepta y por lo tanto no existe diferencia significativa en la respuesta promedio al resolver funciones utilizando una topología completamente conectada o utilizando una topología en estrella.

Es posible utilizar el valor de  $P$  para probar la misma hipótesis. La regla nos indica que debemos rechazar a  $H_0$  si  $\alpha > P$ , para este ejemplo  $P = 0.009663$  y  $\alpha = 0.05$  entonces  $0.05 > 0.009663$ , por lo tanto la hipótesis  $H_0$  sería rechazada, contrario a lo anteriormente dicho. Para fines de este experimento no se ha utilizado el valor de  $P$  como prueba de significancia, sólo el valor del estadístico  $t$ .

Entonces, continuando con el valor de  $t_0$  para aceptar o rechazar la hipótesis nula, es posible seguir con el mismo análisis en cada una de las tablas correspondientes a los algoritmo de PSO. Sin embargo, para facilitar la visualización de aquellos resultados significativos, se ha colocado un símbolo  $\surd$  en las casillas de las tablas, si los promedios de las topologías a los que hace referencia la fila,  $i$ , y la columna,  $j$ , son significativos. Contrario al símbolo  $\surd$ , aquellas casillas con el símbolo  $\times$  serán las topologías que no presentaron significancia alguna al compararse con otra.

Es posible determinar fácilmente aquella topología que mejora el desempeño del algoritmo PSO por medio de la prueba  $t$ , de la siguiente forma:

- Tomar aquella columna que presente el menor promedio obtenido (debido a que las funciones presentadas en este trabajo son problemas de minimización).
- Posteriormente descender en esa columna hasta un nivel abajo de la diagonal principal y recorrer

la fila hacia la derecha, todas aquellas topologías que presentaron significancia en sus promedios (aquellas casillas que posean el símbolo  $\checkmark$ ) serán las topologías que mejor se desempeñaron.

### 5.2.6 Conclusiones del análisis estadístico

Con el procedimiento anteriormente descrito, es fácil identificar que:

- Para el algoritmo básico de PSO (mostrado en la tabla 5.3) la topología que presentó mejor resultado fue la toroidal.
- Para el algoritmo PSOBB (mostrado en la tabla 5.4) las topologías que presentaron mejores resultados fueron: estrella, completamente conectada y el algoritmo de agrupamiento jerárquico.
- Para el algoritmo PSOBBEXP (mostrado en la tabla 5.5), las topologías que presentaron mejores resultados fueron: completamente conectada, estrella y el algoritmo de agrupamiento jerárquico.
- Para el algoritmo PSOBBDE (mostrado en la tabla 5.6) es difícil determinar la topología que mejor se desempeño pues todas presentaron significancia en sus promedios, sin embargo es posible determinar la topología que peor se comportó si aplicamos el anterior procedimiento con una ligera modificación, escoger aquellas topologías que no presentaron significancia en su promedios (las casillas que posean el símbolo  $\times$ ). Siguiendo este procedimiento las topologías que peor se comportaron fueron: malla y árbol.

Topología	Malla 0.1702423	Toroidal 0.1597458	Completa 0.3325060	Estrella 0.3703755	Árbol 0.2439002	<i>k-means</i> 0.2338646	Jerárquica 0.3715466
Anillo 0.2206369	$t_0 = 3.6542$ $p = 0.000271$ VD x	$t_0 = 4.5191$ $p = 6.956^{-06}$ VD x	$t_0 = -7.4339$ $p = 2.097e^{-13}$ VI x	$t_0 = -9.8472$ $p < 2.2e^{-16}$ VI x	$t_0 = -1.5859$ $p = 0.1131$ VD ✓	$t_0 = -0.9661$ $p = 0.3342$ VD ✓	$t_0 = -9.8867$ $p < 2.2e^{-16}$ VI x
Malla 0.1702423		$t_0 = 0.9296$ $p = 0.3528$ VI ✓	$t_0 = -12.3584$ $p < 2.2e^{-16}$ VI x	$t_0 = -15.0359$ $p < 2.2e^{-16}$ VI x	$t_0 = -5.8029$ $p = 8.488e^{-09}$ VI x	$t_0 = -5.5082$ $p = 4.501e^{-08}$ VI x	$t_0 = -15.0495$ $p < 2.2e^{-16}$ VI x
Toroidal 0.1597458			$t_0 = -13.5004$ $p < 2.2e^{-16}$ VI x	$t_0 = -16.2249$ $p < 2.2e^{-16}$ VI x	$t_0 = -6.815$ $p = 1.540e^{-11}$ VI x	$t_0 = -6.6353$ $p = 5.045e^{-11}$ VI x	$t_0 = -16.2308$ $p < 2.2e^{-16}$ VI x
Completa 0.3325060				$t_0 = -2.5921$ $p = 0.009663$ VI ✓	$t_0 = 6.3067$ $p = 4.101^{-10}$ VI x	$t_0 = 7.5726$ $p = 7.952e^{-14}$ VD x	$t_0 = -2.6614$ $p = 0.007894$ VI ✓
Estrella 0.3703755					$t_0 = 8.8953$ $p < 2.2e^{-16}$ VI x	$t_0 = 10.3354$ $p < 2.2e^{-16}$ VD x	$t_0 = -0.079$ $p = 0.937$ VI ✓
Árbol 0.2439002						$t_0 = 0.7974$ $p = 0.4254$ VD ✓	$t_0 = -8.9389$ $p < 2.2e^{-16}$ VI x
<i>k-means</i> 0.2338646							$t_0 = -10.372$ $p < 2.2e^{-16}$ VI x

Tabla 5.3: Análisis estadístico de topologías para PSO.

Topología	Malla 0.3184229	Toroidal 0.2891353	Completa 0.1127955	Estrella 0.1200146	Árbol 0.5037570	<i>k-means</i> 0.2267049	Jerárquica 0.1265078
Anillo 0.5946722	$t_0 = 16.0759$ $p < 2.2e^{-16}$ VI x	$t_0 = 18.2095$ $p < 2.2e^{-16}$ VD x	$t_0 = 29.166$ $p < 2.2e^{-16}$ VD x	$t_0 = 28.6558$ $p < 2.2e^{-16}$ VD x	$t_0 = 5.0141$ $p = 7.176e^{-07}$ VI x	$t_0 = 20.7634$ $p < 2.2e^{-16}$ VI x	$t_0 = 28.1309$ $p < 2.2e^{-16}$ VD x
Malla 0.3184229		$t_0 = 1.8167$ $p = 0.0698$ VI ✓	$t_0 = 12.9702$ $p < 2.2e^{-16}$ VI x	$t_0 = 12.4802$ $p < 2.2e^{-16}$ VI x	$t_0 = -10.5752$ $p < 2.2e^{-16}$ VI x	$t_0 = 5.3634$ $p = 1.2e^{-07}$ VI x	$t_0 = 12.0102$ $p < 2.2e^{-16}$ VI x
Toroidal 2891353			$t_0 = 11.4404$ $p < 2.2e^{-16}$ VI x	$t_0 = 10.9399$ $p < 2.2e^{-16}$ VI x	$t_0 = -12.5302$ $p < 2.2e^{-16}$ VI x	$t_0 = 3.7398$ $p = 0.0002033$ VI x	$t_0 = 10.4631$ $p < 2.2e^{-16}$ VI x
Completa 0.11279955				$t_0 = -0.4756$ $p = 0.6346$ VI ✓	$t_0 = -23.1659$ $p < 2.2e^{-16}$ VI x	$t_0 = -6.9308$ $p = 1.163e^{-11}$ VI x	$t_0 = -0.8983$ $p = 0.3694$ VI ✓
Estrella 0.1200146					$t_0 = -22.6826$ $p < 2.2e^{-16}$ VI x	$t_0 = -6.4748$ $p = 2.089e^{-10}$ VI x	$t_0 = -0.4241$ $p = 0.6717$ VI ✓
Árbol 0.5037570						$t_0 = 15.3466$ $p < 2.2e^{-16}$ VI x	$t_0 = 22.1982$ $p < 2.2e^{-16}$ VD x
<i>k-means</i> 0.2267049							$t_0 = 6.0519$ $p = 2.662e^{-09}$ VD x

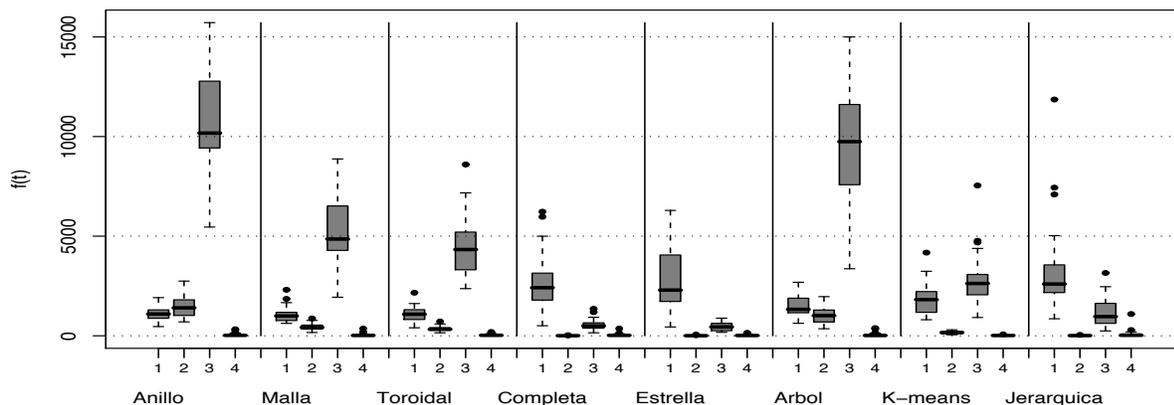
Tabla 5.4: Análisis estadístico de topologías para PSOBB.

Topología	Malla 0.4199163	Toroidal 0.3833513	Completa 0.1480655	Estrella 0.1425615	Árbol 0.5565467	<i>k-means</i> 0.3196852	Jerárquica 0.1861203
Anillo 0.6176645	$t_0 = 12.11$ $p < 2.2e^{-16}$ VI x	$t_0 = 14.3236$ $p < 2.2e^{-16}$ VI x	$t_0 = 29.698$ $p < 2.2e^{-16}$ VI x	$t_0 = 30.0215$ $p < 2.2e^{-16}$ VI x	$t_0 = 3.5531$ $p = 0.0004131$ VI x	$t_0 = 17.3534$ $p < 2.2e^{-16}$ VI x	$t_0 = 25.6694$ $p < 2.2e^{-16}$ VI x
Malla 0.4199163		$t_0 = 2.1991$ $p = 0.02828$ VI ✓	$t_0 = 16.8955$ $p < 2.2e^{-16}$ VI x	$t_0 = 17.224$ $p < 2.2e^{-16}$ VI x	$t_0 = -7.8267$ $p = 2.542e^{-14}$ VI x	$t_0 = 5.7514$ $p = 1.46e^{-08}$ VI x	$t_0 = 13.6939$ $p < 2.2e^{-16}$ VI x
Toroidal 0.3833513			$t_0 = 14.5962$ $p < 2.2e^{-16}$ VI x	$t_0 = 14.9259$ $p < 2.2e^{-16}$ VI x	$t_0 = -9.9059$ $p < 2.2e^{-16}$ VI x	$t_0 = 3.6475$ $p = 0.0002897$ VI x	$t_0 = 11.5334$ $p < 2.2e^{-16}$ VI x
Completa 0.1480655				$t_0 = 0.3533$ $p = 0.724$ VI ✓	$t_0 = -24.0646$ $p < 2.2e^{-16}$ VI x	$t_0 = -10.1287$ $p < 2.2e^{-16}$ VI x	$t_0 = -2.2953$ $p = 0.02209$ VI ✓
Estrella 0.1425615					$t_0 = -24.3716$ $p < 2.2e^{-16}$ VI x	$t_0 = -10.4461$ $p < 2.2e^{-16}$ VI x	$t_0 = -2.6253$ $p = 0.008894$ VI ✓
Árbol 0.5565467						$t_0 = 12.981$ $p < 2.2e^{-16}$ VI x	$t_0 = 20.6841$ $p < 2.2e^{-16}$ VI x
<i>k-means</i> 0.3196852							$t_0 = 7.4701$ $p = 3.12e^{-13}$ VI x

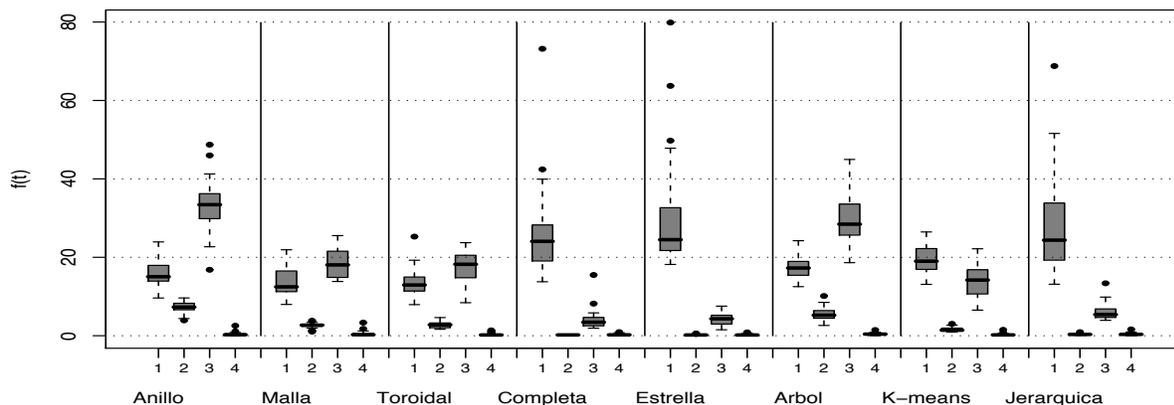
Tabla 5.5: Análisis estadístico de topologías para PSOBEXP.

Topología	Malla 0.2352985	Toroidal 0.2155060	Completa 0.2069735	Estrella 0.1998743	Árbol 0.2411042	<i>k-means</i> 0.1992788	Jerárquica 0.2226245
Anillo 0.2098741	$t_0 = -1.4338$ $p = 0.1522$ VI ✓	$t_0 = -0.3306$ $p = 0.741$ VI ✓	$t_0 = 0.1682$ $p = 0.8665$ VI ✓	$t_0 = 0.6051$ $p = 0.5454$ VI ✓	$t_0 = -1.7791$ $p = 0.07576$ VI ✓	$t_0 = 0.6435$ $p = 0.5201$ VI ✓	$t_0 = -0.7375$ $p = 0.4611$ VI ✓
Malla 0.2352985		$t_0 = 1.0896$ $p = 0.2764$ VI ✓	$t_0 = 1.5425$ $p = 0.1235$ VI ✓	$t_0 = 2.0024$ $p = 0.04574$ VD x	$t_0 = -0.3112$ $p = 0.7557$ VI ✓	$t_0 = 2.0427$ $p = 0.04156$ VD x	$t_0 = 0.5886$ $p = 0.4913$ VI ✓
Toroidal 0.2155060			$t_0 = 0.4824$ $p = 0.6297$ VI ✓	$t_0 = 0.92$ $p = 0.3580$ VI ✓	$t_0 = -1.4228$ $p = 0.1553$ VI ✓	$t_0 = 0.9585$ $p = 0.3382$ VI ✓	$t_0 = -0.4015$ $p = 0.6882$ VI ✓
Completa 0.2069735				$t_0 = 0.4127$ $p = 0.68$ VI ✓	$t_0 = -1.8764$ $p = 0.06113$ VI ✓	$t_0 = 0.4489$ $p = 0.6537$ VI ✓	$t_0 = 0. - 8727$ $p = 0.3832$ VI ✓
Estrella 0.1998743					$t_0 = -2.3544$ $p = 0.0189$ VI ✓	$t_0 = 0.0363$ $p = 0.971$ VI ✓	$t_0 = -1.3192$ $p = 0.1877$ VI ✓
Árbol 0.2411042						$t_0 = 2.3964$ $p = 0.01689$ VD x	$t_0 = 1.0136$ $p = 0.3112$ VI ✓
<i>k-means</i> 0.1992788							$t_0 = -1.3584$ $p = 0.1749$ VI ✓

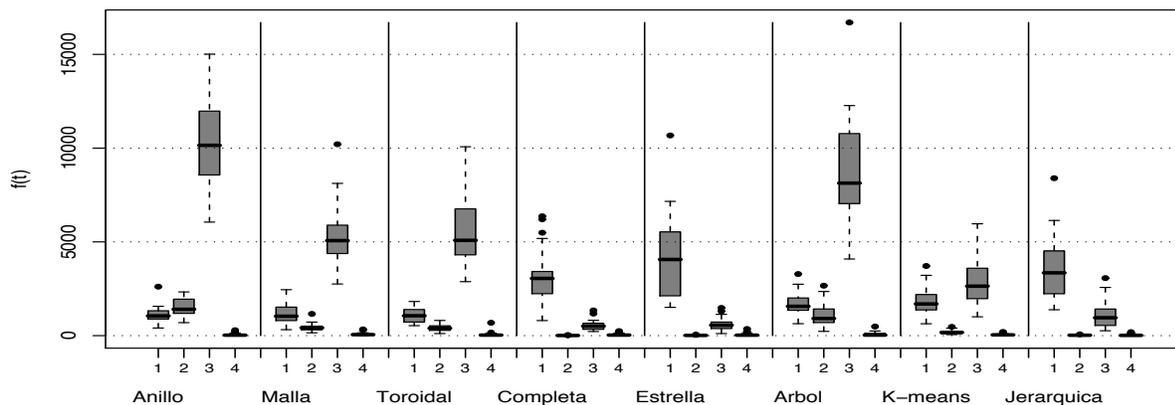
Tabla 5.6: Análisis estadístico de topologías para PSOBDE.



(a) Función *sphere*



(b) Problema de *Schwefel*



(c) Función *step*

Figura 5.1: Comparación entre topologías de interconexión implementadas en: 1) PSO, 2) PSOBB, 3) PSOBBEXP, 4) PSOBBDE.

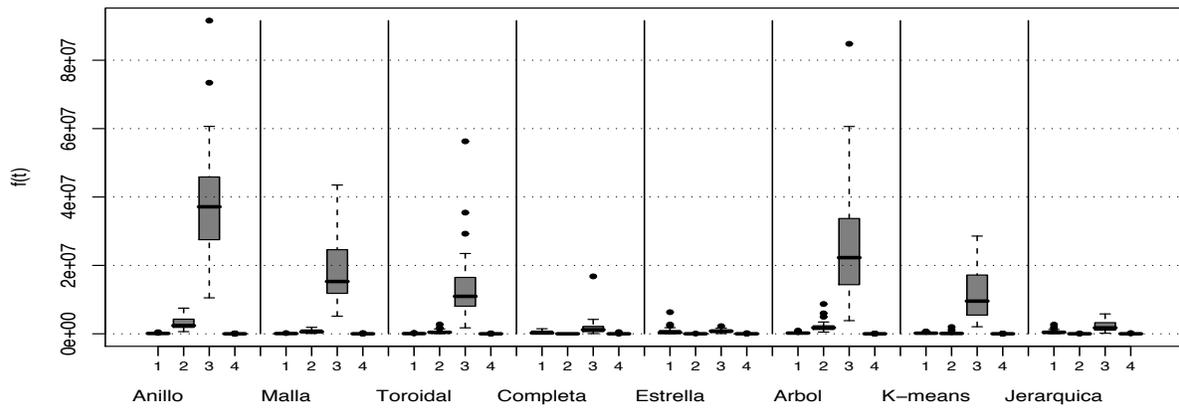
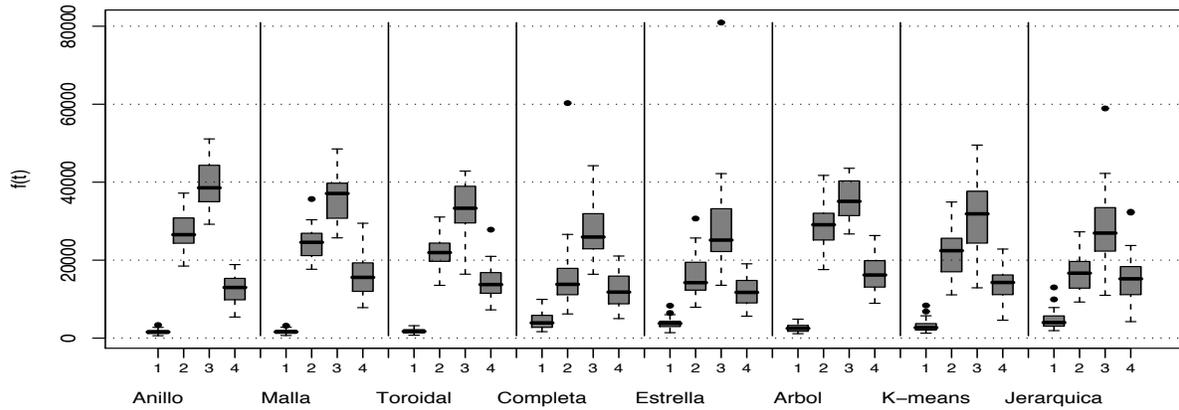
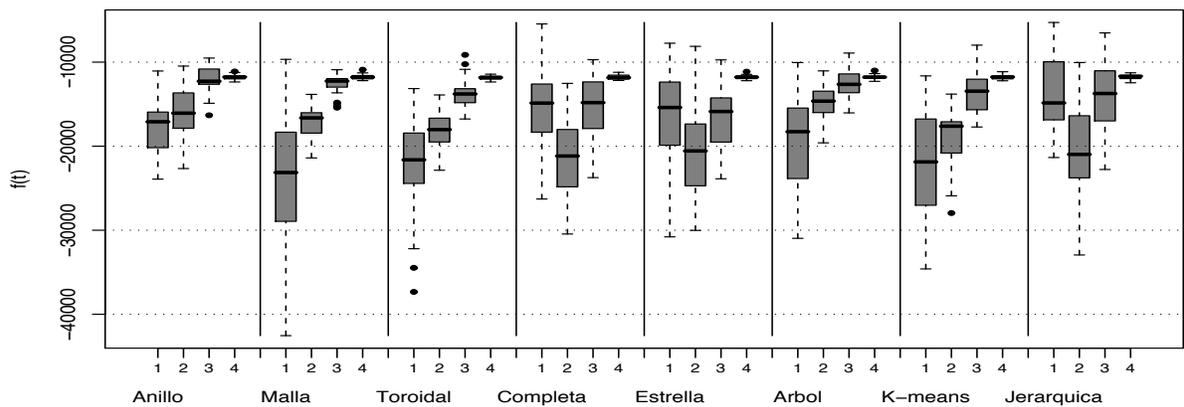
(a) Función de *Rosenbrock*(b) Función *rotated hyper-ellipsoid*(c) Problema generalizado de *Schwefel 2.26*

Figura 5.2: Comparación entre topologías de interconexión implementadas en: 1) PSO, 2) PSOBB, 3) PSOBBEXP, 4) PSOBBDE.

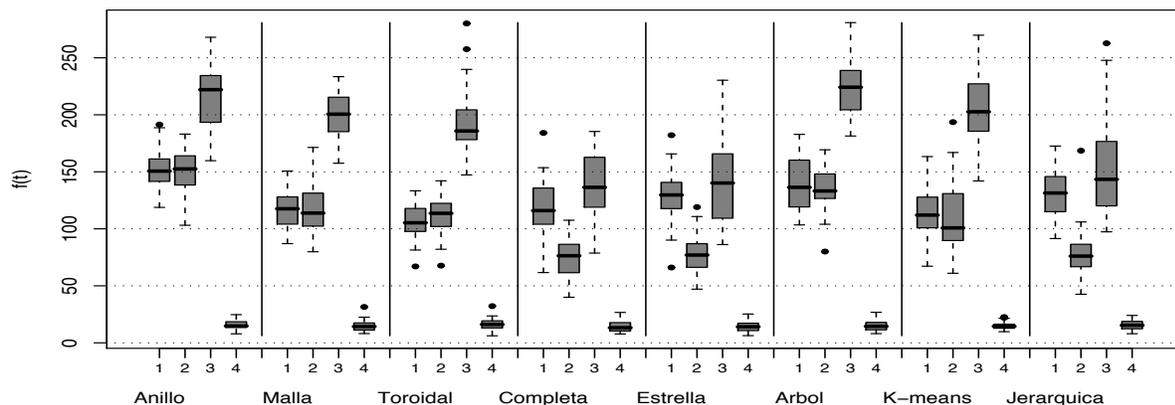
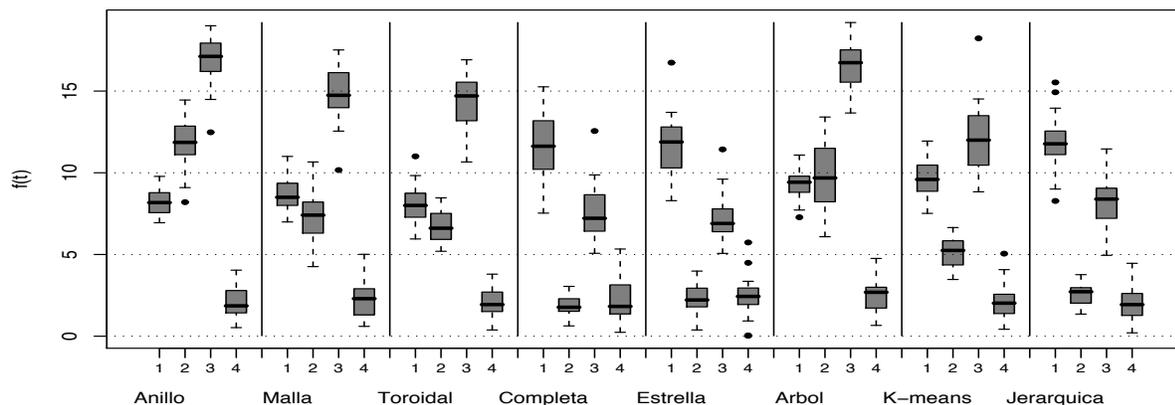
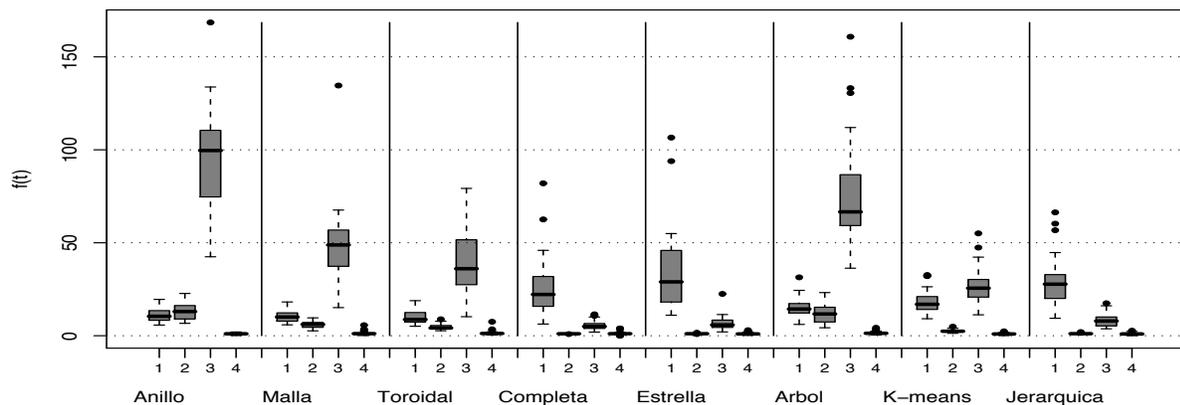
(a) Función de *Rastrigin*(b) Función de *Ackley*(c) Función de *Griewank*

Figura 5.3: Comparación entre topologías de interconexión implementadas en: 1) PSO, 2) PSOB, 3) PSOBEXP, 4) PSOBDE.

## 5.3 Experimento 2

Después de llevar a cabo el primer experimento, se puede observar que la versión básica de PSO presenta un buen comportamiento cuando se usan las topologías toroidal y malla, seguida en algunas ocasiones por la topología en anillo; y presenta su peor comportamiento cuando se utilizan las topologías completamente conectada y en estrella, contrario a lo que tradicionalmente se ha venido usando en PSO.

A partir de esto, se puede concluir que PSO se desempeña correctamente (en promedio), cuando se presentan topologías que superponen los vecinos de una partícula con otra.

Sin embargo, a diferencia de los otros métodos (PSOBB, PSOBBEXP, y PSOBBDE), la convergencia y desempeño de PSO no depende sólo de la topología de interconexión implementada, sino que también está fuertemente ligada a los valores asignados a la inercia,  $w$  y a los coeficientes de aceleración,  $c_1$  y  $c_2$ . Pensamos que la sintonización adecuada de estos parámetros permitirá que el algoritmo básico mejore su desempeño, conjuntamente con las topologías de interconexión usadas en el primer experimento.

El objetivo general de este experimento es determinar si por medio del diseño de experimentos y los métodos estadísticos computacionales se puede encontrar un conjunto de parámetros ( $w$ ,  $c_1$ ,  $c_2$  y la topología de interconexión) que mejore el desempeño promedio de PSO.

La metodología propuesta para la calibración de parámetros del algoritmo PSO iniciará con un diseño factorial completo (descrito en la sección 4.2). En este experimento, los diseños factoriales funcionarán como un método de búsqueda a grano grueso en el rango de valores establecidos para cada parámetro (definidos en la tabla 5.7).

Debe notarse que el dominio de valores que puede tomar cada uno de los parámetros de PSO está en el conjunto de  $\mathbb{R}$ , de tal manera, implementar una combinación exhaustiva de dichos valores es prohibitivo (*v.g.*, un diseño factorial completo que involucre un extenso número de niveles), debido a esto se establecerá un número determinado de niveles para cada factor como se muestra en la tabla 5.7. A partir de la tabla 5.7, la columna **Inicia** muestra límite inferior de cada parámetro, la columna **Termina** muestra el límite superior de cada parámetro, obteniendo así el rango de parámetros para

Parámetros	Inicia	Termina	Incrementos	Niveles	Total niveles
$c_1$	1.0	3.0	0.2	1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0	11
$c_2$	1.0	3.0	0.2	1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0	11
$w$	0.0	1.0	0.1	0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0	11

Tabla 5.7: Rangos y niveles para cada factor en el experimento factorial completo.

cada factor. La columna **Incrementos**, es el tamaño de paso establecido para crear un nuevo nivel. La columna **Niveles**, muestra los niveles obtenidos por cada factor.

La combinación de los niveles permitirá conocer el número de tratamientos, *v.g.*,  $11 \times 11 \times 11 = 1,331$  tratamientos. Cada tratamiento será ejecutado 30 veces independientes, con un tamaño de cúmulo de  $s = 50$  y 200 iteraciones del algoritmo. Cada uno de los tratamientos será ejecutado en las nueve funciones de prueba (las cuales utilizan 30 variables) y en cada una de las topologías de interconexión.

Para las topologías en malla y toroidal, la distribución de las partículas será de  $10 \times 5$ . Para los algoritmos de agrupamiento *k-means* y jerárquico, se establecerán 4 grupos.

Al finalizar este experimento, se espera obtener un conjunto de parámetros y la topología que mejor se desempeñe en cada una de las funciones de prueba

### 5.3.1 Resultados

Los resultados obtenidos del experimento factorial completo se muestran en las tablas 5.8, 5.9 y 5.10. Las tablas muestran los mejores valores obtenidos para los parámetros de PSO, para cada una de las funciones de prueba. Donde la columna **Min** representa la menor aptitud conseguida por cada topología, la columna **Valores Min** representa los tres parámetros de PSO:  $w$ ,  $c_1$  y  $c_2$  con los cuales se consiguió la menor aptitud. También se muestra la aptitud máxima conseguida (expresada en la tabla como **Max**), así como los valores de los parámetros correspondientes para conseguir dicha aptitud. Se muestra también el **Promedio** de las observaciones, la **Mediana** y la desviación estándar (referida en la tabla como **StDev**).

Es claro ver que la topología toroidal fue la que presentó el mejor desempeño en la mayoría de las funciones de prueba. Esto concuerda con los resultados obtenidos previamente en el experimento

5.2, donde las topologías que mostraron los mejores resultados fueron la topología toroidal y malla.

Para la función de *Rastrigin* y el problema generalizado de *Schwefel* 2.26, el algoritmo de agrupamiento *k-means* fue el que presentó mejor desempeño (mostrados en las tablas 5.9(c) y 5.9(b), respectivamente).

Por otro lado, el algoritmo de agrupamiento jerárquico fue el que presentó el peor comportamiento en cuatro de las nueve funciones de prueba (mostrado en las tablas 5.8(b), 5.8(d), 5.9(a) y 5.9(d)), seguido de la topología en malla en tres de las funciones de prueba (mostrado en las tablas 5.7(a), 5.9(c), 5.8(a), 5.8(c)). Para la función de *Rastrigin* y el problema generalizado de *Schwefel* 2.26 la topología en anillo fue la que peor se comportó (mostrados en las tablas 5.9(c) y 5.9(b), respectivamente).

(a) Función de *Griewank*

Topología	Mín	Valores Mín			Promedio	Mediana	StDev	Máx	Valores Máx		
		$c_1$	$c_2$	$w$					$c_1$	$c_2$	$w$
1	3.675796	1.0	2.2	0.7	14.9393173959	9.75228622897	12.7462263397	69.080658	3	2.8	1
2	2.493115	3.0	1.4	0.6	11.0320245815	8.45344091026	7.47095204099	45.002048	3	2.4	1
3	<b>1.802567</b>	<b>2.8</b>	<b>1.0</b>	<b>0.7</b>	<b>9.35038973178</b>	<b>7.29267939446</b>	<b>6.5876901806</b>	<b>51.029095</b>	<b>2.6</b>	<b>3</b>	<b>1</b>
4	2.117833	2.6	1.0	0.8	24.2064244515	16.6145419988	20.9977743838	106.539955	1.6	1.8	0.1
5	2.463366	1.8	1.4	0.9	25.0512899955	17.1454101602	21.4637490286	111.953079	1	1.6	0
6	4.414125	1.8	1.4	0.8	17.8990881886	14.5020566129	10.7517141685	66.748062	2.8	2.8	1
7	2.024352	2.6	1.6	0.6	13.9977836251	11.8005157839	9.82091828463	53.345337	2	1.8	0
8	2.379519	1.0	1.0	0.9	24.7032012885	17.6695550364	19.7628878272	101.583954	1	1.4	0

Tabla 5.8: Diseño factorial completo para las topologías: 1) Anillo, 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol y; algoritmos de agrupamiento: 7) *K-means*, 8) Jerárquico.

(a) Función *sphere*

Topología	Mín	Valores Mín			Promedio	Mediana	StDev	Máx	Valores Máx		
		$c_1$	$c_2$	$w$					$c_1$	$c_2$	$w$
1	275.785492	1.8	1.2	0.8	1561.91796887	992.657687443	1457.95318538	8273.092773	2.4	3	1
2	168.895004	1.2	2.0	0.7	1111.37790437	840.707957267	822.469350286	4828.603027	2.8	2.6	1
3	<b>114.628555</b>	<b>3.0</b>	<b>1.8</b>	<b>0.5</b>	<b>917.663034274</b>	<b>683.953269538</b>	<b>708.952335957</b>	<b>4949.989746</b>	<b>2</b>	<b>3</b>	<b>1</b>
4	159.010666	1.0	1.0	0.9	2587.20308993	1729.11707091	2334.59630614	11094.94043	2.6	1.4	0.1
5	130.17894	2.4	1.8	0.7	2658.81599027	1803.91109501	2334.92572483	13059.5625	2.4	1.8	0
6	423.104095	2.8	2.0	0.5	1863.27766061	1458.85662306	1193.00933489	7308.746094	2.8	3	1
7	135.449219	2.0	2.8	0.5	1406.36205848	1135.75700592	1032.18345398	5312.214355	1.8	1.2	0.2
8	149.269501	1.6	1.4	0.8	2640.38340089	1862.43417697	2250.25526037	13016.339844	1.8	1.2	0.1

(b) Problema de *Schwefel's*

Topología	Mín	Valores Mín			Promedio	Mediana	StDev	Máx	Valores Máx		
		$c_1$	$c_2$	$w$					$c_1$	$c_2$	$w$
1	7.311751	2.6	1.8	0.4	17.6289795109	14.3966785575	8.39435601017	47.463039	2.6	2.8	1
2	6.057532	1.8	2.0	0.7	14.5417586777	13.0862512357	5.927787076	40.462048	3	2.8	1
3	<b>5.424879</b>	<b>2.6</b>	<b>1.8</b>	<b>0.5</b>	<b>13.1560653997</b>	<b>11.9023009942</b>	<b>5.54786317764</b>	<b>40.045792</b>	<b>2.4</b>	<b>3</b>	<b>1</b>
4	6.385514	2.4	1.2	0.8	22.4057440391	20.5845691588	9.21746394952	56.850895	1	2.4	0.1
5	6.238167	2.0	2.0	0.8	22.8122846649	21.0429133571	9.24482556214	53.870342	1.8	1.4	0
6	7.822499	1.2	1.6	0.8	18.8604151773	17.0074553543	6.90829491788	47.424297	2	2.8	1
7	4.008269	2.8	1.8	0.5	16.8900971285	16.2011085732	6.21011212445	34.268444	2	1.6	0
8	7.326743	2.0	1.0	0.9	23.4798308835	21.6162465555	9.38279145643	56.219322	1.2	1.4	0.1

(c) Función *step*

Topología	Mín	Valores Mín			Promedio	Mediana	StDev	Máx	Valores Máx		
		$c_1$	$c_2$	$w$					$c_1$	$c_2$	$w$
1	280.0	1.0	1.2	0.7	1548.96393689	979.01023	1410.97934534	8795.0	2.2	3	1
2	162.0	3.0	1.2	0.6	1121.93313298	830.8640718	850.858303078	5541.0	2.8	3	1
3	<b>116.0</b>	<b>1.2</b>	<b>1.2</b>	<b>0.8</b>	<b>912.399699474</b>	<b>700.259297667</b>	<b>691.782051688</b>	<b>4672.0</b>	<b>2.6</b>	<b>3</b>	<b>0.9</b>
4	197.0	2.6	2.2	0.6	2469.01803156	1744.0319368	2102.2466721	10513.0	1	2	0.1
5	164.0	1.2	2.0	0.8	2616.57776108	1771.22496975	2280.25885925	10885.0	2	1.2	0.1
6	427.0	2.8	1.0	0.8	1854.40270473	1475.250299	1182.18584476	7215.0	2.8	2.8	1
7	119.0	2.6	3.0	0.5	1398.43200601	1155.88665687	1000.87169023	5454.0	1	2	0
8	236.0	2.0	2.0	0.8	2599.48835462	1905.18906042	2139.7665231	13271.0	1	1.2	0.1

(d) Función de *Rosenbrock*

Topología	Mín	Valores Mín			Promedio	Mediana	StDev	Máx	Valores Máx		
		$c_1$	$c_2$	$w$					$c_1$	$c_2$	$w$
1	5802.686035	2.0	1.6	0.5	297717.209066	65145.7432328	580802.791613	4321272.0	2.6	3	1
2	3113.348633	1.4	1.0	0.8	132752.002946	45959.8145728	215327.936019	2094496.5	3	3	1
3	<b>1574.668823</b>	<b>1.6</b>	<b>1.2</b>	<b>0.8</b>	<b>91533.9233418</b>	<b>30077.6602171</b>	<b>168861.704553</b>	<b>1768680.5</b>	<b>2.2</b>	<b>3</b>	<b>1</b>
4	2017.114868	1.6	1.0	0.9	688009.106106	203509.057105	1074087.71371	8270695.0	2.6	1.6	0.1
5	1742.234009	3.0	1.6	0.7	706761.963401	227409.269339	1097796.60189	7880076.5	2	1.6	0
6	8313.881836	1.0	2.8	0.7	320051.548273	137765.971972	442212.156061	3527299.75	3	3	1
7	1871.720581	2.2	2.0	0.7	203024.904532	95021.2398398	260826.562365	1958665.5	1.4	1.4	0
8	6165.892578	1.6	1.8	0.8	715087.553763	259982.142887	1089368.52908	8959198.0	3	1.6	0

Tabla 5.9: Diseño factorial completo para las topologías: 1) Anillo, 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol y; algoritmos de agrupamiento: 7) *K-means*, 8) Jerárquico.

(a) Función *rotated hyper-ellipsoid*

Topología	Mín	Valores Mín			Promedio	Mediana	StDev	Máx	Valores Máx		
		$c_1$	$c_2$	$w$					$c_1$	$c_2$	$w$
1	388.3284	3.0	1.0	0.7	3054.03404458	1590.33147047	3285.50182549	18502.880859	3	3	1
2	284.589966	1.8	1.8	0.6	2435.60386869	1565.40762128	2316.97713604	14847.439453	2.2	3	1
3	<b>262.48642</b>	<b>1.6</b>	<b>1.2</b>	<b>0.8</b>	<b>2133.24057403</b>	<b>1365.64449644</b>	<b>2032.31982777</b>	<b>12091.981445</b>	<b>2.6</b>	<b>3</b>	<b>1</b>
4	562.676941	1.8	1.2	0.8	5007.48607142	3989.68987242	3467.02852452	17852.533203	2.2	1.6	0.1
5	427.528656	2.4	2.0	0.7	4407.93784203	3750.59213293	2721.30942871	13884.950195	1	1.4	0.1
6	431.714081	1.2	2.0	0.7	3356.30757135	2294.86720375	2787.90981292	16515.953125	3	3	1
7	460.028534	1.0	1.0	0.9	3097.508378	2645.29937643	1792.00863642	9722.485352	3	2.8	1
8	574.80188	1.2	1.0	0.9	5234.36286221	4303.42043164	3449.06692918	24429.480469	2.2	2	0

(b) Problema generalizado de *Schwefel 2.26*

Topología	Mín	Valores Mín			Promedio	Mediana	StDev	Máx	Valores Máx		
		$c_1$	$c_2$	$w$					$c_1$	$c_2$	$w$
1	-81158.140625	1.2	1.4	1.0	-25547.2146633	-22103.9562597	14590.5746843	-4267.397461	1	1	0
2	-99357.921875	1.4	1.2	1.0	-29040.9096129	-24973.8326951	16993.8798631	-4260.662109	1	1	0
3	-101061.007812	1.6	1.2	1.0	-30087.69299	-25611.7490431	17824.9214901	-4626.637207	1.2	1	0.1
4	-118340.1875	1.6	1.0	1.0	-30302.1602615	-22216.0070089	22960.8735417	-4259.627441	1	1	0
5	-109803.210938	2.2	1.8	1.0	-30466.8559273	-21731.0408421	23042.6929172	-3958.572754	1	1	0
6	-90815.6875	1.4	1.4	1.0	-26383.3461789	-22684.1146353	15012.363263	-4290.693359	1	1	0
7	<b>-116431.820312</b>	<b>1.2</b>	<b>1.0</b>	<b>1.0</b>	<b>-33936.586289</b>	<b>-26296.3112341</b>	<b>25043.4550441</b>	<b>-4470.788086</b>	<b>1.4</b>	<b>1</b>	<b>0</b>
8	-119115.710938	2.6	1.0	1.0	-30029.775526	-21177.0057445	23604.0982985	-4235.068848	1.2	1	0

(c) Función de *Rastrigin*

Topología	Mín	Valores Mín			Promedio	Mediana	StDev	Máx	Valores Máx		
		$c_1$	$c_2$	$w$					$c_1$	$c_2$	$w$
1	84.663101	1.8	2.4	0.3	142.699893865	132.587161667	32.4416460272	234.852051	3	2.8	1
2	42.969467	2.2	1.4	0.6	119.639955174	108.233599371	38.9918738838	224.985016	3	2.8	1
3	46.589672	2.4	1.6	0.6	113.732891888	102.546858714	40.8253294707	229.27507	2.4	3	1
4	45.526001	1.4	1.4	0.8	119.263635415	116.644334021	37.2795845432	225.002747	1	1.2	0
5	44.133953	2.4	1.2	0.8	119.881130782	114.951466612	37.7316862538	216.057083	3	2.8	0.9
6	66.326897	2.8	1.6	0.4	137.01447112	128.129363731	36.0215831009	237.727829	2.4	2.8	0.9
7	<b>36.177319</b>	<b>2.0</b>	<b>1.2</b>	<b>0.8</b>	<b>112.571146192</b>	<b>105.565581727</b>	<b>39.2612196347</b>	<b>213.101685</b>	<b>2.8</b>	<b>2.8</b>	<b>0.9</b>
8	51.827091	2.2	1.0	0.8	123.359123636	119.370728761	36.5836713866	219.518494	1.8	1.4	0

(d) Función de *Ackley's*

Topología	Mín	Valores Mín			Promedio	Mediana	StDev	Máx	Valores Máx		
		$c_1$	$c_2$	$w$					$c_1$	$c_2$	$w$
1	5.34212	1.8	2.4	0.5	8.46275466116	7.77723779854	2.06478379746	14.636568	2	2.8	1
2	4.109392	2.8	2.0	0.5	7.73313408715	7.42267088741	1.67605840475	13.836149	2.6	2.8	1
3	<b>3.261667</b>	<b>1.4</b>	<b>1.2</b>	<b>0.8</b>	<b>7.18392646056</b>	<b>6.93756187945</b>	<b>1.64638558398</b>	<b>12.797731</b>	<b>2.6</b>	<b>3</b>	<b>1</b>
4	4.830596	2.2	1.6	0.8	9.93441998948	9.70015832349	2.41130288865	16.381605	1	1.8	0
5	4.450777	2.6	1.2	0.8	10.0227835162	9.82460677888	2.41619942539	15.708537	1	2.4	0.1
6	5.998648	2.6	2.0	0.2	9.24098712397	8.95754312854	1.66622211728	14.344028	2.8	2.8	1
7	4.380044	2.6	2.0	0.6	8.42242290008	8.50660179871	1.93771713912	13.475773	1.2	1.6	0
8	5.057732	2.0	2.4	0.7	10.1779903013	9.93987598531	2.34216989577	15.92556	1	1.4	0

Tabla 5.10: Diseño factorial completo para las topologías: 1) Anillo, 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol y; algoritmos de agrupamiento: 7) *K-means*, 8) Jerárquico.

Los resultados arrojados por el diseño factorial completo muestran que existen diferentes conjuntos de parámetros en los cuales cada una de las topologías mejoran el desempeño de PSO.

Es importante notar que la topología que presentó mejores resultados fue la **toroidal**. Un aspecto curioso en esta topología es que presentó un conjunto de parámetros muy similar en la mayoría de las funciones de prueba, *v.g.*, se puede determinar intuitivamente el rango de valores en los cuales presenta buenos resultados, estos valores son:

$$c_1 = [2.6, 2.8]$$

$$c_2 = [2.8, 3]$$

$$w = [0.9, 1]$$

Los cuales corresponden a siete de las nueve funciones de prueba (ver tablas 5.8(a), 5.8(c), 5.8(d), 5.9(a), 5.9(c), 5.9(d) y 5.7(a)) mientras que en las dos restantes fueron los algoritmos de agrupamiento quienes presentaron mejores resultados, con valores para los parámetros  $c_1$  y  $c_2$  muy diferentes entre sí:

$$c_1 = [1.2, 2.0]$$

$$c_2 = [1.0, 1.6]$$

$$w = [0.0]$$

La única característica entre estas dos últimas topologías fue el valor común que obtuvieron para  $w = 0$ , lo que dificulta de cierta manera la posibilidad de encontrar un conjunto de valores para los parámetros de PSO que mejoren el desempeño promedio del algoritmo en cualquier función, cuando estos se localizan en extremos opuestos del rango de valores admitido.

Nuestra hipótesis acerca de este comportamiento en los valores, es que la metodología hasta ahora seguida sólo toma en cuenta aquellos valores que presentaron los mejores resultados en todo el experimento, dejando de lado otros valores que posiblemente puedan presentar buenos resultados. Sustentamos esta suposición si observamos los resultados obtenidos por el diseño factorial en las gráficas mostradas en las figuras 5.4 y 5.5, las cuales presentan los resultados del diseño factorial completo para la topología toroidal.

En estas gráficas, el eje de las “x” representa las 1,331 combinaciones o tratamientos del diseño factorial, mientras que el eje “y” representa la mejor aptitud lograda por cada combinación. En estas gráficas se nota que existen conjuntos de valores que hacen que el algoritmo se comporte caóticamente (v.g., diverja y por ende arroje resultados alejados de la mejor solución), y otros conjuntos de parámetros hacen que se estabilice (v.g., que presente un comportamiento convergente y por ende sus resultados son buenos). De tal manera, se desprende la posibilidad que haya una fuerte interacción entre parámetros que haga convergentes ciertas combinaciones de los mismos.

El comportamiento gráfico en el resto de las funciones es muy similar al presentado en las gráficas de la topología toroidal, sin embargo por razones de espacio sólo se ha presentado el conjunto de funciones correspondientes a esta topología. Adicionalmente a partir de las gráficas se pudo intuir, que aunque el diseño factorial completo haya encontrado un buen conjunto de parámetros que induzcan un buen comportamiento en PSO, también existen otros “varios” conjuntos que potencialmente podrían inducir un buen comportamiento si se hiciera una búsqueda a grano fino en su vecindad.

### 5.3.2 Interacción entre variables

Es común observar la interacción entre variables cuando se utilizan los diseños factoriales, básicamente existen tres tipos de efectos en los diseños factoriales: simples, principales y de interacción. Los que interesan en este estudio son los efectos que pueden tener las variables en la respuesta cuando presentan interacción entre ellas.

Para analizar la interacción entre variables se llevo a cabo un análisis de varianza en las funciones de prueba para conocer la integración existente entre variables ( $c_1$ ,  $c_2$  y  $w$ ), las gráficas 5.7 a 5.6 muestran los resultados de este análisis.

Las líneas paralelas en las gráficas indican que no existe interacción entre variables, por otro lado si estas no son paralelas significa que existe interacción entre variables. Pero basar esta afirmación simplemente en la observación de sus gráficas es una forma subjetiva de decir que existe interacción entre variables, es debido a esto que se utiliza en análisis de varianza para determinar la interacción entre variables.

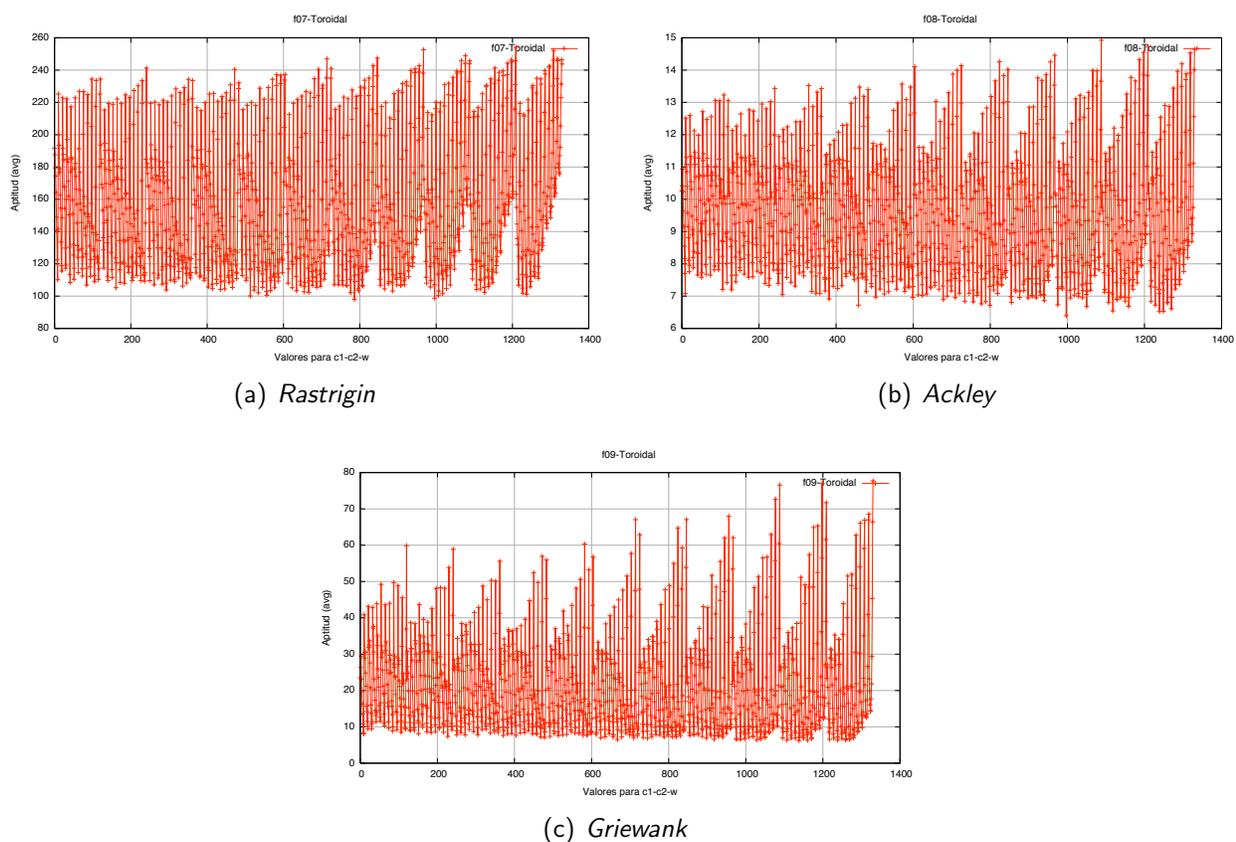


Figura 5.4: Gráficas del diseño factorial completo en todas las funciones de prueba con la topología toroidal.

En el análisis de varianza, se dice que existe interacción entre variables si la suma de cuadrados promedio de un factor, es mayor a la suma de cuadrados promedio debido al error (también llamado residuo). Alternativamente, se puede utilizar el enfoque del valor de  $P$ , si el valor obtenido de  $P$  es mejor que  $\alpha$  comúnmente igual a 0.05.

Los resultados del análisis de varianza mostrados en las gráficas 5.7 a 5.6 muestran que existe alta interacción de la variable  $w$  en todas las funciones de prueba. Por otro lado las variables  $c_1$  y  $c_2$  sólo se ven afectadas cuando  $w$  interactúa con ellas.

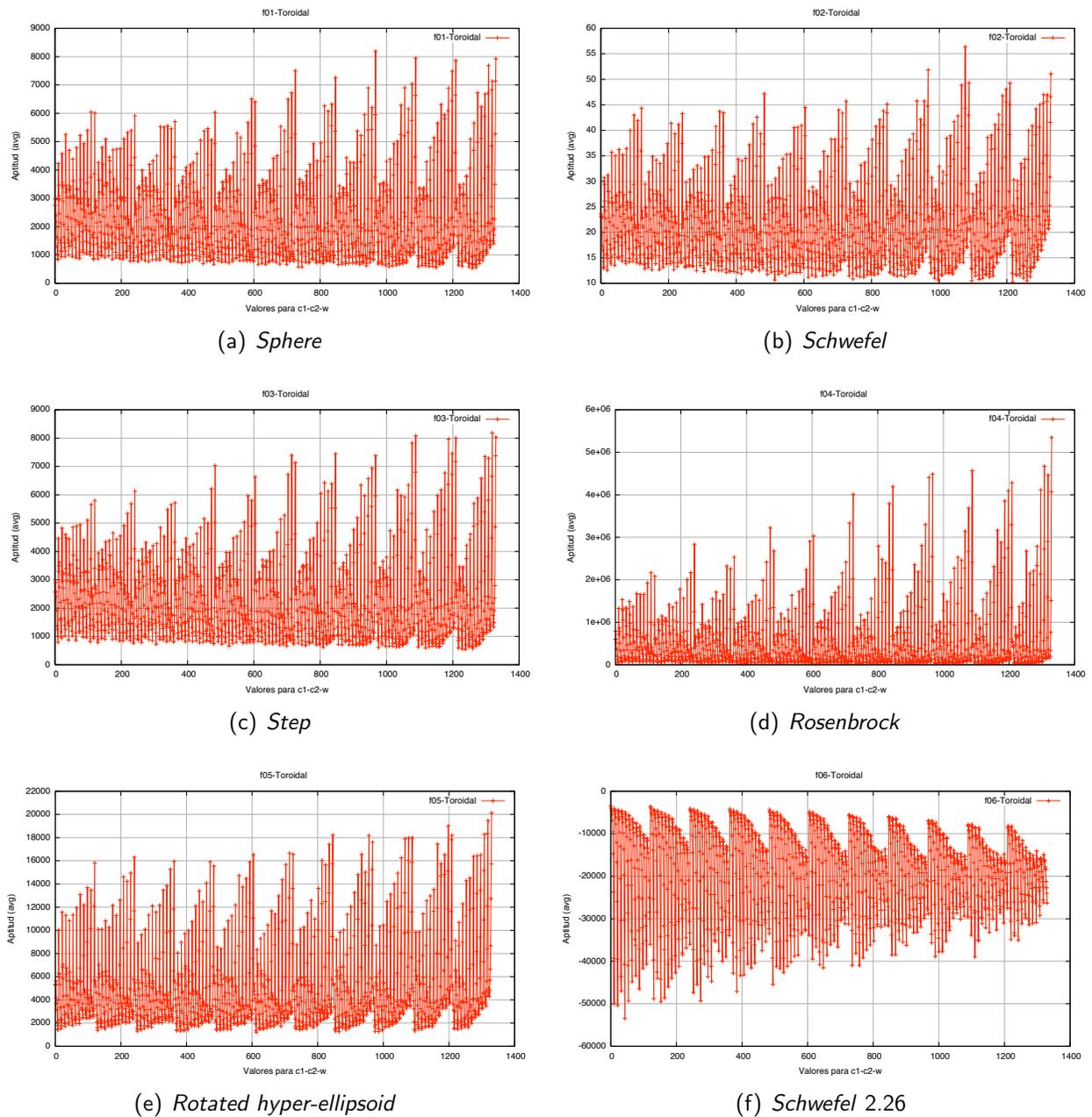


Figura 5.5: Gráficas del diseño factorial completo en todas las funciones de prueba con la topología toroidal.

### 5.3.3 Conclusiones

A partir del análisis realizado anteriormente, se puede concluir satisfactoriamente que la topología que mejora el desempeño promedio de PSO es la topología toroidal.

Partiendo del análisis del diseño factorial es posible concluir que existen dos grupos de parámetros en los cuales PSO se desempeña correctamente. El primer grupo es:

$$c_1 = [2.6, 2.8]$$

$$c_2 = [2.8, 3]$$

$$w = [0.9, 1]$$

el cual mejoró el comportamiento de PSO en siete de las nueve funciones de prueba. El segundo grupo es:

$$c_1 = [1.2, 2.0]$$

$$c_2 = [1.0, 1.6]$$

$$w = [0.0]$$

el cuál se comportó bien en las dos funciones restantes. Sin embargo, podemos decir que el diseño factorial es insuficiente en la calibración de parámetros. Suponemos que es posible suavizar la distancia existente entre ambos grupos para obtener un valor intermedio que presente buenos resultados en todos los problemas.

También se observó que la interacción entre variables hace que el algoritmo presente múltiples combinaciones convergentes, siendo cada una de un conjunto potencial para obtener mejores resultados. De igual manera, estos resultados sugieren realizar búsqueda local al rededor de más de una configuración de parámetros.

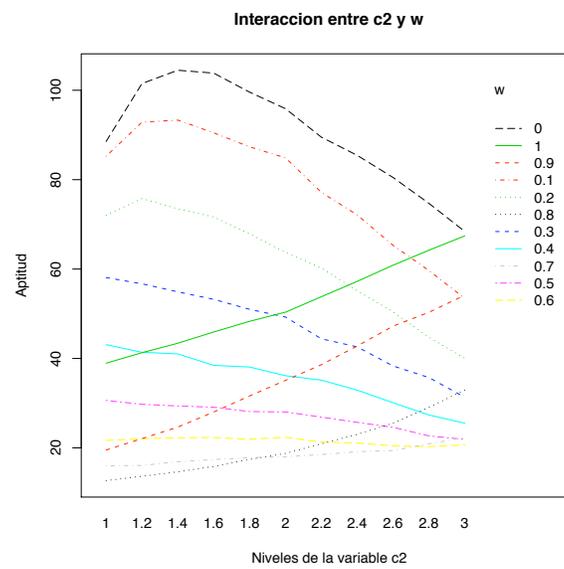
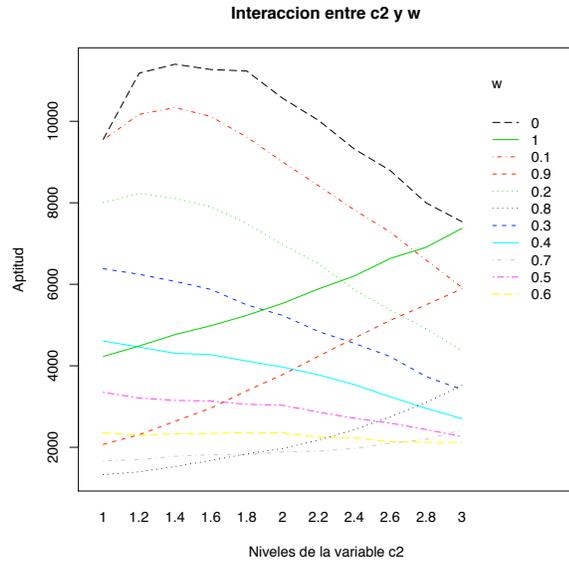
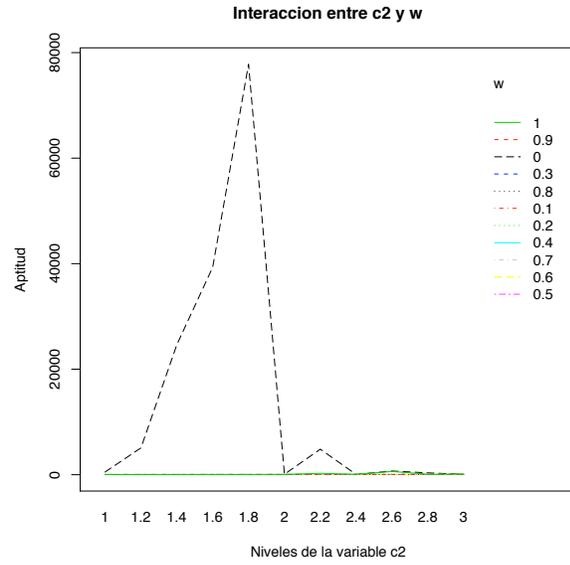
(a) *Griewank*

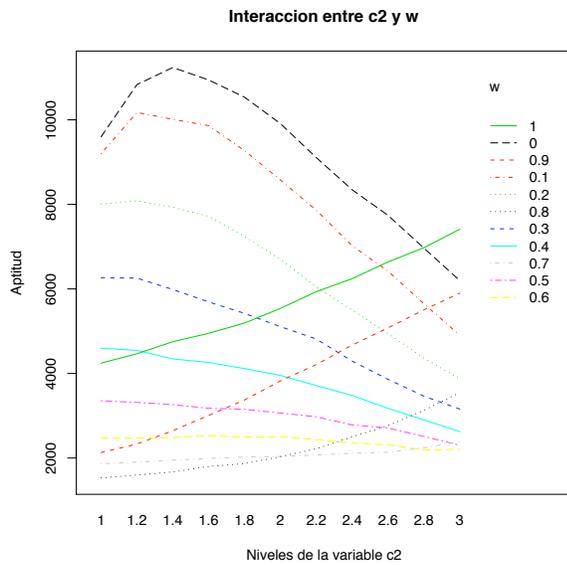
Figura 5.6: Análisis de varianza (ANOVA) en cada una de las funciones de prueba.



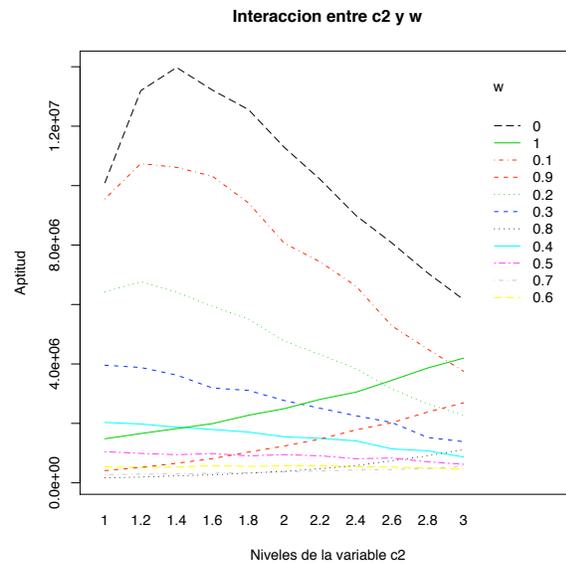
(a) *Sphere*



(b) *Schwefel*

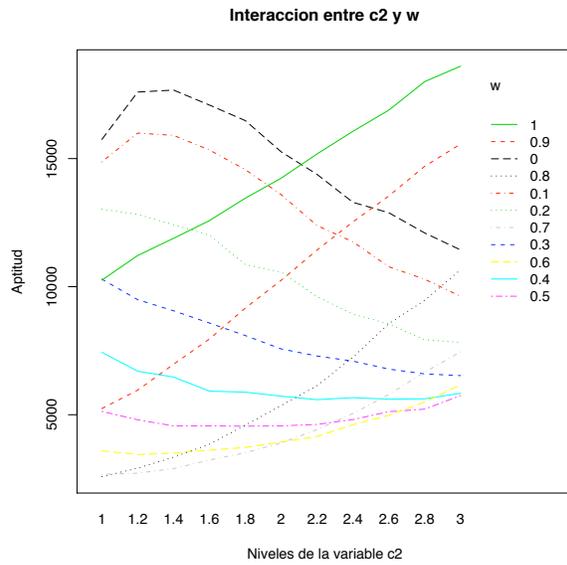


(c) *Step*

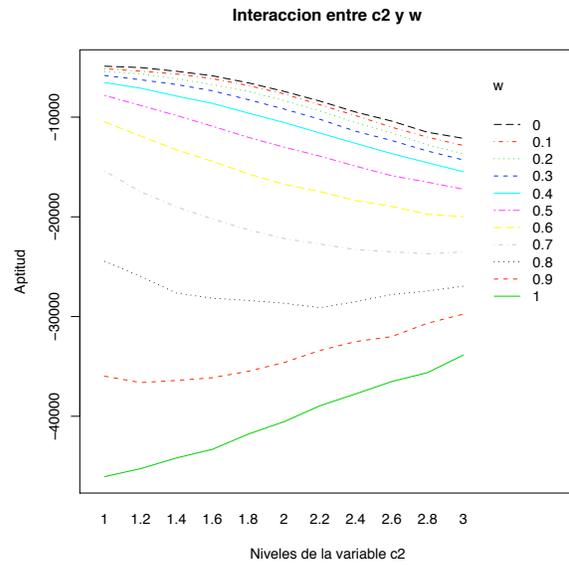


(d) *Rosenbrock*

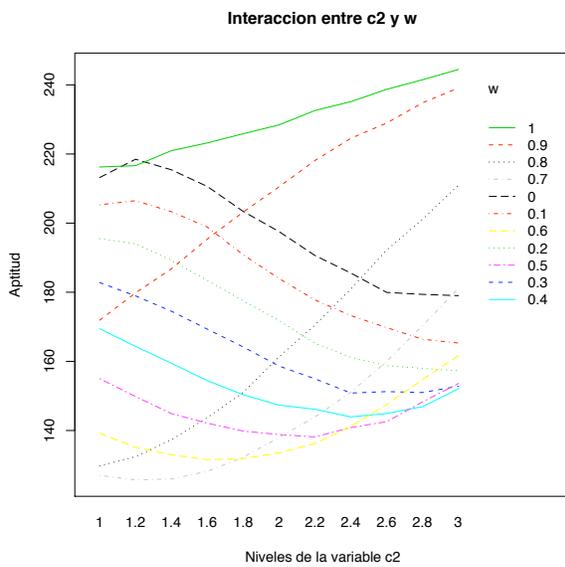
Figura 5.7: Análisis de varianza (ANOVA) en cada una de las funciones de prueba.



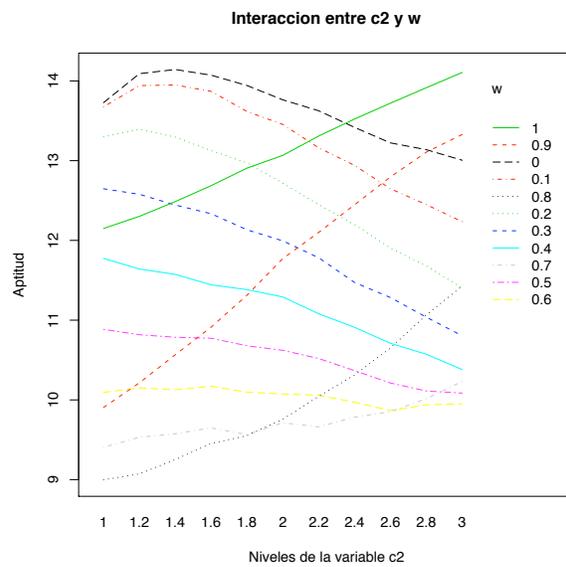
(a) *Rotated hyper-ellipsoid*



(b) *Schwefel 2.26*



(c) *Rastrigin*



(d) *Ackley*

Figura 5.8: Análisis de varianza (ANOVA) en cada una de las funciones de prueba.

## 5.4 Experimento 3

Los diseños factoriales permiten conocer los efectos que los factores tendrán en la respuesta del proceso, así como la interacción entre ellos. Sin embargo, los diseños factoriales no permiten realizar búsquedas en el interior de la superficie, debido a que suponen que la superficie de respuesta es lineal (o plana).

Es posible que los diseños factoriales exploren el interior de la superficie de respuesta (*v.g.*, una búsqueda a grano fino) si se incrementan los niveles de cada factor, pero esto traería consigo un costo computacional excesivamente alto. Por lo tanto, se requiere un método que permita explorar eficazmente el interior de la superficie.

El siguiente paso en la metodología propuesta, es analizar la superficie de respuesta (definida por los valores de los parámetros de PSO) por medio de la metodología de superficie de respuesta (descrita en la sección 4.3). RSM se utilizará como una técnica de búsqueda a grano fino sobre el mejor conjunto de valores arrojados por el experimento factorial completo, para cada una de las topologías en las funciones de prueba.

### 5.4.1 Metodología de superficie de respuesta

Para iniciar el proceso de RSM, se supondrá que los puntos de diseño iniciales son aquellos tratamientos arrojados por el experimento de factorial completo. Asumiendo que estamos lo suficientemente lejos del óptimo y en una región sin curvatura, el proceso de RSM iniciará con un modelo polinomial de primer orden (mostrado en la ecuación (4.4)). Esta es la única clase de diseños que minimiza la varianza de los coeficientes de regresión,  $\hat{\beta}_i$ .

El diseño utilizado para recolectar los datos (muestras) será un diseño factorial  $2^k$ , donde  $k$  representa el número de factores del algoritmo básico de PSO:  $w$ ,  $c_1$  y  $c_2$  (véase la sección 4.2.1 en caso que se requiera una explicación más completa). Este tipo de diseños son llamados ortogonales (ver 4.2.1 para más información). Para permitir estimar el error experimental y poder revisar la adecuación del modelo, se aumentarán las replicas en el centro de diseño en  $n_C = 6$  y se agregarán

$n_F = 2^k$  puntos. La adición de puntos en el centro de diseño no altera la propiedad ortogonal del diseño (véase la sección 4.6.1 para más información).

El siguiente paso será ajustar el modelo polinomial de primer orden con el método de mínimos cuadrados (en la sección 4.4 se encuentra una descripción detallada). La adecuación del modelo de primer orden se obtendrá por medio de la prueba estadística  $F$  (véase 4.5 para mayor información). Si  $F_0$  excede a  $F_{\alpha, k, n-k-1}$  entonces el modelo ha sido ajustado correctamente. Donde,  $\alpha = 0.25$ , por lo tanto  $F_0 = 1.60$  (extraído de [35]) y  $n$  es el número de observaciones denotado por la suma de  $n_F + n_C$ .

Usando el método del descenso más rápido (en la sección 4.3.2 se describe ampliamente este método) nos moveremos en un camino desde el centro de diseño en la dirección donde la respuesta decrece más rápidamente. Se ha decidido usar un tamaño de paso básico para cada factor: para  $w$ ,  $step\_size_1 = 0.025$ ; para  $c_1$  y  $c_2$ ,  $step\_size_2 = step\_size_3 = 0.05$ .

El método del descenso más rápido se detendrá hasta observar un incremento en la respuesta. Entonces un nuevo modelo de primer orden será ajustado alrededor del último punto obtenido y nuevamente un diseño factorial  $2^k$  con seis puntos de diseño centrales se implementará. Si el modelo de primer orden presenta una falta de ajuste, implicará que no es una aproximación adecuada de la superficie real y a que probablemente la superficie presenta una curvatura. La curvatura en la superficie podría indicar que nos encontramos cerca del óptimo, de tal manera, un modelo de orden superior deberá ser usado (ecuación (4.5)).

El modelo de segundo orden será ajustado usando un diseño de composición central (CCD) (ver sección 4.6.2), con  $n_F$  ejecuciones,  $2k$  ejecuciones en estrella o axiales, y  $n_C$  ejecuciones centrales, donde  $\alpha = (n_F)^{1/4}$  y  $n_C = 6$ . Las ejecuciones axiales son agregadas para permitir que los términos cuadráticos sean incorporados en el modelo. Es importante señalar dos parámetros en el diseño CCD: la distancia  $\alpha = (n_F)^{1/4}$  de las ejecuciones axiales desde el punto central del diseño y el número de puntos centro  $n_C = 6$ . Esto hace al diseño rotatable, lo cual significa que provee igual precisión de estimación en todas las direcciones, ya que la localización del óptimo es desconocido.

Basado en la prueba estadística  $F$ , decidiremos ajustar el modelo de segundo orden para la respuesta del proceso. Si el modelo es ajustado correctamente procederemos a la localización del

punto estacionario usando el análisis canónico para caracterizar la superficie de respuesta (ver sección 4.3.3 para más información), en el cual será necesario expresar el modelo ajustado en su forma canónica. Los eigenvalores  $\lambda_k$  determinarán el tipo de punto estacionario que se ha encontrado. Si los  $\lambda_i$  son todos positivos, se tratará de un punto de mínima respuesta; si los  $\lambda_i$  son todos negativos, será un punto de máxima respuesta; y si los  $\lambda_i$  son de signo diferente, se tratará de un punto silla.

Por otro lado, si el modelo de segundo orden no fuera ajustado correctamente, se generará el informe final a partir del último mejor punto de diseño encontrado.

La metodología de RSM será ejecutada para cada una de las funciones de prueba, con 30 dimensiones cada una, para cada una de las topologías propuestas.

Notemos que la función a ser optimizada se trata de una medida de desempeño que nos permitirá conocer la eficacia es el valor del parámetro encontrado, lo que significa que la forma de la superficie que deseamos encontrar no es la función como tal, si no las forma de la superficie que las variables de PSO forman al interactuar unas con otras y con la medida de desempeño (v.g., la función de prueba).

## 5.4.2 Resultados

Las tablas 5.11, 5.12 y 5.13 muestran los valores de los parámetros calibrados después de aplicar RSM sobre cada uno de los tratamientos del experimento factorial completo.

Para comparar el desempeño de los valores de PSO calibrados con RSM, se ejecutaron dos experimentos independientes: uno con los parámetros arrojados por RSM y otro con los valores predeterminados en PSO. Los resultados de este experimento se muestran en las tablas, donde la columna **aptitud con RSM** es el valor mínimo obtenido al ejecutar la función de prueba (30 veces) con los valores calibrados con RSM y la columna **valores por omisión** (llamado VPO en la tabla) es el valor mínimo obtenido al ejecutar la función de prueba (30 veces) con los valores predeterminados de los parámetros de PSO.

A partir de las gráficas anteriores, es posible observar que la topología toroidal fue la que mejor se desempeño cuando se calibraron los parámetros de PSO con RSM (ver tablas 5.10(a), 5.10(b),

5.10(d), 5.11(a), 5.11(d) y 5.12(a)).

La topología en malla (ver la tabla 5.10(c)) y los algoritmo de agrupamiento *K-means* (ver la tabla 5.11(b)) y jerárquico (ver la tabla 5.11(c)) también presentaron un comportamiento favorable con los valores arrojados por RSM.

Por otro lado, las topologías que no presentaron un desempeño favorable después de la calibración fueron: malla y árbol (ver tabla 5.10(d)), de igual manera, las topologías en malla y estrella en la tabla 5.11(a).

### 5.4.3 Conclusiones

La calibración de parámetros llevada a cabo por RSM mejora significativamente el desempeño del algoritmo de PSO, cuando se ejecuta en el conjunto de funciones propuesto.

En general, la gran mayoría de las topologías mejoraron su desempeño cuando los valores de PSO fueron calibrados con RSM. Aunque la topología toroidal presento el mejor desempeño, todas las topologías mejoraron su desempeño y superaron el desempeño de PSO cuando éste utiliza los valores por omisión.

Es importante observar que hay algunos valores de los parámetros que no cambiaron su valor mientras se aplicaba el procedimiento de RSM, *v.g.*, permanecieron con el valor proveniente del experimento factorial completo. Muy probablemente la explicación detrás de este efecto, es que el modelo de segundo orden no se ajustó correctamente y por lo tanto regresó al último punto de diseño (obtenido por los diseños factoriales  $2^k$ ) donde experimentó el mejor desempeño, a pesar de haber descendido por la superficie con el método del **descenso más rápido**.

Si recordamos, la adecuación de los modelos de regresión de primer y segundo orden está dado por la hipótesis estadística  $F$ , la cual requiere que la varianza las muestras sea igual (lo que significa repetir las ejecuciones para determinar su varianza), si no se aproxima esta igualdad no es posible definir la adecuación del modelo. Por otro lado este tipo de hipótesis son criticadas duramente por depender de un valor  $\alpha$  a veces considerado un tanto subjetivo.

En este experimento se tomó un conjunto de parámetros del diseño factorial completo, no obstante

en el experimento 3 se mostró que existen múltiples conjuntos de parámetros convergentes, por lo tanto sería deseable contar con una prueba o métrica que ayude a filtrar estos conjuntos de parámetros.

(a) Función *sphere*

Topología	Valores para los parámetros			Aptitud con RSM	VPO $c_1 = c_2 = 1.49, w = 0.72$
	$c_1$	$c_2$	$w$		
1	1.8	1.2	0.8	333.959625	527.755188
2	1.21750982914917	2.00352380805622	0.706874125840002	397.817139	437.971008
3	<b>3</b>	<b>1.8</b>	<b>0.5</b>	<b>252.769775</b>	<b>386.847687</b>
4	1.03874546893363	1.04939347304942	0.889351730410215	357.221405	1106.123169
5	2.41921805201334	1.77836836553516	0.692191037747123	349.158752	513.906982
6	2.8	2	0.5	484.839966	1045.103516
7	2	2.8	0.5	340.714874	1020.850037
8	1.58201908218393	1.37451805130885	0.847531299903634	459.898071	760.887695

(b) Problema de *Schwefel*

Topología	Valores para los parámetros			Aptitud con RSM	VPO $c_1 = c_2 = 1.49, w = 0.72$
	$c_1$	$c_2$	$w$		
1	2.61809191469735	1.78250952777956	0.37856219151781	10.574624	11.72756
2	1.8	2	0.7	8.283688	9.557344
3	<b>2.6</b>	<b>1.8</b>	<b>0.5</b>	<b>7.139532</b>	<b>9.500757</b>
4	2.4	1.2	0.8	10.217173	13.362272
5	2	2	0.8	10.252662	11.644361
6	1.2	1.6	0.8	11.336485	13.409315
7	2.8	1.8	0.5	8.231	11.867805
8	2	1	0.9	12.81024	17.87634

(c) Función *Step*

Topología	Valores para los parámetros			Aptitud con RSM	VPO $c_1 = c_2 = 1.49, w = 0.72$
	$c_1$	$c_2$	$w$		
1	1	1.2	0.7	579	584
2	<b>3</b>	<b>1.2</b>	<b>0.6</b>	<b>276</b>	<b>393</b>
3	1.2	1.2	0.8	343	411
4	2.61535079155129	2.1909197717088	0.577433810579639	619	905
5	1.23912111468382	2.02229367631297	0.825	727	1291
6	2.8	1	0.8	645	966
7	2.6	3	0.5	455	1118
8	2.07390323567224	1.92406205976573	0.821251416829964	396	967

(d) Función de *Rosenbrock*

Topología	Valores para los parámetros			Aptitud con RSM	VPO $c_1 = c_2 = 1.49, w = 0.72$
	$c_1$	$c_2$	$w$		
1	1.94150233632737	1.66442043958148	0.508571441843373	15919.494141	27681.880859
2	1.4	1	0.8	15633.451172	9084.208984
3	<b>1.6</b>	<b>1.2</b>	<b>0.8</b>	<b>5625.679688</b>	<b>9652.208008</b>
4	1.57724394155946	0.98733985501459	0.889772060917157	14084.302734	21629.357422
5	3	1.6	0.7	24628.189453	35979.867188
6	1	2.8	0.7	110571.734375	22475.363281
7	2.2	2	0.7	14771.779297	21053.283203
8	1.6	1.8	0.8	41773.324219	78786.578125

Tabla 5.11: Valores de los parámetros de PSO obtenidos por RSM. Las topologías son: 1) Anillo, 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol, 7) *K-means* y 8) Jerárquico.

(a) Función *Rotated hyper-ellipsoid*

Topología	Valores para los parámetros			Aptitud con RSM	VPO $c_1 = c_2 = 1.49, w = 0.72$
	$c_1$	$c_2$	$w$		
1	2.96678216093847	0.977333115600322	0.686859332580901	964.244263	471.190033
2	1.8	1.8	0.6	871.589417	600.369507
3	<b>1.6</b>	<b>1.2</b>	<b>0.8</b>	<b>493.241241</b>	<b>848.927856</b>
4	1.8	1.2	0.8	1036.453125	1336.614746
5	2.4	2	0.7	1861.062134	1378.052124
6	1.2	2	0.7	937.154175	943.408264
7	1.07287628674127	1.0194745360128	0.905544276932977	930.705139	1427.613525
8	1.2	1	0.9	963.967957	1254.936157

(b) Problema generalizado de *Schwefel 2.26*

Topología	Valores para los parámetros			Aptitud con RSM	VPO $c_1 = c_2 = 1.49, w = 0.72$
	$c_1$	$c_2$	$w$		
1	1.2	1.4	1	-63900.667969	-32190.126953
2	1.36859203043900	1.21905687889905	1.01615124495400	-67742.757812	-35047.148438
3	1.63315569983381	1.25	1.01480875098068	-73881.671875	-30601.765625
4	1.6	1	1	-90158.5	-30741.560547
5	2.21313815822114	1.78394118849961	1.025	-89296.46875	-37037.554688
6	1.4	1.4	1	-66740.664062	-31070.546875
7	<b>1.2</b>	<b>1</b>	<b>1</b>	<b>-104836.820312</b>	<b>-33222.855469</b>
8	2.63831089599305	0.954957332681001	1.05	-87016.6562	-28232.105469

(c) Función de *Rastrigin*

Topología	Valores para los parámetros			Aptitud con RSM	VPO $c_1 = c_2 = 1.49, w = 0.72$
	$c_1$	$c_2$	$w$		
1	1.69227276478486	2.32590365264891	0.405433806779257	118.306084	108.054176
2	2.19164081760383	1.45775994157551	0.593729260597854	59.616932	93.7957
3	2.35135088915071	1.5879363026885	0.596953954541176	69.46373	88.038666
4	1.39741644110377	1.36895476349706	0.803142715319011	62.921871	80.135696
5	2.4	1.2	0.8	83.950829	64.349007
6	2.8	1.6	0.4	107.878464	91.25943
7	2	1.2	0.8	69.105125	74.679459
8	<b>2.2</b>	<b>1</b>	<b>0.8</b>	<b>49.217339</b>	<b>77.868446</b>

(d) Función de *Ackley*

Topología	Valores para los parámetros			Aptitud con RSM	VPO $c_1 = c_2 = 1.49, w = 0.72$
	$c_1$	$c_2$	$w$		
1	1.91516347813823	2.31824829006885	0.511901577526519	6.430874	6.324037
2	2.8	2	0.5	5.301189	5.560427
3	<b>1.40738003249954</b>	<b>1.20284093001617</b>	<b>0.799792864358377</b>	<b>4.835998</b>	<b>5.884015</b>
4	2.2	1.6	0.8	6.140133	8.6338
5	2.6	1.2	0.8	6.095523	9.01023
6	2.49510647227609	1.52572540588193	-0.116522835573364	9.149312	7.783676
7	2.6	2	0.6	5.387611	7.324994
8	2	2.4	0.7	8.170965	8.380889

Tabla 5.12: Valores de los parámetros de PSO obtenidos por RSM. Las topologías son: 1) Anillo, 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol, 7) *K-means* y 8) Jerárquico.

(a) Función de *Griewank*

Topología	Valores para los parámetros			Aptitud con RSM	VPO $c_1 = c_2 = 1.49, w = 0.72$
	$c_1$	$c_2$	$w$		
1	1	2.2	0.7	5.330857	5.418095
2	3	1.4	0.6	3.621432	3.825122
3	<b>2.78741153640953</b>	<b>0.939366534282044</b>	<b>0.708613241530622</b>	<b>3.054165</b>	<b>4.891899</b>
4	2.6430568632999	1.05	0.777593320227243	4.029626	7.965417
5	1.80035941458992	1.38892828914994	0.88450616698354	4.706392	8.747098
6	1.8	1.4	0.8	8.945205	8.175719
7	2.58290208791464	1.69406523154543	0.59340098833803	6.510141	7.068118
8	1	1	0.9	3.995842	9.90715

Tabla 5.13: Valores de los parámetros de PSO obtenidos por RSM. Las topologías son: 1) Anillo, 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol, 7) *K-means* y 8) Jerárquico.

## 5.5 Metodología propuesta para la calibración de parámetros en AEs

Motivados por los resultados obtenidos previamente, decidimos proponer la siguiente metodología, con el objetivo de verificar la existencia de un conjunto de parámetros que incremente el desempeño del algoritmo de PSO.

Los objetivos planteados son:

- Determinar si existe un conjunto de valores para los parámetros de PSO ( $w$ ,  $c_1$  y  $c_2$ ) que presenten un buen desempeño cuando se trata de optimizar una función de prueba en específico usando cualquiera de las topologías utilizadas en esta tesis.
- Determinar si existe un conjunto de valores para los parámetros de PSO ( $w$ ,  $c_1$  y  $c_2$ ) que presenten un buen desempeño cuando se utiliza una topología de interconexión específica en cualquiera de las funciones de prueba utilizadas en esta tesis.
- Determinar si existe un conjunto de valores para los parámetros de PSO ( $w$ ,  $c_1$  y  $c_2$ ) que presenten un buen desempeño independientemente de la topología o de la función de prueba (de las utilizadas en esta tesis).

La metodología propuesta se muestra en el Algoritmo 7, la cual se ejemplifica con la calibración de parámetros en PSO descrita a continuación:

1. Llevar a cabo un diseño factorial completo sobre el rango de parámetros establecido del AE: en PSO para  $c_1 = [0, 1]$ ,  $c_2 = [1, 3]$ , funciones de prueba  $[1, 9]$ , topologías de interconexión y algoritmos de agrupamiento =  $[1, 8]$ , con la finalidad de encontrar un conjunto de parámetros robusto.
2. Normalizar los datos obtenidos de ejecutar el AE en cada función de prueba de acuerdo a lo siguiente:

$$x^* = \frac{x - \min_j}{\max_j - \min_j}$$

donde  $\max_j$  y  $\min_j$  con  $j = 1, \dots, 9$  representan los valores máximos y mínimos obtenidos al ejecutar cada una de las funciones objetivo  $j$ .

3. Se seleccionará el conjunto de parámetros que presente el mejor resultado de acuerdo a la prueba estadística  $t$  (explicada en la sección 5.2.4), la cual identificará aquellos conjuntos de parámetros que presenten el mejor promedio y que sean similares entre sí.
4. Como siguiente paso, se aplicará la metodología de superficie de respuesta sobre aquellos conjuntos de parámetros seleccionados por la prueba estadística  $t$ , con el objetivo de realizar un búsqueda minuciosa sobre ellos.

En este caso, el diseño factorial completo puede verse como un mecanismo de búsqueda a grano grueso, mientras que la metodología de superficie de respuesta puede verse como un mecanismo de búsqueda a grano fino sobre un conjunto de parámetros aceptable.

5. Como paso final, se presentará aquel conjunto de parámetros que haya incrementando el desempeño promedio del algoritmo.

**Algoritmo 7** Metodología propuesta.

- 1: Llevar a cabo un diseño factorial completo sobre el rango de parámetros establecido del algoritmo evolutivo.
- 2: Normalizar los datos obtenidos de ejecutar el AE en cada función de prueba,  $j$ , de acuerdo a lo siguiente:  $x^* = \frac{x - \min_j}{\max_j - \min_j}$
- 3: Seleccionar el mejor conjunto de parámetros conseguido en el diseño factorial.
- 4: Realizar una prueba estadística  $t$  a partir del mejor conjunto de parámetros seleccionado por el diseño factorial, para seleccionar aquellos conjuntos similares.
- 5: Aplicar la metodología de superficie de respuesta sobre aquellos conjuntos de parámetros seleccionados por la prueba  $t$
- 6: Seleccionar el conjunto de parámetros que haya incrementando el desempeño promedio del algoritmo.

**5.5.1 Experimento 4a**

Después de realizar los experimentos anteriores, se pudo observar que efectivamente existen conjuntos de valores para los parámetros de PSO con los cuales mejora su desempeño promedio. Motivados por estos resultados, se decidió llevar a cabo un cuarto experimento utilizando la metodología propuesta, con la finalidad de encontrar aquel conjunto de parámetros que trabaje bien en cada función de prueba independientemente de la topología seleccionada.

De la misma forma que el experimento 5.4, iniciaremos el proceso con un diseño factorial completo que involucre una única función objetivo denotada por:

$$\mathbf{T}_j(\mathbf{x}) = \sum_{i=1}^n t_i(x^*) \quad (5.3)$$

donde  $n = 8$ , es el número de topologías implementadas en cada una de las funciones de prueba  $j$ , con  $j = 1, \dots, 9$ . Para realizar dicha sumatoria, es necesario normalizar previamente los valores de las funciones, de la misma forma en que se realizó en la prueba estadística  $t$ .

$$x^* = \frac{x - \min}{\max - \min} \quad (5.4)$$

donde  $\min$  y  $\max$ , representan el valor mínimo y máximo obtenido al ejecutar la función  $j$  en todas las topologías implementadas (cada una de las topologías será ejecutada 30 veces). Esta normalización

se realiza con la finalidad de reducir la variabilidad en la respuesta de la función  $\mathbf{T}(\mathbf{x})$ .

Posteriormente se escogerá aquellos conjunto de valores ( $c_1$ ,  $c_2$  y  $w$ ) que arroje los mejores resultados de acuerdo a su desempeño promedio y la prueba estadística  $t$  para aplicarles la metodología de superficie de respuesta.

### 5.5.1.1. Resultados

Los mejores conjuntos de valores obtenidos por el diseño factorial completo para cada una de las funciones se muestran en la tabla 5.14, donde la columna **función** corresponde a la función ejecutada para obtener dichos parámetros, el valor mostrado en la última columna corresponde al valor promedio de la función objetivo,  $\mathbf{T}_j(\mathbf{x})$ , alcanzado con los valores obtenidos.

Función	Valores de los parámetros			Promedio
	$c_1$	$c_2$	$w$	$T_j(x)$
f01	2.4	1	0.8	0.161450715499766
f02	2.4	1	0.8	$4.65402194850756e - 07$
f03	2.8	1.6	0.7	0.131685799805874
f04	1.8	1	0.8	0.00497160597773529
f05	1.8	1	0.8	0.104288319767772
f06	1	1	1	4.70788370276065
f07	1.4	1.2	0.8	1.80818251838716
f08	3	1.4	0.7	2.50598321904561
f09	2.6	1	0.8	0.154207966455378

Tabla 5.14: Diseño factorial completo para el experimento 4a. f01) *Sphere*, f02) *Schwefel*, f03) *Step*, f04) *Rosenbrock*, f05) *Rotated hyper-ellipsoid*, f06) *Schwefel 2.26*, f07) *Rastrigin*, f08) *Ackley* y f09) *Griewank*.

El siguiente paso en la metodología nos indica que se debe aplicar la prueba estadística  $t$  para filtrar un mayor conjunto de parámetros, lo cual nos permitirá realizar una búsqueda a grano fino en diferentes regiones del espacio de búsqueda por medio de la metodología de superficie de respuesta.

Sin embargo, en ocasiones el número de conjuntos que pasan la prueba  $t$  puede ser elevado. Para fines de este trabajo, se han elegido los cinco mejores valores que superaron esta prueba. A estos conjuntos se les aplicará la metodología de superficie de respuesta.

La tabla 5.15, también muestra los valores elegidos por la prueba estadística  $t$  y aquellos obtenidos por RSM al sintonizar los valores elegidos por la prueba  $t$ , la penúltima columna (llamada Apt. Norm. RSM) de la tabla muestra la aptitud conseguida (aún normalizada) por los parámetros sintonizados

Fun	Prueba $t$ $c_1-c_2-w$	Valores obtenidos por RSM		Apt. Norm. RSM	Apt. Norm. VPO
		$c_1-c_2-w$		$c_1-c_2-w$	1.49 - 1.49 - 0.72
f01	2.2 - 1 - 0.8	2.21451476531964 - 0.907555572516606 - 0.822021003316988		0.163843103669761	0.271561972490627
f01	2.4 - 1 - 0.8	2.40949662967633 - 1.02298329799542 - 0.795047021050783		0.158874420649999	0.271561972490627
f01	2.4 - 1.2 - 0.8	2.13630464023929 - 1.08198425492703 - 0.823093630339378		0.175573345182795	0.271561972490627
f01	2.6 - 1 - 0.8	2.60127943809384 - 1.02169288802524 - 0.775877284893101		0.171628999203111	0.271561972490627
f01	3 - 1.4 - 0.7	3 - 1.4 - 0.7		0.164716471352136	0.271561972490627
f02	1.8 - 1 - 0.8	1.38449369790386 - 0.543751021606914 - 0.753149026716469		5.37298953729009e <sup>-07</sup>	6.47164097568182e <sup>-07</sup>
f02	2.2 - 1 - 0.8	2.15339153080608 - 1.34726327759184 - 0.793348786943337		5.17664580022684e <sup>-07</sup>	6.47164097568182e <sup>-07</sup>
f02	2.4 - 1 - 0.8	2.32838042704828 - 0.962613274037767 - 0.785322490974405		4.73530465670641e <sup>-07</sup>	6.47164097568182e <sup>-07</sup>
f02	2.8 - 1 - 0.7	2.66738443691436 - 0.91341346189978 - 0.760684159382997		4.652233962357e <sup>-07</sup>	6.47164097568182e <sup>-07</sup>
f02	3 - 1 - 0.7	3.03007313751582 - 1.03118862377461 - 0.710865707715548		5.00041305361701e <sup>-07</sup>	6.47164097568182e <sup>-07</sup>
f03	2.4 - 1 - 0.8	2.42323504213372 - 1.03003366437235 - 0.809912016301178		0.140569877883311	0.249935206500694
f03	2.8 - 1.6 - 0.7	f2.82941031737428 - 1.65374181455255 - 0.69007791118398		0.14198059752311	0.249935206500694
f03	3 - 1.2 - 0.7	3.03276655936265 - 1.22045994014023 - 0.697798091375323		0.136883508911012	0.249935206500694
f03	3 - 1.4 - 0.7	2.9637192210095 - 1.43758970449227 - 0.70451811613357		0.136972441164962	0.249935206500694
f03	3 - 2 - 0.6	3.01093032280782 - 2.07191681218162 - 0.591259827245677		0.160846939968188	0.249935206500694
f04	1.8 - 1 - 0.8	1.79487721190437 - 1.02662550174832 - 0.825		0.00488141576397087	0.0119111513154515
f04	2 - 1.2 - 0.8	2 - 1.2 - 0.8		0.00656405911870423	0.0119111513154515
f04	2.4 - 1 - 0.8	2.14479961590651 - 1.26094464046551 - 0.905007319035324		0.0213789360050679	0.0119111513154515
f04	2.6 - 1 - 0.8	2.4588825208491 - 1.06232457208472 - 0.785889611285266		0.00679891567420341	0.0119111513154515
f04	3 - 1.4 - 0.7	2.97186082345052 - 1.44116983132891 - 0.700371325532907		0.00619103793629565	0.0119111513154515
f05	1.2 - 1 - 0.8	1.19818092830281 - 1.15965113362146 - 0.775271677699338		0.130378404342354	0.153248503004497
f05	1.4 - 1 - 0.8	1.29340531879525 - 1.07996875768895 - 0.803872348280245		0.119459815905499	0.153248503004497
f05	1.6 - 1 - 0.8	1.42714104707806 - 1.01959170027061 - 0.81251459083572		0.109462970210997	0.153248503004497
f05	1.8 - 1 - 0.8	1.77862858548854 - 1.01511152471722 - 0.791857665290023		0.109084634531024	0.155327231567671
f05	2 - 1 - 0.8	1.97460311412484 - 1.09363563830273 - 0.780455907451104		0.116021242017331	0.153248503004497
f06	1 - 1 - 1	0.793623549490487 - 0.887049102213912 - 1.42206230499874		2.56175142782188	6.83680691561986
f06	1 - 1.2 - 1	0.918748382493475 - 1.27200749260529 - 1.08835999436854		4.08135620141639	6.83680691561986
f06	1 - 1.4 - 1	1.01803778108099 - 1.23074010322477 - 1.19046640845253		3.93955742498676	6.83680691561986
f06	1.2 - 1 - 1	0.0887817355506794 - 0.601877016404683 - 1.83803248098864		-0.723575146944011	6.83680691561986
f06	1.4 - 1.2 - 1	1.63364436463818 - 1.37120999810862 - 0.941236012567527		5.24948434983392	6.83680691561986
f07	1.4 - 1.2 - 0.8	1.4 - 1.2 - 0.8		1.80818251838716	1.92205591170985
f07	1.6 - 1 - 0.8	1.60363476009694 - 0.972272892915832 - 0.803893982350912		1.87913840895491	1.92205591170985
f07	1.6 - 1.2 - 0.8	1.58599205924967 - 1.29965185560531 - 0.796635793743438		1.86035494900791	1.92205591170985
f07	2.4 - 1.2 - 0.7	2.39205442608635 - 1.14260329792791 - 0.70692568961757		1.84116247176435	1.92205591170985
f07	2.6 - 1.2 - 0.7	2.60080654483344 - 1.23660162633127 - 0.68460681604567		1.86810977331321	1.92205591170985
f08	2.4 - 1 - .8	2.42951581920396 - 0.926384790264021 - 0.823822256858496		2.65130871525832	3.09303659362475
f08	2.6 - 1.2 - 0.8	2.61018431254102 - 1.18398186809147 - 0.781912426161626		2.58430885914407	3.09303659362475
f08	2.8 - 1 - 0.8	8.35292297130691 - 1.09685619607629 - 0.899898096564582		4.76365101461184	3.09303659362475
f08	2.8 - 1.2 - 0.8	2.75877852105007 - 1.17481867231291 - 0.774065945246512		2.56072071408898	3.09303659362475
f08	3 - 1.4 - 0.7	3 - 1.4 - 0.7		2.50598321904561	3.09303659362475
f09	1.8 - 1 - 0.8	1.78418152970666 - 0.957626044724262 - 0.841669068637909		0.161240017296071	0.262723599868832
f09	2.2 - 1.2 - 0.8	2.2 - 1.2 - 0.8		0.165949217041632	0.262723599868832
f09	2.6 - 1 - 0.8*	2.30475214978299 - 0.433148181586842 - 0.768956159507187*		0.659672192906018*	0.262723599868832*
f09	3 - 1 - 0.7	2.99828325373653 - 1.00301115092339 - 0.732099971792927		0.167603103562231	0.262723599868832
f09	3 - 1.2 - 0.7	3.00031800002121 - 0.699307044573668 - 0.629622802239423		0.274728181696422	0.262723599868832

Tabla 5.15: Resultados de la prueba estadística  $t$  para el experimento 4a. f01) *Sphere*, f02) *Schwefel*, f03) *Step*, f04) *Rosenbrock*, f05) *Rotated hyper-ellipsoid*, f06) *Schwefel 2.26*, f07) *Rastrigin*, f08) *Ackley* y f09) *Griewank*.

con la metodología propuesta, la cual es mucho menor que la aptitud conseguida (aún normalizada) utilizando los parámetros por defecto de PSO (los cuales aparecen en la última columna de la tabla denominada Apt. Norm. VPO, ). Las filas sombreadas representan los mejores valores alcanzados por cada una de las funciones.

Las tablas 5.16 y 5.17 muestran los valores reales para cada una de las funciones en cada una de las topologías implementadas, la última fila de las tablas (valores por omisión) representan el valor alcanzado por PSO cuando se utilizan los valores por omisión en la topología completamente conectada (t4).

T	f01	f02	f03	f04	f05
t1	1509.16692506667	16.5052347333333	1818.9	115730.358854167	1733.98820196667
t2	775.021921866667	12.2691332666667	812.433333333333	48440.2347656667	1509.91233116667
t3	629.900292466667	11.5806023333333	616.833333333333	41492.9458741333	1235.21725253333
t4	1030.08641463333	16.6829387333333	1088.86666666667	118490.008398467	2836.2754435
t5	1418.31449273333	17.7430518666667	1649.2	235694.412565067	2824.19504396667
t6	1500.4006999	18.1581688333333	1490.86666666667	149578.278906267	2065.00989576667
t7	890.649061066667	13.8089732333333	844.233333333333	82882.0350260667	2169.55816656667
t8	1225.79520163333	20.8075386	1585.26666666667	192858.493424467	3116.1247599
Valores por omisión					
t4	2421.94218343333	26.5807742333333	3247.86666666667	579759.2723958	4724.29145096667

Tabla 5.16: Valores de las funciones objetivo para cada una de las topologías, obtenidas con la metodología propuesta.

T	f06	f07	f08	f09
t1	-125532.761197967	149.164799833333	9.58107453333333	13.5246507
t2	-112033.334114633	114.7573221	7.49837433333333	8.909483
t3	-110983.1190104	104.9755199	6.8788688	7.58604466666667
t4	-142190.3726562	113.695642333333	8.87828926666667	13.0947816333333
t5	-127832.903385367	109.0470473	9.0945725	13.0654098
t6	-131983.980208367	141.570542866667	9.1688702	14.421682
t7	-134973.465364567	93.8706772333333	7.63855766666667	10.6191236
t8	-152200.450781267	122.8806475	9.63613616666667	12.028027
Valores por omisión				
t4	-18256.8306641333	111.187207833333	11.3465951333333	21.4075525666667

Tabla 5.17: Valores de las funciones objetivo para cada una de las topologías, obtenidas con la metodología propuesta.

### 5.5.1.2. Conclusiones

Se puede concluir, que los valores de los parámetros sintonizados mediante el diseño factorial y RSM muestran una mejora en el desempeño de PSO con respecto al desempeño logrado utilizando los valores por omisión, ya que en todas las combinaciones sugeridas por la prueba *t*, mejoraron el desempeño de PSO, excepto aquella fila con un símbolo de estrella (\*) la cual fue la única que no mejoró el desempeño de PSO.

A partir de este experimento es posible derivar intuitivamente el rango de valores para los parámetros que incrementen el desempeño de PSO en cada una de las funciones de prueba, lo cual se resume en la tabla 5.18. Puede observarse que mientras el valor de  $w$  permanece constante, el valor de  $c_1$  es el que presenta mayor variabilidad, mientras que  $c_2$  permanece en un nivel intermedio entre ambos.

Por lo tanto, mediante los resultados del experimento se lograron identificar aquellos conjuntos de parámetros que independientemente de la topología (se probaron todas las topologías simultáneamente), funcionaron bien en cada función de prueba (v.g., lograron mejorar los resultados por omisión).

Función	Valores de los parámetros		
	$c_1$	$c_2$	$w$
f01	[2.4, 2.6]	[1.0, 1.2]	[0.7, 0.8]
f02	2.4	1	[0.7, 0.8]
f03	[2.8, 3.0]	[1.2, 1.6]	[0.7, 0.8]
f04	[2.0, 2.6]	[1.0, 1.2]	[0.7, 0.8]
f05	[1.2 – 1.8]	1	0.8
f06	[1.0, 1.4]	[1.0, 1.4]	1.0
f07	[1.4, 2.6]	[1.0, 1.2]	[0.7, 0.8]
f08	[2.4, 3.0]	[1.0, 1.4]	[0.7, 0.8]
f09	[1.8, 3.0]	[1.0, 1.2]	[0.7, 0.8]

Tabla 5.18: Rango de valores para los parámetros de PSO derivados del experimento 4a. f01) *Sphere*, f02) *Schwefel*, f03) *Step*, f04) *Rosenbrock*, f05) *Rotated hyper-ellipsoid*, f06) *Schwefel 2.26*, f07) *Rastrigin*, f08) *Ackley* y f09) *Griewank*.

### 5.5.2 Experimento 4b

El objetivo de este experimento es determinar si existe un conjunto de parámetros para una topología específica que induzca un buen comportamiento de PSO en todas las funciones de prueba simultáneamente.

Para ello, se seguirá el mismo procedimiento que en el experimento 4a, la única diferencia será la función objetivo a ser minimizada que estará denotada por:

$$\mathbf{F}_j(\mathbf{x}) = \sum_{i=1}^n f_i(x^*) \quad (5.5)$$

donde  $n = 9$ , es el número de funciones implementadas en cada una de las topologías  $j$ , con  $j = 1, \dots, 8$ .

## 5.5.2.1. Resultados

Los resultados obtenidos por la metodología propuesta, se muestran en la tabla 5.19. La columna **Top** corresponde a la topología utilizada para obtener el valor de  $F_j(x)$ . Los valores elegidos por la prueba  $t$  también se especifican en la tabla, así como la aptitud alcanzada por PSO cuando utiliza los valores calibrados (aún normalizados) por RSM y la aptitud alcanzada cuando utilizan los parámetros por omisión (aún normalizados).

Se eligieron los cinco mejores conjuntos de valores seleccionados por la prueba  $t$  a los cuales se les aplico la metodología de superficie de respuesta. Los mejores promedios encontrados se muestran en la tabla 5.19,

Top	Prueba $t$ $c_1-c_2-w$	Valores por RSM $c_1-c_2-w$	Apt. Norm RSM $c_1-c_2-w$	Apt. Norm. VPO 1.49 - 1.49 - 0.72
1	1.4 - 1.4 - 0.8	1.4462964384578 - 1.36131297513153 - 0.7859326227271	0.0344176806399611	0.0352089650837478
2	2.8 - 1.4 - 0.7	2.71493656332118 - 1.40651944134776 - 0.713671855619998	0.055378807653558	0.0554241661996675
3	2.2 - 1.2 - 0.8	1.9218833376558 - 0.998076262357089 - 0.804620942200519	0.0549775505004811	0.0566664590837956
4	2.2 - 1.2 - 0.8	2.2015258722751 - 1.20031671522485 - 0.792299268597809	0.0562527137852061	0.0599346706150676
5	<b>2.8 - 1.2 - 0.8</b>	<b>2.76677819697712 - 1.25 - 0.793467376726383</b>	<b>0.00538879572976524</b>	<b>0.00911298166331154</b>
6	1.8 - 1 - 0.8	1.8 - 1 - 0.8	0.0334426833997241	0.0368645506382759
7	2.8 - 1.4 - 0.7	2.62949840471689 - 1.3378530001005 - 0.682813929579398	0.0152975921598228	0.0162063099099204
8	2.6 - 1 - 0.8	2.55963569329976 - 0.964641289346756 - 0.776336013327914	0.00708816653541264	0.0105709762192114

Tabla 5.19: Resultados de la prueba estadística  $t$  y RMS para el experimento 4b. 1) Anillo, 2) Malla, 3) Toroidal, 4) Completamente conectada, 5) Estrella, 6) Árbol, 7) K-means y 8) Jerárquico.

Los valores promedio reales para cada una de las funciones objetivo de  $F_5(x) = \text{estrella}$ , se muestran en la tabla 5.20.

Función	Topología 5 = estrella	
	Aptitud con RSM	VPO
f01	1335.05272326667	3478.44792686667
f02	16.8173742666667	29.2653532333333
f03	1164.03333333333	3774.76666666667
f04	145057.424674533	887987.788281267
f05	3878.19549153333	4557.2819987
f06	-33162.5414062	-16729.0982096333
f07	130.557693933333	118.152414933333
f08	9.30245406666667	12.0331747
f09	11.9666826666667	29.3892676333333

Tabla 5.20: Valores de las funciones objetivo despues de aplicar RSM en el experimento 4b. f01) Sphere, f02) Schwefel, f03) Step, f04) Rosenbrock, f05) Rotated hyper-ellipsoid, f06) Schwefel 2.26, f07) Rastrigin, f08) Ackley y f09) Griewank.

Aunque la diferencia es mínima, se logra percibir que existen ciertos valores para los parámetros

de PSO donde el algoritmo se comporta mejor. Para poder visualizar mejor este comportamiento, se puede ver la tabla 5.20 de valores individuales para cada una de las funciones, en ellas se puede notar que el promedio de las funciones arrojados por PSO, con sus parámetros calibrados con RSM, superan el promedio de las funciones con el algoritmo de PSO con los valores por omisión. La única función que no mostró mejoría fue la función *Rastrigin* denotada como f07 en la tabla.

### 5.5.2.2. Conclusiones

Se puede observar que los valores para el parámetro  $w$  son similares al experimento 4a, v.g.,  $w = [0.7, 0.8]$  y  $c_2 = [1, 1.4]$  lo que acota el rango de valores que podría tomar  $c_2$ , sin embargo  $c_1$  muestra una mayor variabilidad estando su rango de valores entre  $[1.4, 2.8]$ . Lo que indica que existe clara diferencia en los valores de los parámetros cuando se utilizan diferentes topologías.

Un aspecto interesante a notar en este experimento es que los mejores resultados los obtuvo la topología en estrella, cuando ésta había mostrado un pobre comportamiento en el primer experimento correspondiente a la comparación entre topologías (en la sección 5.2).

De tal manera, con este experimento se mostró empíricamente que existe una gran interacción entre la topología y los parámetros por omisión de PSO. También queda claro que se obtuvo un conjunto de parámetros para cada topología que indujo un buen comportamiento de PSO en todas las funciones de prueba simultáneamente.

### 5.5.3 Experimento 4c

A pesar de la experimentación realizada, aún persiste la pregunta: ¿Existe un conjunto de parámetros que presente un desempeño promedio aceptable de PSO sin importar la topología o la función de prueba que se pretenda optimizar?

Persiguiendo la respuesta de esta pregunta, se decidió utilizar nuevamente la metodología planteada variando la función objetivo, que a diferencia de los experimentos 4a y 4b, ahora estará dada

por la combinación de topologías y funciones, es decir:

$$\mathbf{FT}(\mathbf{x}) = \sum_{i=1, j=1}^{nm} f_i(x^*) t_i(x^*) \quad (5.6)$$

donde  $n = 9$ , es el número de funciones objetivo y  $m = 8$  es el número de topologías implementadas en el experimento. De la misma forma que los experimentos anteriores se ha normalizado cada uno de los valores de la función objetivo. Seleccionando el valor máximo y mínimo de todo el experimento.

### 5.5.3.1. Resultados

Los resultados del experimento se muestran en la tabla 5.21

Prueba $t$ $c_1-c_2-w$	Valores por RSM $c_1-c_2-w$	Aptitud RSM $c_1-c_2-w$	Apt. Norm. VPO 1.49 – 1.49 – 0.72
1.6 – 1.2 – 0.8	1.5893427870613 – 1.17710473300143 – 0.825	0.00594201532147192	0.00690041930666526
2 – 1 – 0.8	2.08575918363248 – 0.334575276514631 – 0.782353684826672	0.00607641662568923	0.00653524379321951
<b>2.4-1-0.8</b>	<b>2.36939895817006-0.968382849951114-0.794557130080742</b>	<b>0.00571967890015592</b>	<b>0.00717569352342433</b>
2.4 – 1.4 – 0.8	2.5093475579857 – 1.35837482847946 – 0.765116508710288	0.00586266054753414	0.00682759512683047
3 – 1.2 – 0.7	3.00543035302514 – 1.19512376483193 – 0.71642848636002	0.00580884177757228	0.00734878775190634

Tabla 5.21: Resultados de la prueba estadística  $t$  y RMS para el experimento 4c.

Los mejores valores arrojados por el diseño factorial completo y aceptados por la prueba estadística  $t$ , se muestran en la columna **Prueba  $t$** . Se puede concluir por medio de estos resultados que existe un conjunto de valores para los parámetros de PSO en los cuales el desempeño del algoritmo mejora. Podría establecerse valores para estos parámetros como:

$$c_2 = [1.0, 1.4]$$

$$w = [0.7, 0.8]$$

Mientras que  $c_2$  aún permanece variable.

Aunque la diferencia es mínima, se logra percibir que existen ciertos valores para los parámetros de PSO donde el algoritmo se comporta mejor ( $c_1 = 2.4$ ,  $c_2 = 1$ ,  $w = 0.8$ ). Para poder visualizar mejor este comportamiento, se puede ver la tabla 5.22 los valores individuales para cada una de las funciones para el mejor conjunto de parámetros calibrados con metodología propuesta. La topología a la que hace referencia la tabla 5.22 es el algoritmo de agrupamiento  $k$ -means, en ellas se puede

notar que el promedio de las funciones arrojados por PSO con sus parámetros calibrados con RSM superan el promedio de las funciones resueltas con el algoritmo de PSO con los valores por omisión. La única función que no mostró mejoría fue la función *Rastrigin* denotada como f07 en la tabla.

Función	Aptitud con RSM	VPO
f01	1159.20013078333	1971.8178485875
f02	15.8898679416667	20.2828288916667
f03	1400.94583333333	2230.67916666667
f04	123882.804893004	412749.068375617
f05	2316.6342875125	2870.3511779625
f06	-26288.2677246083	-18334.6560506542
f07	126.4683448625	125.874663654167
f08	8.90426654583333	9.91315940833333
f09	10.7958049875	17.9582681625

Tabla 5.22: Valores de las funciones objetivo despues de aplicar RSM en el experimento 4c. f01) *Sphere*, f02) *Schwefel*, f03) *Step*, f04) *Rosenbrock*, f05) *Rotated hyper-ellipsoid*, f06) *Schwefel 2.26*, f07) *Rastrigin*, f08) *Ackley* y f09) *Griewank*.

### 5.5.3.2. Conclusiones

Los resultados arrojados por PSO cuando se calibran sus parámetros por medio de la metodología propuesta fueron favorables, comparados con los resultados cuando los parámetros de este permanecen sin calibrar.

Con este experimento se logró obtener un conjunto de parámetros que brindan buenos resultados independientemente de las topología o función de prueba utilizada.



# 6

## Conclusiones y trabajo futuro

### 6.1 Conclusiones

En este trabajo se implementaron métodos estadísticos computacionales clásicos para calibrar los parámetros en AEs, particularmente, se implementaron métodos del diseño de experimentos (DOE), análisis de varianza (ANOVA), modelos de regresión y métodos de superficie de respuesta (RSM), con la finalidad de determinar si permitían la calibración adecuada de parámetros en AEs. En esta tesis se seleccionó la optimización mediante cúmulo de partículas (PSO) como algoritmo base.

Se llevaron a cabo una serie de experimentos, con la finalidad de configurar los parámetros de PSO y por lo tanto mejorar el desempeño promedio del algoritmo. Los resultados demostraron que PSO es altamente dependiente de sus parámetros (topología de comunicación,  $w$ ,  $c_1$  y  $c_2$ ).

El objetivo del primer experimento fue estudiar los efectos que una topología de comunicación puede tener sobre PSO. Se implementaron seis topologías de comunicación: anillo, malla, toroidal, completamente conectada, estrella y árbol; y dos algoritmos de agrupamiento: *k-means* y jerárquico. Así mismo, se implementaron tres versiones del algoritmo de PSO libres de parámetros: PSobb, PSobbexp y PSobbde, con la intención de comparar su desempeño en cada una de las topologías

propuestas.

Cada uno de los algoritmos se ejecutó 30 veces independientes en las nueve funciones de prueba con la finalidad de obtener un estadístico robusto de los resultados

Los resultados de este primer experimento demostraron que el desempeño de PSO recae fuertemente en la topología de interconexión. Este comportamiento fue opuesto a la versión de PSOBBD, la cual fue altamente independiente de la forma de comunicación entre partículas.

Un resultado muy interesante de este experimento, fue descubrir que el desempeño de PSO mejora significativamente cuando se implementa una topología en malla y toroidal. Contrario a lo que tradicionalmente se ha venido usando en PSO, cuando se utilizaron las topologías completamente conectada y estrella PSO presentó el peor comportamiento.

Por otro lado, PSOBDEXP fue la versión con peor desempeño y la que presentó mayor dependencia en topología, siendo únicamente la topología completamente conectada y en estrella, las que relativamente incrementaron su desempeño. Por otro lado, PSOBBD presentó una alta dependencia en la topología de comunicación. Sin embargo, análogamente a la versión básica de PSO, su desempeño mejoró cuando se implementaron las topologías malla y toroidal.

De acuerdo a este experimento, se pudo observar que el comportamiento de PSO depende significativamente de la topología de interconexión empleada. Sin embargo, es posible mejorar el desempeño de PSO si se calibran sus parámetros intrínsecos.

Basados en esta hipótesis, se llevó a cabo un segundo experimento con la finalidad de calibrar los parámetros de PSO. En este experimento se usó un diseño factorial completo con tres factores:  $c_1$  y  $c_2$ , con 6 niveles cada uno; y  $w$ , con 11 niveles. En total fueron 396 tratamientos. De la misma forma que el primer experimento, se implementaron las topologías de interconexión.

Los resultados obtenidos por este segundo experimento demostraron que las topologías en malla y toroidal siguieron siendo las mejores en desempeñarse cuando se calibran los parámetros con diseños factoriales. La topología en árbol fue la que obtuvo el peor desempeño. Sin embargo, se pudo notar que cada función de prueba mejoró sus resultados (aptitud) con respecto a los del primer experimento. En este experimento se pudo observar que la gran mayoría de las topologías implementadas mejoraron sus resultados.

El diseño factorial fue visto en este trabajo como un método de búsqueda a grano grueso sobre un conjunto de valores, mientras que RSM fue visto como un método de búsqueda de fina granularidad en el conjunto de valores arrojados por los diseños factoriales. Es decir, mientras que los diseños factoriales realizaban una búsqueda en los límites de la superficie, RSM realizaba una búsqueda en el interior de la superficie.

Como resultados de este tercer experimento obtuvimos valores calibrados de los parámetros de PSO que mejoraron significativamente su desempeño. La comparación de resultados entre el experimento 1 con respecto a los obtenidos en el experimento 3, demuestran que la metodología propuesta es capaz de obtener resultados favorables con respecto a los originales en PSO.

Es importante resaltar que todas las topologías de interconexión mejoraron su convergencia en todas las funciones de prueba, aún cuando estas habían comportado mal en los experimentos 1 y 2.

En general, con respecto a las topologías de interconexión, se puede concluir que la mejor forma de comunicar a las partículas dentro del cúmulo es superponiendo los vecinos de una partícula con otra, lo cual origina un nivel de redundancia factible en el proceso de búsqueda, que no cae en los extremos de una conexión completa (como es el caso de la topología completamente conectada) o bien escasamente conectada (como es el caso de la topología en estrella).

Por otro lado, los algoritmos de agrupamiento parecen ser una buena alternativa para mejorar la convergencia de PSO. Sin embargo, es posible mejorar su desempeño si se emplean diferentes mecanismos para generar los grupos (*p.ej.*, la forma de mover el centroide en el algoritmo de *k-means*).

En general, los diseños factoriales y la metodología de superficie de respuesta parecen ser alternativas eficaces cuando se requieren calibrar los parámetros de una meta-heurística. Sin embargo, con respecto a RSM, se pudo observar que es muy propensa a quedar atrapada en óptimos locales.

Otro aspecto interesante a considerar cuando se utiliza RSM, son las hipótesis de adecuación de los modelos de regresión, las cuales a veces se consideran muy subjetivas, sobre todo cuando se utilizan los enfoques del *P - value* y *F - statistic* para verificar la adecuación de los modelos lineales de regresión<sup>1</sup>.

---

<sup>1</sup>Para más información sobre esta discusión refierase a Montgomery [35].

## 6.2 Contribuciones obtenidas

Las principales contribuciones de este trabajo de tesis son las siguientes:

- Se realizó un estudio comparativo para evaluar el desempeño de PSO con respecto a otros enfoques libres de parámetros. En este estudio se implementaron diversas topologías de interconexión, que nos permitieron comparar diferentes enfoques. Colateralmente, de este estudio derivó que la topología de comunicación influye significativamente en el desempeño de PSO.
- Una metodología para calibrar los parámetros en AE, la cual inicia con un diseño factorial completo (vista como una búsqueda a grano grueso sobre el conjunto de valores establecido), continuando con una pruebas estadística  $t$  (la cual funciona como un filtro de mejores soluciones sobre todo el conjunto obtenido por el diseño factorial) y termina con una metodología de superficie de respuesta (vista como una búsqueda a grano grueso sobre el conjunto de valores arrojados por la prueba estadística  $t$ ).
- Se obtuvieron valores para los parámetros de PSO ( $w$ ,  $c_1$  y  $c_2$ ) los cuales trabajan bien para una función y una topología determinada.
- Se obtuvieron valores para los parámetros de PSO ( $w$ ,  $c_2$ ,  $c_2$ ) los cuales trabajan bien cuando se trata de optimizar una función de prueba en específico usando cualquiera de las topologías utilizadas en esta tesis.
- Se obtuvieron valores para los parámetros de PSO ( $w$ ,  $c_2$ ,  $c_2$ ) los cuales trabajan bien cuando se utiliza una topología de interconexión específica en cualquiera de las funciones de prueba utilizadas en esta tesis.
- Se obtuvieron valores para los parámetros de PSO ( $w$ ,  $c_2$ ,  $c_2$ ) los cuales trabajan bien independientemente de la topología o de la función de prueba (de las utilizadas en esta tesis).
- Se obtuvo un nuevo enfoque para comunicar las partículas en PSO el cual trabaja bastante bien en con varias funciones de prueba. Contrario a las topologías recomendadas tradicionalmente en

PSO (completamente conectada y estrella), la topología toroidal presenta el mejor desempeño mientras que las topologías completamente conectada y estrella presentan los peores resultados.

- Publicación del artículo titulado: *A Comparative Study of Neighborhood Topologies for Particle Swarm Optimizers*. ICEC 09, Madeira Portugal.

### 6.3 Trabajo futuro

Como trabajo a futuro se pueden distinguir ciertas rutas importantes:

- Se podría extender la calibración de parámetros usado en esta tesis a multi-objetivo, usando estadística computacional con la finalidad de verificar si existe un conjunto de parámetros robusto que sea válido para los ambos enfoques.
- En este trabajo se utilizó el algoritmo de PSO como motor principal de nuestros experimentos. Sin embargo, existen otros AEs que podrían implementarse con la finalidad de verificar la robustez del método propuesto.
- Sería interesante desarrollar un mecanismo, que permita auto-adaptar los parámetros de un AE, basado en la metodología de superficie de respuesta.
- Sería interesante revisar e implementar algunos mecanismos de elitismo que puedan ser implementados en RSM para mejorar la calibración de parámetros.
- De igual forma, se podría buscar algún mecanismo que permita salir de un óptimo local cuando la superficie de respuesta se implementa.
- Revisar otras técnicas estadísticas para la calibración de parámetros, para comparar su eficiencia con la metodología propuesta en este trabajo.





## Diseños factoriales $2^2$

Consideremos la conversión (respuesta) de un proceso químico donde intervienen dos factores, el efecto de concentración de reacción y la cantidad del catalizador. Denotemos la concentración de reacción como  $A$  con dos niveles de interés: 15 y 25 por ciento; y denotemos el catalizador como  $B$  con dos niveles: 2 y 1 libras. Los datos mostrados en la tabla A.1 muestran el experimento replicado tres veces.

Factor		Tratamiento	Réplica			Total
$A$	$B$	Combinación	I	II	III	
-	-	A bajo, B bajo	28	25	27	80
+	-	A alto, B bajo	36	32	32	100
-	+	A bajo, B alto	18	19	23	60
+	+	A alto, B alto	31	30	29	90

Tabla A.1: Datos del diseño factorial  $2^2$

La combinación de tratamientos de este diseño se muestra en la figura A.1. Por convención, se denotará el efecto del factor  $A$  por " $A$ ", el efecto del factor  $B$  por " $B$ " y la interacción entre ambos factores  $AB$  como " $AB$ ". En un diseño factorial  $2^2$ , el nivel alto y bajo de los factores  $A$  y  $B$  son denotados por "-" y "+", respectivamente. Entonces en el eje  $A$ , "-" representa el nivel bajo de

concentración (15 %) y “+” el nivel alto (25 %). Para el eje B, el nivel bajo del catalizador esta representado por “-” y el nivel alto por “+”.

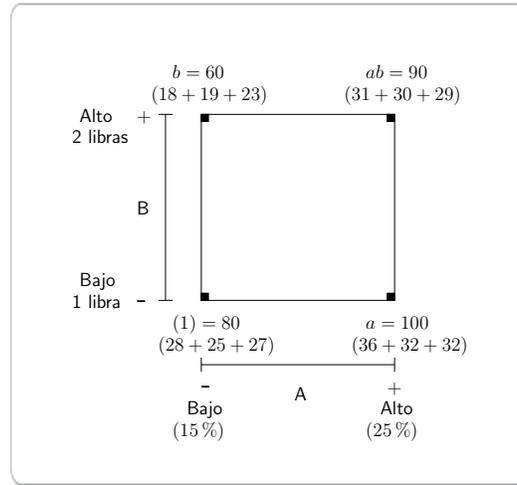


Figura A.1: Combinación de tratamientos en el diseño  $2^2$

Las combinaciones de los cuatro tratamientos se representan mediante letras minúsculas (como en la figura A.1). La combinación del tratamiento en su nivel alto se representa por su correspondiente letra minúscula, y en su nivel bajo por la ausencia de la misma. Entonces,  $a$  representa la combinación del tratamiento del factor  $A$  en su nivel alto,  $b$  la combinación del tratamiento del factor  $B$  en su nivel alto y  $ab$  representa ambos factores en su nivel alto. Por convención,  $(1)$  denota ambos factores en su nivel bajo y los símbolos  $(1)$ ,  $a$ ,  $b$ ,  $ab$  representan el total de las  $n$  réplicas tomadas en la combinación de tratamientos.

El efecto de  $A$ , en el nivel bajo de  $B$  es  $[a - (1)]/n$  y el efecto de  $A$ , en el nivel alto de  $B$  es  $[ab - b]/n$ , promediando estas cantidades el **efecto principal** de  $A$  está dado por:

$$\begin{aligned} A &= \frac{1}{2n} \{ [ab - b] + [a - (1)] \} \\ &= \frac{1}{2n} [ab + a - b - (1)] \end{aligned} \quad (\text{A.1})$$

El efecto principal promedio de  $B$ , se obtiene a partir del efecto de  $B$  en el nivel bajo de  $A$  (v.g.,

$[b - (1)]/n$ ) y en el nivel alto de  $A$  (v.g.,  $[ab - a]/n$ ), es decir:

$$\begin{aligned} B &= \frac{1}{2n} \{ [ab - a] + [b - (1)] \} \\ &= \frac{1}{2n} [ab + b - a - (1)] \end{aligned} \quad (\text{A.2})$$

El **efecto de interacción**  $AB$ , es la diferencia promedio entre el efecto de  $A$  en el nivel alto de  $B$  y el efecto de  $A$  en el nivel bajo de  $B$ , esto es:

$$\begin{aligned} AB &= \frac{1}{2n} \{ [ab - b] - [a - (1)] \} \\ &= \frac{1}{2n} [ab + (1) - a - b] \end{aligned} \quad (\text{A.3})$$

Similarmente a la ecuación (A.3), se puede definir  $AB$  como la diferencia promedio entre el efecto de  $B$  en el nivel alto de  $A$  y el efecto de  $B$  en el nivel bajo de  $A$ .

Aunque, las fórmulas de los efectos de  $A$ ,  $B$  y  $AB$  podrían derivarse de otra forma, v.g., el efecto de  $A$  puede obtenerse de la diferencia en la respuesta promedio de las dos combinaciones de tratamientos, en el lado derecho del cuadro de la figura A.1 (llamado  $\bar{y}_{A+}$ , por que es la respuesta promedio en las combinaciones de tratamientos cuando  $A$  está en su nivel alto) y las dos combinaciones de tratamientos, en el lado izquierdo del cuadro de la figura (o  $\bar{y}_{A-}$ ), es decir:

$$\begin{aligned} A &= \bar{y}_{A+} - \bar{y}_{A-} \\ &= \frac{ab + a}{2n} - \frac{b + (1)}{2n} \\ &= \frac{1}{2n} [ab + a - b - (1)] \end{aligned}$$

Este es exactamente el mismo resultado de la ecuación (A.1). Similarmente, se puede obtener el efecto de  $B$  a partir de la figura A.1, como la diferencia entre el promedio de las dos combinaciones de tratamientos en la parte superior del cuadro ( $\bar{y}_{B+}$ ) y el promedio de las dos combinaciones del

tratamiento en la parte inferior del cuadro ( $\bar{y}_{B-}$ ), esto es:

$$\begin{aligned} B &= \bar{y}_{B+} - \bar{y}_{B-} \\ &= \frac{ab + b}{2n} - \frac{a + (1)}{2n} \\ &= \frac{1}{2n}[ab + b - a - (1)] \end{aligned}$$

Finalmente, el efecto de interacción  $AB$  es el promedio de las combinaciones de tratamientos en diagonal, de derecha a izquierda en la figura A.1 (v.g.,  $[ab$  y  $(1)]$ ), menos el promedio de las combinaciones de tratamientos en diagonal, de izquierda a derecha en la misma figura, esto es:

$$\begin{aligned} AB &= \frac{ab + (1)}{2n} - \frac{a + b}{2n} \\ &= \frac{1}{2n}[ab + (1) - a - b] \end{aligned}$$

lo cual es igual a la ecuación (A.3).

Usando el experimento en la figura A.1 se pueden estimar los efectos promedios, como:

$$\begin{aligned} A &= \frac{1}{2(3)}(90 + 100 - 60 - 80) = 8.33 \\ A &= \frac{1}{2(3)}(90 + 60 - 100 - 80) = -5.00 \\ AB &= \frac{1}{2(3)}(90 + 80 - 100 - 60) = 1.67 \end{aligned}$$

Además, podemos observar que el efecto de  $A$  (concentración de reacción) es positivo y sugiere que al incrementar  $A$  desde el nivel bajo (15 %) al nivel alto (25 %) se puede incrementar la respuesta. Por otro lado, efecto de  $B$  (catalizador) es negativo y sugiere que incrementando la cantidad de catalizador se puede decrementar la respuesta. Además, el efecto de interacción parece ser pequeño.

Se puede examinar la magnitud y la dirección de los efectos de los factores para determinar que variable probablemente es la más importante. Generalmente, se puede usar el análisis de varianza para obtener esta interpretación.

A partir de la ecuación (A.1), notemos que se usa un contraste para estimar  $A$ , usualmente llamado **efecto total** de  $A$ .

$$\text{Contraste}_A = ab + a - b - (1) \quad (\text{A.4})$$

De igual forma, existen contrastes para las ecuaciones A.2 y A.3. Es importante señalar que los tres contrastes son **ortogonales**, lo que hace característico a los modelos  $2^2$  y en general a los diseños  $2^k$ .

La suma de cuadrados para cualquier contraste puede obtenerse elevando al cuadrado el contraste y dividirlo por el número de observaciones:

$$SS_A = \frac{[ab + a - b - (1)]^2}{4n} \quad (\text{A.5})$$

$$SS_B = \frac{[ab + b - a - (1)]^2}{4n} \quad (\text{A.6})$$

$$SS_{AB} = \frac{[ab + (1) - a - b]^2}{4n} \quad (\text{A.7})$$

Se puede obtener la suma de cuadrados de las ecuaciones A.5, A.6 y A.7, usando el experimento de la figura A.1, como:

$$SS_A = \frac{(50)^2}{4(3)} = 208.33$$

$$SS_B = \frac{(-30)^2}{4(3)} = 75.00 \quad (\text{A.8})$$

$$SS_{AB} = \frac{(10)^2}{4(3)} = 8.33$$

Por lo tanto, la suma total de cuadrados es:

$$SS_T = \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^n y_{ijk}^2 - \frac{y_{..}^2}{4n} \quad (\text{A.9})$$

En general,  $SS_T$  tiene  $4n - 1$  grados de libertad, por lo tanto la suma de cuadrados del error, con  $4(n - 1)$  grados de libertad, se obtiene a partir de una sustracción:

$$SS_E = SS_T - SS_A - SS_B - SS_{AB} \quad (\text{A.10})$$

Y a partir del experimento de la figura A.1 obtenemos:

$$\begin{aligned} SS_T &= \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^n y_{ijk}^2 - \frac{y_{..}^2}{4(3)} \\ &= 9398.00 - 9075.00 = 323.00 \end{aligned}$$

y usando  $SS_A$ ,  $SS_B$  y  $SS_{AB}$  de la ecuación (A.8), tenemos:

$$\begin{aligned} SS_E &= SS_T - SS_A - SS_B - SS_{AB} \\ &= 323.00 - 208.33 - 75 - 00 - 8.33 \\ &= 31.34 \end{aligned}$$

Basados en el  $P$ -value, se puede concluir que los efectos principales son estadísticamente significantes y que no existe interacción entre factores. Esto confirma la interpretación inicial de los datos basada en la magnitud de los efectos de los factores.

# B

## Ejemplo de la metodología de superficie de respuesta

Un ejemplo sencillo que utiliza la metodología de superficie de respuesta puede encontrarse en [35], el cual detallamos a continuación.

Un ingeniero químico está interesado en determinar las condiciones de operación que maximicen la respuesta de un proceso, el cual se ve afectado por dos variables controlables: tiempo de reacción y temperatura de reacción. Actualmente el ingeniero opera el proceso con un tiempo de reacción de 35 minutos y una temperatura de  $155^{\circ}F$ , lo cual arroja un rendimiento al rededor del 40%. Ya que es improbable encontrar el óptimo en esta región, primero se aplicará un modelo de primer orden (descrito en la sección 3.4.1.3) y posteriormente el método de ascenso empinado (descrito en la sección 4.3.2).

El ingeniero decide que la región de exploración para ajustar el modelo de primer orden puede estar en el rango de  $[30, 40]$  minutos de tiempo de reacción y  $[150, 160]^{\circ}F$  de temperatura de reacción. Para simplificar los cálculos, las variables independientes se codifican en el intervalo usual de  $[-1, 1]$ , entonces si  $\xi_1$  denota la **variable natural de tiempo** y  $\xi_2$  denota la **variable natural de**

**temperatura**, las variables codificadas son:

$$x_1 = \frac{\xi_1 - 35}{5} \quad y \quad x_2 = \frac{\xi_2 - 155}{5}$$

El diseño experimental del proceso se muestra en la tabla B.1. Notemos que el diseño utilizado para recolectar los datos es un diseño factorial  $2^2$  aumentado con cinco puntos centrales, estos diseños están ubicados al rededor de las condiciones de operación actuales del proceso, las réplicas en el centro se usan para estimar el error experimental y verificar la adecuación del modelo de primer orden.

Variables Naturales		Variables codificadas		Respuesta
$\xi_1$	$\xi_2$	$x_1$	$x_2$	$y$
30	150	-1	-1	39.3
30	160	-1	1	40.0
40	150	1	-1	40.9
40	160	1	1	41.5
35	155	0	0	40.3
35	155	0	0	40.5
35	155	0	0	40.7
35	155	0	0	40.2
35	155	0	0	40.6

Tabla B.1: Datos del proceso para ajustar el modelo de primer orden

Un modelo de primer orden con dos variables de diseño (como el mostrado en la ecuación (4.4)) puede ser ajustado usando el método de mínimos cuadrados (descrito en la sección 4.4), con el cual obtenemos el modelo de la ecuación (B.1) expresado en variables codificadas.

$$\hat{y} = 40.44 + 0.775x_1 + 0.325x_2 \quad (\text{B.1})$$

El modelo de primer orden asume que las variables  $x_1$  y  $x_2$  tienen un efecto aditivo en la respuesta, entonces la interacción entre variables podría representarse por el coeficiente  $\beta_{12}$ , el cual expresa que un término de productos cruzados,  $x_1x_2$  fue agregado al modelo. El estimado de mínimos cuadrados de este coeficiente es sólo una mitad del efecto de interacción, calculado como en un diseño factorial

ordinario  $2^2$ , o

$$\begin{aligned}\widehat{\beta}_{12} &= \frac{1}{4}[(1 \times 39.3) + (1 \times 41.5) + (-1 \times 40.0) + (-1 \times 40.9)] \\ &= \frac{1}{4}(-0.1) \\ &= -0.025\end{aligned}$$

(B.2)

Para verificar la adecuación del modelo de primer orden recurrimos a la prueba estadística  $F$  (descrita en la sección 4.5) donde se rechaza la hipótesis nula  $H_0$  si excede a  $F_{\alpha, k, n-k-1}$ . Alternativamente se puede usar el enfoque  $P$ -value, para verificar la hipótesis y entonces rechazar  $H_0$  si el  $P$ -value para el estadístico  $F_0$  es menor que  $\alpha$ , el valor de  $\alpha$  puede ser extraído de [35].

Si  $F_0$  es mayor a  $F(\alpha, k, n - k - 1)$ , entonces el modelo de primer orden es el correcto y no hay razón para cuestionar la adecuación del modelo de primer orden, por lo tanto se busca un camino a partir del centro de diseño (el punto  $(x_1 = 0, x_2 = 0)$ ) que nos permita descender rápidamente. Podemos movernos 0.775 unidades en la dirección de  $x_1$  por cada 0.325 unidades en la dirección de  $x_2$ , lo que significa que la ruta de descenso más rápido pasa a través del punto  $(x_1 = 0, x_2 = 0)$  con una pendiente de  $0.325/0.775$ . El ingeniero decide usar 5 minutos de tiempo de reacción como tamaño de paso básico. Usando la relación entre  $\xi_1$  y  $x_1$ , se puede ver que 5 minutos de tiempo de reacción es equivalente a un paso, en la variable codificada  $x_1$  de  $\Delta x_1 = 1$ . Por lo tanto, los pasos a lo largo de la ruta de más rápido descenso son  $\Delta x_1 = 1.0000$  y  $\Delta x_2 = (0.325/0.775) = 0.42$ . Un algoritmo sencillo para determinar las coordenadas de un punto, en la ruta de más rápido descenso, puede observarse en la sección 4.3.2.

Posteriormente, el ingeniero genera puntos a lo largo de esta ruta hasta observar un decremento en la respuesta. Los resultados de este procedimiento se muestran en la tabla B.2 junto con sus variables codificadas y naturales. Aunque matemáticamente las variables codificadas son fáciles de manipular, las variables naturales pueden usarse en un proceso de ejecución.

Pasos	Variables codificadas		Variables naturales		Respuesta
	$\xi_1$	$\xi_2$	$x_1$	$x_2$	$y$
Origen	0	0	35	155	
$\Delta$	1.00	0.42	5	2	
Origen + $\Delta$	1.00	0.42	40	157	41.0
Origen +2 $\Delta$	2.00	0.84	45	159	42.9
Origen +3 $\Delta$	3.00	1.26	50	161	47.1
Origen +4 $\Delta$	4.00	1.68	55	163	49.7
Origen +5 $\Delta$	5.00	2.10	60	165	53.8
Origen +6 $\Delta$	6.00	2.52	65	167	59.9
Origen +7 $\Delta$	7.00	2.94	70	169	65.0
Origen +8 $\Delta$	8.00	3.36	75	171	70.4
Origen +9 $\Delta$	9.00	3.78	80	173	77.6
Origen +10 $\Delta$	10.00	4.20	85	175	80.3
Origen +11 $\Delta$	11.00	4.62	90	179	76.2
Origen +12 $\Delta$	12.00	5.04	95	181	75.1

Tabla B.2: Datos del proceso para ajustar el modelo de primer orden

Se pueden observar diez incrementos a través de la ruta y en el onceavo punto un decremento, entonces otro modelo de primer orden es ajustado en las vecindades generales de este punto ( $\xi_1 = 85, \xi_2 = 175$ ), donde la región de exploración para  $\xi_1$  es  $[80, 90]$  y para  $\xi_2$  es  $[170, 180]$ , entonces las variables codificadas son:

$$\xi_1 = \frac{\xi_1 - 85}{5} \quad \text{y} \quad \xi_2 = \frac{\xi_2 - 175}{5}$$

Nuevamente, se implementa un diseño factorial  $2^2$  con cinco puntos agregados en el centro, los datos del diseño experimental se muestra en la tabla B.3, con los cuales se obtiene el siguiente modelo ajustado de primer orden para las variables codificadas:

$$\hat{y} = 78.97 + 1.00x_1 + 0.50x_2$$

Nuevamente, la adecuación del modelo de primer orden se determina con la prueba estadística  $F$ . Si esta prueba falla, significa que el modelo no es una adecuada aproximación y probablemente existe curvatura en la superficie verdadera, lo cual podría indicar que nos encontramos cerca del óptimo, entonces se necesita un análisis adicional que nos permita localizar el óptimo con más precisión.

Variables Naturales		Variables codificadas		Respuesta
$\xi_1$	$\xi_2$	$x_1$	$x_2$	$y$
80	170	-1	-1	76.5
80	180	-1	1	77.0
90	170	1	-1	78.0
90	180	1	1	79.5
85	175	0	0	79.9
85	175	0	0	80.3
85	175	0	0	80.0
85	175	0	0	79.7
85	175	0	0	79.8

Tabla B.3: Datos para el segundo modelo de primer orden.

Cuando nos encontramos relativamente cerca del óptimo, un modelo que incorpore curvatura puede ser implementado para aproximar la respuesta, en muchos casos el modelo de segundo orden (mostrado en la ecuación (4.5)) puede ser el adecuado.

No es posible ajustar el modelo de segundo orden con dos variables,  $x_1$  y  $x_2$ , con los datos de diseño de la tabla B.3, por lo tanto es necesario aumentar los datos del diseño con suficientes puntos para ajustar el modelo de segundo orden, para lo cual se obtienen cuatro observaciones adicionales en  $(x_1 = 0, x_2 = \pm 1.414)$  y  $(x_1 = \pm 1.414, x_2 = 0)$ . Los datos de este experimento se muestran en la tabla B.4.

El modelo de segundo orden será ajustado usando el diseño de composición central (CCD). Este modelo ha sido explicado en la sección 4.6.2.

Basados en la prueba estadística  $F$  se decide si el modelo de segundo orden ha sido ajustado correctamente. Si el modelo ha sido ajustado correctamente, significa que es una aproximación adecuada a la superficie de respuesta verdadera y entonces procedemos a localizar el punto estacionario usando la solución general en la ecuación (4.9). El método para localizar el punto estacionario se describe en la sección 4.3.3.

Notemos que:

$$\mathbf{b} = \begin{bmatrix} 0.995 \\ 0.515 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} -1.376 & 0.1250 \\ 0.1250 & -1.001 \end{bmatrix}$$

Variables Naturales		Variables codificadas		Respuesta
$\xi_1$	$\xi_2$	$x_1$	$x_2$	$y$
80	170	-1	-1	76.5
80	180	-1	1	77.0
90	170	1	-1	78.0
90	180	1	1	79.5
85	175	0	0	79.9
85	175	0	0	80.3
85	175	0	0	80.0
85	175	0	0	79.7
85	175	0	0	79.8
92.07	175	1.414	0	78.4
77.93	175	-1.414	0	75.6
85	182.07	0	1.414	78.5
85	167.93	0	-1.414	77.0

Tabla B.4: Datos para el diseño de composición central.

y a partir de la ecuación (4.9) el punto estacionario es:

$$\mathbf{x}_s = -\frac{1}{2}\mathbf{B}^{-1}\mathbf{b} = -\frac{1}{2} \begin{bmatrix} -0.7345 & -0.0917 \\ -0.0917 & -1.0096 \end{bmatrix} \begin{bmatrix} 0.995 \\ 0.515 \end{bmatrix} = \begin{bmatrix} 0.389 \\ 0.306 \end{bmatrix}$$

Esto es,  $x_{1,s} = 0.389$  y  $x_{2,s} = 0.306$ . En términos de las variables naturales, el punto estacionario es:

$$0.389 = \frac{\xi_1 - 85}{5} \quad 0.306 = \frac{\xi_2 - 175}{5}$$

entonces la respuesta  $\xi_1 = 86.95 \simeq 87$  minutos de tiempo de reacción y  $\xi_2 = 176.53 \simeq 176.5^\circ F$ . Usando la ecuación (4.10) se puede encontrar la respuesta predicha en el punto estacionario como  $\hat{y}_s = 80.21$  y posteriormente usar el análisis canónico (descrito en la sección 4.3.3) para caracterizar la superficie de respuesta. El primer paso para realizar este análisis, es expresar el modelo ajustado en forma canónica, de acuerdo a la ecuación (4.11), donde los *eigen-valores*  $\lambda_1$  y  $\lambda_2$  son las raíces de la ecuación:

$$|\mathbf{B} - \lambda\mathbf{I}| = 0$$

$$\begin{vmatrix} -1.376 - \lambda & 0.1250 \\ 0.1250 & -1.001 - \lambda \end{vmatrix} = 0$$

lo cual se reduce a

$$\lambda^2 + 2.3788\lambda + 1.3639 = 0$$

Las raíces de la ecuación cuadrática son  $\lambda_1 = -0.9641$  y  $\lambda_2 = -1.4147$ , entonces la forma canónica del modelo ajustado es

$$\hat{y} = 80.21 - 0.9641w_1^2 - 1.4147w_2^2$$

Ya que  $\lambda_1$  y  $\lambda_2$  son negativos y el punto estacionario esta dentro de la región de exploración, podemos concluir que el punto estacionario es un punto de máxima respuesta.

Para una descripción más detallada de los métodos mencionados en este ejemplo referirse a Montgomery [35].





## Funciones de prueba

A continuación se muestra la definición matemática de cada problema de prueba utilizado:

A. Función *sphere*, definida como:

$$f(\mathbf{x}) = \sum_{i=1}^{N_d} x_i^2,$$

donde  $\mathbf{x}^* = 0$  y  $f(\mathbf{x}^*) = 0$  para  $-100 \leq x_i \leq 100$

B. Problema de *Schwefel*, definido como:

$$f(\mathbf{x}) = \sum_{i=1}^{N_d} |x_i| + \prod_{i=1}^{N_d} |x_i|,$$

donde  $\mathbf{x}^* = 0$  y  $f(\mathbf{x}^*) = 0$  para  $-10 \leq x_i \leq 10$

C. Función *step*, definida como:

$$f(\mathbf{x}) = \sum_{i=1}^{N_d} (\lfloor x_i + 0.5 \rfloor)^2,$$

donde  $\mathbf{x}^* = 0$  y  $f(\mathbf{x}^*) = 0$  para  $-100 \leq x_i \leq 100$

D. Función de *Rosenbrock*, definida como:

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

donde  $\mathbf{x}^* = (1, 1, \dots, 1)$  y  $f(\mathbf{x}^*) = 0$  para  $-30 \leq x_i \leq 30$

E. Función *rotated hyper-ellipsoid*, definida como:

$$f(\mathbf{x}) = \sum_{i=1}^{N_d} (\sum_{j=1}^i x_j)^2,$$

donde  $\mathbf{x}^* = 0$  y  $f(\mathbf{x}^*) = 0$  para  $-100 \leq x_i \leq 100$

F. Problema generalizado de *Schwefel* 2.26, definido como:

$$f(\mathbf{x}) = - \sum_{i=1}^{N_d} (x_i \text{sen}(\sqrt{|x_i|})),$$

donde  $\mathbf{x}^* = (420.9687, \dots, 420.9687)$  y  $f(\mathbf{x}^*) = -4426.407721$  para  $-500 \leq x_i \leq 500$

G. Función de *Rastrigin*, definida como:

$$f(\mathbf{x}) = - \sum_{i=1}^{N_d} (x_i^2 - 10\cos(2\pi x_i) + 10),$$

donde  $\mathbf{x}^* = 0$  y  $f(\mathbf{x}^*) = 0$  para  $-5.12 \leq x_i \leq 5.12$

H. Función de *Ackley*, definida como:

$$f(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{N_d} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{N_d} \cos(2\pi x_i)\right) + 20 + e,$$

donde  $\mathbf{x}^* = 0$  y  $f(\mathbf{x}^*) = 0$  para  $-32 \leq x_i \leq 32$

I. Función de *Griewank*, definida como:

$$f(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^{N_d} x_i^2 - \prod_{i=1}^{N_d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

donde  $\mathbf{x}^* = 0$  y  $f(\mathbf{x}^*) = 0$  para  $-600 \leq x_i \leq 600$

# D

## Gráficas de caja

Las **gráficas de caja** (o *Box Plots* en Inglés) mostradas en la figura D.1, permiten visualizar la distribución de una muestra a través de la información de cuartiles. Los **cuartiles** son los tres valores que dividen una muestra en cuatro partes porcentualmente iguales, comúnmente  $Q_1$ ,  $Q_2$  o mediana y  $Q_3$ . El primer cuartil, es el valor en el cual o por debajo del cual queda un cuarto (25 %) de todos los valores de una muestra; y el tercer cuartil, es el valor en el cual o por debajo del cual quedan las tres cuartas partes (75 %) de la muestra.

Además de la información de cuartiles de una muestra, las gráficas de caja permiten visualizar de manera compacta los valores atípicos, y los valores mínimo y máximo alcanzados por cada una de las muestras, debido a ello se les considera excelentes herramientas para comparar diferentes grupos de datos.

En este trabajo se desea comparar el desempeño de diversos algoritmos de optimización basados en PSO, para ello se utilizan las gráficas de caja. Supongamos que la figura D.1 muestra las cajas de desempeño mostrado en un problema de minimización por dos algoritmos de optimización. La caja mostrada en el lado izquierdo de la figura presentan una muestra cuyos datos son más similares entre sí que los datos representados por la caja derecha. De tal manera, se puede decir que el algoritmo

representado por la caja izquierda presentó un comportamiento robusto<sup>1</sup>. No obstante, sus mejores resultados son inferiores a los mejores resultados mostrados por el segundo algoritmo, a pesar que la distribución de los datos de este último se presenta en un rango más amplio (que indica que tuvo un comportamiento errático).

De esta sección en adelante, se considera que un algoritmo obtiene un buen desempeño si este se aproxima al mejor valor reportado en la literatura, para lo cual se han considerado únicamente problemas de minimización.

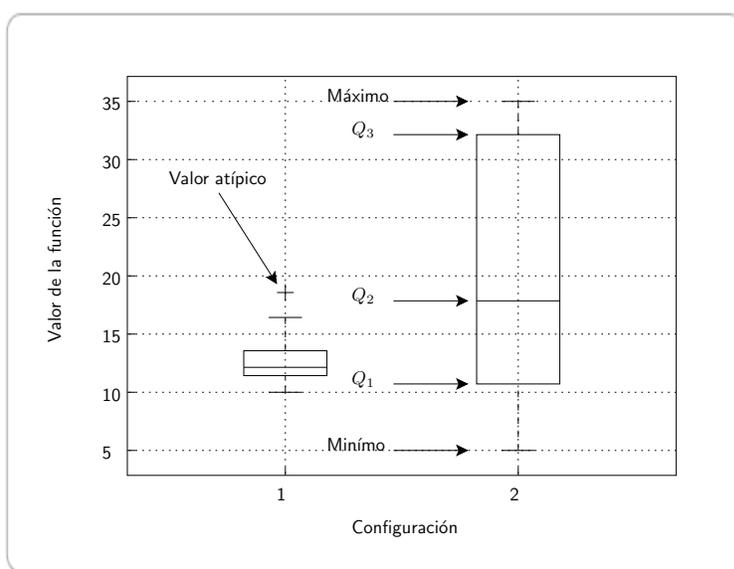


Figura D.1: Gráfica de cajas.

<sup>1</sup>En este caso el término robusto se refiere a que la ejecución repetida de los algoritmos produce sistemáticamente soluciones similares entre sí.

# Bibliografía

- [1] H. Abbass. The self-adaptive pareto differential evolution algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 831–836, 2002.
- [2] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.
- [3] Wolfgang Banzhaf, Peter Nordin, Robert E. Keller, and Frank D. Francone. *Genetic Programming-An Introduction; On the Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann, dpunkt.verlag, January 1998.
- [4] Thomas Bartz-Beielstein. Experimental analysis of evolution strategies-overview and comprehensive introduction. In *Interner Bericht des Sonderforschungsbereichs 531 Computational Intelligence CI-157/03*, Universitat Dortmund, Germany, 2003.
- [5] Thomas Bartz-Beielstein. Experimental research in evolutionary computation. In G. Rozenberg, editor, *The New Experimentalism, Natural Computing*. Springer, Germany, 2006. ISBN 3-540-32026-1.
- [6] Thomas Bartz-Beielstein, Christian W. G. Lasarczyk, and Mike Preuss. Secuential parameter optimization. In *IEEE Congress on Evolutionary Computation, 2005*, volume 1, pages 773–780, Edinburgh, Scotland, September 2005.
- [7] Thomas Bartz-Beielstein and Sandor Markon. Tuning search algorithms for real-world applications: A regression tree based approach. In *Congress on Evolutionary Computation, 2004. CEC2004*, volume 1, pages 1111–1118, June 2004.
- [8] Thomas Bartz-Beielstein, K. E. Parsopoulos, and M.Ñ. Vrahatis. Analysis of particle swarm optimization using computational statistics. In *Proceedings of the International Conference of Numerical Analysis and Applied Mathematics, ICNAAM 2004*, pages 34–37, 2004.

- [9] Thomas Bartz-Beielstein, K. E. Parsopoulos, and M.Ñ. Vrahatis. Design and analysis of optimization algorithms using computational statistics. *Applied Numerical Analysis and Computational Mathematics*, 1(2):413–433, 2004.
- [10] C. J. A. Bastos-Filho, M. P. Caraciolo, P. B. C. Miranda, and D. F. Carvalho. Multi-ring dispersed particle swarm optimization. In *Eighth International Conference on Hybrid Intelligent Systems, 2008. HIS '08*, pages 25–30, September 2008.
- [11] G. E. P. Box, W. G. Hunter, and J. S. Hunter. *Statistics for Experimenters*. Wiley, New York NY, 1978.
- [12] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Monterrey CA, 1984.
- [13] Bill P. Buckles and Fred E. Petry. *Genetic Algorithms*. IEEE Computer Society Press, 1992.
- [14] M. Clerc and J. Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.
- [15] N. L. Cramer. A representation for the adaptive generation of simple. In *Proceedings of an International Conference on Genetic*, pages 183–187, Pittsburgh, PA, 1985. Carnegie-Mellon University.
- [16] A. Czarn, C. MacNish, K. Vijayan, and B. Turlach. Statistical exploratory analysis of genetic algorithms: The importance of interaction. In *Congress on Evolutionary Computation, 2004. CEC2004*, volume 2, pages 2288–2295, June 2004.
- [17] Charles Robert Darwin. *On the Origin of Species by Means of Natural Selection Or the Preservation of Favoured Races in the Struggle for Life*. Cambridge University Press, UK, sixth edition, 1964. Originally published in 1859.

- [18] A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, July 1999.
- [19] R. A. Fisher. *The Design of Experiments*. Oliver and Boyd, Edinburgh, 1935.
- [20] Lawrence J. Fogel. *On the organization of intellect*. PhD thesis, University of California, Los Angeles, California, 1964.
- [21] M. Gardner. Mathematical games: On the remarkable császár polyhedron and its applications in problem solving. In *Sci. Amer*, pages 102–107, May 1975.
- [22] J. J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics, SMC*, 16:122–128, 1986.
- [23] Nikolaus Hansen. The cma evolution strategy: A comparing review. In *Towards a New Evolutionary Computation*, Studies in Fuzziness and Soft Computing, pages 75–102. Springer Berlin / Heidelberg, 2006.
- [24] A. Hoffman. *Arguments on Evolution: A Paleontologist's Perspective*. Oxford University Press, New York, 1989.
- [25] John H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, Massachusetts, second edition, 1992.
- [26] Wei Jian, Yuncan Xue, and Jixin Qian. Improved particle swarm optimization algorithms study based on the neighborhoods topologies. In *The 30th Annual Conference of the IEEE Industrial Electronics Society, 2004. IECON 2004*, volume 3, pages 2192–2196, Busan, Korea, November 2004.
- [27] J. Kennedy. Small worlds and mega-mind: Effects of neighborhood topology on particle swarm performance. In *Proceedings of the IEEE Congress on Evolutionary Computation, 1999. CEC 99*, volume 3, pages 1931–1938, Washington, DC, USA, 1999.

- [28] J. Kennedy. Bare bones particle swarms. In *Proceedings of the IEEE Swarm Intelligence Symposium, 2003. SIS '03*, pages 80–87, Piscataway, NJ, April 2003. IEEE Press.
- [29] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 1942–1948, Piscataway, NJ, 1995. IEEE Press.
- [30] J. Kennedy and R. Mendes. Population structure and particle performance. In *Proceedings of the IEEE Congress on Evolutionary Computation, 2002. CEC '02*, pages 1671–1676, Washington, DC, USA, 2002. IEEE Computer Society.
- [31] James Kennedy and Russell C. Eberhart, editors. *Swarm Intelligence*. Morgan Kaufmann, San Francisco, California, 2001.
- [32] J. P. C. Kleijnen. *Statistical Tools for Simulation Practitioners*. Marcel Dekker, New York NY, 1987.
- [33] John R. Koza. Hierarchical genetic algorithms operating on populations of computer programs. In *Proceedings of 11th International Joint Conference on Artificial Intelligence*, pages 768–774, San Mateo, California, 1989. Morgan Kaufmann.
- [34] Gregor Johann Mendel. Experiments in plant hybridisation. In *Journal of the Royal Horticultural Society*, page 1–32, 1901. Traducción al inglés de un artículo publicado originalmente en 1865.
- [35] Douglas C. Montgomery. *Design and analysis of experiments*. John Wiley, USA, fifth edition, 2001. ISBN 0-471-31649-0.
- [36] Max D. Morris. The design and analysis of computer experiments. thomas j. santner and brian j. williams and william i. notz. *Journal of the American Statistical Association*, 99:1203–1204, December 2004.
- [37] R. H. Myers and Douglas C. Montgomery. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley, New York, 1995.

- [38] Volker Nannen and A. E. Eiben. Relevance estimation and value calibration of evolutionary algorithm parameters. In *Proceedings of the Joint International Conference for Artificial Intelligence (IJCAI)*, pages 975–980, 2007.
- [39] M. Omran, A. Engelbrecht, and A. Salman. Bare bones differential evolution. *European Journal of Operational Research*, 196(1):128–139, 2008.
- [40] M. Omran, A. Salman, and A. Engelbrecht. Self-adaptive differential evolution. In *Lecture Notes in Artificial Intelligence*, pages 192–199, 2005.
- [41] Q. T. Pham. Using statistical analysis to tune an evolutionary algorithm for dynamic optimization with progressive step reduction. *Computers and Chemical Engineering*, 31(11):1475 – 1483, 2007.
- [42] Shi and R. Eberhart. A modified particle swarm optimizer. In *Proceedings of the IEEE Congress on Evolutionary Computation, 1998*, pages 69–73, Anchorage, AK, USA, May 1998.
- [43] S. K. Smit and A. E. Eiben. Comparing parameter tuning methods for evolutionary algorithms. In *IEEE Congress on Evolutionary Computation 2009, CEC 2009*, pages 399–406, May 2009.
- [44] Inc. Stat-Ease. Stat-ease. statistics mode easy, August 2009. [www.statease.com/articles.html](http://www.statease.com/articles.html).
- [45] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, International Computer Science Institute, 1995.
- [46] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):359–431, 1997.
- [47] F. van den Bergh. *An Analysis of Particle Swarm Optimizers*. PhD thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, November 2002.
- [48] F. van den Bergh and A. Engelbrecht. A study of particle swarm optimization particle trajectories. *Information sciences*, 176(8):937–971, 2006.

- [49] August Weismann. *The Germ Plasm: A Theory of Heredity*. Scott, London, UK, 1906.