



COLEGIO DE POSTGRADUADOS

INSTITUCIÓN DE ENSEÑANZA E INVESTIGACIÓN
EN CIENCIAS AGRÍCOLAS

CAMPUS MONTECILLO

POSTGRADO DE SOCIOECONOMÍA, ESTADÍSTICA E INFORMÁTICA
ESTADÍSTICA

**ESTUDIO COMPARATIVO PARA LA
OBTENCIÓN DE ÓPTIMOS PARA SUPERFICIES
DE RESPUESTA MULTIVARIADAS EN DISEÑOS
EXPERIMENTALES**

ABEL BARRIOS CÓRDOVA

T E S I S

PRESENTADA COMO REQUISITO PARCIAL PARA
OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS

MONTECILLO, TEXCOCO, ESTADO DE MÉXICO
2019



COLEGIO DE POSTGRADUADOS

INSTITUCIÓN DE ENSEÑANZA E INVESTIGACIÓN EN CIENCIAS AGRÍCOLAS

CARTA DE CONSENTIMIENTO DE USO DE LOS DERECHOS DE AUTOR Y DE LAS REGALÍAS COMERCIALES DE PRODUCTOS DE INVESTIGACIÓN

En adición al beneficio ético, moral y académico que he obtenido durante mis estudios en el Colegio de Postgraduados, el que suscribe, **“Abel Barrios Córdova”**, Alumno de esta Institución, estoy de acuerdo en ser partícipe de las regalías económicas y/o académicas, de procedencia nacional e internacional, que se deriven del trabajo de investigación que realicé en esta institución, bajo la dirección del Profesor **“Dr. Humberto Vaquera Huerta”**, por lo que otorgo los derechos de autor de mi tesis **“Estudio comparativo para la obtención de óptimos para superficies de respuesta multivariadas en diseños experimentales”**, y de los productos de dicha investigación al Colegio de Postgraduados. Las patentes y secretos industriales que se puedan derivar serán registrados a nombre del Colegio de Postgraduados y las regalías económicas que se deriven serán distribuidas entre la Institución, El Consejero o Director de Tesis y el que suscribe, de acuerdo a las negociaciones entre las tres partes, por ello me comprometo a no realizar ninguna acción que dañe el proceso de explotación comercial de dichos productos a favor de esta Institución.

Montecillo, Texcoco, Estado de México a 11 de Noviembre de 2019.

Abel Barrios Córdova

Vo. Bo. Dr. Humberto Vaquera Huerta

La presente tesis titulada: **Estudio comparativo para la obtención de óptimos para superficies de respuesta multivariadas en diseños experimentales**, realizada por el alumno: **Abel Barrios Córdova**, bajo la dirección del Consejo Particular indicado ha sido aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

MAESTRO EN CIENCIAS

SOCIOECONOMÍA, ESTADÍSTICA E INFORMÁTICA ESTADÍSTICA

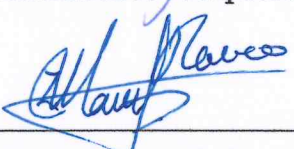
CONSEJO PARTICULAR

CONSEJERO



Dr. Humberto Vaquera Huerta

ASESOR



Dr. Juan Manuel Romero Padilla

ASESOR



Dr. José Luis Francisco Crossa Hiriart

Montecillo, Texcoco, Estado de México, Diciembre 2019

ESTUDIO COMPARATIVO PARA LA OBTENCIÓN DE ÓPTIMOS PARA SUPERFICIES DE RESPUESTA MULTIVARIADAS EN DISEÑOS EXPERIMENTALES.

Abel Barrios Córdova, M. en C.

Colegio de Postgraduados, 2019

RESUMEN

En este trabajo se desarrolla una propuesta para comparar diferentes metodologías de optimización multiobjetivo que tienen la finalidad de aportar herramientas para la solución de problemas en el área industrial principalmente, pero con extensiones a cualquier área del conocimiento que así lo requieran. Se trabajan las siguientes metodologías: función de deseabilidad, métodos de toma de decisiones con criterios múltiples (MCDM): MOORA, TOPSIS, MMOORA, y en este tipo de métodos se propone emplear una propiedad aditiva que aporta información sobre el efecto de las variables independientes en sus diferentes niveles que da origen a dos técnicas más llamadas en este trabajo MOORA AD y TOPSIS AD. Además se emplea la técnica de redes neuronales multicapa (con los paquetes Neuralnet (NEU) y Nnet (NET)) debido al auge e importancia que tienen este tipo de instrumentos de inteligencia artificial en la actualidad. Se aplican cada una de estas técnicas en tres casos de interés comercial o industrial reportados en la literatura con diferentes diseños experimentales (Taguchi, Box-Behnken y Diseño Central Compuesto), en un estudio de simulación Monte Carlo donde se controlan diferentes correlaciones, obteniendo en cada método una métrica basada en el GPE (Global Percentage Error) que se menciona en [Rocha et al. \(2015\)](#) y una segunda métrica que aparece en [Costa y Lourenço \(2016\)](#) llamada RTD (Relative Target Deviation), que evalúan la distancia de la respuesta estimada respecto a su valor ideal o deseado, con el fin de poder analizar las ventajas y desventajas de cada método. Todo el desarrollo se realizó con el software R ([R Core Team, 2019](#)) promoviendo el uso de este software libre con fines de investigación o desarrollo comercial.

Palabras clave: Optimización, Optimización Multiobjetivo, Optimización Multirespuesta, Deseabilidad, Optimización MCDM, Redes Neuronales Artificiales, Diseños Experimentales.

COMPARATIVE STUDY TO OBTAIN OPTIMA FOR MULTIVARIATE RESPONSE SURFACES IN EXPERIMENTAL DESIGNS

Abel Barrios Córdova, M. of S.

Colegio de Postgraduados, 2019

ABSTRACT

In this research work, a proposal was developed to compare different multiobjective optimization methodologies that have the purpose of contributing with tools to the solution of these problems mainly in the industrial area, but with extensions to any area of knowledge that so require. Eight methodologies are worked on: the desirability function, multi-criteria decision making methods (MCDM): MOORA, TOPSIS, MMOORA, and in this type of methods it was proposed to use an additive property that provides information on the effect of the independent variables at different levels that generates rise to two more techniques called in this work MOORA AD and TOPSIS AD. In addition, the multilayer neural network technique is used (with Neuralnet (NEU) and Nnet (NET) packages) because of the boom and importance of these types of artificial intelligence instruments today. Each one of these techniques is applied in three cases of commercial or industrial interest reported in the literature with different experimental designs (Taguchi, Box-Behnken and Composite Central Design), in a Monte Carlo simulation study where different correlations are controlled, obtaining in each method a metric based on the Global Percentage Error (GPE) mentioned in [Rocha et al. \(2015\)](#) and a second metric that proposed by [Costa y Lourenço \(2016\)](#) called Relative Target Deviation (RTD), which evaluate the distance of the estimated response with respect to its ideal or desired value, in order to analyze the advantages and disadvantages of each method. All the programs were carried out with R ([R Core Team, 2019](#)) to promote the use of free software for commercial research or development purposes.

Key words: Optimization, Multiobjective Optimization, Multi-Response Optimization, Desirability, MCDM Optimization, Artificial Neural Networks, Experimental Designs.

AGRADECIMIENTOS

A CONACYT.

A COLPOS.

A los integrantes de mi Consejo Particular:

Dr. Humberto Vaquera Huerta

Dr. José Luis Francisco Crossa Hiriart

Dr. Juan Manuel Romero Padilla

Dr. Esteban Burguete Hernández

A Nany por apoyarme en todo y a pesar de todo. Te amo.

A Melissa y Emmanuel por ser mi motor e inspiración.

A mis compañeros Rodolfo, Lizeth y Laura, por tan excelente convivencia.

A mi mamá y mi hermano Rubén por su apoyo tan importante.

A mis profesores.

A Tom por tu valiosa ayuda y soporte.

A Isa, Gris, Lau, Juan, Itzel, Gus, y todo el personal administrativo por su ayuda y aliento.

Y todos aquellos que me han apoyado para lograr este sueño.

DEDICATORIA

Al **Universo**.

A mis **hijos** Melissa y Emmanuel.

A mi **esposa** María del Carmen.

A mi **mamá** María Eugenia.

A mi **hermano** Rubén.

A toda mi **familia**.

CONTENIDO

PORTADA	i
RESUMEN	iv
ABSTRACT	v
LISTA DE TABLAS	xiii
LISTA DE FIGURAS	xv
1. Introducción	1
2. Objetivos	3
2.1. Particular	3
2.2. Específicos	3
3. Marco Teórico	4
3.1. Metodología de Superficie de Respuesta	4
3.2. Diseños experimentales	6
3.2.1. Etapas en el diseño de experimentos	7

CONTENIDO

3.2.2.	Principios en el diseño de experimentos	8
3.3.	Diseños experimentales en RSM	8
3.3.1.	Propiedades	8
3.3.2.	Superficies de Respuesta de primer orden	9
3.3.3.	Superficies de Respuesta de segundo orden	9
3.3.4.	Diseño Box–Behnken	10
3.3.5.	Diseño Central Compuesto	11
3.3.6.	Diseños Taguchi	11
3.4.	Técnicas de optimización	13
3.4.1.	Optimización caso univariado (una respuesta)	13
3.4.1.1.	Análisis canónico	14
3.4.1.2.	Análisis de cordillera	15
3.4.2.	Optimización caso multivariado	16
3.4.2.1.	Técnicas de optimizacion Multirespuesta	19
3.5.	Técnicas de optimización seleccionadas	19
3.5.1.	Función de Deseabilidad	20
3.5.2.	MOORA	23
3.5.3.	MOORA AD	25
3.5.4.	TOPSIS	25
3.5.5.	TOPSIS AD	27
3.5.6.	MULTIMOORA	27
3.5.7.	Redes Neuronales Artificiales	30
3.5.7.1.	La neurona artificial	30

CONTENIDO

3.5.7.2.	Funciones de activación	31
3.5.7.3.	Arquitecturas de Redes Neuronales	36
3.5.7.4.	Entrenamiento y aprendizaje	39
3.5.7.5.	Método de Optimización multiobjetivo con Redes Neuronales	41
4.	Metodología	42
4.1.	Estudio comparativo	42
4.2.	Simulación	42
4.2.1.	Factores de estudio	42
4.3.	Métrica usada para comparar los métodos	43
4.4.	Implementación en software R	44
5.	Resultados	46
5.1.	Resultados de la simulación	46
5.1.1.	Caso 1: Diseño Taguchi	46
5.1.2.	Caso 2: Diseño Box–Behnken	48
5.1.3.	Caso 3: Diseño Central Compuesto	49
5.2.	Resultados de Casos	52
5.2.1.	Caso 1: Diseño Taguchi	52
5.2.2.	Caso 2: Diseño Box–Behnken	54
5.2.3.	Caso 3: Diseño Central Compuesto	55
5.3.	Discusión	59
6.	Conclusiones	60

CONTENIDO

6.1. Recomendaciones	61
Referencias	62
ANEXOS	66
Función de Deseabilidad	67
MOORA (MOO)	69
TOPSIS (TOP)	70
MOORA AD (MAD)	71
TOPSIS AD (TAD)	72
MMOORA (MMO)	73
NEURALNET (NEU)	74
NNET (NET)	77
Simulación	79

LISTA DE TABLAS

3.1. Arreglos ortogonales estándar	12
3.2. Razones señal/ruido	13
5.1. Respuestas medias estimadas en la simulación para el Caso 1	46
5.2. Varianzas individuales para el Caso 1	47
5.3. Varianzas por método para el Caso 1	47
5.4. GPE/RTD por método para el Caso 1	47
5.5. Respuestas medias estimadas en la simulación para el Caso 2	48
5.6. Varianzas individuales para el Caso 2	48
5.7. Varianzas por método para el Caso 2	48
5.8. GPE/RTD por método para el Caso 2	49
5.9. Respuestas medias estimadas en la simulación para el Caso 3	50
5.10. Varianzas individuales para el Caso 3	50
5.11. Varianzas por método para el Caso 3	51
5.12. GPE/RTD por método para el Caso 3	51
5.13. Diseño experimental Taguchi L_8 , con transformación S/N.	52
5.14. Modelos ajustados para el Caso 1	53
5.15. Respuestas para Caso 1	53

LISTA DE TABLAS

5.16. Diseño experimental Box–Behnken	54
5.17. Modelos ajustados para el Caso 2	54
5.18. Respuestas para Caso 2	55
5.19. Diseño Central Compuesto	56
5.20. Errores individuales Caso 3	57
5.21. Modelos ajustados para el Caso 3	58
5.22. Respuestas para Caso 3	58

LISTA DE FIGURAS

3.1. Metodología de superficie de respuesta.	6
3.2. Modelo de un sistema o proceso.	7
3.3. Superficies de respuesta	10
3.4. Diseño Box–Behnken para tres factores.	11
3.5. Diseño Central Compuesto	11
3.6. Óptimo simultáneo	16
3.7. Funciones de deseabilidad individual	22
3.8. Neurona McCulloch-Pitts.	30
3.9. Neurona Rosenblatt.	31
3.10. Función de activación escalón binaria	32
3.11. Función de activación escalón bipolar	32
3.12. Función de rampa simétrica.	33
3.13. Función logística.	34
3.14. Función tangente hiperbólica.	34
3.15. Función gaussiana.	35
3.16. Función lineal.	35
3.17. Función ReLU.	36

LISTA DE FIGURAS

3.18. Red MLP.	37
3.19. Red feedback (Recurrente).	38
3.20. Red de Malla.	39
5.1. Correlación Caso 3.	57

Capítulo 1

Introducción

El continuo progreso del mundo actual basado en desarrollos científicos y tecnológicos logra mejoras importantes en la vida del ser humano, ampliando su expectativa de vida e incluso su salud y comodidad. En ese afán se busca desarrollar nuevos materiales, innovar y generar procesos, mejorar la calidad de productos, aumentar la vida en anaquel, mejorar servicios, disminuir pérdidas, reducir costos y un muy largo etcétera. Es importante aclarar que todos ellos buscan de manera intrínseca la optimización de procesos; en diversos ámbitos estos métodos son de vital importancia, teniendo aplicaciones tan diversas como en medicina, genética, astrofísica, etcétera, pero con un interés inherente en el área de producción industrial donde uno de los retos más importantes es encontrar condiciones óptimas de operación, que permitan maximizar calidad, utilidad económica, optimizar recursos, y simultáneamente minimizar desperdicios, costos de producción y por las preocupaciones ambientales actuales reducir sustancias contaminantes y residuos peligrosos.

Típicamente estas situaciones se estudian por medio de diseños de experimentos y superficies de respuesta; [Berger et al. \(2018\)](#) define a un diseño experimental como la agregación de variables independientes, sus magnitudes (niveles), y las combinaciones de estos niveles que se eligen con fines experimentales. Es decir, la finalidad de un diseño experimental es responder qué factores debemos estudiar, cómo deben variar los niveles de estos factores y la configuración de estos niveles en conjunto. Por otro lado [Myers et al. \(2016\)](#) define a la metodología de superficie de respuesta (RSM, por sus siglas en inglés) como una colección de técnicas estadísticas y matemáticas útiles para desarrollar, mejorar y optimizar procesos.

De acuerdo a [Khuri \(2017\)](#) la metodología de superficie de respuesta (RSM) incluye tres fases: a) la selección y construcción de un diseño experimental apropiado que genere información adecuada y confiable de ciertas variables respuesta denotadas como y ; b) la determinación de un modelo estadístico que dé mejor ajuste que se genera a partir del diseño experimental empleado. Tal modelo dará una aproximación

1. Introducción

funcional de las relaciones entre las variables respuesta y un conjunto de variables de control (factores de estudio, x_1, x_2, \dots, x_k) que el investigador considera afectan la variable de respuesta y ; c) se desea obtener los niveles óptimos en las variables de control (factores de estudio) que producen los valores de respuesta(s) máximas (o mínimas) en ciertas regiones de interés.

La RSM tiene importantes aplicaciones en el diseño, desarrollo y formulación de nuevos productos, así como en la mejora de los mismos. Es ampliamente usada en muchas áreas como la industria manufacturera, ciencias agropecuarias, ciencia de alimentos, e ingenierías. Las variables de entrada a veces se denominan variables independientes y están sujetas al control del ingeniero o científico, al menos para fines de una prueba o un experimento (Myers et al., 2016).

En muchas situaciones reales intervienen más de una variable respuesta, por lo que la optimización involucrará varias variables, en este caso, el problema no es trivial. El enfoque univariado de optimización es la estrategia mas común que usan los investigadores, pero tiene el problema que no necesariamente esto nos conduce a un óptimo común a todos los factores de estudio. A pesar de ello, la idea es buscar el mejor balance entre las variables involucradas buscando una solución de compromiso que permita optimizar el proceso y mejore la calidad del producto o servicio de interés (Gutiérrez-Pulido y De la Vara-Salazar, 2012).

En el caso multirespuesta en RSM, existen varias propuestas para resolver el problema de optimización. Khuri y Conlon (1981) desarrollaron un algoritmo para la optimización simultánea de varias funciones de respuesta que dependen del mismo conjunto de variables de control y están adecuadamente representadas por modelos de regresión polinómica del mismo grado. Antony (2000) propone utilizar el análisis de componentes principales en funciones de pérdida de Taguchi. Otro enfoque es el uso de funciones de deseabilidad Akteke-Öztürk et al. (2015), las funciones de deseabilidad desempeñan un papel cada vez mayor para resolver la optimización.

En el presente trabajo se analizarán algunas de las metodologías más utilizadas que se han propuesto para trabajar con RSM multiobjetivo y determinar cuál de ellas arroja mejores resultados en comparación con trabajos previos donde se ha logrado identificar los valores óptimos. Se comparan los siguientes métodos: Función de Deseabilidad (DES), métodos de toma de decisiones de criterios múltiples (MCDM o MCDA por sus siglas en inglés, (de los que se eligieron MOORA (MOO), TOPSIS (TOP) y Multi-MOORA (MMO) y se proponen dos métodos denominados MOORA AD (MAD) y TOPSIS AD (AD)) y Redes Neuronales Artificiales (ANN) (con los paquetes Neuralnet (NEU) y Nnet (NET)). Además se generó un experimento Monte Carlo exploratorio considerando como factores de estudio los métodos mencionados anteriormente, diseño experimental y correlación entre las respuestas. Por último se comparan las técnicas por medio de una métrica que evalúa la distancia de cada respuesta estimada respecto a su valor ideal o deseado, con el fin de analizar las ventajas y desventajas de cada método.

Capítulo 2

Objetivos

2.1. Particular

- Realizar un análisis comparativo del desempeño entre diferentes metodologías para la obtención del óptimo en superficies de respuesta multivariadas.

2.2. Específicos

- Por medio de simulación Monte Carlo realizar un estudio comparativo del desempeño de algunos métodos para obtener óptimos con respuesta multivariada.
- Conocer el efecto de la correlación entre las respuestas en la efectividad de cada metodología y del diseño.
- Desarrollar programas en software R ([R Core Team \(2019\)](#)), de cada una de las metodologías seleccionadas.

Capítulo 3

Marco Teórico

En el presente capítulo se muestran las herramientas y metodologías necesarias en la optimización multiobjetivo, iniciando con la Metodología de Superficie de Respuesta para una y varias respuestas. Además se mencionan los diseños experimentales más utilizados en la RSM y se culmina con los métodos de optimización multiobjetivo seleccionados para la elaboración de la presente investigación.

3.1. Metodología de Superficie de Respuesta

Esta metodología como se mencionó anteriormente comprende una colección de técnicas estadísticas y matemáticas útiles para desarrollar, mejorar y optimizar procesos; en los que se presentan una o varias respuestas. Todo inició con el trabajo de [Box y Wilson \(1951\)](#) que describe un procedimiento para determinar la combinación óptima de factores en un experimento multifactorial. Primero proponen emplear una técnica de ascenso pronunciado para llegar al vecindario del punto estacionario empleando diseños de primer orden, y después utilizando un diseño de segundo orden con puntos de diseño adicionales para explorar la superficie y localizar candidatos a óptimo. Esta metodología es la base para las posteriores investigaciones en este tema de las cuales surgieron algunas secuencias que en concreto siguen la idea original, para el caso de una sola respuesta se considera como una optimización simple.

3.1. Metodología de Superficie de Respuesta

Por ejemplo [Gutiérrez-Pulido y De la Vara-Salazar \(2012\)](#) propone una metodología con tres etapas:

1. Cribado: Si se tienen muchos factores, se identifican los factores que tienen mayor influencia.
2. Búsqueda I: Se corre un diseño de primer orden que caracterice la superficie e identifique si existe curvatura.
3. Búsqueda II: Al detectar curvatura se corre un diseño de segundo orden para modelar la curvatura y determinar las condiciones óptimas.

Podemos mencionar además la metodología propuesta por [Burguete-Hernández \(2000\)](#):

1. Hacer un diagrama de Ishikawa, eliminando los factores no importantes.
2. Filtrar los factores, hacer un diseño factorial fraccionado y eliminar nuevamente los factores no importantes.
3. Hacer un experimento factorial y determinar el camino de máximo ascenso.
4. Hacer observaciones en el máximo ascenso hasta encontrar curvatura, o bien un máximo o un mínimo.
5. Hacer un experimento factorial para estimar la superficie de respuesta y determinar el valor óptimo.

Observando las metodologías anteriores podemos resumir los pasos como: diseño, modelación y modelación-optimización. En la figura [3.1](#) podemos observar gráficamente los pasos secuenciales de la metodología, sin embargo, es prioritario mencionar que en realidad no se conoce el valor óptimo ni las curvas de nivel, la metodología permite caracterizar y modelar la superficie de respuesta.

3.2. Diseños experimentales

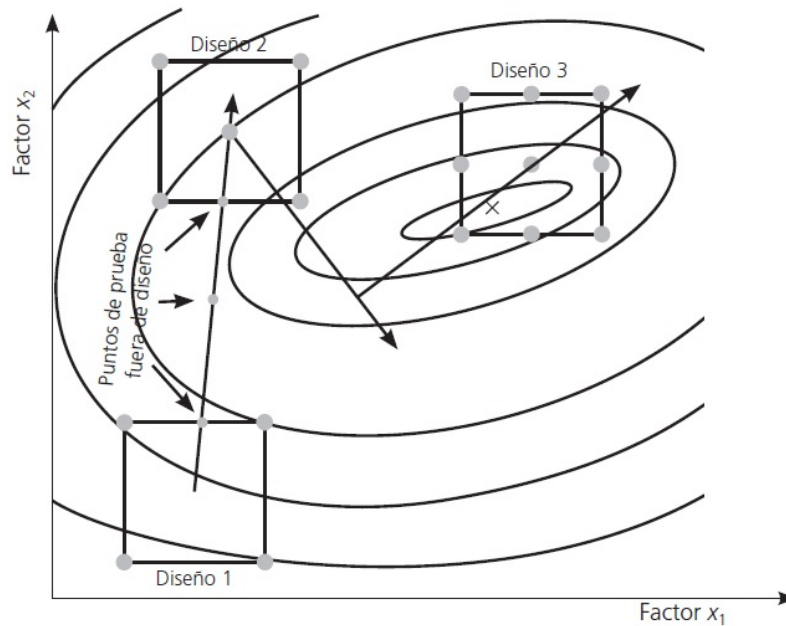


Figura 3.1: Metodología de superficie de respuesta. Fuente: [Gutiérrez-Pulido y De la Vara-Salazar \(2012\)](#).

3.2. Diseños experimentales

Previamente se menciona que los diseños experimentales son técnicas que permiten estudiar como la agregación de variables independientes, sus niveles y combinaciones influyen sobre una variable o variables de interés. Los primeros métodos propuestos en el diseño de experimentos aparecen en [Fisher \(1926\)](#) y en [Fisher \(1935\)](#), en los cuales Ronald Fisher realiza investigaciones sobre agronomía en la estación experimental Rothamsted, Inglaterra. Donde sentó las bases teóricas para planificar y analizar experimentos. Los objetivos de Fisher eran eliminar la mayor cantidad de variación natural posible, evitar que la variación resultante sesgue los efectos que se prueban y detectar causa y efecto con la mínima cantidad de esfuerzo experimental. Preocupaciones que hasta el día de hoy son necesarias y fundamentales en un diseño experimental ([Lawson, 2014](#)).

Es muy difícil pensar en investigación científica dejando de lado a la experimentación, ya que la información generada por esta permite comprender mejor un fenómeno para poder comprobar hipótesis científicas y tomar decisiones en su solución o propuesta ([Montgomery, 2017](#)). Un diseño experimental incluye los factores de diseño que intervienen en el proceso, que pueden ser controlables o no controlables (factores de ruido) y con esto obtener la medición para las características de interés, este esquema se muestra en la figura 3.2.

3.2. Diseños experimentales

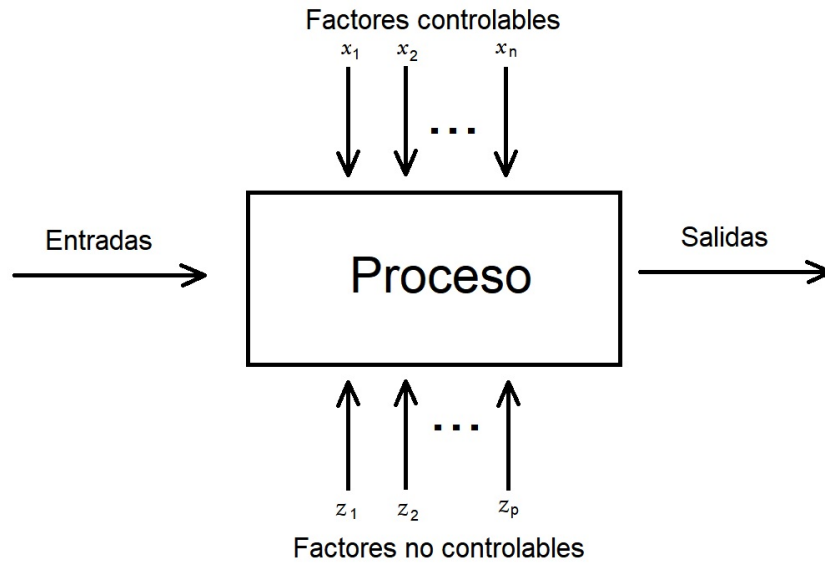


Figura 3.2: Modelo de un sistema o proceso.

3.2.1. Etapas en el diseño de experimentos

Así como en el método científico, los diseños experimentales siguen una serie de etapas iterativas ([Montgomery, 2017](#)):

Previo al diseño:

1. Reconocimiento y planteamiento del problema.
2. Selección de la variable de respuesta.

Directrices en el diseño:

1. Elección de factores, niveles y rangos.
2. Elección del diseño experimental.
3. Realizar el experimento.
4. Análisis estadístico de los datos.
5. Conclusiones y recomendaciones.

3.3. Diseños experimentales en RSM

3.2.2. Principios en el diseño de experimentos

Montgomery (2017) propone tres principios básicos en el diseño de experimentos, pero que algunas veces se puede considerar agregar el principio factorial. Los principios son:

1. Aleatorización: Es el fundamento de poder utilizar métodos estadísticos en esta metodología, ya que los métodos estadísticos requieren que las observaciones o los errores sean variables aleatorias con distribuciones independientes.
2. Replicación: Es una repetición independiente de cada combinación de factores con el fin de distinguir que parte de la variabilidad total se debe a cada factor elegido.
3. Bloqueo: Antes de realizar los tratamientos se agrupan las unidades experimentales tomando como parámetro su homogeneidad en la respuesta para mejorar la precisión de las comparaciones.

3.3. Diseños experimentales en RSM

El análisis de la superficie de respuesta sobre la región de interés implica la recopilación de datos experimentales, elegir un modelo adecuado que se ajuste a esos datos y probar la idoneidad del modelo ajustado. Para lograrlo se emplea un modelo polinomial y se selecciona un conjunto de tratamientos sobre los que se harán observaciones experimentales (Bayramov et al., 2004).

3.3.1. Propiedades

Las propiedades más relevantes de los diseños empleados en RSM son:

Ortogonalidad: esta propiedad indica que la información de un coeficiente estimado no contiene información de otro coeficiente, es decir no están correlacionados. Un diseño es ortogonal si los vectores columna de la matriz de factores son independientes entre sí.

Rotabilidad: esta propiedad está relacionada con el centro del diseño, ya que si se rota el diseño es probable que los óptimos estén a diferentes distancias y la predicción sea complicada. Por este hecho se eligen diseños que contemplen esta propiedad donde la varianza de los valores estimados sea constante respecto al centro del diseño.

3.3. Diseños experimentales en RSM

3.3.2. Superficies de Respuesta de primer orden

Para poder estudiar una superficie de respuesta de primer orden se ajusta un modelo donde el grado del polinomio es $d = 1$, y k es el número de factores, tenemos un modelo de primer orden:

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \epsilon \quad (3.1)$$

note que en 3.1 solo considera los efectos principales y no toma en cuenta las interacciones de los factores. Los diseños de primer orden más usados son: factoriales 2^k , arreglos ortogonales 2^k , factoriales fraccionados, Plackett-Burman y simplex.

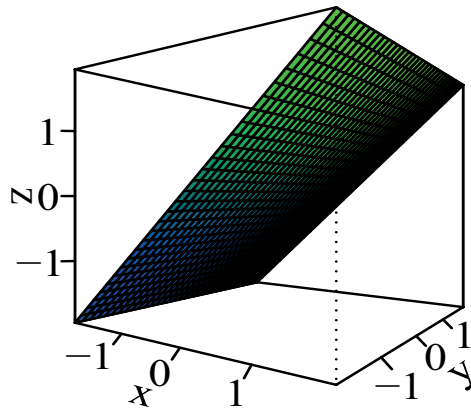
3.3.3. Superficies de Respuesta de segundo orden

En el análisis de superficies de respuesta de segundo orden típicamente se ajustan modelos que permitan estudiar la curvatura (efectos cuadráticos) y las interacciones entre los factores, el modelo de segundo orden es:

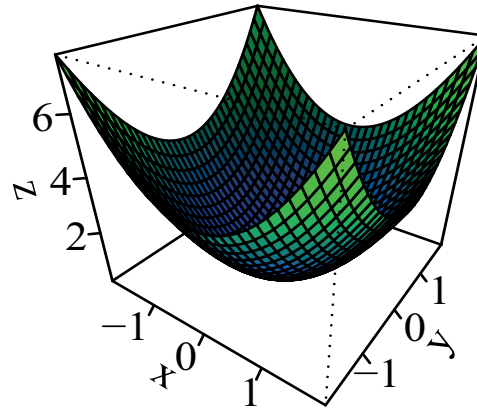
$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \sum_{j=1}^k \beta_{ij} x_i x_j + \sum_{i=1}^k \beta_{ii} x_i^2 + \epsilon \quad (3.2)$$

los diseños de segundo orden más utilizados en RSM son el diseño central compuesto (DCC) propuestos por [Box y Wilson \(1951\)](#) y los diseños Box-Behnken propuestos por [Box y Behnken \(1960\)](#). En la figura 3.3 podemos observar gráficas de superficies de respuesta obtenidas con el software [R Core Team \(2019\)](#), en la figura 3.3 a) se muestra una superficie caracterizada por un modelo de primer orden, y en el resto de las figuras se observan diferentes superficies caracterizadas por modelos de segundo orden.

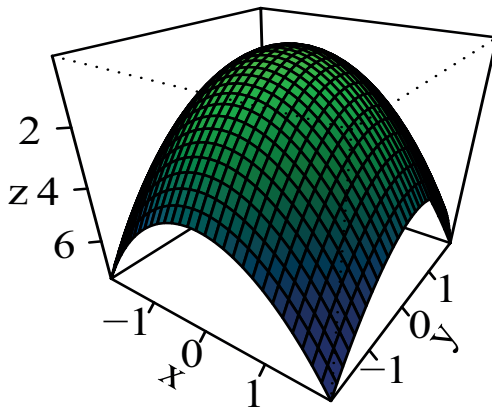
3.3. Diseños experimentales en RSM



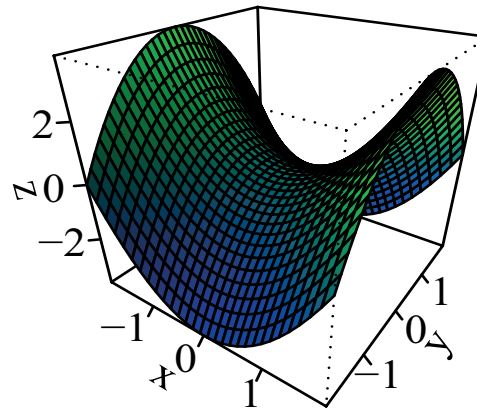
(a) modelo de primer orden



(b) modelo de segundo orden (mínimo)



(c) modelo de segundo orden (máximo)



(d) modelo de segundo orden (minimax)

Figura 3.3: Superficies de respuesta

3.3.4. Diseño Box–Behnken

Estos diseños se emplean a partir de tres variables independientes, y necesita pocas corridas experimentales (por lo que es elegible por su bajo costo), cada variable independiente tiene tres niveles que se pueden codificar con los valores: $-1, 0, 1$. No se puede usar en experimentos secuenciales debido a que su base no es un diseño factorial y además no incluye experimentos en los extremos, por lo que la rotabilidad es parcial aunque no varía demasiado respecto a un diseño rotatable. Gráficamente se puede observar en la figura 3.4

3.3. Diseños experimentales en RSM

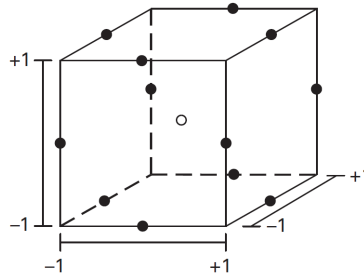


Figura 3.4: Diseño Box–Behnken para tres factores. Fuente: [Montgomery \(2017\)](#).

3.3.5. Diseño Central Compuesto

Este tipo de diseños son muy utilizados en RSM porque pueden estudiar fenómenos que requieren experimentos con una secuencia ya que se basan en diseños factoriales, sólo que se le agregan puntos axiales y puntos centrales. Se pueden considerar desde dos variables independientes con hasta cinco niveles cada una, por lo que a partir de tres variables independientes considera más puntos experimentales que el Diseño Box-Behnken. En la figura 3.5 se observa geoméricamente el diseño para dos y tres variables independientes.

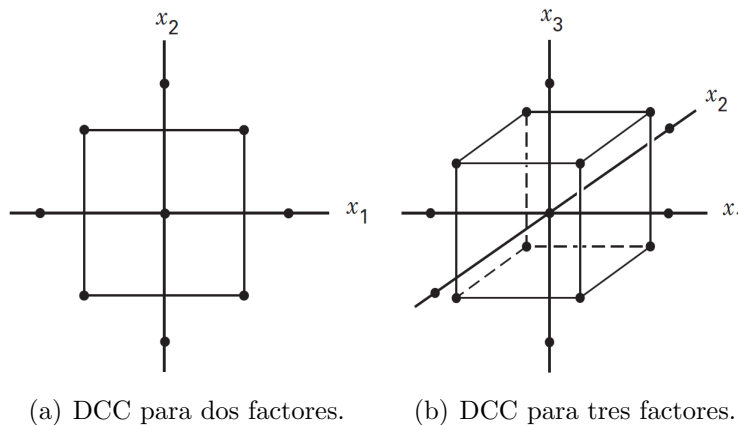


Figura 3.5: Diseño Central Compuesto. Fuente: [Montgomery \(2017\)](#).

3.3.6. Diseños Taguchi

El Dr. Genichi Taguchi propuso en la década de los años 50 nuevas ideas en el mejoramiento de calidad de las industrias japonesas, y no fue sino hasta la década de

3.3. Diseños experimentales en RSM

los años 80 cuando sus aportaciones llegaron a los Estados Unidos, ocasionando una gran controversia debido a cuestiones de sustento teórico y algunas excepciones a sus métodos. Pero la prueba irrefutable de que sus métodos funcionaron y aún lo hacen, es el éxito de diversas industrias de su país y de varias partes del mundo. Su filosofía de calidad es que la calidad debe diseñarse en el producto y no después de una revisión. Además de que la calidad de un producto se logra minimizando la variabilidad en un valor objetivo en su fabricación y en la robustez a factores de ruido, que se pueden observar en la figura 3.2.

Taguchi propone las matrices o diseños ortogonales para diseño de experimentos, que cuentan con la propiedad de ortogonalidad. Estos arreglos ortogonales proveen de la información necesaria pero con un menor número de corridas experimentales, aunque en algunos casos no se pueden estudiar las interacciones de las variables independientes y se recomienda que la variable más difícil de manipular al cambiar de nivel se coloque en la primer columna por que se muestran niveles graduales, pero que trae como consecuencia perder la aleatoriedad, por lo que estas dos últimas ideas fueron los principales puntos de controversia con sus ideas.

La nomenclatura de los arreglos Taguchi es:

$$L_a(b^c) \tag{3.3}$$

donde: L hace referencia a cuadro latino (Latin square), ya que estos se pueden representar por matrices ortogonales, a es el número de experimentos, b es el número de niveles de las variables independientes y c es la cantidad de variables independientes, y como nota los grados de libertad se obtienen de $a - 1$.

Ya definida la nomenclatura podemos elegir de los arreglos ortogonales más usados que se muestran en la tabla 3.1, obtenida de Krishnaiah y Shahabudeen (2012).

Tabla 3.1: Arreglos ortogonales estándar.

Dos niveles	Tres niveles	Cuatro niveles	Niveles mixtos
$L_4(2^3)$	$L_9(3^4)$	$L_{15}(4^5)$	$L_{18}(2^1, 3^7)^\dagger$
$L_8(2^7)$	$L_{27}(3^{13})$	$L_{64}(4^{21})$	$L_{36}(2^1, 3^1, 2)$
$L_{16}(2^{15})$	$L_{81}(3^{40})$		
$L_{12}(2^{11})^*$			

† Se puede estudiar una interacción entre factor de 2 niveles y uno de 3 niveles.

* Las interacciones no se pueden estudiar.

Para el análisis de los diseños Taguchi se emplea la razón señal ruido (S/N), los factores de señal son aquellos factores o variables independientes que cambian la magnitud de la variable respuesta y que se controlan para disminuir los efectos de

3.4. Técnicas de optimización

los factores de ruido. Estos últimos como ya se mencionó, no se pueden controlar y actúan sobre el proceso. Este cociente se calcula en cada combinación de variables de control y se analiza como cualquier respuesta.

La tabla 3.2 muestra la razón S/N de acuerdo al tipo de característica de calidad de interés:

Tabla 3.2: Razones señal/ruido. Fuente: modificado de [Gutiérrez-Pulido y De la Vara-Salazar \(2012\)](#).

Característica	Razón (S/N)
Mientras más pequeña mejor	$-10 \log \left[\frac{1}{n} \sum_{i=1}^n y_i^2 \right]$
Mientras más grande mejor	$-10 \log \left[\frac{1}{n} \sum_{i=1}^n \frac{1}{y_i^2} \right]$
Valor target (tipo I)	$10 \log \left(\frac{\bar{Y}^2}{S^2} \right)$
Valor target (tipo II)	$-10 \log(S^2)$
Proporción de defectuosos	$-10 \log \left(\frac{p}{(1-p)} \right)$

3.4. Técnicas de optimización

3.4.1. Optimización caso univariado (una respuesta)

Llevando a cabo el ajuste de los modelos se procede a tratar de encontrar el óptimo de la respuesta. Note que si el modelo no explica un mínimo de 70 % del comportamiento de la respuesta, en términos del R_{aj}^2 , no se recomienda emplearlo porque su capacidad de predecir es mala.

La técnica de optimización para el modelo de primer orden es el escalamiento ascendente (o descendente). Y para los modelos de segundo orden se emplea análisis canónico o análisis de cordillera. Todo inicia explorando la región experimental con los diseños de primer orden mencionados en la sección anterior y tratar de identificar la ruta de ascenso o descenso deseada, estos pasos son iterativos hasta encontrar el punto estacionario y posteriormente aplicar diseños de segundo orden para poder

3.4. Técnicas de optimización

identificar el óptimo o la región de operación donde se encuentra optimizada la respuesta, en este punto se utilizan las técnicas de análisis canónico o de cordillera. En el presente trabajo sólo se emplean situaciones donde se presenta curvatura, por lo que se emplea la regresión con modelos de segundo orden.

3.4.1.1. Análisis canónico

El punto estacionario es aquel punto donde la derivada del modelo es igual a cero, por lo que se aplica el análisis canónico reescribiendo el modelo ajustado de segundo orden en términos de nuevas variables llamadas canónicas, las cuales son transformaciones de las variables codificadas. Se deriva el modelo respecto a cada variable independiente, se iguala a cero y se resuelve el sistema de ecuaciones. La importancia de esta forma de optimización es que se puede observar a simple vista que tipo de superficie representa al fenómeno o situación de estudio (Montgomery, 2017).

Si tenemos un modelo ajustado de segundo orden como el siguiente:

$$\hat{y} = \hat{\beta}_0 + \sum_{i=1}^k \hat{\beta}_i x_i + \sum_{i=1}^k \sum_{j=1}^k \hat{\beta}_{ij} x_i x_j + \sum_{i=1}^k \hat{\beta}_{ii} x_i^2 \quad (3.4)$$

y reescribimos la ecuación anterior en forma matricial tenemos:

$$\hat{y} = \hat{\beta}_0 + \mathbf{x}'\mathbf{b} + \mathbf{x}'\mathbf{B}\mathbf{x} \quad (3.5)$$

donde \mathbf{x}' es cualquier punto en la región de experimentación, \mathbf{b} son los coeficientes lineales y \mathbf{B} es la matriz de coeficientes de las interacciones y términos cuadráticos. Ahora derivando con respecto a \mathbf{x} e igualando a cero:

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

si la ecuación anterior se resuelve para \mathbf{x} , se obtiene el punto estacionario:

$$\mathbf{x}_0 = \frac{-\mathbf{B}^{-1}\mathbf{b}}{2} \quad (3.6)$$

para casos donde están involucradas diversas variables independientes es recomendable usar la ecuación canónica siguiente:

3.4. Técnicas de optimización

$$\hat{y} = \hat{y}_0 + \lambda_1 w_1^2 + \lambda_2 w_2^2 + L + \dots + \lambda_k w_k^2 \quad (3.7)$$

donde los coeficientes λ son los valores característicos de la matriz \mathbf{B} de la ecuación 3.5 y de acuerdo a los signos definen el tipo de punto estacionario:

1. Si λ_i es positivo para toda i , es un mínimo.
2. Si λ_i es negativo para toda i , es un máximo.
3. Si se encuentran ambos signos es un punto silla.

3.4.1.2. Análisis de cordillera

Hoerl (1959) introduce este modelo y Draper (1963) desarrolló la teoría matemática detrás de la técnica, la cual consiste en optimizar \hat{y} de la ecuación 3.5 sujeto a que x se encuentra en la superficie de una hiperesfera de radio r y centro en el origen:

$$\sum_{i=1}^k x_i^2 = r^2 \quad (3.8)$$

Esta optimización emplea varios valores de r con el fin de obtener información sobre el óptimo a varias distancias del origen, y considera optimización con restricción en el método de Lagrange. Se considera la función:

$$H = \hat{\beta}_0 + \mathbf{x}'\mathbf{b} + \mathbf{x}'\mathbf{B}\mathbf{x} - \lambda(\mathbf{x}'\mathbf{x} - r^2) \quad (3.9)$$

donde λ es un multiplicador de Lagrange. Derivando H con respecto a \mathbf{x} e igualando a cero:

$$\mathbf{b} + 2(\mathbf{B}\mathbf{x} - \lambda\mathbf{x}) = 0 \quad (3.10)$$

y resolviendo para x , obtenemos:

$$\mathbf{x} = -\frac{1}{2}(\mathbf{B} - \lambda\mathbf{I}_n)^{-1}\mathbf{b} \quad (3.11)$$

y la solución a la ecuación anterior representa un punto estacionario de \hat{y} , y por medio

3.4. Técnicas de optimización

de la matriz Hessiana podemos determinar si es un máximo, mínimo o punto silla. La matriz Hessiana es la matriz de derivadas parciales de segundo orden de H con respecto a \mathbf{x} es negativo definido (positivo definido) (Khuri y Mukhopadhyay, 2010). De la ec. 3.10 la matriz está dada por:

$$\frac{\partial}{\partial \mathbf{x}} \left[\frac{\partial H}{\partial \mathbf{x}'} \right] = 2(\mathbf{B} - \lambda \mathbf{I}_n) \quad (3.12)$$

3.4.2. Optimización caso multivariado

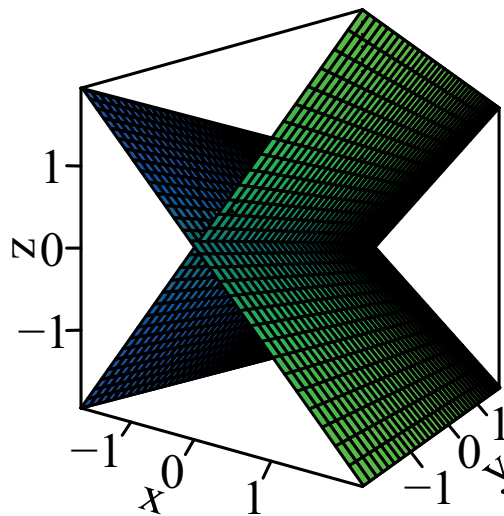


Figura 3.6: Óptimo simultáneo

Las aplicaciones y la teoría sobre RSM fue desarrollada inicialmente para casos de una sola respuesta, donde la metodología anteriormente comentada tiene una completa aplicación y eficacia, pero las necesidades reales han forzado a que la investigación avance sobre nuevos métodos de solución a problemas más complejos como lo son las respuestas duales que se observa gráficamente en la figura 3.6 pero incluso existen situaciones que implican la solución y estudio de fenómenos multirespuesta tan comunes en diversas áreas de estudio. En este sentido surge la idea de desarrollar el presente trabajo para aportar conocimiento sobre estas novedosas y útiles herramientas, que si bien hace falta seguir trabajando en esa dirección han cimentado toda una área de formación. La RSM multirespuesta tiene el objetivo de seleccionar los niveles de las variables independientes que optimicen todas las respuestas a la vez, pero este óptimo global no coincide comúnmente con los óptimos individuales, por lo que se busca una solución de compromiso que identifique la mejor combinación de factores y permita optimizar de la mejor manera al proceso, cabe mencionar que en la búsqueda

3.4. Técnicas de optimización

de este óptimo se consideren restricciones de las variables involucradas lo que vuelve a esta tarea aún más complicada.

El modelo multirespuesta fue propuesto inicialmente por [Khuri y Cornell \(1987\)](#) que mencionan prestar atención a la correlación de los datos dentro de las corridas experimentales pero asumir independencia entre corridas, el modelo se define a continuación:

Primero, para describir la relación inicialmente desconocida entre la variable respuesta y y un número k de variables independientes, x_1, x_2, \dots, x_k , se aproxima empleando un modelo polinomial grado d , ($d \geq 1$) de la forma:

$$y = f'(\underline{x})\beta + \epsilon \quad (3.13)$$

donde $\underline{x} = (x_1, x_2, \dots, x_k)'$, $f'(\underline{x})$ es una función vector de p elementos que consiste en potencias y productos cruzados de potencias de x_1, x_2, \dots, x_k , β es el vector de p parámetros desconocidos y ϵ es el error experimental aleatorio con media cero. La cantidad $f'(\underline{x})$ representa la respuesta media, que es el valor esperado de y y se denota por $\mu(x)$ ([Khuri y Mukhopadhyay, 2010](#)).

Este modelo univariado permite establecer una relación entre la variable respuesta y y las variables de control, determinar la significancia de las variables de control y poder determinar el conjunto óptimo de variables de control que minimiza o maximiza a la respuesta.

Para el caso multivariado se llevan a cabo n experimentos, donde se observan las r respuestas, y un número k de variables independientes, a este conjunto de valores de las variables de control en los que se lleva a cabo el experimento se le llama diseño, y puede ser representado por una matriz de dimensiones $n \times r$, y cada renglón de esa matriz diseño representa una respuesta obtenida de experimentar en el i -ésimo valor de x , por lo que la ecuación 3.13 se puede escribir como:

$$y_i = f'(\underline{x}_i)\beta + \epsilon_i, \quad i = 1, 2, \dots, r \quad (3.14)$$

y los r modelos se pueden escribir en forma matricial como:

$$Y = \mathbf{X}\beta + \epsilon \quad (3.15)$$

asumiendo que ϵ tiene media cero y matriz de varianzas y covarianzas $\sigma^2 I_n$, y además si los modelos de las respuestas son de la misma forma y consideran la misma matriz diseño, el estimador de mínimos cuadrados de β es:

3.4. Técnicas de optimización

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \quad (3.16)$$

vea que este estimador $\widehat{\boldsymbol{\beta}}$ es un estimador lineal insesgado de $\boldsymbol{\beta}$ por lo que la matriz de covarianzas de $\widehat{\boldsymbol{\beta}}$ es:

$$\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\beta}}} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \quad (3.17)$$

y ahora se puede calcular un estimado de la respuesta media en el punto x dentro de la región experimental \mathfrak{R} , empleando el estimador $\widehat{\boldsymbol{\beta}}$, esto es,

$$\widehat{y}(\mathbf{x}) = \mathbf{f}'(\mathbf{x})\widehat{\boldsymbol{\beta}}, \quad \mathbf{x} \in R \quad (3.18)$$

donde $\mathbf{f}'(\mathbf{x})$ es de dimensión p con el primer componente de unos y el resto de componentes son los productos cruzados de potencias de x_1, x_2, \dots, x_r de acuerdo al grado del polinomio. Y ya que $\widehat{\boldsymbol{\beta}}$ es un estimador insesgado de $\boldsymbol{\beta}$, $\widehat{y}(\mathbf{x})$ es también un estimador insesgado de $\mathbf{f}'(\mathbf{x})\boldsymbol{\beta}$, que es la respuesta media de $\mathbf{x} \in \mathfrak{R}$. Ahora empleando la ecuación 3.17 podemos escribir su varianza como:

$$Var[\widehat{y}(\mathbf{x})] = \sigma^2 \mathbf{f}'(\mathbf{x})(\mathbf{X}'\mathbf{X})^{-1} \mathbf{f}(\mathbf{x}) \quad (3.19)$$

esta varianza debe ser lo más pequeña posible ya que es directamente proporcional a la calidad de la predicción buscada y esto es afectado por la elección del diseño y que además tenga un buen ajuste a las condiciones reales estudiadas. Después de que se tenga un buen ajuste de los modelos es ahora imperativo localizar la región óptima del proceso o fenómeno de estudio, pero hay que tomar en cuenta que cada una de las variables independientes involucradas cuentan con una región de operación que dependen de las especificaciones del proceso, de la maquinaria y del dispositivo de medición. Además existen situaciones donde no es posible realizar experimentos en toda la región por lo que se restringe aún más el espacio de parámetros.

Existen aportaciones de solución a problemas de optimización de RSM multiobjetivo por medio de un análisis gráfico que se muestran en [Myers et al. \(2016\)](#) y [Gutiérrez-Pulido y De la Vara-Salazar \(2012\)](#) pero que tienen la limitante de ser efectivas para tres o menos respuestas, por lo que no se abordará en el presente trabajo.

Para lograr la optimización simultánea de varias respuestas se emplea la técnica de superficies de respuesta comentada anteriormente y se complementa con otras técnicas matemáticas que reducen de alguna manera el problema de optimización multivariado a un problema univariado. En este sentido los métodos emplean algún mecanismo

3.5. Técnicas de optimización seleccionadas

que permite ponderar las variables de interés de acuerdo a su importancia relativa o de acuerdo a su variabilidad. Existen diversos métodos que se intentan clasificar a continuación:

3.4.2.1. Técnicas de optimización Multirespuesta

La siguiente clasificación es una modificación de [Šibalija y Majstorović \(2016\)](#):

1. Métodos convencionales. Técnicas estadísticas o matemáticas:
 - a) Técnicas de diseño experimental
 - 1) RSM
 - 2) Función de Deseabilidad
 - 3) Taguchi (diseño de parámetros robustos S/N)
 - b) Técnicas de búsqueda matemática iterativa
 - c) Programación por metas
 - d) Toma de decisiones de criterios múltiples. Multiple-criteria decision-making (MCDM)
2. Métodos no convencionales. Técnicas de inteligencia artificial:
 - a) Fuzzy Logic
 - b) Redes Neuronales
 - c) Técnicas de búsqueda metaheurística
 - 1) Algoritmos genéticos
 - 2) Simulated Annealing (recocido simulado)
 - 3) Optimización de enjambre de partículas, colonias de hormigas, etc.
 - 4) Búsqueda tabú
 - 5) Algoritmos evolutivos recientes (colonia de abejas artificiales, optimización basada en biogeografía, sistema inmunológico y la optimización basada en la enseñanza – aprendizaje)
 - d) Sistemas expertos

3.5. Técnicas de optimización seleccionadas

En el presente trabajo se consideran las siguientes técnicas:

3.5. Técnicas de optimización seleccionadas

1. Función de Deseabilidad (DES)
2. Multiple-criteria decision-making (MCDM)
 - a) MOORA (MOO)
 - b) TOPSIS (TOP)
 - c) MMOORA (MMO)
 - d) MOORA AD (MAD)
 - e) TOPSIS AD (TAD)
3. Redes Neuronales (NEU y NET)

3.5.1. Función de Deseabilidad

Esta es la técnica más utilizada en la industria actualmente ya que se encuentra programada en diversos software comerciales por lo que se retoma en el presente trabajo para compararla con otras metodologías. Fue propuesta originalmente por [Harrington \(1965\)](#) proponiendo transformaciones exponenciales, posteriormente [Derringer y Suich \(1980\)](#) y [Derringer \(1994\)](#) mejoran la propuesta con funciones más flexibles, estas funciones son las que utilizaremos. La idea detrás de esta metodología consiste en convertir primeramente cada respuesta individual y_i en una función de deseabilidad d_i que se encuentra en el rango:

$$0 \leq d_i \leq 1$$

si $d_i = 1$, indica que la respuesta \hat{y}_i estimada alcanza su máximo valor deseable, si por el contrario $d_i = 0$ indica que la respuesta toma un valor no deseado. Las funciones de deseabilidad individual se presentan en tres casos, cuando el objetivo es minimizar, maximizar o lograr un objetivo; en algunos textos se mencionan como transformaciones de una cola para los primeros dos casos y de dos colas para el caso de lograr un objetivo. Las funciones son:

Para el caso de que y_i se desea maximizar:

3.5. Técnicas de optimización seleccionadas

$$d_i(x) = \begin{cases} 0 & y < L \\ \left(\frac{y-L}{T-L}\right)^r & L \leq y \leq T \\ 1 & y > T \end{cases} \quad (3.20)$$

donde L es el límite inferior y T es el valor objetivo en la respuesta.

Para el caso de que y_i se desea minimizar:

$$d_i(x) = \begin{cases} 1 & y < T \\ \left(\frac{U-y}{U-T}\right)^r & T \leq y \leq U \\ 0 & y > U \end{cases} \quad (3.21)$$

donde U es el límite superior de la respuesta.

Para el caso donde se busca obtener un valor objetivo T para y_i que se encuentra entre un límite inferior L y superior U :

$$d_i(x) = \begin{cases} 0 & y < L \\ \left(\frac{y-L}{T-L}\right)^{r_1} & L \leq y \leq T \\ \left(\frac{U-y}{U-T}\right)^{r_2} & T \leq y \leq U \\ 0 & y > U \end{cases} \quad (3.22)$$

vea que en las tres expresiones anteriores aparecen términos r que son ponderaciones de las respuesta por ejemplo si $r = 1$, la función de deseabilidad será lineal. Si se elige $r > 1$ pone énfasis en estar en el valor objetivo y $0 < r < 1$ le da menos importancia. En la figura 3.7 se pueden observar cada una de las funciones de deseabilidad y los efectos que tienen las ponderaciones en lograr el objetivo (Montgomery, 2017).

3.5. Técnicas de optimización seleccionadas

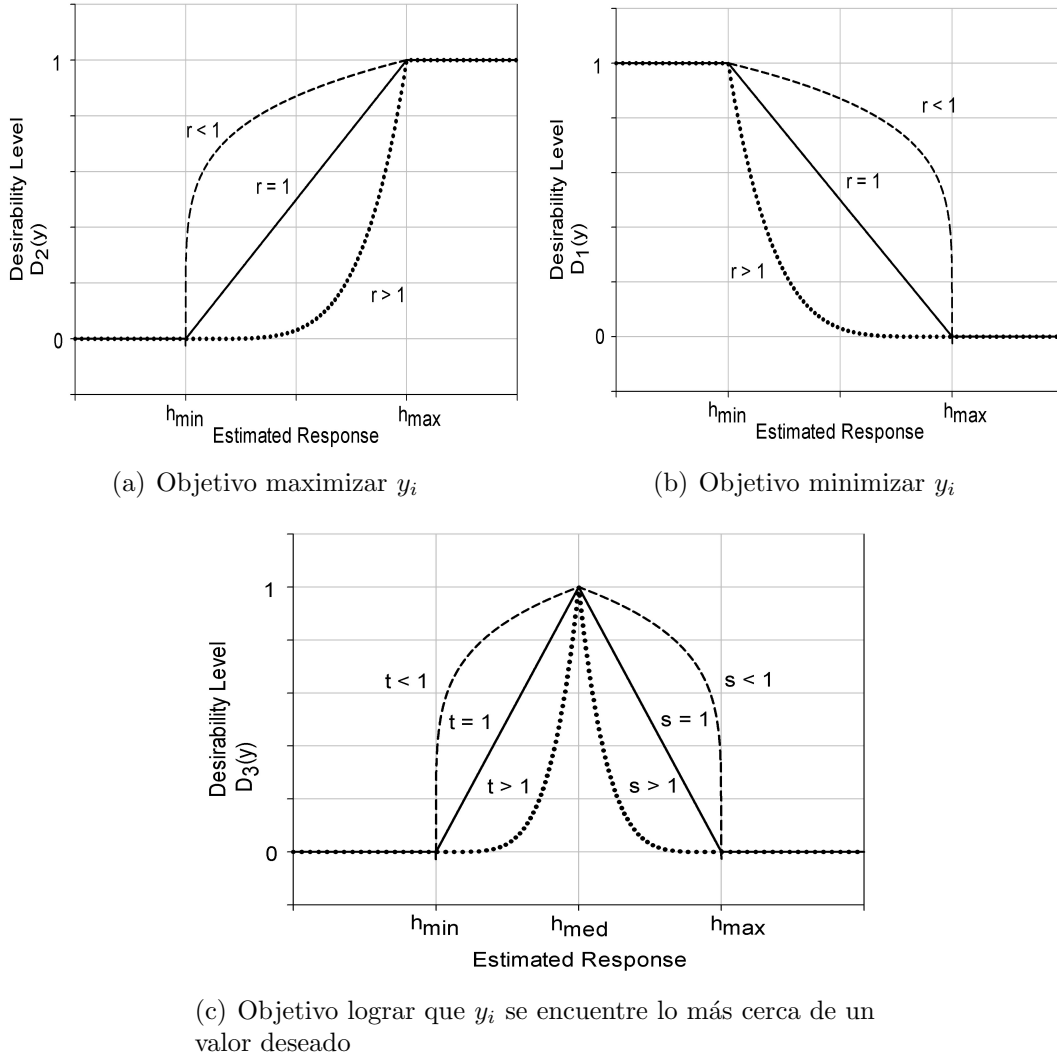


Figura 3.7: Funciones de deseabilidad individual. Fuente: [Altinoz et al. \(2015\)](#)

Después de haber transformado cada una de las respuestas se procede a combinarlas empleando una media geométrica que tiene la ventaja de ser menos sensible que la media aritmética a valores extremos, pero que por poseer una propiedad multiplicativa al tomar un valor cero toda la deseabilidad se haría cero, esto se puede considerar como una ventaja si lo que se desea es mejorar un proceso y la idea es obtener respuestas deseadas, por lo que un resultado como el mencionado nos da idea de que el producto o proceso es inaceptable. Esta media geométrica ponderada se le conoce como deseabilidad global,

$$DG(x) = (d_1^{w_1} \times d_2^{w_2} \times \dots \times d_k^{w_k})^{\frac{1}{\sum w_i}} \quad (3.23)$$

3.5. Técnicas de optimización seleccionadas

donde w_i son pesos que balancean la importancia de cada variable, entre más grande sea un peso respecto a los otros, implica que más grande será la exigencia para que el punto óptimo global beneficie a dicha variable (Gutiérrez-Pulido y De la Vara-Salazar, 2012). Si se dá el caso de que todas tengan la misma importancia, entonces, $w_i = 1$ para $i = 1, 2, \dots, k$ y la deseabilidad global resulta:

$$DG(x) = (d_1 \times d_2 \times \dots \times d_k)^{\frac{1}{k}} \quad (3.24)$$

y para localizar el óptimo global $x_0 = (x_{10}, x_{20}, \dots, x_{k0})$ donde $DG(x)$ es máxima se recurre a métodos numéricos, de los que se emplean comúnmente los algoritmos Levenberg-Marquardt o Nelder-Mead.

Este método se resume en los siguientes pasos:

Paso 1. Convertir cada respuesta y_i en función de deseabilidad d_i , considerando si se desea maximizar, minimizar o lograr un valor objetivo.

Paso 2. Combinar las deseabilidades individuales por medio de la Deseabilidad Global que es una media geométrica. Considerar un vector de pesos que pondera la importancia de cada respuesta.

Paso 3. Maximizar la deseabilidad global por medio de un algoritmo como Levenberg-Marquardt o Nelder-Mead, considerando los límites de control para cada variable.

Paso 4. Consiguiendo la maximización de la deseabilidad global se localiza el vector óptimo global de las variables de control que se usa para poder predecir las respuestas óptimas deseadas.

3.5.2. MOORA

Dentro de los métodos de optimización multiobjetivo se encuentran los llamados problemas de toma de decisiones con criterios múltiples (MCDM), que cuentan con diversas técnicas de acuerdo a las características del problema en cuestión. Y por medio de implementar programación matemática y un método de toma de decisiones de una manera entrelazada logran resolver esta situación de balanceo de varias respuestas. En la mayoría de los métodos del MCDM, el tomador de decisiones desempeña un papel importante en el suministro de información para construir un modelo de preferencia que se explota mediante el método de programación matemática para encontrar soluciones que se adapten mejor a las preferencias de la toma de decisiones, y teniendo como resultado un orden o también llamado ranking de las mejores respuestas encontradas. (Miettinen, 2008).

3.5. Técnicas de optimización seleccionadas

El método MOORA (Multi-Objective Optimization on the Basis of Ratio Analysis) propuesto por Brauers y Zavadskas (2006) es un método muy simple de implementar, incluso en una hoja de cálculo. El método realiza las sumas de rendimientos normalizados de costo beneficio y calcula su diferencia para representar el rendimiento global de cada una de las alternativas en forma de un índice o ranking. Los pasos son los siguientes:

Paso 1. Se identifican las variables independientes $x_{ij}, i = 1, 2, \dots, n; j = 1, 2, \dots, r$ donde n es el número de situaciones experimentales y r es el número de respuestas. Después se procede a construir la matriz de decisión.

$$D = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1r} \\ x_{21} & x_{22} & \cdots & x_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nr} \end{bmatrix} \quad (3.25)$$

Paso 2. Se normaliza la matriz de decisión por medio de dividir cada valor de x_{ij} entre su norma euclidiana para normalizar cada elemento x_{ij}^* y obteniendo la matriz normalizada D^* (Yang y Chou, 2005):

$$x_{ij}^* = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n x_{ij}^2}}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, r \quad (3.26)$$

Paso 3. Se obtiene la matriz de decisión normalizada ponderada V que se calcula multiplicando cada elemento de la matriz D^* por el peso correspondiente a la importancia de cada respuesta w_j y que cumple $\sum_{j=1}^r w_j = 1$,

$$V = [X_{ij}]_{m \times r}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, r \quad (3.27)$$

$$X_{ij} = x_{ij}^* w_j \quad (3.28)$$

Paso 4. Calcular el índice MOORA Y_i :

$$Y_i = \sum_{j=1}^t X_{ij} - \sum_{j=t+1}^r X_{ij} \quad (3.29)$$

donde t es el número de respuestas a maximizar, $(r-t)$ es el número de respuestas

3.5. Técnicas de optimización seleccionadas

a minimizar y Y_i es el ranking del i -ésimo caso experimental con respecto a todas las respuestas. A mayor valor de Y_i se produce un mejor rendimiento de respuesta múltiple.

3.5.3. MOORA AD

En el presente trabajo además se propone el uso de una propiedad aditiva que aparece en Phadke (1989) en diseño robusto, y se utiliza en el método MOORA en Kuo et al. (2008) para casos de optimización empleando técnicas multicriterio, que calcula un promedio del efecto de cada factor en sus diferentes niveles y elige el mayor de ellos como mejor respuesta. Por ejemplo si se tiene un factor A con dos niveles y tres experimentos, se calcula el efecto de la respuesta media de A en el nivel 1, m_{A1} :

$$m_{A1} = \frac{1}{3}(Y_1 + Y_2 + Y_3) \quad (3.30)$$

donde Y_1, Y_2, Y_3 son los índices MOORA en donde el factor A se encuentra en el nivel 1. Para el efecto de la respuesta media de A en el nivel 2, m_{A2} :

$$m_{A2} = \frac{1}{3}(Y_4 + Y_5 + Y_6) \quad (3.31)$$

donde Y_4, Y_5, Y_6 son los índices MOORA en donde el factor A se encuentra en el nivel 2. Una vez teniendo ambos promedio se elige el de mayor valor como el efecto que tiene mayor impacto en la respuesta.

Considerando la selección de factores se propone un método denominado MOORA AD añadiendo un paso 5 en el método MOORA.

Paso 5. Calcular un promedio del efecto de cada factor en sus diferentes niveles y elegir el mayor de ellos como mejor respuesta.

3.5.4. TOPSIS

El método TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) desarrollado por Hwang y Yoon (1981), es un ranking basado en dimensiones que elige la alternativa que simultáneamente tienen la distancia más corta de la solución ideal positiva y la distancia más alejada de la solución ideal negativa. La solución ideal positiva maximiza los beneficios y minimiza el costo, y la solución ideal negativa

3.5. Técnicas de optimización seleccionadas

maximiza el costo y minimizan los beneficios (Majid et al., 2012). Los pasos son los siguientes:

Paso 1. Se identifican las variables independientes $x_{ij}, i = 1, 2, \dots, n; j = 1, 2, \dots, r$ donde n es el número de situaciones experimentales y r es el número de respuestas. Después se procede a construir la matriz de decisión.

$$D = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1r} \\ x_{21} & x_{22} & \cdots & x_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nr} \end{bmatrix} \quad (3.32)$$

Paso 2. Se normaliza la matriz de decisión por medio de dividir cada valor de x_{ij} entre su norma euclidiana para normalizar cada elemento x_{ij}^* y obteniendo la matriz normalizada D^* (Yang y Chou, 2005):

$$x_{ij}^* = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n x_{ij}^2}}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, r \quad (3.33)$$

Paso 3. Se obtiene la matriz de decisión normalizada ponderada V que se calcula multiplicando cada elemento de la matriz D^* por el peso correspondiente a la importancia de cada respuesta w_j y que cumple $\sum_{j=1}^r w_j = 1$,

$$V = [X_{ij}]_{m \times r}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, r \quad (3.34)$$

$$X_{ij} = x_{ij}^* w_j \quad (3.35)$$

Paso 4. Calcular la solución ideal positiva y negativa:

$$d^+ = (X_1^+, X_2^+, \dots, X_r^+) \quad (3.36)$$

$$X_j^+ = \{(max_i X_{ij} \mid j \in J), i = 1, 2, \dots, n\} \quad (3.37)$$

$$d^- = (X_1^-, X_2^-, \dots, X_r^-) \quad (3.38)$$

$$X_j^- = \{(min_i X_{ij} \mid j \in J'), i = 1, 2, \dots, n\} \quad (3.39)$$

3.5. Técnicas de optimización seleccionadas

donde J está relacionado con algún criterio de beneficio y J' está relacionado con algún criterio de costo

Paso 5. Calcular la distancia del i -ésimo experimento a la solución ideal positiva y negativa (Yang y Chou, 2005):

$$d_i^+ = \sqrt{\sum_{j=1}^r (X_{ij} - X_j^+)^2}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, r \quad (3.40)$$

$$d_i^- = \sqrt{\sum_{j=1}^r (X_{ij} - X_j^-)^2}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, r \quad (3.41)$$

Paso 4. Calcular el índice TOPSIS C_i , también llamado coeficiente de cercanía:

$$C_i = \frac{d_i^-}{d_i^- + d_i^+}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, r \quad (3.42)$$

3.5.5. TOPSIS AD

Así como en el método MOORA, se propone utilizar la propiedad aditiva que aparece en Phadke (1989) y Kuo et al. (2008) en casos de optimización empleando técnicas multicriterio, para calcular el promedio del efecto de cada factor en sus diferentes niveles. El método TOPSIS AD surge añadiendo esta propiedad como paso 7 en el método TOPSIS.

Paso 7. Calcular un promedio del efecto de cada factor en sus diferentes niveles y elegir el mayor de ellos como mejor respuesta.

3.5.6. MULTIMOORA

El método MOORA puede construirse con una variante en el paso 4 mencionado, en lugar de basarse en el cálculo del "ratio system", se realiza un "Reference point", y la combinación de estos índices junto con el índice "Full multiplicative form" genera el índice MULTIMOORA o MMOORA propuesto por Brauers y Zavadskas (2010). El Ratio system emplea a la sumatoria como operador de agregación con los datos normalizados, el Reference point calcula un punto ideal de referencia y mide las dis-

3.5. Técnicas de optimización seleccionadas

tancias de las alternativas a este punto ideal. El Full multiplicative form usa como operador de agregación a la multiplicación, de cada índice se calcula el ranking y finalmente se calcula el ranking MULTIMOORA por medio de la teoría de dominancia. La metodología se menciona en [Ceballos et al. \(2015\)](#) y los pasos son los siguientes:

Paso 1. Se identifican las variables independientes $x_{ij}, i = 1, 2, \dots, n; j = 1, 2, \dots, r$ donde n es el número de situaciones experimentales y r son el número de respuestas. Después se procede a construir la matriz de decisión.

$$D = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1r} \\ x_{21} & x_{22} & \cdots & x_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nr} \end{bmatrix} \quad (3.43)$$

Paso 2. Se normaliza la matriz de decisión por medio de dividir cada valor de x_{ij} entre su norma euclidiana para normalizar cada elemento x_{ij}^* y obtener la matriz normalizada D^* ([Yang y Chou, 2005](#)):

$$x_{ij}^* = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n x_{ij}^2}}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, r \quad (3.44)$$

Paso 3. Se obtiene la matriz de decisión normalizada ponderada V que se calcula multiplicando cada elemento de la matriz D^* por el peso correspondiente a la importancia de cada respuesta w_j y que cumple $\sum_{j=1}^r w_j = 1$,

$$V = [X_{ij}]_{m \times r}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, r \quad (3.45)$$

$$X_{ij} = x_{ij}^* w_j \quad (3.46)$$

Paso 4. Calcular el índice de Ratio system, Y_i :

$$Y_i = \sum_{j=1}^t X_{ij} - \sum_{j=t+1}^r X_{ij} \quad (3.47)$$

donde t es el número de respuestas a maximizar, $(r-t)$ es el número de respuestas a minimizar y Y_i es el ranking del i -ésimo caso experimental con respecto a todas las respuestas. A mayor valor de Y_i se produce un mejor rendimiento de respuesta

3.5. Técnicas de optimización seleccionadas

múltiple.

Paso 5. Cálculo del índice de Reference point P_i , primero se calcula el punto de referencia como $RF = \{v_1^+, v_2^+, \dots, v_n^+\}$, donde $v_j^+ = \max_i(v_{ij})$ si el j -ésimo criterio es de beneficio, y $v_j^- = \min_i(v_{ij})$ si es de coste. Posteriormente se aplica la métrica min-max de Chebyshev:

$$P_i = \underset{\min}{i} \left(\underset{\max}{j} | v_j^+ - v_{ij} | \right) \quad (3.48)$$

y las alternativas se ordenan de forma creciente y las mejores alternativas son las de menor valor.

Paso 6. Cálculo de Full multiplicative form, U_i :

$$U_i = \frac{\prod_{j=1}^t x_{ij}^{w_j}}{\prod_{j=t+1}^r x_{ij}^{w_j}} \quad (3.49)$$

donde $\prod_{j=1}^t x_{ij}^{w_j}$ son los criterios de beneficio y $\prod_{j=t+1}^r x_{ij}^{w_j}$ son los de coste. Después las alternativas se ordenan de forma decreciente, siendo las mejores alternativas las de valor mayor.

Paso 7. Teoría de dominancia. Se unen los rankings por medio de la teoría de dominancia y se genera un ranking final. Se parte de la base que los tres índices tienen la misma importancia. Esto se aplica de la siguiente forma:

1. Dominancia

- a) Absoluta: Alternativa en primera posición en los tres rankings.
- b) General: cuando una alternativa supera a otra en al menos dos de los tres índices.
- c) Transitividad: si a domina a b y b domina a c , entonces a domina a c .
- d) Dominancia total: una alternativa domina a otra en los tres índices.

2. Igualdad

- a) Absoluta: dos alternativas son exactamente iguales si tiene los mismos rankings en los tres índices.
- b) Parcial: cuando dos alternativas tienen el mismo ranking en un índice y en otro la primera domina a la segunda y en el tercer ranking la segunda domina a la primera.
- c) Razonamiento circular: cuando no se cumple la transitividad.

3.5. Técnicas de optimización seleccionadas

3.5.7. Redes Neuronales Artificiales

Dentro de la inteligencia artificial y el Machine Learning encontramos a las Redes Neuronales Artificiales (ANN, Artificial Neural Networks), que son además parte fundamental del área del Deep Learning. Las redes neuronales artificiales son modelos computacionales inspirados en el sistema nervioso de los seres vivos. Tienen la capacidad de adquirir y mantener conocimientos (basados en información) y pueden definirse como un conjunto de unidades de procesamiento, representadas por neuronas artificiales, interconectadas por muchas interconexiones (sinapsis artificiales), implementadas por vectores y matrices de pesos sinápticos (Da Silva et al., 2017).

3.5.7.1. La neurona artificial

El trabajo pionero en el desarrollo de una neurona artificial aparece con el modelo neuronal de McCulloch y Pitts (1943), donde un modelo matemático al que se le ingresan datos y los procesa, devuelve un resultado, esta neurona tiene salida binaria, a continuación se muestra su estructura:

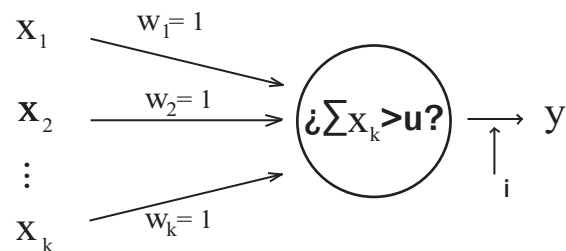


Figura 3.8: Neurona McCulloch-Pitts.

donde, y es la salida binaria $\{0, 1\}$, además, si $y = 0$ la neurona permanece en reposo y si $y = 1$ la neurona se activa, las entradas x_1, x_2, \dots, x_k son las entradas excitativas también binarias $\{0, 1\}$ con pesos igual a la unidad, i es una entrada de carácter inhibitorio y si está encendido la neurona no se activa, por último si la suma de las entradas excitativas es mayor a u (umbral) la neurona se activa, de lo que resulta:

$$g(x) = \begin{cases} 1 & \text{si } \sum_{k=1}^n x_k > u, i = 0, \\ 0 & \text{d.o.m.} \end{cases} \quad (3.50)$$

Posteriormente esta neurona fue mejorada por Rosenblatt (1957), el cual propuso el

3.5. Técnicas de optimización seleccionadas

perceptrón que era una neurona capaz de aprender:

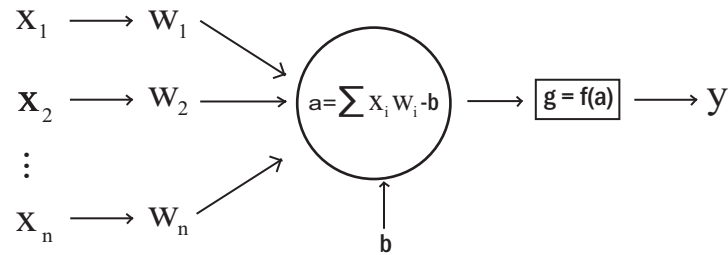


Figura 3.9: Neurona Rosenblatt.

donde, x_1, x_2, \dots, x_n son las señales de entrada que generalmente se normalizan para mejorar la eficiencia computacional de los algoritmos de aprendizaje, w_1, w_2, \dots, w_n son los pesos sinápticos que ponderan (cuantifican su relevancia respecto a la funcionalidad de la neurona) cada una de las variables de entrada, Σ es el agregador lineal que reúne las señales de entrada ponderadas por los pesos sinápticos para producir un voltaje de activación, b es el umbral de activación o sesgo. El potencial de activación a es el resultado producido por la diferencia entre el agregador lineal y el umbral de activación, si este valor es positivo, es decir, si $a \geq b$, entonces la neurona produce un potencial excitador; de lo contrario, será inhibitorio. Función de activación g , y la señal de salida y que es el valor final producido por la neurona dado un conjunto particular de señales de entrada (Da Silva et al., 2017).

3.5.7.2. Funciones de activación

Note que en el caso de clasificación la salida puede tomar valores discretos, y en el caso de regresión puede tomar valores de $-\infty$ a $+\infty$, por lo que se hace necesario el empleo de un mecanismo de mapeo entre la entrada y la salida de la neurona. Este mecanismo que limita la salida a un rango de valores razonable se le conoce como función de activación.

Existen dos tipos de funciones de activación de acuerdo a sus dominios: funciones parcialmente diferenciables y funciones completamente diferenciables (Da Silva et al., 2017).

Funciones parcialmente diferenciables

Las funciones de activación parcialmente diferenciables son funciones con puntos cuyas derivadas de primer orden no existen. Las tres funciones principales son:

* **Función escalón:** (también conocida como función umbral, limitador estricto o

3.5. Técnicas de optimización seleccionadas

función Heaviside), se usó en el modelo neuronal de McCulloch y Pitts. Hay dos versiones de esta función de activación, binaria y bipolar (función signo o limitador estricto simétrico). la función binaria es la siguiente:

$$g(a) = \begin{cases} 1 & \text{si } a \geq 0, \\ 0 & \text{si } a < 0 \end{cases} \quad (3.51)$$

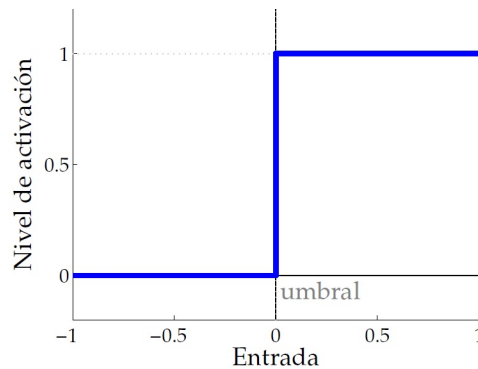


Figura 3.10: Función de activación escalón binaria. Fuente: [Berzal \(2018\)](#)

La función bipolar es la siguiente:

$$g(a) = \begin{cases} 1 & \text{si } a \geq 0, \\ -1 & \text{si } a < 0 \end{cases} \quad (3.52)$$

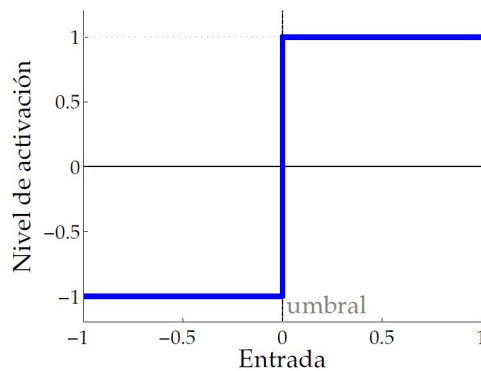


Figura 3.11: Función de activación escalón bipolar. Fuente: [Berzal \(2018\)](#)

* **Función de rampa simétrica:** (también conocida como función lineal a tramos), es una función de activación continua, los valores devueltos por esta función son

3.5. Técnicas de optimización seleccionadas

iguales a los valores del potencial de activación en sí mismos cuando se definen dentro del rango $[-c, c]$:

$$g(a) = \begin{cases} 1 & \text{si } a > 1, \\ a & \text{si } -1 \leq a \leq 1 \\ -1 & \text{si } a < -1 \end{cases} \quad \text{para el rango} [-1, 1] \quad (3.53)$$

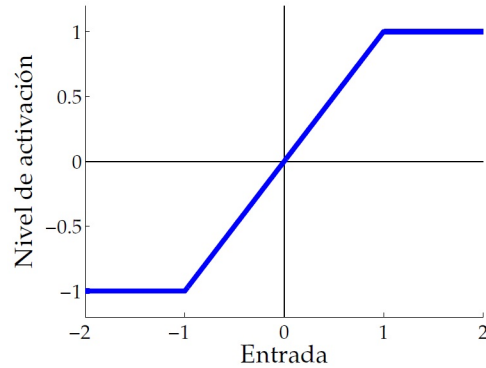


Figura 3.12: Función de rampa simétrica. Fuente: [Berzal \(2018\)](#)

Funciones completamente diferenciables

Las funciones de activación completamente diferenciables son funciones cuyas derivadas de primer orden existen para todos los puntos de su dominio. Las funciones más usadas en redes neuronales artificiales, son las de tipo sigmoidal:

* **Función logística:** es una función que produce una salida de valores siempre reales con rango entre $[0, 1]$. Su expresión es:

$$g(a) = \frac{1}{1 + e^{-a}} \quad (3.54)$$

3.5. Técnicas de optimización seleccionadas

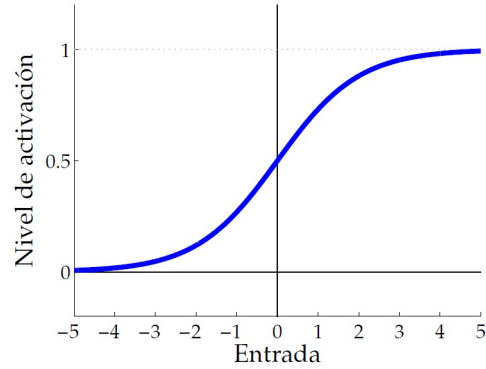


Figura 3.13: Función logística. Fuente: Berzal (2018)

* **Función tangente hiperbólica:** el resultado de salida de esta función siempre asumirá valores reales entre $[-1, 1]$:

$$g(a) = \tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} = \frac{1 - e^{-2a}}{1 + e^{-2a}} = \frac{1}{1 + e^{-2a}} - 1 \quad (3.55)$$

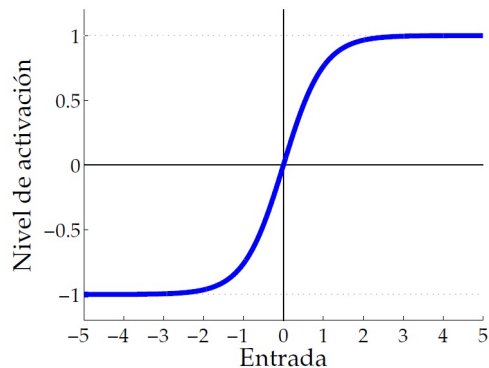


Figura 3.14: Función tangente hiperbólica. Fuente: Berzal (2018)

* **Función gaussiana:** la salida de neuronas producirá resultados iguales para aquellos valores potenciales de activación a colocados a la misma distancia de su centro (media). La curva es simétrica a este centro y la función gaussiana viene dada por:

$$g(a) = e^{-\frac{(a - c)^2}{2\sigma^2}} \quad (3.56)$$

donde, c es un parámetro que indica el centro de la función Gaussiana y σ denota la

3.5. Técnicas de optimización seleccionadas

desviación estándar.

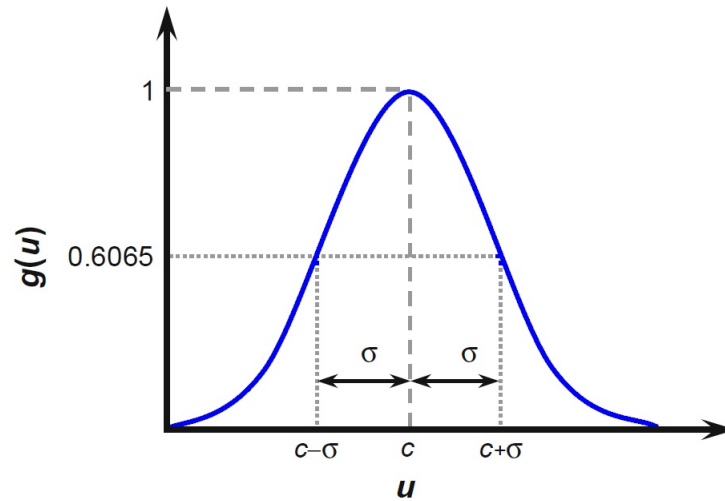


Figura 3.15: Función gaussiana. Fuente: [Da Silva et al. \(2017\)](#)

* **Función lineal:** es la función identidad:

$$g(a) = a \tag{3.57}$$

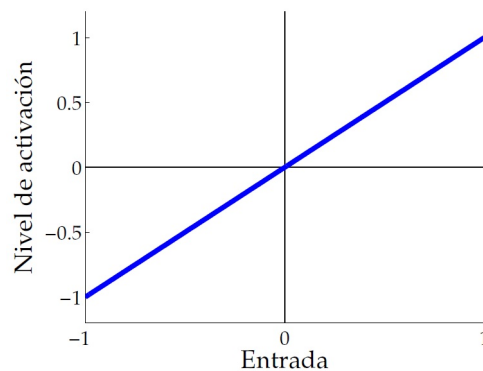


Figura 3.16: Función lineal. Fuente: [Berzal \(2018\)](#)

Función de activación lineal rectificada: (también conocida como ReLU, Rectified Linear Units), la cual no requiere de funciones trascendentales ni de operaciones aritméticas, por lo que resulta extremadamente eficiente [Berzal \(2018\)](#):

3.5. Técnicas de optimización seleccionadas

$$g(a) = \begin{cases} a & \text{si } a \leq 0, \\ 0 & \text{si } a < 0 \end{cases} \quad (3.58)$$

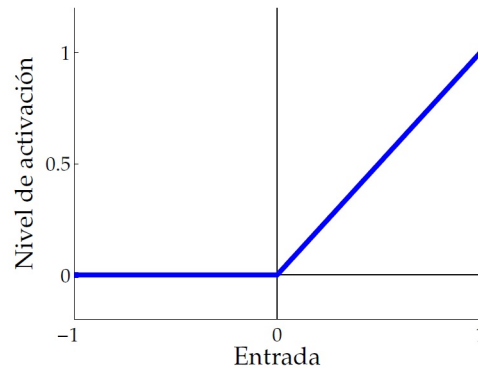


Figura 3.17: Función ReLU. Fuente: [Berzal \(2018\)](#)

Usualmente, es de interés que la función de activación de una neurona sea no lineal, estrictamente creciente, continua y derivable. Y las funciones de activación sigmoideas satisfacen esos requisitos, lo que las hace especialmente útiles en redes neuronales que se entrenan usando backpropagation. Backpropagation es un algoritmo de propagación de errores que, dados los errores observados en la capa de salida, propaga esos errores hacia atrás en la red para ir ajustando sus parámetros internos ([Berzal, 2018](#)).

3.5.7.3. Arquitecturas de Redes Neuronales

Las redes neuronales en general constan de tres partes, denominadas capas y son:

a) Capa de entrada: esta capa contiene neuronas que reciben sus entradas del ambiente externo de estudio, que generalmente se normalizan para lograr un mejor desempeño de la red.

b) Capa oculta: contiene neuronas que son las responsables de extraer patrones asociados con el proceso de estudio. En estas capas se realiza la mayor parte del procesamiento de una red.

c) Capa de salida: esta capa de neuronas se encarga de presentar las salidas de la red, como resultado del procesamiento de las neuronas en capas anteriores.

Las principales arquitecturas de redes neuronales artificiales, se pueden dividir como

3.5. Técnicas de optimización seleccionadas

sigue: red de alimentación (feedforward) de una sola capa y múltiples capas, redes recurrentes y redes de malla.

Redes feedforward de capa única

Son las redes neuronales más simples, compuestas de dos capas, una capa de entrada que no realiza ningún proceso u operación y una capa de salida donde se desarrollan los procesos internos de la red, por eso se le llama red neuronal de una sola capa, ya que la capa de salida es la única que realmente hace algo en la red. Un ejemplo de este tipo de red es la neurona de Rosenblatt (perceptrón) que se observa en la figura 3.9.

Redes feedforward multicapa

Estas redes neuronales constan de al menos tres capas, capa de entrada, capa oculta y capa de salida, cada capa puede contener varias neuronas. El número de neuronas de la capa de entrada coincide con las variables de entrada. En este caso la capa oculta es donde se realiza el procesamiento mayoritario de la red neuronal y el número de neuronas ocultas dependerá de la complejidad del problema o proceso de estudio y de la cantidad de información alimentada. Y la cantidad de neuronas de salida depende de las señales de respuesta estudiadas.

Dentro de este tipo de redes se encuentran las redes Perceptrón Multicapa (MLP, Multi-Layer Perceptron) y las redes de Función de Base Radial (RBF, Radial Basis Function), que se utilizan en la solución de problemas como son la clasificación, control de procesos, aproximación de funciones, optimización, robótica, etc. Si una red neuronal tiene más de una capa oculta se le conoce comúnmente como redes neuronales profundas, haciendo alusión al *deep learning*.

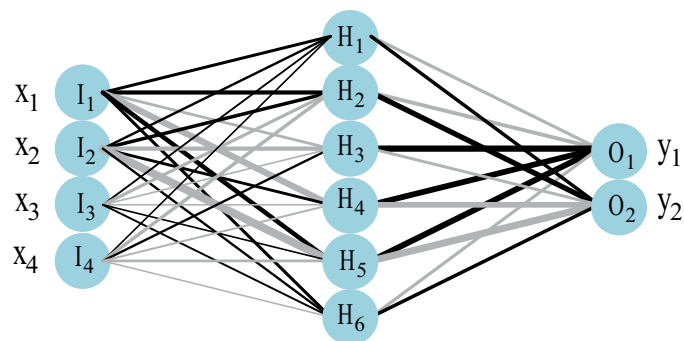


Figura 3.18: Red MLP.

La figura 3.18 muestra una red MLP con múltiples capas, compuestas de una capa de entrada con 4 neuronas, una capa oculta con 6 neuronas y una capa de salida con

3.5. Técnicas de optimización seleccionadas

2 neuronas. Este tipo de red es la empleada en esta propuesta de investigación.

Redes feedback (Recurrentes)

Estas redes presentan retroalimentación de las salidas de neuronas de una capa hacia neuronas de una capa anterior, lo que les permite un procesamiento dinámico de la información, que se puede usar en sistemas con predicción de series de tiempo, identificación y optimización de sistemas, control de procesos, etc. Entre las mas usuales se encuentran las redes Hopfield y el Perceptron con retroalimentación entre neuronas de distintas capas, cuyos algoritmos de aprendizaje en sus entrenamientos se basan respectivamente en la minimización de la función energética y la regla delta generalizada (Da Silva et al., 2017).

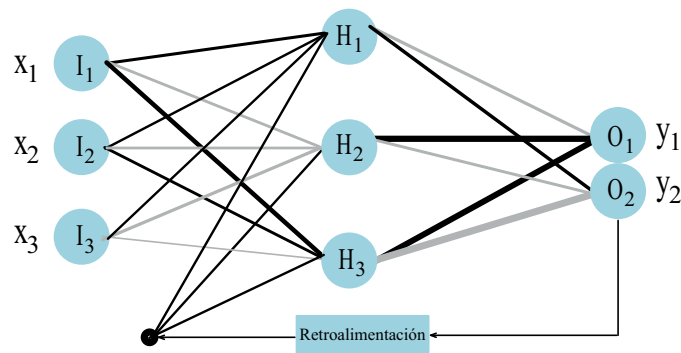


Figura 3.19: Red feedback (Recurrente).

La figura 3.19 se muestra una red perceptrón con retroalimentación, donde una de sus señales de salida se retroalimenta a la capa oculta.

Redes de Malla

Estas redes con estructuras de malla consideran la disposición espacial de las neuronas para extraer patrones, ya que esta localización de las neuronas está directamente relacionada con el ajuste de pesos y umbrales sinápticos. Estas redes se emplean en reconocimiento de patrones, agrupación de datos, gráficos, etc. La red Kohonen es la red de malla más conocida, y basa su entrenamiento por medio de un proceso competitivo. En la figura 3.20 se muestra un ejemplo de esta red con neuronas dispuestas dentro de un espacio bidimensional (Da Silva et al., 2017).

3.5. Técnicas de optimización seleccionadas

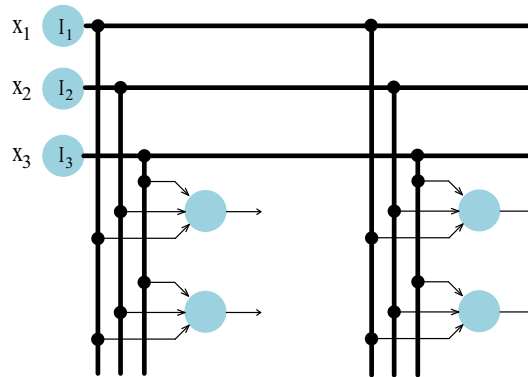


Figura 3.20: Red de Malla. Fuente: Modificado de [Da Silva et al. \(2017\)](#).

3.5.7.4. Entrenamiento y aprendizaje

Ya que se eligió la arquitectura de la red neuronal a implementar según nuestro tipo de problema y cantidad de información se hace mención a una de las características más importantes de las redes neuronales artificiales, nos referimos a su capacidad de aprender de las muestras que se le proveen, que nos da una idea del comportamiento del sistema o proceso. Este entrenamiento consiste en ajustar pesos y umbrales entre las neuronas para que la red aprenda las relaciones entre las entradas y salidas, para posteriormente generalizar soluciones para cualquier otro valor de entrada y que esa solución sea muy cercana a la solución deseada. Esta serie de pasos de entrenamiento de la red se conoce como algoritmo de aprendizaje, que inicia con dividir nuestra muestra de entrada en dos subconjuntos llamados de entrenamiento y prueba; divididos del 60 a 80% y del 10 a 40% respectivamente ([Da Silva et al., 2017](#)). La idea es que la red aprenda del subconjunto de entrenamiento y verifique posteriormente la capacidad de solución de la red en el subconjunto de prueba, es importante mencionar que las redes neuronales aprenden por repetición (a cada ciclo de entrenamiento de la muestra se le llama época), y entre más datos se tengan en la entrada y más veces se entrene la red mejores resultados se espera obtener.

Según el método de aprendizaje podemos mencionar:

Aprendizaje supervisado

Este proceso de aprendizaje está basado en un entrenamiento controlado por un experto que conoce las salidas deseadas a partir de los datos de entrada, además de que conoce el funcionamiento del proceso o fenómeno de estudio. El experto revisa y controla los datos de salida modificando los pesos y umbrales y realizando comparaciones hasta encontrar la menor discrepancia con respecto a las respuestas deseadas. Las redes MLP emplean este tipo de aprendizaje y es el que se desarrolla en el presente

3.5. Técnicas de optimización seleccionadas

trabajo.

Aprendizaje no supervisado

En este tipo de aprendizaje no es necesaria la intervención de un experto para ajustar los pesos y umbrales, por lo que no se requiere conocer las respuestas deseadas. La idea es encontrar las características, relaciones y similitudes de los datos de entrada, identificando subconjuntos que presentan estas similitudes dentro de la red.

Aprendizaje por refuerzo

Es una variante del aprendizaje supervisado ya que se analiza y se le indica a la red si la respuesta es aceptable o no, comparando con un resultado deseado y así el algoritmo de aprendizaje por ejemplo un algoritmo estocástico selecciona el ajuste probabilísticamente considerando un conjunto finito de posibles soluciones que pueden recompensarse si tienen posibilidades de generar resultados satisfactorios.

Entrenamiento de redes multicapa

Las redes neuronales feedforward multicapa en ocasiones llamadas MLP (Perceptrón Multicapa) se entrenan por medio de un algoritmo de propagación de errores hacia atrás llamado *backpropagation* o simplemente *backprop* y la combinación con una técnica de optimización para obtener un algoritmo de entrenamiento para redes multicapa. La propagación de errores hacia atrás permite calcular el gradiente del error conforme varían los parámetros de la red. Esta forma de entrenamiento se puede explicar de la siguiente manera: primeramente se lleva a cabo una propagación forward que va desde la capa de entrada y termina en la capa de salida, cada neurona realiza una suma ponderada de las entradas de acuerdo a los pesos y se le aplica una función de activación, hasta llegar al resultado de la red. Una vez realizado esto se procede a calcular el gradiente del error por medio de una función auxiliar *error o función de coste*, que puede ser por ejemplo la raíz cuadrada media del error o el error absoluto medio; con la idea de optimizar los parámetros de la red. Una vez que tenemos listo el gradiente se procede a realizar el *backpropagation* que se encarga de calcular los deltas necesarios para ajustar los parámetros de la red y determinar la aportación de cada neurona al error. Estos pasos se repiten muchas veces (épocas) buscando disminuir el error hasta un criterio establecido previamente. Posteriormente se puede emplear un conjunto de valores diferentes (validación) para evaluar la predicción de la red.

Si se observa que el error en el conjunto de validación sube gradualmente se sospecha de que se empieza a sobreaprender o a tener lo denominado *overfitting* para lo que se recomienda detener el entrenamiento. Además se debe tener presente que en la optimización de la función auxiliar de error algunos métodos de optimización pueden quedar atrapados en un óptimo local por lo que se aconseja reiniciar la red con diferentes parámetros, es por esto que cuando se lleva a cabo el entrenamiento varias

3.5. Técnicas de optimización seleccionadas

veces no siempre se llega al mismo resultado de manera exacta. ([Berzal, 2018](#)).

Con lo anterior tenemos claro que para poder entrenar una red neuronal es necesario el control de diversos parámetros que modulan la red, llamados hiperparámetros como números de capas, número de neuronas por capa, tasa de aprendizaje, la técnica de optimización, etc. Por lo que en la literatura se hacen algunas recomendaciones pero el objetivo es realizar diversas pruebas e ir encontrando la mejor configuración de los valores de los hiperparámetros.

Es importante mencionar a los algoritmos de optimización de la función objetivo para minimizar los errores, para el caso unidimensional se cuenta con: búsqueda ternaria, búsqueda de la sección áurea, interpolación parabólica inversa, método Brent, etc. Y en el caso multidimensional con relevancia en las problemáticas abordadas en la propuesta de trabajo se tienen: Gradiente descendente (método Nelder-Mead, método Powell, tasa de aprendizaje), Gradiente conjugado (algoritmo de Fletcher-Reeves, algoritmo Polak-Ribière), Gradiente estocástico (Promediado de Polyak, tasas de aprendizaje adaptativas), Momentos (Momentos tradicionales, Nesterov), Técnicas de optimización de segundo orden (Newton, Gauss-Newton, métodos quasi-Newton como Davidon-Fletcher-Powell o Broyden-Fletcher-Goldfarb-Shanno (BFGS), Levenberg-Marquardt) y otros métodos más recientes son: AdaGrad, AdaDelta, Rprop, AdaSe-cant, Adam, AdaMax, entre otros. Para una revisión más exhaustiva de los métodos mencionados anteriormente y otros más, vea [Berzal \(2018\)](#).

3.5.7.5. Método de Optimización multiobjetivo con Redes Neuronales

[Tong y Hsieh \(2001\)](#) muestran una metodología en la que se establece el uso de redes neuronales multicapa en optimización multiobjetivo, la cuál se resume en los siguientes pasos:

Paso 1. (Red inversa). Entrenar una red neuronal con las respuestas observadas como entradas y los factores de control como salidas.

Paso 2. Obtenida la red entrenada se predice con los datos de las respuestas deseadas para obtener los factores de entrada optimizados.

Paso 3. (Red directa). Entrenar una red neuronal con los factores de control como entradas y las respuestas observadas como salidas.

Paso 4. Obtenida la red entrenada se realiza la predicción con los factores de entrada optimizados del paso 2 y así obtener los valores objetivos optimizados.

Capítulo 4

Metodología

En esta sección se desarrolla la metodología del experimento Monte Carlo para cada uno de los métodos y se plantea una métrica para identificar cuál o cuáles de ellos tienen un mejor desempeño.

4.1. Estudio comparativo

Se ilustrará el estudio comparativo con el análisis de tres casos reportados en la literatura y que son de interés comercial e industrial. Los métodos empleados se mencionaron en la sección 3.5 y son: Función de Deseabilidad (DES), MOORA (MOO), TOPSIS (TOP), MMOORA (MMO), MOORA AD (MAD), TOPSIS AD (TAD) y Redes Neuronales (NEU y NET).

4.2. Simulación

Se realizó un estudio de simulación para analizar la efectividad de los métodos, el experimento Monte Carlo consistió en lo siguiente:

4.2.1. Factores de estudio

- Factor método: Se emplearon ocho métodos de estudio para comparar los métodos mencionados en la sección 3.5.
- Factor tipo de Diseño Experimental: Se usan tres situaciones con diferentes

4.3. Métrica usada para comparar los métodos

diseños experimentales:

- Diseño Taguchi ($L8$): con cuatro factores de control y dos respuestas. (Ver sección 5.2.1).
- Diseño Box–Behnken: con tres factores de control y dos respuestas. (Ver sección 5.2.2).
- Diseño Central Compuesto: con tres factores de control y seis respuestas. (Ver sección 5.2.3).
- Factor correlación en superficie de respuesta: con 7 niveles (0.9, 0.5, 0.2, 0, -0.2, -0.5, -0.9). Que indican correlación entre respuestas alta, media y baja, tanto positiva como negativa, y considera al centro el caso independiente.

De los anteriores factores se obtuvieron 168 combinaciones que se estudiaron con 5000 réplicas por cada experimento, obteniendo como respuesta un vector de respuestas medias y un vector de varianzas. Al vector de respuestas medias se le aplicó una métrica para poder comparar la efectividad de cada método.

4.3. Métrica usada para comparar los métodos

Se emplearon dos métricas:

La primer métrica es el GPE, Error Porcentual Global (Global Percentage Error), que se emplea para poder comparar los métodos y su desempeño, el GPE evalúa la distancia de la respuesta estimada respecto a su valor ideal o deseado (Rocha et al., 2015). El mejor valor de esta métrica es el cero o muy cercano a este, su expresión es la siguiente:

$$GPE = \sum_{i=1}^m \left| \frac{\hat{y}_i}{T_i} - 1 \right| \quad (4.1)$$

donde, \hat{y}_i es la respuesta media estimada, T_i es el valor objetivo deseado y m es la cantidad de respuestas.

La segunda métrica aparece en Costa y Lourenço (2016) llamada desviación relativa del objetivo, RTD (Relative Target Deviation), que mide la distancia de la respuesta estimada con respecto al valor deseado, es adimensional y entre más bajo el valor mejor, cero es el valor ideal, e indica que todas las respuestas están en el objetivo. Su expresión es:

$$RTD = \sum_{i=1}^m \frac{|\hat{y}_i - T_i|}{T_i} \quad (4.2)$$

En el cálculo de ambas métricas se obtienen los mismos resultados, por lo que se reportan los resultados como GPE/RTD.

4.4. Implementación en software R

Todos los métodos y simulaciones fueron programados en el software R ([R Core Team, 2019](#)); el código desarrollado se muestra en [ANEXOS](#). Los cuatro primeros pasos listados se emplearon en la solución de cada uno de los casos descritos en la sección [5.2](#). El método consistió en los siguientes pasos:

1. Se programó la función de Deseabilidad con ayuda del paquete desirability de [Kuhn \(2016\)](#).
2. Después se programaron los métodos multicriterio MOORA, TOPSIS, MMOORA, MOORA AD y TOPSIS AD, con ayuda del paquete MCDM de [Ceballos \(2016\)](#).
3. Se programaron redes neuronales feedforward multicapa siguiendo la idea de [Tong y Hsieh \(2001\)](#), para el primer método llamado NEU se usó el paquete Neuralnet de [Fritsch et al. \(2019\)](#) obteniendo redes multicapa con dos capas ocultas, algoritmo rprop+, y funciones de activación sigmoide logística, el entrenamiento termina en el momento de obtener el mínimo error de entrenamiento, consiguiendo esto en varios ciclos de aprendizaje (épocas). La cantidad de neuronas por cada capa oculta se obtuvo por prueba y error con pruebas de 3 a 12 neuronas (esto para todos los casos). Para el segundo método llamado NET se empleó el paquete Nnet de [Venables y Ripley \(2016\)](#), obteniendo redes neuronales de una sola capa oculta.
 - a) Caso 1:
 - 1) NEURALNET tuvo una red inversa y directa con una arquitectura (2,10,8,4) y (4,10,8,2) respectivamente.
 - 2) NNET tuvo una red inversa y directa con una arquitectura (2,3,4) y (4,4,2).
 - a) Caso 2:
 - 1) NEURALNET tuvo una red inversa y directa con una arquitectura (2,10,8,3) y (3,10,8,2) respectivamente.

4.4. Implementación en software R

- 2) NNET tuvo una red inversa y directa con una arquitectura (2,3,3) y (3,4,2).
- a) Caso 3:
- 1) NEURALNET tuvo una red inversa y directa con una arquitectura (6,3,3) y (3,4,6) respectivamente.
 - 2) NNET tuvo una red inversa y directa con una arquitectura (6,3,3) y (3,3,6).
4. Se realizó una regresión para generar los modelos que ajustan a cada una de las respuestas, ya que se usan en los métodos DES, MAD y TAD; y para llevar a cabo la generación de modelos con residuales normales multivariados que serán donde trabajarán los métodos.
 5. Con ayuda del paquete mvtnorm de [Genz et al. \(2019\)](#), se generan normales multivariadas con media = 0, varianza = 1 y con matriz de correlación donde se controla la correlación entre las respuestas, para tener diferentes escenarios y verificar el comportamiento de los métodos planteados, estas normales multivariadas se agregan en forma de residuales a los modelos ajustados, y sobre estos nuevos modelos se implementan cada una de las metodologías. Se realizaron 5000 repeticiones en esta simulación obteniendo un vector de medias y un vector de varianzas para cada respuesta y para cada método.
 6. Por último se emplea la métrica GPE/RTD en cada uno de los métodos, usando los datos de respuestas óptimas deseadas indicadas en cada caso estudiado, reportando una tabla con los valores GPE/RTD para cada método, considerando a los valores más pequeños y cercanos a cero como los mejores valores y por lo tanto representantes de los métodos mas eficientes.

Capítulo 5

Resultados

5.1. Resultados de la simulación

En las siguientes tablas se muestran los resultados obtenidos después de aplicar la metodología descrita para la simulación en la sección anterior, estos resultados son para cada uno de los casos abordados y se emplea la métrica GPE/RTD para comparar la eficiencia de los métodos bajo diferentes escenarios de correlación.

5.1.1. Caso 1: Diseño Taguchi

La tabla 5.1 incluye los vectores de respuestas medias obtenidos en la simulación Monte Carlo (estas respuestas son del tipo señal ruido taguchi); la tabla 5.2 muestra los vectores de varianzas para cada método y para cada variable, y la tabla 5.3 muestra la suma de varianzas en la obtención de los óptimos para cada método haciendo uso de la propiedad de ortogonalidad de los diseños.

Tabla 5.1: Respuestas medias estimadas en la simulación para el Caso 1

Corr.	DES		MOO		TOP		MAD		TAD		MMO		NEU		NET	
	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2
0.9	-33.74	64.08	-33.55	63.55	-33.55	63.54	-33.67	63.82	-33.68	63.73	-33.59	63.66	-34.53	63.98	-34.63	64.00
0.5	-33.64	64.38	-33.52	63.88	-33.52	63.86	-33.66	64.08	-33.65	63.97	-33.55	63.96	-34.48	64.09	-34.58	64.07
0.2	-33.59	64.58	-33.53	64.15	-33.53	64.12	-33.67	64.27	-33.66	64.17	-33.55	64.20	-34.42	64.14	-34.53	64.13
0	-33.56	64.70	-33.53	64.32	-33.53	64.30	-33.68	64.38	-33.67	64.29	-33.54	64.36	-34.36	64.18	-34.48	64.16
-0.2	-33.50	64.82	-33.50	64.47	-33.49	64.45	-33.66	64.51	-33.65	64.44	-33.50	64.49	34.32	64.25	-34.43	64.21
-0.5	-33.44	64.99	-33.49	64.73	-33.49	64.72	-33.64	64.68	-33.64	64.63	-33.49	64.74	-34.26	64.32	-34.37	64.29
-0.9	-33.39	65.16	-33.48	65.01	-33.48	65.01	-33.64	64.89	-33.63	64.87	-33.48	65.01	-34.19	64.45	-34.25	64.40

5.1. Resultados de la simulación

Tabla 5.2: Varianzas individuales para el Caso 1

Corr.	DES		MOO		TOP		MAD		TAD		MMO		NEU		NET	
	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2	SN_1	SN_2
0.9	0.411	0.441	0.581	1.295	0.572	1.246	0.839	1.021	0.823	1.104	0.605	1.135	0.440	0.511	0.573	0.724
0.5	0.464	0.605	0.544	1.351	0.537	1.287	0.818	1.112	0.799	1.171	0.562	1.228	0.385	0.441	0.603	0.720
0.2	0.525	0.697	0.555	1.267	0.548	1.232	0.822	1.166	0.815	1.230	0.567	1.163	0.334	0.412	0.512	0.546
0.0	0.503	0.692	0.529	1.167	0.521	1.151	0.789	1.095	0.787	1.129	0.540	1.100	0.253	0.345	0.472	0.513
-0.2	0.536	0.701	0.540	1.071	0.537	1.054	0.818	1.060	0.818	1.089	0.550	1.021	0.249	0.286	0.420	0.444
-0.5	0.518	0.673	0.498	0.912	0.497	0.907	0.794	1.011	0.793	1.047	0.503	0.886	0.135	0.206	0.272	0.265
-0.9	0.505	0.536	0.495	0.590	0.495	0.591	0.802	0.864	0.802	0.878	0.496	0.588	0.041	0.046	0.105	0.106

Tabla 5.3: Varianzas por método para el Caso 1

Correlación	DES	MOO	TOP	MAD	TAD	MMO	NEU	NET
0.9	0.853	1.877	1.819	1.860	1.927	1.740	0.951	1.298
0.5	1.070	1.895	1.823	1.930	1.969	1.790	0.825	1.323
0.2	1.222	1.822	1.779	1.989	2.045	1.730	0.746	1.058
0.0	1.196	1.696	1.672	1.883	1.916	1.640	0.598	0.985
-0.2	1.238	1.611	1.591	1.878	1.907	1.571	0.535	0.864
-0.5	1.192	1.411	1.403	1.805	1.840	1.389	0.341	0.537
-0.9	1.042	1.085	1.086	1.665	1.680	1.084	0.086	0.212

Por último, la tabla 5.4 presenta los valores de GPE/RTD conseguidos de la simulación para el caso de la mejora de las condiciones de operación de una lavadora donde se desea minimizar el ruido y maximizar la velocidad de giro, que se describe en la sección 5.2.1.

Tabla 5.4: GPE/RTD por método para el Caso 1

Correlación	DES	MOO	TOP	MAD	TAD	MMO	NEU	NET
0.9	0.019	0.032	0.032	0.025	0.026	0.030	0.019	0.022
0.5	0.017	0.028	0.029	0.021	0.023	0.026	0.016	0.020
0.2	0.017	0.024	0.024	0.018	0.020	0.022	0.014	0.017
0	0.020	0.021	0.022	0.016	0.017	0.020	0.011	0.015
-0.2	0.024	0.020	0.020	0.015	0.016	0.019	0.009	0.013
-0.5	0.028	0.023	0.023	0.018	0.017	0.023	0.006	0.010
-0.9	0.033	0.027	0.027	0.021	0.021	0.027	0.002	0.005

Observando la tabla 5.4 es claro que los valores más bajos de GPE/RTD los tienen las redes neuronales NEU y NET, después los métodos DES, MAD y TAD con un desempeño intermedio, y los valores más altos se tienen en MMO, MOO y TOP. Respecto a la varianza de la estimación, NEU y NET muestran los valores más bajos seguidos de DES, MMO, MOO y TOP; y los valores más altos se observan en MOA y TOA. Considerando el desempeño en diferentes correlaciones, al aumentar positivamente la correlación los valores del GPE/RTD tienen un aumento gradual para todas las metodologías, excepto para la DES donde se observa una ligera disminución del error; en el caso de aumentar negativamente la correlación los valores del GPE/RTD tienen un

5.1. Resultados de la simulación

aumento gradual en todos los métodos, excepto para ambas redes neuronales donde al aumentar la correlación negativa se observa una disminución del error. De acuerdo al análisis anterior podemos decir que los mejores métodos para este caso en diseño Taguchi con dos respuestas y correlación negativa entre las variables observadas, son las redes neuronales.

5.1.2. Caso 2: Diseño Box–Behnken

La tabla 5.5 muestra los vectores de respuestas medias; la tabla 5.6 muestra los vectores de varianzas para cada método y para cada variable, y la tabla 5.7 exhibe la suma de varianzas en la obtención de los óptimos para cada método.

Tabla 5.5: Respuestas medias estimadas en la simulación para el Caso 2

Corr.	DES		MOO		TOP		MAD		TAD		MMO		NEU		NET	
	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2
0.9	1.755	3.540	1.871	3.565	1.871	3.563	1.293	3.017	1.307	2.984	1.874	3.571	0.208	2.281	0.175	2.156
0.5	1.570	3.346	1.788	3.219	1.791	3.202	1.250	2.773	1.282	2.682	1.810	3.224	0.197	2.210	0.168	2.128
0.2	1.403	3.215	1.753	2.897	1.754	2.879	1.220	2.580	1.249	2.468	1.723	2.949	0.187	2.149	0.163	2.105
0	1.301	3.079	1.720	2.663	1.725	2.633	1.172	2.442	1.216	2.298	1.674	2.736	0.180	2.118	0.159	2.097
-0.2	1.181	2.951	1.696	2.406	1.703	2.368	1.172	2.244	1.219	2.077	1.621	2.487	0.171	2.077	0.156	2.077
-0.5	0.974	2.730	1.670	1.913	1.664	1.898	1.145	1.945	1.202	1.767	1.504	2.135	0.158	2.026	0.149	2.061
-0.9	0.578	2.338	1.744	0.853	1.664	0.993	1.184	1.286	1.224	1.129	1.270	1.481	0.143	1.962	0.140	2.013

Tabla 5.6: Varianzas individuales para el Caso 2

Corr.	DES		MOO		TOP		MAD		TAD		MMO		NEU		NET	
	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2	y_1	y_2
0.9	0.401	0.409	0.308	0.394	0.307	0.393	0.546	0.623	0.548	0.646	0.325	0.421	0.004	0.077	0.003	0.067
0.5	0.404	0.426	0.354	0.704	0.339	0.594	0.562	0.719	0.554	0.735	0.365	0.634	0.007	0.144	0.003	0.068
0.2	0.407	0.444	0.381	0.813	0.371	0.772	0.577	0.795	0.566	0.839	0.371	0.695	0.010	0.208	0.003	0.069
0	0.391	0.413	0.399	0.874	0.379	0.814	0.579	0.789	0.561	0.842	0.369	0.709	0.011	0.229	0.003	0.072
-0.2	0.366	0.384	0.411	0.963	0.389	0.872	0.577	0.823	0.584	0.869	0.379	0.693	0.013	0.261	0.003	0.072
-0.5	0.307	0.342	0.404	0.989	0.375	0.856	0.556	0.821	0.540	0.846	0.341	0.587	0.013	0.314	0.003	0.070
-0.9	0.193	0.217	0.370	0.733	0.387	0.706	0.526	0.726	0.512	0.705	0.208	0.300	0.016	0.357	0.003	0.074

Tabla 5.7: Varianzas por método para el Caso 2

Correlación	DES	MOO	TOP	MAD	TAD	MMO	NEU	NET
0.9	0.809	0.702	0.700	1.169	1.194	0.746	0.081	0.070
0.5	0.830	1.058	0.933	1.281	1.289	0.999	0.150	0.072
0.2	0.851	1.193	1.143	1.372	1.405	1.066	0.218	0.073
0	0.804	1.274	1.193	1.368	1.403	1.077	0.240	0.075
-0.2	0.750	1.375	1.261	1.400	1.452	1.072	0.274	0.075
-0.5	0.649	1.393	1.231	1.376	1.386	0.927	0.327	0.074
-0.9	0.411	1.103	1.093	1.252	1.218	0.508	0.373	0.077

Por último se presenta la tabla 5.8 con los valores de GPE/RTD conseguidos de la simulación para el caso de donde se desea maximizar la extracción de carotenoides y clorofila en un alga de interés comercial que se describe en la sección 5.2.2.

5.1. Resultados de la simulación

Tabla 5.8: GPE/RTD por método para el Caso 2

Correlación	DES	MOO	TOP	MAD	TAD	MMO	NEU	NET
0.9	6.425	6.894	6.895	4.371	4.413	6.909	0.247	0.430
0.5	5.607	6.424	6.431	4.103	4.190	6.514	0.323	0.470
0.2	4.892	6.157	6.153	3.905	3.977	6.056	0.385	0.501
0	4.430	5.928	5.939	3.682	3.915	5.777	0.426	0.519
-0.2	3.900	5.785	5.828	3.760	4.017	5.466	0.480	0.539
-0.5	2.986	5.880	5.861	3.776	4.073	5.130	0.550	0.575
-0.9	1.356	6.601	6.227	4.197	4.419	4.462	0.636	0.629

Observando la tabla 5.8 podemos indicar que las redes neuronales tienen el mejor desempeño. Es evidente que el resto de métodos se encuentran fuera de los límites de cada respuesta. La varianza de las redes neuronales es la más baja respecto a las demás.

En el desempeño a diferentes correlaciones al aumentar positivamente la correlación los valores del *GPE/RTD* tienen un aumento gradual para todas las metodologías, excepto para las redes neuronales donde se observa una ligera disminución del error; en el caso de aumentar negativamente la correlación los valores del *GPE/RTD* tienen un aumento gradual en todos los métodos, excepto para la DES donde al aumentar la correlación negativa se observa una disminución del error.

Tomando en cuenta el análisis anterior encontramos que los métodos sufren de errores muy altos, pero se observa baja variabilidad en la estimación y poca afectación en la correlación, lo que hace pensar que las fallas en los métodos se pueden deber al tipo de diseño utilizado (Box-Behnken), que no es completamente rotatable puede estar afectando las estimaciones de los métodos además de que el ruido agregado como una normal multivariada con las diferentes correlaciones también puede afectar la estimación de los parámetros. Por lo que, con base a los resultados obtenidos podemos concluir que este caso con diseño Box-Behnken con dos respuestas y correlación positiva entre las variables observadas, las redes neuronales tienen el mejor desempeño.

5.1.3. Caso 3: Diseño Central Compuesto

En este caso se desarrollará un método de HPLC para la determinación simultánea de domperidona (DP) y pantoprazol (PP) en dos preparaciones farmacéuticas comerciales, vea la sección 5.2.3 para más información. En la tabla 5.9 se incluyen los vectores de respuestas medias. La tabla 5.10 expone los vectores de varianza para cada método y para cada variable, además la tabla 5.11 muestra la suma de varianzas para cada método.

5.1. Resultados de la simulación

Tabla 5.9: Respuestas medias estimadas en la simulación para el Caso 3

Corr.	DES						MOO						TOP					
	y_1	y_2	y_3	y_4	y_5	y_6	y_1	y_2	y_3	y_4	y_5	y_6	y_1	y_2	y_3	y_4	y_5	y_6
0.9	0.862	2.360	3.444	5.324	2.992	9.316	-0.267	1.924	2.397	5.676	0.666	10.655	-0.146	2.130	2.550	6.270	0.257	11.665
0.5	0.994	2.496	3.464	5.456	3.032	9.256	0.100	2.292	2.781	6.109	1.030	11.025	0.263	2.563	3.016	6.821	0.474	12.184
0.2	1.082	2.596	3.461	5.532	3.079	9.212	0.426	2.592	3.107	6.434	1.365	11.272	0.653	2.931	3.431	7.354	0.635	12.738
0	1.415	3.038	3.491	5.424	3.137	8.683	0.722	2.869	3.394	6.750	1.637	11.504	0.969	3.305	3.801	7.821	0.743	13.231
-0.2	1.411	3.033	3.486	5.415	3.152	8.639	1.070	3.212	3.841	7.349	2.027	11.759	1.434	3.796	4.395	8.729	0.762	14.142
-0.5	1.400	3.020	3.471	5.391	3.193	8.516	1.019	3.191	3.839	7.526	1.864	11.940	1.380	3.764	4.374	8.843	0.618	14.177
-0.9	1.386	3.003	3.452	5.361	3.247	8.357	1.010	3.172	3.821	7.786	1.641	12.190	1.326	3.702	4.362	9.005	0.444	14.320

Corr.	MAD						TAD						MMO					
	y_1	y_2	y_3	y_4	y_5	y_6	y_1	y_2	y_3	y_4	y_5	y_6	y_1	y_2	y_3	y_4	y_5	y_6
0.9	1.379	3.302	3.907	6.604	2.677	11.345	1.608	3.687	4.199	7.910	1.884	13.564	-0.072	2.189	2.627	6.253	0.387	11.586
0.5	1.404	3.305	3.884	6.491	2.792	11.101	1.668	3.737	4.213	7.950	1.888	13.614	0.334	2.553	3.019	6.641	0.722	11.864
0.2	1.418	3.298	3.846	6.342	2.923	10.797	1.737	3.805	4.241	8.026	1.855	13.759	0.659	2.863	3.381	7.012	1.004	12.230
0	1.437	3.284	3.816	6.179	3.011	10.452	1.787	3.825	4.235	8.006	1.842	13.771	0.902	3.158	3.686	7.350	1.270	12.516
-0.2	1.450	3.287	3.773	5.973	3.111	9.998	1.877	3.910	4.267	8.129	1.738	13.987	1.207	3.523	4.165	7.898	1.588	12.827
-0.5	1.444	3.284	3.771	6.031	3.098	10.069	1.847	3.893	4.253	8.121	1.779	13.913	1.169	3.503	4.165	8.067	1.464	12.992
-0.9	1.448	3.279	3.784	6.041	3.087	10.115	1.839	3.894	4.255	8.120	1.792	13.841	1.185	3.466	4.127	8.329	1.289	13.274

Corr.	NEU						NET					
	y_1	y_2	y_3	y_4	y_5	y_6	y_1	y_2	y_3	y_4	y_5	y_6
0.9	1.721	4.062	4.522	7.569	2.115	11.359	1.712	3.975	4.434	7.318	2.205	11.293
0.5	1.700	3.964	4.433	7.292	2.253	11.103	1.711	3.933	4.401	7.190	2.318	11.163
0.2	1.715	3.923	4.387	7.149	2.354	11.001	1.720	3.922	4.387	7.129	2.404	11.073
0	1.724	3.904	4.361	7.042	2.443	10.904	1.726	3.922	4.374	7.057	2.481	10.955
-0.2	1.742	3.886	4.336	6.913	2.550	10.716	1.744	3.915	4.368	6.989	2.585	10.804
-0.5	1.741	3.909	4.362	7.005	2.566	10.712	1.744	3.944	4.401	7.083	2.566	10.846
-0.9	1.748	3.943	4.403	7.072	2.546	10.738	1.749	3.972	4.436	7.170	2.553	10.861

Tabla 5.10: Varianzas individuales para el Caso 3

Corr.	DES						MOO						TOP					
	y_1	y_2	y_3	y_4	y_5	y_6	y_1	y_2	y_3	y_4	y_5	y_6	y_1	y_2	y_3	y_4	y_5	y_6
0.9	0.427	0.685	0.296	0.816	0.275	2.566	0.357	0.436	0.420	1.890	1.139	5.364	0.421	0.557	0.508	2.529	0.796	5.549
0.5	0.465	0.713	0.295	0.797	0.234	2.142	0.595	0.680	0.659	2.121	1.320	5.531	0.639	0.798	0.753	2.793	0.919	5.848
0.2	0.482	0.710	0.297	0.788	0.214	1.780	0.762	0.868	0.866	2.100	1.410	5.398	0.808	0.964	0.952	3.046	0.998	6.152
0	0.004	0.005	0.006	0.021	0.051	0.448	0.885	0.977	0.981	2.136	1.479	5.288	0.873	1.055	1.038	3.053	1.075	6.017
-0.2	0.004	0.005	0.006	0.020	0.051	0.444	1.018	1.041	1.127	1.701	1.289	4.157	0.971	1.174	1.101	2.862	1.118	5.799
-0.5	0.004	0.006	0.007	0.020	0.054	0.474	1.251	1.358	1.326	2.011	1.519	4.561	1.233	1.393	1.331	3.143	1.322	6.183
-0.9	0.004	0.006	0.007	0.019	0.055	0.486	1.633	1.684	1.678	2.484	1.906	5.275	1.494	1.671	1.629	3.374	1.449	6.535

Corr.	MAD						TAD						MMO					
	y_1	y_2	y_3	y_4	y_5	y_6	y_1	y_2	y_3	y_4	y_5	y_6	y_1	y_2	y_3	y_4	y_5	y_6
0.9	0.161	0.332	0.189	1.520	0.665	4.509	0.222	0.492	0.293	2.755	0.803	6.755	0.405	0.523	0.507	2.161	0.934	4.738
0.5	0.167	0.322	0.173	1.263	0.564	3.751	0.241	0.494	0.288	2.656	0.833	6.774	0.509	0.766	0.757	2.502	1.111	5.539
0.2	0.166	0.295	0.152	0.927	0.432	2.798	0.247	0.497	0.282	2.514	0.834	6.647	0.560	0.949	0.967	2.716	1.138	5.866
0	0.172	0.277	0.123	0.546	0.275	1.570	0.258	0.465	0.248	2.296	0.809	6.427	0.604	1.047	1.065	2.860	1.170	6.143
-0.2	0.170	0.264	0.103	0.236	0.111	0.278	0.260	0.439	0.209	1.953	0.802	5.939	0.723	1.114	1.157	2.542	1.204	5.942
-0.5	0.209	0.316	0.128	0.316	0.150	0.435	0.289	0.508	0.238	2.108	0.849	6.230	0.786	1.376	1.340	2.891	1.299	6.242
-0.9	0.258	0.417	0.164	0.418	0.200	0.590	0.361	0.625	0.286	2.319	0.901	6.513	1.002	1.597	1.643	3.340	1.424	6.855

Corr.	NEU						NET					
	y_1	y_2	y_3	y_4	y_5	y_6	y_1	y_2	y_3	y_4	y_5	y_6
0.9	0.034	0.112	0.112	0.568	0.420	1.206	0.026	0.074	0.072	0.352	0.336	1.220
0.5	0.031	0.096	0.100	0.547	0.427	1.407	0.025	0.071	0.069	0.360	0.370	1.343
0.2	0.033	0.090	0.094	0.479	0.454	1.438	0.027	0.069	0.070	0.333	0.357	1.281
0	0.037	0.091	0.090	0.391	0.454	1.406	0.028	0.070	0.068	0.298	0.336	1.146
-0.2	0.041	0.094	0.093	0.330	0.436	1.263	0.030	0.070	0.070	0.250	0.305	0.842
-0.5	0.042	0.102	0.099	0.372	0.477	1.327	0.030	0.073	0.072	0.263	0.317	0.833
-0.9	0.046	0.115	0.111	0.425	0.496	1.333	0.031	0.083	0.078	0.302	0.328	0.828

5.1. Resultados de la simulación

Tabla 5.11: Varianzas por método para el Caso 3

Correlación	DES	MOO	TOP	MAD	TAD	MMO	NEU	NET
0.9	5.065	9.607	10.360	7.376	11.320	9.268	2.453	2.080
0.5	4.646	10.906	11.749	6.241	11.286	11.184	2.608	2.237
0.2	4.271	11.405	12.919	4.770	11.021	12.196	2.588	2.138
0	0.536	11.746	13.111	2.964	10.505	12.889	2.469	1.946
-0.2	0.530	10.333	13.026	1.161	9.603	12.684	2.257	1.566
-0.5	0.564	12.026	14.604	1.553	10.221	13.934	2.419	1.587
-0.9	0.576	14.662	16.151	2.046	11.005	15.860	2.524	1.651

Por último, la tabla 5.12 presenta los valores de GPE/RTD conseguidos de la simulación.

Tabla 5.12: GPE/RTD por método para el Caso 3

Correlación	DES	MOO	TOP	MAD	TAD	MMO	NEU	NET
0.9	1.320	2.902	3.038	0.914	1.487	2.863	1.314	1.442
0.5	1.158	2.243	2.497	0.939	1.564	2.251	1.167	1.416
0.2	1.063	1.719	2.033	0.944	1.694	1.763	0.998	1.394
0	0.924	1.264	1.718	0.924	1.743	1.352	0.937	1.396
-0.2	0.937	0.787	2.199	0.887	1.947	1.238	0.895	1.341
-0.5	0.975	0.939	2.236	0.892	1.884	1.371	0.884	1.363
-0.9	1.024	1.131	2.301	0.899	1.863	1.501	0.856	1.335

Analizando la tabla 5.9 se puede constatar que los métodos DES, MOO, TOP y MMO están fuera del rango inferior en las respuestas 1 y 2; a su vez los métodos MOO, TOP y MMO están fuera de rango inferior en la respuesta 2. En la varianza podemos ver que los métodos NEU, NET, DES y MAD tienen las varianzas más bajas y el resto de métodos tienen alta variabilidad en la estimación.

En la tabla 5.12 se observa que los métodos con los errores más pequeños son MAD y NEU, seguidos de DES y NET, el resto con errores más grandes.

En el desempeño a diferentes correlaciones al aumentar positivamente la correlación los valores del GPE/RTD tienen un aumento gradual en las metodologías, excepto para MAD y TAD donde se observa una ligera disminución del error; en el caso de aumentar negativamente la correlación los valores del GPE/RTD tienen un aumento gradual en todos los métodos, excepto para NEU y NET que sufren una disminución del error.

En este caso las mejores metodologías son MAD y NEU, pero esta última tiene varianza más baja; después siguen los métodos DES y NET, pero la función de Deseabilidad se encuentra fuera de rango.

5.2. Resultados de Casos

A continuación se muestran detalladamente los casos estudiados y los resultados obtenidos aplicando cada método propuesto de forma directa.

5.2.1. Caso 1: Diseño Taguchi

Este caso aparece en [Ic y Yildirim \(2012\)](#) donde se muestra una solución con los métodos multicriterio GRA, TOPSIS y VIKOR. También aparece en [Ic y Yildirim \(2013\)](#) en donde se muestra la solución con el método MOORA y una variante del método MOORA propuesta por [Kuo et al. \(2008\)](#), esta variante basada en una propiedad aditiva se expone en [Phadke \(1989\)](#) con la misma idea sobre la estimación del efecto de cada factor, esta variante genera que se implemente en los métodos denominados MOORA AD y TOPSIS AD.

El estudio trata sobre un caso en el que se desea mejorar la calidad de un modelo de lavadora para un fabricante de lavadoras en Turquía. Donde se emplea el diseño Taguchi $L8$ que se muestra en la tabla 5.13, se consideran cuatro factores controlables con dos niveles cada uno, los cuales caracterizan la calidad de la lavadora, que son: x_1 la profundidad del panel lateral del cuerpo de la lavadora, x_2 el grosor del panel lateral del cuerpo de la lavadora, x_3 el tipo de aislamiento del motor de la lavadora y x_4 el grosor de la correa del motor de la lavadora. Las dos respuestas deseadas son: y_1 nivel de ruido de la lavadora (dB) (minimizar) y y_2 velocidad de giro (rpm) (maximizar). Además se consideran 0.7 y 0.3 como pesos de cada respuesta, ya que es más importante la disminución del ruido. Note que las respuestas son transformadas a señal ruido Taguchi dependiendo de la necesidad de maximizar, minimizar o lograr un target.

Tabla 5.13: Diseño experimental Taguchi $L8$, con transformación SN.

Experimento	Factores				Valor medio		Razón S/N	
	x_1	x_2	x_3	x_4	y_1	y_2	SN_1	SN_2
1	1	1	1	1	63.70	1320	-36.08	62.41
2	1	1	2	2	64.30	1305	-36.16	62.31
3	1	2	1	2	57.40	1470	-35.18	63.35
4	1	2	2	1	57.10	1470	-35.13	63.35
5	2	1	1	2	59.60	1410	-35.50	62.98
6	2	1	2	1	59.70	1380	-35.52	62.80
7	2	2	1	1	51.00	1680	-34.15	64.51
8	2	2	2	2	52.00	1620	-34.32	64.19

Los modelos implementados se muestran en la tabla 5.14.

5.2. Resultados de Casos

Tabla 5.14: Modelos ajustados para el Caso 1

Modelo de regresión	R^2 aj.	p-valor modelo
$SN_1 = 37.2068 - 0.7655x_1 + 1.1222x_2 + 0.5175x_3 + 0.5329x_4 - 0.3084x_3x_4$	0.9974	0.0018
$SN_2 = 61.1785 + 0.7655x_1 + 1.5234x_2 - 0.8596x_3 - 0.4637x_4 - 0.2017x_2x_4 + 0.4728x_3x_4$	0.9998	0.0095

El óptimo deseado es:

- y_1 = nivel de ruido de la lavadora (dB) = -34.15 (SN) = 51 dB
- y_2 = velocidad de giro (rpm) = 64.51 (SN) = 1680 rpm

El óptimo reportado en [Ic y Yildirim \(2013\)](#) es:

- y_1 = nivel de ruido de la lavadora (dB) = -34.15 (SN) = 51 dB
- y_2 = velocidad de giro (rpm) = 64.51 (SN) = 1680 rpm

En la tabla [5.15](#) se muestran los resultados de cada método y su correspondiente RTD/GPE:

Tabla 5.15: Respuestas para Caso 1

Método	Factores				Respuestas S/N		GPE/RTD
	x_1	x_2	x_3	x_4	SN_1	SN_2	
DES	2	2	1	1	-34.15	64.50	0
MOO	2	2	1	1	-34.15	64.50	0
TOP	2	2	1	1	-34.15	64.50	0
MAD	2	2	1	1	-33.93	63.86	0.01
TAD	2	2	1	1	-33.93	63.86	0.01
MMO	2	2	1	1	-34.15	64.50	0
NEU	1.99	2	1	0.99	-34.15	64.50	0
NET	2	2	1	1	-34.14	64.51	0

De acuerdo a los resultados obtenidos vea que todas las metodologías obtienen excelentes resultados al estar prácticamente en cero o muy cercano a cero los valores de GPE/RTD y por lo tanto respuestas prácticamente son las deseadas.

5.2. Resultados de Casos

5.2.2. Caso 2: Diseño Box–Behnken

Este caso se muestra en Lu et al. (2013), donde se estudia la extracción de carotenoides y clorofila del alga *Laminaria japonica* Aresch, que es un alga de interés comercial; emplean la optimización simultánea usando la función de deseabilidad programada en el software comercial Design Expert (version 8.0.6). Se usa un diseño Box–Behnken que se puede ver en la tabla 5.16 y los factores de control son: x_1 temperatura de extracción (K), x_2 presión de extracción (MPa) y x_3 cantidad de cosolvente (%). Las respuestas son: y_1 g de carotenoides kg^{-1} masa seca de biomasa (max) y y_2 g de clorofila kg^{-1} masa seca de biomasa (max). Se consideran 0.5 y 0.5 como pesos de cada respuesta.

Tabla 5.16: Diseño experimental Box–Behnken

Experimento	Factores			Respuestas	
	x_1	x_2	x_3	y_1	y_2
1	1	1	0	0.206	2.122
2	0	-1	1	0.107	1.888
3	-1	0	-1	0.138	1.519
4	0	-1	-1	0.159	1.75
5	-1	1	0	0.123	1.468
6	0	1	1	0.196	2.473
7	1	-1	0	0.185	1.612
8	1	0	-1	0.167	1.956
9	0	0	0	0.244	2.113
10	0	0	0	0.238	1.987
11	-1	-1	0	0.058	1.348
12	-1	0	1	0.014	1.485
13	1	0	1	0.098	2.286
14	0	1	-1	0.251	1.992
15	0	0	0	0.226	2.031

La tabla 5.17 exhibe los modelos cuadráticos usados, que coinciden con los que reportan los autores del presente caso.

Tabla 5.17: Modelos ajustados para el Caso 2

Modelo de regresión	R^2 aj.	p-valor modelo
$y_1 = 0.2302 + 0.0404x_1 + 0.0334x_2 - 0.0375x_3 - 0.0828x_1^2 - 0.0475x_3^2$	0.8643	0.0085
$y_2 = 2.0437 + 0.2695x_1 + 0.1821x_2 + 0.1144x_3 - 0.3102x_1^2 - 0.096x_2^2 + 0.078x_3^2 + 0.0975x_1x_2 + 0.091x_1x_3 + 0.0858x_2x_3$	0.9278	0.0019

El óptimo deseado es:

- $y_1 =$ g de carotenoides kg^{-1} masa seca de biomasa = 0.251

5.2. Resultados de Casos

- $y_2 = \text{g de clorofila } kg^{-1} \text{ masa seca de biomasa} = 2.473$

El óptimo reportado en [Lu et al. \(2013\)](#) obtenido de un experimento comprobatorio:

- $y_1 = \text{g de carotenoides } kg^{-1} \text{ masa seca de biomasa} = 0.233$
- $y_2 = \text{g de clorofila } kg^{-1} \text{ masa seca de biomasa} = 2.335$

En la tabla 5.18 se muestran los resultados de cada método y su correspondiente RTD/GPE:

Tabla 5.18: Respuestas para Caso 2

Método	Factores			Respuestas		GPE/RTD
	x_1	x_2	x_3	y_1	y_2	
DES	0.408	1	0.366	0.239	2.325	0.105
MOO	0	0	0	0.244	2.113	0.173
TOP	0	0	0	0.244	2.113	0.173
MAD	0	1	0	0.259	2.129	0.174
TAD	0	1	0	0.259	2.129	0.174
MMO	0	0	0	0.244	2.113	0.173
NEU	0.523	1.13	0.677	0.231	2.390	0.113
NET	0.841	0.999	0.467	0.216	2.367	0.182

Revisando la tabla de resultados se observa que todas las metodologías obtienen buenos resultados al presentar valores de GPE/RTD cercanos a cero, pero la función de Deseabilidad (DES) y la red neuronal NEU muestran los valores más bajos. Incluso son los más cercanos a los reportados en [Lu et al. \(2013\)](#). Pero además se puede comentar que los métodos multicriterio no logran encontrar valores intermedios en los factores de control por lo que sólo reportan los valores discretos más cercanos al óptimo, a diferencia del resto de los métodos que se acercan más a los valores de los factores reportados: $x_1 = 0.7 = 324 \text{ K}$, $x_2 = 1 = 17 \text{ MPa}$ y $x_3 = 0.683 = 4.73 \%$.

5.2.3. Caso 3: Diseño Central Compuesto

Esta situación aparece en [Sivakumar et al. \(2007\)](#), donde se quiere desarrollar un método de HPLC de fase inversa para la determinación simultánea de domperidona (DP) y pantoprazol (PP) en dos preparaciones farmacéuticas comerciales. Para la optimización simultánea usan la función de deseabilidad programada en el software comercial Design Expert (version 7.0.0). Se usa un diseño Central Compuesto con 14 experimentos por duplicado y 6 experimentos con puntos centrales que se puede ver en la tabla 5.19 y los factores de control son: x_1 composición de la fase móvil

5.2. Resultados de Casos

(%, v/v), x_2 molaridad del buffer (mM) y x_3 velocidad de flujo (ml/min), con cinco niveles. Respuestas: y_1 factor de retención de PP, y_2 tiempo de retención de PP, y_3 tiempo de retención de IS (Acetofenona), y_4 tiempo de retención de DP, y_5 resolución entre PP-IS, y_6 resolución entre IS-DP. Ponderación (2,1,1,4,2,1) para cada respuesta.

Tabla 5.19: Diseño Central Compuesto

Experimento	Factores			Respuestas					
	x_1	x_2	x_3	y_1	y_2	y_3	y_4	y_5	y_6
1	-1	-1	-1	2.08	5.24	5.72	12.34	2.11	18.84
				2.02	5.14	5.65	12.29	2.28	18.91
2	1	-1	-1	1.40	4.09	4.90	8.12	4.11	12.02
				1.43	4.13	4.94	8.16	4.08	12.00
3	-1	1	-1	2.12	5.30	5.69	11.30	1.73	16.91
				2.11	5.29	5.67	11.30	1.70	16.92
4	1	1	-1	1.49	4.24	4.92	7.94	3.39	11.41
				1.51	4.27	4.98	8.00	3.49	11.29
5	-1	-1	1	2.10	4.16	4.54	9.88	2.01	17.92
				2.12	4.19	4.56	9.92	1.97	17.99
6	1	-1	1	1.44	3.27	3.91	6.48	3.76	11.32
				1.43	3.26	3.89	6.46	3.73	11.25
7	-1	1	1	2.12	4.18	4.48	8.95	1.55	15.99
				2.10	4.15	4.44	8.92	1.53	16.05
8	1	1	1	1.48	3.32	3.87	6.23	3.19	10.53
				1.48	3.32	3.87	6.27	3.21	10.59
9	-1.682	0	0	2.35	5.02	5.02	11.25	0	16.85
	-1.682	0	0	2.38	5.06	5.06	11.42	0	17.07
10	1.682	0	0	1.26	3.38	3.99	5.94	3.49	8.88
	1.682	0	0	1.25	3.38	3.99	5.92	3.50	8.77
11	0	-1.682	0	1.65	3.98	4.42	7.94	2.33	13.53
	0	-1.682	0	1.67	4.00	4.44	8.00	2.34	13.61
12	0	1.682	0	1.81	4.22	4.57	7.70	1.85	12.07
	0	1.682	0	1.81	4.22	4.57	7.68	1.82	12.00
13	0	0	-1.682	1.74	5.14	5.66	9.87	2.26	13.56
	0	0	-1.682	1.76	5.18	5.70	9.84	2.27	13.50
14	0	0	1.682	1.75	3.43	3.76	6.58	1.95	12.25
	0	0	1.682	1.71	3.39	3.72	6.51	1.94	12.22
15	0	0	0	1.72	4.09	4.51	8.09	2.22	13.35
16	0	0	0	1.74	4.11	4.53	8.11	2.19	13.32
17	0	0	0	1.73	4.10	4.52	8.09	2.21	13.38
18	0	0	0	1.72	4.09	4.51	8.07	2.19	13.37
19	0	0	0	1.72	4.08	4.50	8.06	2.20	13.41
20	0	0	0	1.71	4.06	4.48	8.03	2.22	13.43

5.2. Resultados de Casos

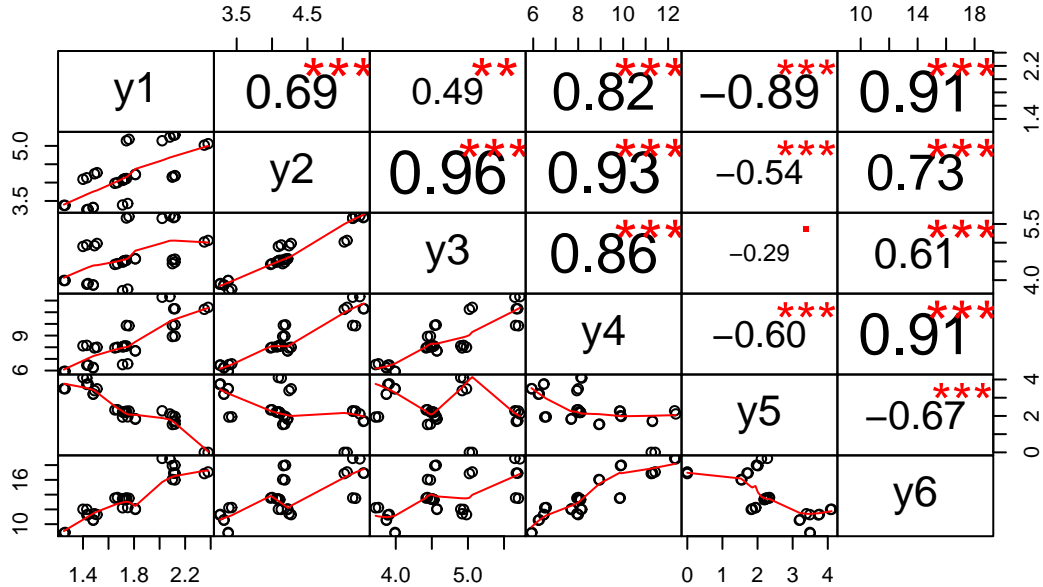


Figura 5.1: Correlación Caso 3.

La figura 5.1 exhibe la correlación entre las respuestas y se observa que existe alta correlación negativa de la variable y_5 con las variables y_1 y y_6 , lo que puede provocar que un cambio en la variable y_5 afecte los valores de las otras variables y viceversa. La imagen fue obtenida por el paquete PerformanceAnalytics de Peterson y Carl (2019). Y efectivamente si mostramos los cálculos de los errores individuales de la métrica RTD/GPE en la tabla 5.20 se observa que existen mayores errores en la variable y_1 , y_5 y y_6 debido a este hecho.

Tabla 5.20: Errores individuales Caso 3

Método	Respuestas						GPE/RTD
	y_1	y_2	y_3	y_4	y_5	y_6	
DES	0.270	0.003	0.005	0.000	0.264	0.196	0.737
MOO	0.368	0.040	0.000	0.100	0.030	0.393	0.931
TOP	0.880	0.540	0.349	0.900	1.000	0.921	4.591
MAD	0.168	0.009	0.011	0.002	0.550	0.127	0.866
TAD	0.168	0.009	0.011	0.002	0.550	0.127	0.866
MMO	0.680	0.273	0.194	0.507	0.235	0.830	2.718
NEU	0.092	0.007	0.019	0.059	0.389	0.108	0.673
NET	0.192	0.019	0.019	0.067	0.461	0.169	0.926

La tabla 5.21 contiene los seis modelos cuadráticos usados, que se reportan en este caso.

5.2. Resultados de Casos

Tabla 5.21: Modelos ajustados para el Caso 3

Modelo de regresión	R^2 aj.	p-valor modelo
$y_1 = 1.738 - 0.3238x_1 + 0.0327x_2 + 0.0299x_1^2 - 0.0131x_2x_3$	0.9942	<2.2e-16
$y_2 = 4.0992 - 0.4881x_1 + 0.0499x_2 - 0.5029x_3 + 0.0436x_1^2 + 0.0701x_3^2 + 0.0456x_1x_3 - 0.0256x_2x_3$	0.9972	<2.2e-16
$y_3 = 4.5723 - 0.3296x_1 - 0.5651x_3 + 0.0923x_3^2$	0.9724	<2.2e-16
$y_4 = 8.1136 - 1.6628x_1 - 0.2080x_2 - 1.0058x_3 + 0.3290x_1^2 + 0.1761x_3^2 + 0.1988x_1x_2$	0.957	<2.2e-16
$y_5 = 2.3771 + 0.9459x_1 - 0.2175x_2$	0.8221	9.081e-13
$y_6 = 13.6621 - 2.80x_1 - 0.5756x_2 - 0.4033x_3$	0.9069	3.527e-16

El óptimo deseado es:

- y_1 = factor de retención de PP = 1.25
- y_2 = tiempo de retención de PP = 3.26
- y_3 = tiempo de retención de IS (Acetofenona)= 3.72
- y_4 = tiempo de retención de DP = 5.92
- y_5 = resolución entre PP-IS = 2.00
- y_6 = resolución entre IS-DP = 8.77

El óptimo reportado en [Sivakumar et al. \(2007\)](#):

- y_1 = factor de retención de PP = 1.59
- y_2 = tiempo de retención de PP = 3.27
- y_3 = tiempo de retención de IS (Acetofenona)= 3.72
- y_4 = tiempo de retención de DP = 5.97
- y_5 = resolución entre PP-IS = 2.54
- y_6 = resolución entre IS-DP = 10.40

Tabla 5.22: Respuestas para Caso 3

Método	Factores			Respuestas						GPE/RTD
	x_1	x_2	x_3	y_1	y_2	y_3	y_4	y_5	y_6	
DES	0.55	1.68	1.68	1.59	3.25	3.70	5.92	2.53	10.49	0.737
MOO	0	0	1.68	1.71	3.39	3.72	6.51	1.94	12.22	0.931
TOP	-1.68	0	0	2.35	5.02	5.02	11.25	0.00	16.85	4.591
MAD	1	1	1	1.46	3.29	3.76	5.91	3.10	9.88	0.866
TAD	1	1	1	1.46	3.29	3.76	5.91	3.10	9.88	0.866
MMO	-1.00	1	1	2.10	4.15	4.44	8.92	1.53	16.05	2.718
NEU	0.73	0.64	0.73	1.37	3.28	3.79	6.27	2.78	9.71	0.673
NET	0.97	1.00	0.90	1.49	3.32	3.79	6.32	2.92	10.25	0.926

5.3. Discusión

Revisando la tabla de resultados 5.22, vea que todas las metodologías obtienen buenos resultados, excepto los métodos TOP y MMO que tienen el peor desempeño. La función de Deseabilidad (DES) y la red neuronal NEU muestran los valores más bajos y el mejor desempeño, que incluso son los más cercanos a los reportados. Pero nuevamente los métodos multicriterio no logran encontrar valores intermedios en los factores de control por lo que sólo reportan los valores discretos más cercanos al óptimo, a diferencia del resto de los métodos que se acercan más a los valores óptimos de los factores reportados.

5.3. Discusión

La función de Deseabilidad (DES) tiene ciertas desventajas ya que no considera la correlación de las respuestas y además la elección de los pesos se hace de forma manual o por medio de un experto, aunque se han hecho diversas propuestas de como hacerlo aún no se logra unificarlas e incluso una mala elección de las ponderaciones provoca que se seleccionen óptimos fuera de rango. Necesita contar con los modelos ajustados por lo que se debe conocer muy bien el proceso, el resto de las técnicas tienen la ventaja de no requerir los modelos. En los resultados encontrados se visualiza que es afectada por la correlación, que la elección de los pesos es de suma importancia y que se observa en el caso 3 que se seleccionan respuestas fuera de rango. En general DES tiene un buen desempeño pero sufre de errores más grandes si el número de respuestas crece.

Los métodos multicriterio (MCDM) también necesitan que el experto guíe el proceso para tener una adecuada elección de ponderaciones sobre las respuestas. Estos métodos son los más simples de implementar pudiendo programarse en una hoja de cálculo. Los resultados no fueron tan alentadores ya que encontraban soluciones fuera de rango posiblemente debido a la elección de valores nominales en la variables independientes y a que no se especifican los límites de control. Pero una muestra de su efectividad se observa en el caso 3 donde la técnica MAD tiene un resultado sobresaliente.

Por último las redes neuronales NEU y NET muestran en general el mejor desempeño, pero tienen diversas dificultades como la elección de los pesos, el sobreajuste en su aprendizaje, la dificultad de dividir muestras pequeñas en entrenamiento, prueba y validación, la elección del algoritmo de optimización y de los hiperparámetros dificulta su aplicación. En los resultados se observa que es un método robusto al efecto de la correlación y logra además valores bajos en variabilidad y error GPE/RTD .

En base a la métrica empleada y los argumentos anteriores se puede decir que la mejor técnica fue la de redes neuronales artificiales, específicamente el método NEU ya que logra los errores más pequeños y cercanos a cero en los casos estudiados.

Capítulo 6

Conclusiones

Tomando como sustento la métrica usada en el presente trabajo podemos concluir que el mejor método considerando los casos abordados y las condiciones mencionadas durante el desarrollo del experimento de simulación son las redes neuronales artificiales, particularmente el paquete NEU (Neuralnet) de redes multicapa al obtener los valores de GPE/RTD mas pequeños. En segundo lugar se puede mencionar a las redes neuronales multicapa construidas con el paquete NET (Nnet) y la función de Deseabilidad (DES) que logran un buen desempeño, pero al aumentar las respuestas ambos métodos empiezan a sufrir de mayores errores, e incluso DES puede encontrar respuestas fuera de rango. Las técnicas MCDM no presentaron un buen desempeño ya que no son capaces de encontrar valores intermedios en los factores de control; pero es importante resaltar que el método MOORA AD (MAD) logra un excelente resultado equiparable con el método NEU en casos donde se incrementan sustancialmente las respuestas, por lo que se recomienda su aplicación en este tipo de situaciones.

Por último, en la aplicación directa de las metodologías en los datos experimentales originales en cada caso, se refuerza la conclusión de que las redes neuronales del tipo NEU logran el mejor desempeño, seguidas de DES considerando tener los valores más pequeños de GPE/RTD . Y nuevamente las técnicas MCDM y la red neuronal NET obtienen valores mayores en los errores, por lo que tienen el peor desempeño.

Como nota final es importante entender que todos los métodos analizados son técnicas auxiliares y que pueden trabajar en conjunto para tener más solidez en la toma de decisiones considerando sus ventajas y desventajas.

6.1. Recomendaciones

Se puede recomendar utilizar la mayor cantidad de métodos de optimización multi-respuesta con el fin de comparar sus resultados bajo ciertos supuestos y con la guía de un experto en el proceso o fenómeno de estudio, para poder tomar la mejor decisión. Además se pueden hacer las siguientes recomendaciones para trabajos futuros:

- Considerar la correlación entre respuestas y dentro de las respuestas.
- Tomar en cuenta la variabilidad de las respuestas.
- En la parte de simulación verificar el efecto de la adición de otro tipo de ruido diferente a la normal multivariada.
- En los casos de diseños Box-Behnken poder probar los métodos con otras situaciones donde se lleve a cabo este tipo de diseño e indagar más sobre los efectos de usarlo.
- Probar los métodos en otras situaciones de estudio, con diferente número de variables independientes, más respuestas y diferentes diseños experimentales.
- Programar el algoritmo de Levenberg-Marquardt como optimizador en la función de deseabilidad y en redes neuronales artificiales.
- Proponer un método empleando redes neuronales artificiales pero con aprendizaje no supervisado.
- Generar un problema teórico donde el óptimo sea matemáticamente conocido y aplicar las técnicas propuestas en un estudio de simulación para poder calcular el Error Cuadrado Medio y tener otra métrica de comparación entre los métodos.

Referencias

- Akteke-Öztürk, B., Weber, G.-W., y Köksal, G. (2015). Desirability functions in multiresponse optimization. In Plakhov, A., Tchemisova, T., y Freitas, A., editors, *Optimization in the Natural Sciences*, pages 129–146, Cham. Springer International Publishing.
- Altinoz, O. T., Akca, E., Yilmaz, A. E., Duca, A., y Ciuprina, G. (2015). Parallel implementation of desirability function-based scalarization approach for multiobjective optimization problems. *Informatica*, 39(2):115–123.
- Antony, J. (2000). Multi-response optimization in industrial experiments using taguchi's quality loss function and principal component analysis. *Quality and Reliability Engineering International*, 16(1):3–8.
- Bayramov, F., Taşdemir, C., y Taşdemir, M. A. (2004). Optimisation of steel fibre reinforced concretes by means of statistical response surface method. *Cement and Concrete Composites*, 26(6):665–675.
- Berger, P. D., Maurer, R. E., y Celli, G. B. (2018). *Experimental Design: With Application in Management, Engineering, and the Sciences*. Springer.
- Berzal, F. (2018). *Redes Neuronales & Deep Learning*. Independently published, Granada, España.
- Box, G. E. y Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(1):1–45.
- Box, G. E. P. y Behnken, D. W. (1960). Some new three level designs for the study of quantitative variables. *Technometrics*, 2(4):455–475.
- Brauers, W. K. M. y Zavadskas, E. K. (2006). The moora method and its application to privatization in a transition economy. *Control and Cybernetics*, 35(2):445–469.
- Brauers, W. K. M. y Zavadskas, E. K. (2010). Project management by multimooora as an instrument for transition economies. *Technological and Economic Development of Economy*, 16:5–24.

Referencias

- Burguete-Hernández, J. F. (2000). *Introducción a la Ingeniería de Calidad Experimental*. UNACH-UDLAP, México.
- Ceballos, B., Lamata, M. T., y Pelta, D. (2015). Una comparativa de modelos de decisión multi-criterio difusos. In *Actas de la XVI Conferencia CAEPIA*, pages 459–469.
- Ceballos, B. A. (2016). *MCDM: Multi-Criteria Decision Making Methods for Crisp Data*. R package version 1.2.
- Costa, N. R. y Lourenço, J. (2016). Multiresponse problems: Desirability and other optimization approaches. *Journal of Chemometrics*, 30(12):702–714.
- Da Silva, I. N., Spatti, D. H., Flauzino Andrade, R., Bartocci Liboni, L. H., y dos Reis Alves, S. F. (2017). *Artificial Neural Networks*. Springer.
- Derringer, G. y Suich, R. (1980). Simultaneous optimization of several response variables. *Journal of Quality Technology*, 12(4):214–219.
- Derringer, G. C. (1994). A balancing act: Optimizing a products properties. *Quality Progress*, 27(6):51–58.
- Draper, N. R. (1963). Ridge analysis of response surfaces. *Technometrics*, 5:469–479.
- Fisher, R. A. (1926). The arrangement of field experiments. *Journal of the Ministry of Agriculture of Great Britain*, 33:503–513.
- Fisher, R. A. (1935). *The Design of Experiments*. Oliver and Boyd, Edinburgh.
- Fritsch, S., Guenther, F., y Wright, M. N. (2019). *neuralnet: Training of Neural Networks*. R package version 1.44.2.
- Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., y Hothorn, T. (2019). *mvtnorm: Multivariate Normal and t Distributions*. R package version 1.0-11.
- Gutiérrez-Pulido, H. y De la Vara-Salazar, R. (2012). *Análisis y Diseño de Experimentos*. (Tercera edición). McGraw-Hill Interamericana Editores, S.A. de C.V., México.
- Harrington, E. C. (1965). The desirability function. *Industrial Quality Control*, 21:494–498.
- Hoerl, A. E. (1959). Optimum solution of many variables equations. *Chemical Engineering Progress*, 55:67–78.
- Hwang, C.-L. y Yoon, K. (1981). *Multiple Attribute Decision Making*. Springer-Verlag, Berlin Heidelberg.
- Šibaliija, T. V. y Majstorović, V. D. (2016). *Advanced Multiresponse Process Optimization: An Intelligent and Integrated Approach*. Springer International Publishing, Switzerland.

Referencias

- Ic, Y. y Yildirim, S. (2012). Improvement of a product design using multi criteria decision making methods with taguchi method. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 27:447–458.
- Ic, Y. y Yildirim, S. (2013). Moora-based taguchi optimisation for improving product or process quality. *International Journal of Production Research*, 51(11):3321–3341.
- Khuri, A. (2017). Response surface methodology and its applications in agricultural and food science. *Biom Biostat Int J*, 5(5):2–11.
- Khuri, A. I. y Conlon, M. (1981). Simultaneous optimization of multiple responses represented by polynomial regression functions. *Technometrics*, 23(4):363–375.
- Khuri, A. I. y Cornell, J. A. (1987). *Response Surfaces: Designs and Analyses*. 2nd ed. Marcel Dekker, New York.
- Khuri, A. I. y Mukhopadhyay, S. (2010). Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(2):128–149.
- Krishnaiah, K. y Shahabudeen, P. (2012). *Applied Design of Experiments and Taguchi Methods*. PHI Learning Pvt. Ltd., New Delhi.
- Kuhn, M. (2016). *desirability: Function Optimization and Ranking via Desirability Functions*. R package version 2.1.
- Kuo, Y., Yang, T., y Huang, G.-W. (2008). The use of a grey-based taguchi method for optimizing multi-response simulation problems. *Engineering Optimization*, 40:517–528.
- Lawson, J. (2014). *Design and Analysis of Experiments with R*. Chapman and Hall/CRC.
- Lu, J., Feng, X., Han, Y., y Xue, C. (2013). Optimisation of subcritical fluid extraction of carotenoids and chlorophyll a from laminaria japonica aresh by response surface methodology. *Journal of the Science of Food and Agriculture*, 94:139–145.
- Majid, B., S. Khanmohammadi, O., Morteza, Y., y Joshua, I. (2012). A state-of the-art survey of topsis applications. *Expert Systems with Applications*, 39(17):13051–13069.
- McCulloch, W. S. y Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.
- Miettinen, K. (2008). Introduction to multiobjective optimization: Noninteractive approaches. In *Branke J., Deb K., Miettinen K., Słowiński R. (eds) Multiobjective Optimization. Lecture Notes in Computer Science*, pages 1–26. Springer, Berlin Heidelberg.
- Montgomery, D. C. (2017). *Design and Analysis of Experiments*. John wiley & sons.

Referencias

- Myers, R. H., Montgomery, D. C., y Anderson-Cook, C. M. (2016). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley & Sons.
- Peterson, B. G. y Carl, P. (2019). *PerformanceAnalytics: Econometric Tools for Performance and Risk Analysis*. R package version 1.5.3.
- Phadke, M. S. (1989). *Quality Engineering Using Robust Design*. PTR Prentice-Hall Inc, Englewood Cliffs, NJ.
- R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Rocha, L. C. S., de Paiva, A. P., Paiva, E. J., y Balestrassi, P. P. (2015). Comparing dea and principal component analysis in the multiobjective optimization of p-gmaw process. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 38:2513–2526.
- Rosenblatt, F. (1957). The perceptron: A perceiving and recognizing automaton (cornell aeronautical laboratory). Technical report, Cornell Aeronautical Laboratory, Technical Report 85-460-1, Buffalo, New York.
- Sivakumar, T., Manavalan, R., Muralidharan, C., y Valliappan, K. (2007). Multi-criteria decision making approach and experimental design as chemometric tools to optimize hplc separation of domperidone and pantoprazole. *Journal of Pharmaceutical and Biomedical Analysis*, 43(5):1842–1848.
- Tong, L.-I. y Hsieh, K.-L. (2001). A novel means of applying neural networks to optimize the multiresponse problem. *Quality Engineering*, 13:11–18.
- Venables, W. N. y Ripley, B. D. (2016). *nnet: Feed-Forward Neural Networks and Multinomial Log-Linear Models*. R package version 7.3-12.
- Yang, T. y Chou, P. (2005). Solving a multiresponse simulation-optimization problem with discrete variables using a multiple- attribute decision-making method. *Mathematics and Computers in Simulation*, 68:9–21.

ANEXOS

A continuación se muestran los códigos empleados, se ejemplifica con el caso 1; primeramente se deben cargar las variables independientes x y las respuestas y a la memoria, posteriormente se procede a formar las matrices y vectores de interés y realizar la regresión.

Código en R. Necessary information

```
library(PerformanceAnalytics); library(desirability); library(MCDM)
library(neuralnet); library(nnet); library(mvtnorm)

#p is the design matrix (number of independent variables or factors)
#M is the response matrix
#cb is the cost benefit vector which indicates if each answer max or min
#w is a vector of weights. The sum of the weights has to be 1.

p<- cbind(x1,x2,x3,x4); M<-abs(cbind(y1,y2))
cb <- c("min","max"); w <- c(0.7,0.3)

#MODELS
modelo1 <- lm(y1~x1+x2+x3+x4+x3:x4)
summary(modelo1)
model1 <- function(x) {
  modelo1$coefficients[1]+(modelo1$coefficients[2])*x[1]+
    (modelo1$coefficients[3])*x[2]+(modelo1$coefficients[4])*x[3]+
    (modelo1$coefficients[5])*x[4]+(modelo1$coefficients[6])*(x[3]*x[4])}

modelo2 <- lm(y2~x1+x2+x3+x4+x2:x4+x3:x4)
summary(modelo2)
model2<-function(x){
  modelo2$coefficients[1]+(modelo2$coefficients[2])*x[1]+
    (modelo2$coefficients[3])*x[2]+(modelo2$coefficients[4])*x[3]+
    (modelo2$coefficients[5])*x[4]+(modelo2$coefficients[6])*
    (x[2]*x[4])+(modelo2$coefficients[7])*(x[3]*x[4])}
```

Función de Deseabilidad (DES)

Ya teniendo las matrices, vectores y modelos ajustados se procede a cargar cada uno de los métodos propuestos en el presente trabajo como funciones. El primer método es la función de Deseabilidad (DES).

Código en R. Desirability

```

Deseabilidad <- function(datos,p,cb,w){
  #Individual desirability is maximized, minimized or targeted.
  for (i in 1:ncol(r)) {
    if (cb[i] == "min"){
      cadena <- paste("deseab", i, "=dMin(min(r[,i]), max(r[,i]),w[i])",
        ",sep = """)
    } else if(cb[i] == "max"){
      cadena <- paste("deseab", i, "=dMax(min(r[,i]), max(r[,i]),w[i])",
        ",sep = """)
    } else if (is.numeric(as.numeric(cb[i])) == T){
      cadena <- paste("deseab", i, "=dTarget(min(r[,i]),",cb[i],"",max
        (r[,i]),w[i])",sep = """) }
    cadena
    eval(parse(text = cadena))}

  #Overall Desirability
  cadena2 <-""
  for (i in 1:ncol(r)) {
    cadena2 <- paste(cadena2,"deseab",i,"",",sep = """)
  }
  cadena2 <- substring(cadena2,1,nchar(cadena2)-1)
  cf <- paste("DGlobal <- dOverall(",cadena2,")",sep = """)
  eval(parse(text = cf))

  modelos <- ""
  for (i in 1:ncol(r)) {
    modelos <- paste(modelos,"mod",i,"(x)",",sep = """) }
  modelos <- substring(modelos,1,nchar(modelos)-1)

  rsmOpt <- function(x, dObject)
  {models <- paste("out <- predict(dObject, data.frame(",modelos,")",
    sep = "")
  eval(parse(text = models))
  if(any(abs(x) > max(x1))) out <- 0
  out}

  #Search Zone
  parametros <- ""

```

Código en R. Desirability

```

for (i in 1:ncol(p)) {
  parametros <- paste(parametros,"x",i,"= seq(min(x",i,"),
    max(x",i,"), length = 2),"",sep = "") }

parametros <- substring(parametros,1,nchar(parametros)-1)
Sz <- paste("Searchzone <- expand.grid(",parametros,")",sep = "")
eval(parse(text = Sz))

#Nelder-Mead
inferior = ""
superior = ""
for (i in 1:ncol(p)) {
  inferior <- paste(inferior,"min(x",i,")",sep = "")
  superior <- paste(superior,"max(x",i,")",sep = "") }
inferior <- substring(inferior,1,nchar(inferior)-1)
superior <- substring(superior,1,nchar(superior)-1)

limites <- paste("tmp <- optim(as.vector(Searchzone[i, ]),lower=c(",
  inferior,""),upper=c(",superior,
  "), rsmOpt, dObject = DGlobal,
  control = list(fnscale = -1))",sep = "")

for(i in 1:dim(Searchzone)[1]){
  eval(parse(text = limites))
  if(i == 1)
  {best <- tmp} else {if(tmp$value > best$value) best <- tmp}}
xx <- best$par

#Optimal Vector
prediccion <- ""
for (i in 1:ncol(r)) {
  prediccion <- paste(prediccion,"mod",i,"(xx)",sep = "")}
prediccion <- substring(prediccion,1,nchar(prediccion)-1)

pred <- paste("predoptimo <- as.vector(c(",prediccion,")",sep = "")
eval(parse(text = pred))
return(predoptimo)}

```

MOORA (MOO)

Código en R. MOORA

```
MOORA <- function(datos,w,cb){

  #Matrix Normalization
  d = sqrt(colSums(datos^2))
  N <- matrix(nrow = nrow(datos), ncol = ncol(datos))
  for(j in 1:ncol(datos)){
    N[,j] <- (datos[,j] / d[j])
  }

  #weights
  WE <- diag(w)
  NW <- N%*%WE

  #Ratio
  NR <- NW
  for(j in 1:ncol(datos)){
    if (cb[j] == 'min'){
      NR[,j] <- NW[,j]*(-1)
    }
  }

  #MOORA Index
  IM <- apply(NR, 1, sum)

  #Ranking
  Ranking <- rank(-IM, ties.method= "first")
  a <- (data.frame(valor = IM, Ranking))

  #RANKING WITH DESIGN MATRIX
  RD <- cbind(p,datos,a)

  for (i in 1:nrow(datos)) {
    if (Ranking[i] == 1){
      OPM <- (RD[i,])
    }
  }

  OPMM <- c(OPM[,5],OPM[,6])
  return(OPMM)
}
```

TOPSIS (TOP)

Código en R. TOPSIS

```

TOPSIS <- function(datos,w,cb){

  #Matrix Normalization
  d = sqrt(colSums(datos^2))
  N <- matrix(nrow = nrow(datos), ncol = ncol(datos))
  for(j in 1:ncol(datos)){
    N[,j] <- (datos[,j] / d[j]) }

  #weights
  WE <- diag(w)
  NW <- N%*%WE

  # Ideal and Not-Ideal Solution
  Ideal <- as.integer(cb == "max") * apply(NW, 2, max) +
    as.integer(cb == "min") * apply(NW, 2, min)
  noIdeal <- as.integer(cb == "min") * apply(NW, 2, max) +
    as.integer(cb == "max") * apply(NW, 2, min)

  #Distance to the ideal solution
  distance =function(x,y){
    sqrt(sum((x - y) ^ 2))
  }
  Dispos <- apply(NW, 1, distance, Ideal)
  Disneg <- apply(NW, 1, distance, noIdeal)

  #TOPSIS Index
  IT <- Disneg/(Disneg+Dispos)

  # Ranking
  Ranking <- rank(-IT, ties.method= "first")
  a <- (data.frame(valor = IT, Ranking))

  #RANKING WITH DESIGN MATRIX
  RD <- cbind(p,datos,a)

  for (i in 1:nrow(datos)) {
    if (Ranking[i] == 1){
      OPT <- (RD[i,]) }
  }
  OPMT <- c(OPT[,5],OPT[,6])
  return(OPMT)
}

```

MOORA AD (MAD)

Código en R. MOORA AD

```

MOORAAD <- function(datos,w,cb){
  #Matrix Normalization
  d = sqrt(colSums(datos^2))
  N <- matrix(nrow = nrow(datos), ncol = ncol(datos))
  for(j in 1:ncol(datos)){N[,j] <- (datos[,j] / d[j])}

  #weights
  WE <- diag(w); NW <- N%*%WE
  #Ratio
  NR <- NW
  for(j in 1:ncol(datos)){if (cb[j] == 'min'){NR[,j] <- NW[,j]*(-1)}}
  #MOORA Index
  IM <- apply(NR, 1, sum)
  #Ranking
  Ranking = rank(-IM, ties.method= "first")
  a <- (data.frame(IM = IM, Ranking))

  #RANKING WITH DESIGN MATRIX
  RD <- cbind(p,datos,a)
  for (i in 1:nrow(datos)) {if (Ranking[i] == 1){OPM <- (RD[i,])} }

  #Additive property
  cadena <-""
  for (i in 1:ncol(p)) {cadena <- paste(cadena,"A",i," <- tapply
(IM,x",i,",mean)",sep = "", "\n") }
  eval(parse(text = cadena))

  cadena2 <-""
  for (i in 1:ncol(p)) {
  cadena2 <- paste(cadena2,"which.max(A",i,",)",",",sep = "") }
  cadena2 <- substring(cadena2,1,nchar(cadena2)-1)

  op <- paste("optimo <- c(",cadena2,")",sep = "")
  eval(parse(text = op))
  optimo <- as.integer(names(optimo))
  x <- as.vector(optimo)

  cadena3 <- ""
  for(i in 1:ncol(r)){cadena3 <- paste(cadena3,"mod",i,"(x)",",",sep = "")}
  cadena3 <- substring(cadena3,1,nchar(cadena3)-1)
  op1 <- paste("OPMAD <- c(",cadena3,")",sep = "")
  eval(parse(text = op1))
  return(OPMAD)}

```

TOPSIS AD (TAD)

Código en R. TOPSIS AD

```

TOPSISAD <- function(datos,w,cb){
  #Normalization matrix
  d = sqrt(colSums(datos^2))
  N <- matrix(nrow = nrow(datos), ncol = ncol(datos))
  for(j in 1:ncol(datos)){N[,j] <- (datos[,j] / d[j])}

  #weights
  WE <- diag(w); NW <- N%*%WE

  #Ideal and Not-Ideal Solution
  Ideal <- as.integer(cb == "max") * apply(NW, 2, max) +
    as.integer(cb == "min") * apply(NW, 2, min)
  noIdeal <- as.integer(cb == "min") * apply(NW, 2, max) +
    as.integer(cb == "max") * apply(NW, 2, min)

  #Distance to the Ideal Solution
  distance =function(x,y){
    sqrt(sum((x - y) ^ 2))}
  Dispos <- apply(NW, 1, distance, Ideal)
  Disneg <- apply(NW, 1, distance, noIdeal)

  #TOPSIS Index
  IT <- Disneg/(Disneg+Dispos)

  #Ranking
  Ranking <- rank(-IT, ties.method= "first")
  a <- (data.frame(IT = IT, Ranking))

  #RANKING WITH DESIGN MATRIX
  RD <- cbind(p,datos,a)

  for (i in 1:nrow(datos)) {
    if (Ranking[i] == 1){
      OPT <- (RD[i,]) } }
  for (i in 1:ncol(OPT)) {
    OPT1 <- (OPT[,1:i-1]) }

  #Additive property
  cadena <- ""
  for (i in 1:ncol(p)) {
    cadena <- paste(cadena,"A",i," <- tapply(IT,x",i,",mean)",
      sep = " ", "\n") }
  eval(parse(text = cadena))

```

Código en R. TOPSIS AD

```

cadena2 <- ""
for (i in 1:ncol(p)) {
  cadena2 <- paste(cadena2,"which.max(A",i,")",",sep = "" )
  cadena2 <- substring(cadena2,1,nchar(cadena2)-1)

  op <- paste("optimo <- c(",cadena2,")",",sep = "" )
  eval(parse(text = op))
  optimo <- as.integer(names(optimo))
  x <- as.vector(optimo)

  cadena3 <- ""
  for (i in 1:ncol(r)) {
    cadena3 <- paste(cadena3,"mod",i,"(x)",",sep = "" )
    cadena3 <- substring(cadena3,1,nchar(cadena3)-1)
    op1 <- paste("OPTAD <- c(",cadena3,")",",sep = "" )
    eval(parse(text = op1))
    return(OPTAD)
  }
}

```

MMOORA (MMO)

Código en R. MMOORA

```

MMMOORA <- function(datos,w,cb){

  MM <- MMOORA(datos,w,cb)
  Ranking <- (MM$MultiMooraRanking)

  #RANKING WITH DESIGN MATRIX
  RD <- cbind(p,datos,Ranking)

  for (i in 1:nrow(datos)) {
    if (Ranking[i] == 1){OPMM <- (RD[i,])}
  }
  OPMMM <- c(OPMM[5],OPMM[6])
  return(OPMMM)
}

```


NEURALNET (NEU)

Código en R. NEURALNET

```

NEURALNETB <- function(datos,p,cb,w){

  # VARIABLE STANDARDIZATION
  # -----
  M1 <- M
  normalizacion <- ""
  for (i in 1:ncol(M1)) {
    normalizacion <- paste(normalizacion,"M1[,"i,"] <- (M1[,"i,"]-
      min(M1[,"i,"])) / (max(M1[,"i,"])-min(M1[,"i,"]))",
      sep = "","\n")
  }
  eval(parse(text = normalizacion))

  # MODEL.MATRIX1
  # -----
  r <- data.frame(r)

  equis <- ""
  for (i in 1:ncol(p)) {
    equis <- paste(equis,"x",i,"+",sep = "")
  }
  equis <- substring(equis,1,nchar(equis)-1)

  ye1 <- ""
  for (i in 1:ncol(M)) {
    ye1 <- paste(ye1,"\\"yy",i,"\\",",",sep = "")
  }
  ye1 <- substring(ye1,1,nchar(ye1)-1)

  namesr <-""
  namesr <-paste(namesr,"colnames(r) <- c(",ye1,")",sep = "")
  eval(parse(text = namesr))

  ye2 <- ""
  for (i in 1:ncol(M)) {
    ye2 <- paste(ye2,"yy",i,"+",sep = "")
  }
  ye2 <- substring(ye2,1,nchar(ye2)-1)

  eme <- ""
  eme <- paste(eme,"m1 <- model.matrix(~",equis,"+",ye2,"",
    data=r)[,-1]",sep = "")
  eval(parse(text = eme))
}

```

Código en R. NEURALNET

```

normalizacionm1 <- ""
for (i in 1:ncol(m1)) {
  normalizacionm1 <- paste(normalizacionm1,"m1[",i,"] <- (m1[",i,
  "]-min(m1[",i,"])) / (max(m1[",i,"])-min(m1[",i,"]))",
  sep = "","\n")
}
eval(parse(text = normalizacionm1))

#FORMULA 1
namex <- paste(c(colnames(p)),collapse="+")
namey <- paste(c(colnames(r)),collapse="+")
col_list <- paste(c(namex,"~",namey),collapse="")
f <- as.formula(col_list)

#NET 1
#Use "y" as input and "x" as outputs
#Sigmoid use because the values of x are between 0 and 1
sigmoid <- function(x) {1/(1+exp(-x))}

modelo.yx <- neuralnet(f, data=m1, hidden=c(10,8), act.fct=sigmoid,
  threshold = 0.02, startweights = c(0.7,0.3),
  stepmax = 1000000, algorithm = "rprop+")

#Once trained enter desired "y" values.
for (i in 1:ncol(M1)) {
  if (cb[i] == "min"){
    cadena <- paste("yy", i, "=min(M1[",i,"])",sep = "")
  } else if(cb[i] == "max"){
    cadena <- paste("yy", i, "=max(M1[",i,"])",sep = "")
  }
  cadena
  eval(parse(text = cadena))
}

datayy <- ""
for (i in 1:ncol(M1)) {
  datayy <- paste(datayy,"yy",i,"",sep = "")
}
datayy <- substring(datayy,1,nchar(datayy)-1)

datay <- ""
datay <- paste(datay,"data.y <- cbind(",datayy,")",sep = "")
datay
eval(parse(text = datay))

data.y <- data.frame(data.y)

```

Código en R. NEURALNET

```

#These are the predicted optimized "x" values
pr.x <- neuralnet::compute(modelo.yx,data.y)
pr.x$net.result
xopt <- round(pr.x$net.result,2)

# MODEL.MATRIX2 WITH SIMULATED r VALUES
r <- data.frame(r)

#FORMULA 2
col_list2 <- paste(c(namey,"~",namex),collapse="")
f2 <- as.formula(col_list2)

#NET 2
#Use "x" as input y "y" as outputs
modelo.xy <- neuralnet(f2, data=m1, hidden=c(10,8), act.fct=sigmoid,
                      threshold = 0.01, stepmax = 100000,
                      algorithm = "rprop+")

#plot(modelo.xy)

#Enter the optimized "x" values of Network 1 to get "y" optimized
pr.y <- neuralnet::compute(modelo.xy,xopt)
pr.y$net.result

# PREDICTION
# -----
# DECODE
predic <- ""
for (i in 1:ncol(r)) {
  predic <- paste(predic,"y",i,".predict <- pr.y$net.result[,",i,
  "]*(max(y",i,")-min(y",i,))+min(y",i,")",sep="","\n")
}
eval(parse(text = predic))

predic2 <- ""
for (i in 1:ncol(r)) {
  predic2 <- paste(predic2,"y",i,".predict,",sep="")
}
predic2 <- substring(predic2,1,nchar(predic2)-1)

optimo <-""
optimo <- paste(optimo,"OPT.RNB <- round(c(",predic2,"),3)",sep="")
eval(parse(text = optimo))

return(OPT.RNB)
}

```

NNET (NET)

Código en R. NNET

```

NEURALNETF <- function(datos,p,cb,w){
  # VARIABLE STANDARDIZATION
  M1 <- M
  normaliz <- ""
  for (i in 1:ncol(M1)) {
    normaliz <- paste(normaliz,"M1[,"i,"] <- (M1[,"i,"]-min(M1[,"
      i,"])) / (max(M1[,"i,"])-min(M1[,"i,"]))",
      sep = "","\n") }
  eval(parse(text = normaliz))
  normalizacion <- ""
  for (i in 1:ncol(datos)) {
    normalizacion <- paste(normalizacion,"datos[,"i,"] <-(datos[,"i,
      "] -min(datos[,"i,"])) / (max(datos[,"i,"])-min(datos[,"i,"]))",
      sep = "","\n") }
  eval(parse(text = normalizacion))

  ye <- ""
  for (i in 1:ncol(datos)) {ye <- paste(ye,"datos[,"i,"],"sep = """)}
  ye <- substring(ye,1,nchar(ye)-1)

  ye2 <- ""
  ye2 <- paste(ye2,"yes <- cbind(",ye,")",sep = "")
  eval(parse(text = ye2))

  equis1 <- ""
  for (i in 1:ncol(p)) {
    equis1 <- paste(equis1,"x",i,","sep = "") }
  equis1 <- substring(equis1,1,nchar(equis1)-1)

  equis2 <- ""
  equis2 <- paste(equis2,"equis <- cbind(",equis1,")",sep = "")
  eval(parse(text = equis2))

  #NET 1
  mod.yx<-nnet(yes,equis, size=3, linout=T,rang = 0.01, decay = 4e-5,
    trace = F, maxit = 10000)
  #Once trained enter desired "y" values.
  for (i in 1:ncol(M1)) {if (cb[i] == "min"){
    cadena <- paste("yy", i, "=min(M1[,"i,"])",sep = "")
  } else if(cb[i] == "max"){
    cadena <- paste("yy", i, "=max(M1[,"i,"])",sep = "") }
  cadena
  eval(parse(text = cadena))}

```

Código en R. NNET

```

datayy <- ""
for (i in 1:ncol(M1)) {
  datayy <- paste(datayy,"yy",i,",",sep = "")
}
datayy <- substring(datayy,1,nchar(datayy)-1)

datay <- ""
datay <- paste(datay,"data.y <- cbind(",datayy,")",sep = "")
eval(parse(text = datay))

data.y <- data.frame(data.y)

#These are the predicted optimized "x" values
pred.x1 <-predict(mod.yx, data.y, type = "raw")

#NORMALIZE r
normalr <- ""
for (i in 1:ncol(r)) {
  normalr <- paste(normalr,"r[",i,"] <- (r[",i,"]-min(r[",
                    i,"])) / (max(r[",i,"])-min(r[",i,"])",
                    sep = "", "\n")
}
eval(parse(text = normalr))

#NET 2
mod.xy<-nnet(equis,r,size=4,linout=T, rang = 0.01, decay = 4e-5,
            trace = F, maxit = 10000)
pred.y <-predict(mod.xy, pred.x1, type = "raw")

# PREDICTION
# DECODE
predic <- ""
for (i in 1:ncol(r)) {
  predic <- paste(predic,"y",i, ".predict <-pred.y[,",i,"]*(max(y",i,
    ") -min(y",i,") +min(y",i,")",sep="", "\n")
}
eval(parse(text = predic))

predic2 <- ""
for (i in 1:ncol(r)) {
  predic2 <- paste(predic2,"y",i, ".predict",sep="")}
predic2 <- substring(predic2,1,nchar(predic2)-1)

optimo <-""
optimo <- paste(optimo,"OPT.RNF <- round(c(",predic2,"),3)",sep = "")
eval(parse(text = optimo))
return(OPT.RNF)}

```

Simulación

Para la simulación Monte Carlo se programó el siguiente código:

Código en R. Simulación

```
#Predict of each model
predichos <- function(M) {
  cade <- ""
  for (i in 1:ncol(M)) {cade <- paste(cade,"modelo",i,sep = "")}

  pre <- ""
  for (i in 1:ncol(M)){pre <- paste(pre,"predict(modelo",i,")",",",sep = "")}
  pre <- substring(pre,1,nchar(pre)-1)

  predi<-""
  for (i in 1:ncol(M)) {
    predi <- paste("bx <- matrix(c(",pre,")",nrow(p))",sep = "")}
  eval(parse(text = predi))}
predichos(M)

#Regression for r
regresion2 <- function(p,r){
  equis <-""
  equis2 <- ""
  equis3 <- ""
  for (i in 1:ncol(p)) {equis <- paste(equis,"x",i,"+",sep = "")
    equis2 <- paste(equis2,"x",i,i,"+",sep = "")
    equis3 <- paste(equis3,"x",i,i," <- x",i,"^2 \n ",sep = "")}
  equis <- substring(equis,1,nchar(equis)-1)
  equis2 <- substring(equis2,1,nchar(equis2)-1)
  equis3 <- substring(equis3,1,nchar(equis3)-1)
  eval(parse(text = equis3))
  y1<-r[,1]
  y2<-r[,2]

  modelos <-""
  for (i in 1:ncol(r)) {
    modelos <- paste(modelos,"mode",i,"<-lm(r[,",i,"]~(",equis,")^2",
      "+",equis2,")\n",sep = "")}
  eval(parse(text = modelos))
  #####
  dividir <- function(target, index) {index <- sort(index)
    substr(rep(target, length(index) + 1),
      start = c(1, index),
      stop = c(index -1, nchar(target)))}
```

Código en R. Simulación

```

inter <- function(v1,v2){
  ord1 <- 2*(1:length(v1))-1
  ord2 <- 2*(1:length(v2))
  c(v1,v2)[order(c(ord1,ord2))]  }
insertar <- function(target, insert, index) {
  insert <- insert[order(index)]
  index <- sort(index)
  paste(inter(dividir(target, index), insert), collapse="") }
#####
nombres <- ""
for (i in 1:ncol(M)) {
  nombres<-paste(nombres,"names(mode",i,'$coefficients)<-c("", "x1",
                    "x2", "x3", "x1:x1", "x2:x2", "x3:x3",
                    "x1:x2","x1:x3","x2:x3")',"\n",sep = "")}
eval(parse(text = nombres))
cade=""; cade2=""
for (j in 1:ncol(r)) {cade=""
  cade2=""
  cadena=paste("for (i in (2:length(mode",j,"$coefficients))){"
    if (is.na(mode",j,"$coefficients[i])==F)
    {
      cade=paste(cade,mode",j,"$coefficients[i],\"*\",
        paste(insertar(names(mode",j," $coefficients[i]),
          c("\\"),c(1)),\"+\",sep=\\\"\\\"),sep=\\\"\\\" )}else{
      cade2=paste(cade2,names(mode",j,"$coefficients[i]),
        sep=",")
    } }",sep="")

  eval(parse(text=cadena))
  cade <-gsub(":",")*\"",cade)
  cade <- gsub("x","x[\"",cade)
  cade <- substring(cade,1,nchar(cade)-1)
  cade3 <- paste("mode",j,"$coefficients[1]\"",sep="")
  cade3 <- eval(parse(text=cade3))
  eval(parse(text=paste("mod",j,"<- function(x){\"",cade3,\"",cade,\"\"",
    sep="\"\"\"))  }}

nc <- ncol(M); nr <- nrow(p)

#Indicate the number of simulations
NREP <- 5000

```

Código en R. Simulación

```

#Start the result matrices of each method at zero
MD <- matrix(0, NREP,2)
MM <- matrix(0, NREP,2)
MT <- matrix(0, NREP,2)
MMAD <- matrix(0,NREP,2)
MTAD <- matrix(0,NREP,2)
MMM <- matrix(0,NREP,2)
MNNB <- matrix(0,NREP,2)
MNNF <- matrix(0,NREP,2)

##### For cycle that applies each method #####
for (i in 1:NREP){
  mu = c(0,0)
  # rho controls the correlation
  rho <- cbind( c (1, 0.9), c (0.9, 1))
  vars <- 1
  Sigma = vars * rho
  r=y+rmvnorm(nr,mu,Sigma)
  #simuladas[[i]] <- r
  regre <- regresion2(p,r)
  #MD[i,] <- (Deseabilidad(r,p,cb,w))
  #MM[i,] <- (MOORA(r,w,cb))
  #MT[i,] <- (TOPSIS(r,w,cb))
  #MMAD[i,] <- (MOORAAD(r,w,cb))
  #MTAD[i,] <- (TOPSISAD(r,w,cb))
  #MMM[i,] <- (MMMOORA(r,w,cb))
  #MNNB[i,] <- (NEURALNETB(r,p,cb,w))
  #MNNF[i,] <- (NEURALNETF(r,p,cb,w))
}

#Get the vectors of means and variances
Result <- round(cbind(MD,MM,MT,MMAD,MTAD,MMM,MNNB,MNNF),3)
Medias <- (colMeans(Result))
Medias

VAR <- apply(Result, 2, var)
VAR

#METRICS

# Individual RTD
RTDI <- (rep(0,length(Medias)))
k=1

```


Código en R. Desirability

```
for (i in 1:length(RTDI)){
  if (k==1){RTDI[k] <- abs(Medias[k]-OP[1])/OP[1]}
  if (k==2){RTDI[k] <- abs(Medias[k]-OP[2])/OP[2]}
  if (k==3){RTDI[k] <- abs(Medias[k]-OP[1])/OP[1]}
  if (k==4){RTDI[k] <- abs(Medias[k]-OP[2])/OP[2]}
  if (k==5){RTDI[k] <- abs(Medias[k]-OP[1])/OP[1]}
  if (k==6){RTDI[k] <- abs(Medias[k]-OP[2])/OP[2]}
  if (k==7){RTDI[k] <- abs(Medias[k]-OP[1])/OP[1]}
  if (k==8){RTDI[k] <- abs(Medias[k]-OP[2])/OP[2]}
  if (k==9){RTDI[k] <- abs(Medias[k]-OP[1])/OP[1]}
  if (k==10){RTDI[k] <- abs(Medias[k]-OP[2])/OP[2]}
  if (k==11){RTDI[k] <- abs(Medias[k]-OP[1])/OP[1]}
  if (k==12){RTDI[k] <- abs(Medias[k]-OP[2])/OP[2]}
  if (k==13){RTDI[k] <- abs(Medias[k]-OP[1])/OP[1]}
  if (k==14){RTDI[k] <- abs(Medias[k]-OP[2])/OP[2]}
  if (k==15){RTDI[k] <- abs(Medias[k]-OP[1])/OP[1]}
  if (k==16){RTDI[k] <- abs(Medias[k]-OP[2])/OP[2]}
  k = k + 1}
RTDI

# Métrica GPE/RTD
RTD <- c((sum(RTDI[1],RTDI[2])),sum(RTDI[3],RTDI[4]),
         sum(RTDI[5],RTDI[6]),sum(RTDI[7],RTDI[8]),
         (sum(RTDI[9],RTDI[10])),sum(RTDI[11],RTDI[12]),
         sum(RTDI[13],RTDI[14]),sum(RTDI[15],RTDI[16]))
```