

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



**“MODELACIÓN Y OPTIMIZACIÓN APLICADA A LA
SIMULACIÓN DE PROCESOS Y SISTEMAS DE MANUFACTURA”**

POR

MARÍA GUADALUPE VILLARREAL MARROQUÍN

TESIS

EN OPCIÓN AL GRADO DE MAESTRO EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

San Nicolás de los Garza, N. L.

Diciembre de 2007

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



**"MODELACIÓN Y OPTIMIZACIÓN APLICADA A LA
SIMULACIÓN DE PROCESOS Y SISTEMAS DE MANUFACTURA"**

POR

MARÍA GUADALUPE VILLARREAL MARROQUÍN

TESIS

EN OPCIÓN AL GRADO DE MAESTRO EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

San Nicolás de los Garza, N. L.

Diciembre de 2007

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
DIVISIÓN DE ESTUDIOS DE POSGRADO

Los miembros del Comité de Tesis recomendamos que la tesis “**Modelación y Optimización Aplicada a la Simulación de Procesos y Sistemas de Manufactura**”, realizada por la alumna **María Guadalupe Villarreal Marroquín**, con número de matrícula **01126981**, sea aceptada para su defensa como opción al grado de Maestro en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

Dr. Mauricio Cabrera Ríos

Asesor

Dra. Deniz Özdemir

Revisor

Dr. Oscar L. Chacón

Revisor

Dr. Guadalupe Alan Castillo Rodríguez

Subdirector

División de Estudios de Posgrado

DEDICATORIA

Quiero dedicar este trabajo a mis padres Sr. Rogelio Villarreal Silva y Sra. María H. Marroquín Salazar, que me han apoyado incondicionalmente y han confiado en mí.

A mis hermanos Juan José Villarreal Marroquín y Ana Patricia Villarreal Marroquín.

AGRADECIMIENTOS

Primeramente a Dios por estar conmigo siempre y permitirme lograr este sueño.

Un especial agradecimiento a mi asesor el Dr. Mauricio Cabrera Ríos, por haberme apoyado incondicionalmente durante el desarrollo de este trabajo, así como durante mi estancia en el verano científico 2005 y la maestría. Así mismo por haber confiado en mí y brindarme su amistad.

Quiero agradecer a la Dra. Deniz Özdemir y al Dr. Oscar L. Chacón por su apoyo y colaboración en este trabajo.

Al personal de la empresa donde se desarrolló parte de este trabajo, especialmente a Pedro Silva, Tlálóc Maldonado, Juan A. González, Ricardo Castillo y Sergio Alanís.

A mi grupo de trabajo Matilde Sánchez, Lyzett Uribe, Hugo A. Pérez, y Miguel A. Urbano a quien le agradezco enormemente por su apoyo y cuidados durante todos los viajes que realizamos juntos, especialmente durante nuestra estancia en The Ohio State University.

A mis compañeros y profesores del PISIS por todo su apoyo. De manera especial a Jania A. Saucedo.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), a la Facultad de Ingeniería Mecánica y Eléctrica y a la Universidad Autónoma de Nuevo León, por las becas proporcionadas durante mis estudios de maestría.

Quiero agradecer al Dr. José M. Castro y a su esposa Ana por su hospitalidad y apoyo durante mi estancia de investigación en The Ohio State University, así como a su equipo de trabajo Eric Chang, Matt Mulyana, Yottha Srithep y Yunior Hioe; y también al personal de OSU por todo su apoyo.

A mis padres y hermanos por su apoyo y confianza, por todo su amor. De igual manera a toda mi familia.

A Israel Salazar por todo su apoyo, paciencia, confianza y amor, así como a toda su familia.

RESUMEN

La simulación por computadora es una herramienta imprescindible para las compañías manufactureras. En un nivel micro, la simulación contribuye a diseñar procesos de manufactura que permiten producir bienes con múltiples especificaciones y, en un nivel macro, ayuda a diseñar los sistemas que integran a estos procesos de manera eficiente y costeable para asegurar la viabilidad financiera de la compañía. Los modelos de simulación permiten crear representaciones adecuadas del sistema o proceso para tratar diferentes combinaciones de valores de las variables controlables y así generar predicciones sobre varios criterios de desempeño de interés.

Tradicionalmente la simulación se ha utilizado para generar una cantidad limitada de instancias sobre las que finalmente se selecciona una. Esto último es en sí una mejora de proceso y no propiamente una optimización. En optimización la idea es buscar combinaciones de valores de las variables controlables que demostrablemente sean altamente atractivas en términos de los criterios de desempeño de interés, y en su mejor caso, inmejorables.

En este trabajo se propone un método de optimización de simulaciones para apoyar a personas que necesiten realizar la optimización de procesos por medio de un modelo de simulación. El método propuesto es suficientemente sencillo y transparente para ser codificado con recursos computacionales modestos y utilizado con poco entrenamiento en optimización. El método comienza con un diseño de experimentos con el cual se determina una solución incumbente y un metamodelo inicial. En cada iteración, el

metamodelo es utilizado para generar un punto atractivo bajo el cual se ejecuta una simulación. El valor simulado del nuevo punto se compara con el incumbente para actualizarlo, y finalmente, se prueban varios criterios de parada. Si se decide seguir, el nuevo punto se añade al conjunto de puntos disponibles en una nueva iteración y un nuevo metamodelo es construido.

La validación del método se realizó mediante su aplicación a diferentes funciones de prueba para optimización global, así como a funciones no lineales polinomiales y no polinomiales. Los resultados empíricamente apuntan a que el método es capaz de converger a soluciones cercanas al óptimo global con pocas ejecuciones de la simulación. Adicionalmente el método se aplicó con resultados competitivos a diferentes modelos de simulación sencillos los cuales sirven como casos de demostración del método propuesto así como evidencia de la aplicabilidad del método a simulaciones de eventos discretos y simulaciones de fenómenos continuos.

Dentro de la tesis se presenta además un caso de estudio real, la simulación de una línea de pintura de partes automotrices de una compañía de la ciudad de Monterrey. El modelo que se desarrolló en este caso fue utilizado para evaluar cambios propuestos en la línea. La determinación de la mejor configuración del sistema se abordó utilizando el método de optimización para simulaciones propuesto en este trabajo. Este caso sirvió para mostrar la instauración del método para optimizar un sistema a escala real. Similarmente, se presenta un caso de estudio de simulación de fenómenos continuos, específicamente en el terreno del moldeo por inyección.

Durante la realización de este trabajo se puso especial cuidado en que todos los requerimientos computacionales pudieran ser instauradas en MS Excel, por lo que los requerimientos de paquetería computacional son modestos. Esto se hizo pensando en mantener la factibilidad de la aplicación del método aquí desarrollado en la industria mexicana.

ESTRUCTURA DE LA TESIS

La tesis está organizada de la siguiente manera:

En el Capítulo 1 se presenta el objetivo, la motivación y la justificación de este trabajo. Se muestra también la hipótesis de la tesis y un resumen del desarrollo de la misma.

En el Capítulo 2 se presentan los antecedentes teóricos; se incluyen conceptos de optimización de simulaciones, diseño de experimentos, metamodelación y optimización.

En el Capítulo 3 se describe a detalle el método propuesto de optimización aplicado a simulaciones, así como sus características técnicas.

En el Capítulo 4 se muestra la validación del método descrito en el Capítulo 3. La validación comprende la aplicación del método a diferentes funciones de prueba para optimización global, así como a funciones no lineales tanto polinomiales como no polinomiales (Redes Neuronales Artificiales). Del mismo modo se muestra la aplicación del método a diferentes modelos de simulación sencillos. Para estos últimos se muestra la comparación del desempeño contra un optimizador comercial. Con estos casos se muestra la sencillez y competitividad del método propuesto.

En el Capítulo 5 se presenta la descripción y la caracterización de un sistema de pintado de partes automotrices de una compañía de la ciudad de Monterrey. Se muestra también el modelo de simulación desarrollado para la línea de pintura así como la optimización del sistema con el objetivo de incrementar la producción. Esto último se logró mediante la aplicación del método de optimización para simulaciones propuesto en el Capítulo 3.

El Capítulo 6 contiene varios escenarios de optimización de la simulación de un proceso de moldeo por inyección que comprenden diferentes aspectos en la manufactura de una placa rectangular hecha de un material termoplástico. En este caso también se utilizó el método propuesto en el Capítulo 3.

Finalmente en el Capítulo 7 se presentan las conclusiones, las contribuciones y el trabajo a futuro.

Como material adicional, posterior a la sección de capítulos se incluyen las referencias bibliográficas, los apéndices y los índices de tablas y figuras.

ÍNDICE

RESUMEN	vii
ESTRUCTURA DE LA TESIS	ix
CAPÍTULO 1. INTRODUCCIÓN	1
1.1. Objetivo	2
1.2. Motivación y Justificación	2
1.3. Hipótesis	3
1.4. Desarrollo	3
CAPÍTULO 2. ANTECEDENTES	6
2.1. Optimización aplicada a la simulación	6
2.2. Diseño estadístico de experimentos	10
2.3. Metamodelación	15
2.4. Optimización	17
2.4.1. Clasificación de modelos de optimización	18
2.4.2. Criterios de Parada	21
CAPÍTULO 3. MÉTODO PROPUESTO	23
3.1. Descripción del método	24
CAPÍTULO 4. EVALUACIÓN DEL MÉTODO PROPUESTO	30
4.1. Funciones de prueba para optimización global	30

4.2.	Redes Neuronales Artificiales (RNAs)	36
4.3.	Ejemplos de simulación sencillos	39
4.3.1.	Sala de Hospital	39
4.3.2.	Línea de embotellado	50
4.3.3.	Moldeo por Inyección de la Carátula de un teléfono celular	54
4.4.	Conclusiones	57

CAPÍTULO 5. CASO DE ESTUDIO 1 **58**

5.1.	Descripción del sistema	59
5.2.	Modelo de simulación	62
5.3.	Validación y resultados de interés del modelo de simulación	66
5.4.	Optimización del sistema	68
5.5.	Decisión Final Propuesta	77
5.6.	Conclusiones	78

CAPÍTULO 6. CASO DE ESTUDIO 2 **79**

6.1.	Descripción y modelo de simulación del proceso	80
6.2.	Optimización del proceso	81
6.3.	Consideraciones para optimización de problemas multicriterio	86

CAPÍTULO 7. CONCLUSIONES Y TRABAJO FUTURO **89**

7.1.	Conclusiones	89
7.2.	Contribución	91
7.3.	Trabajo a Futuro	92

APÉNDICES	95
Apéndice A	95
Apéndice B	100
Apéndice C	108
Apéndice D	114
Apéndice E	116
LISTA DE TABLAS	117
LISTA DE FIGURAS	120
BIBLIOGRAFÍA	122
AUTOBIOGRAFÍA	127
PUBLICACIONES	128

CAPÍTULO 1

INTRODUCCIÓN

La simulación por computadora es una herramienta imprescindible para las compañías manufactureras. En un nivel micro, la simulación contribuye a diseñar procesos de manufactura que permitan producir bienes con múltiples especificaciones y en un nivel macro, ayuda a diseñar los sistemas que integran a estos procesos de manera eficiente y costeable para asegurar la viabilidad financiera de la compañía.

Los especialistas en simulación tienen por lo general un entrenamiento sólido en modelación de fenómenos físicos y químicos a nivel proceso o bien en la modelación de las interrelaciones entre los diferentes procesos a nivel sistema que les permiten crear representaciones adecuadas para tratar diferentes combinaciones de valores de las variables controlables en sus modelos y así generar predicciones sobre varios criterios de desempeño de interés. Sin embargo, estos mismos especialistas no necesariamente poseen el conocimiento para optimizar sus diseños con base a los modelos de simulación que desarrollan. Por esta razón, tradicionalmente la simulación se ha utilizado para generar una cantidad limitada de instancias sobre las que finalmente se selecciona una. Esto último, vale precisar, es en sí una mejora de proceso y no propiamente una

optimización. En optimización la idea es buscar combinaciones de los valores de las variables controlables que demostrablemente sean altamente atractivos en términos de los criterios de desempeño, y en su mejor caso, inmejorables. Existe, entonces, potencial que explotar en la aplicación adecuada y eficiente de técnicas de optimización en los diseños de procesos y sistemas de producción a través de la simulación.

Aunque este potencial ya ha sido detectado por los productores de paquetería computacional de simulación, las soluciones que han desarrollado toman en general la forma de módulos de optimización con un costo adicional a las licencias -que son ya bastante caras- de los paquetes de simulación. Por otro lado, especialmente a nivel proceso, la flexibilidad de las herramientas de optimización provistas es baja en términos del número de criterios de desempeño que se pueden involucrar (usualmente una sola). A pesar de que el proceso de diseño se beneficia al contar con rutinas de optimización para simulación, también es claro que el beneficio sería aún mayor si estas rutinas fueran más transparentes y flexibles para que el usuario de los paquetes de simulación pudiera capitalizar aún más en las fortalezas de la optimización. Tomando en cuenta estos puntos, se presenta el siguiente objetivo de tesis.

1.1. Objetivo

El objetivo de esta tesis es desarrollar un método de optimización aplicado a la simulación tanto de procesos como de sistemas de manufactura. Se busca que el método desarrollado sea entendible, conveniente de codificar y que tenga la flexibilidad suficiente para poder representar los intereses de los especialistas en simulación con fidelidad.

1.2. Motivación y Justificación

El desarrollo de este trabajo fue motivado por un caso de estudio real donde se requirió caracterizar estadísticamente la línea de pintura de una compañía de la ciudad de Monterrey dedicada a la fabricación de partes automotrices y, con esta información,

crear un modelo de simulación. El modelo resultante debía ser utilizado para evaluar cambios propuestos en esta línea así como para prescribir políticas de operación. Con la toma de decisiones asociada a este caso, quedó al descubierto la necesidad de no solamente elegir una configuración operativa de la línea, sino hacerlo de una manera óptima o, en su defecto, altamente atractiva. Los paquetes que ofrecían esta capacidad, además de tener un costo adicional no despreciable, no eran muy transparentes en la generación de estas decisiones así como tampoco daban mucha flexibilidad de crear funciones objetivo personalizadas a los intereses de los ingenieros de la planta. Los trabajos descritos en la literatura de optimización aplicada a la simulación (*Simulation Optimization*, en inglés) ofrecían varias alternativas, sin embargo, la gran mayoría necesitaban de un alto nivel de habilidad y trabajo de programación para su instauración efectiva. Por otro lado, se detectó que esta situación no era exclusiva de la simulación de eventos discretos –tan aplicada a sistemas de manufactura- sino que también era característica de la simulación de eventos continuos, como la que se aplica para representar fenómenos físicos y químicos en los procesos de manufactura. Por tanto, la necesidad de optimizar modelos de simulación de manera clara y efectiva para potenciar los beneficios del uso de la simulación de procesos y sistemas de manufactura en la industria provee la justificación del trabajo que aquí se presenta.

1.3.Hipótesis

Es posible encontrar soluciones altamente atractivas para modelos de simulación de procesos y sistemas de manufactura utilizando estratégicamente técnicas de diseño de experimentos, metamodelación y optimización; cuidando, además, de no requerir un número muy alto de corridas de simulación.

1.4.Desarrollo

El desarrollo de este trabajo implicó: 1) Obtener conocimientos necesarios de simulación y técnicas de optimización aplicada a la simulación, 2) Aprender a utilizar software de simulación tanto de procesos como de sistemas de manufactura, 3) Realizar

la caracterización de un sistema de manufactura real, 4) Construir el modelo de simulación del sistema real y utilizar el modelo de simulación de diferentes partes reales en un proceso de moldeo por inyección, 5) Llevar a cabo los análisis pertinentes de optimización aplicada a la simulación y 6) Resumir los pasos desarrollados en un método de optimización aplicado a la simulación con las características anteriormente identificadas: ser entendible, conveniente de codificar y flexible para poder representar los intereses de los especialistas en simulación con fidelidad.

Inicialmente se requirió de una búsqueda bibliográfica que incluyó artículos en revistas de investigación y libros especializados de la disciplina. Posteriormente, y tras varias visitas a la planta con el fin de recabar información sobre la línea, se adquirió un software de simulación comercial de eventos discretos (SIMUL8) para abordar la construcción del modelo de simulación de la línea de pintura. Este mismo software se utilizó además para realizar validaciones adicionales del método de optimización propuesto en la variante de sistemas de manufactura. Nuestro grupo de investigación contaba ya con una versión de un software de simulación de flujo de polímeros (MoldFlowTM) para la experimentación en el área de simulaciones de procesos, aunque después, a través de una colaboración con The Ohio State University se tuvo acceso a una versión más poderosa.

Posterior al desarrollo de los modelos de simulación se definieron los problemas de optimización respectivos. Las funciones objetivo estuvieron conformadas por las salidas de los modelos de simulación como el tiempo requerido para producir una parte en el proceso de moldeo por inyección o la cantidad de piezas producidas en una jornada de trabajo en el sistema de manufactura. Las restricciones fueron principalmente cotas inferiores y superiores. Las variables de decisión o controlables representaron aspectos como el número de trabajadores a asignar a una estación de trabajo, la capacidad máxima de determinado buffer o bien, la temperatura del polímero y la presión de inyección. Entonces, los objetivos de los problemas de optimización fueron encontrar los

niveles más adecuados de las variables controlables optimizando una función de las salidas del modelo de simulación.

Como parte del método de optimización aplicada a la simulación que se propone en este trabajo, la construcción de los problemas de optimización involucró la utilización de técnicas tanto de diseño estadístico de experimentos como de modelación empírica (metamodelación). La optimización en sí fue abordada utilizando la técnica de múltiples comienzos con un optimizador local (*Solver* de MSExcel).

Finalmente, después de efectuarse la optimización de los modelos de simulación, se realizaron las interpretaciones pertinente para cada caso.

La descripción detallada de lo puntualizado en este apartado se encuentra organizada en los capítulos posteriores de la siguiente manera: en el Capítulo 2 se presentan los antecedentes; la descripción del método de optimización aplicado a simulaciones propuesto y su validación se encuentran en los Capítulos 3 y 4 respectivamente; la descripción y solución del caso de estudio del sistema de pintado se presenta en el Capítulo 5. El Capítulo 6 presenta la optimización de la simulación de un proceso de moldeo por inyección, y finalmente en el Capítulo 7 se presentan las conclusiones, las contribuciones y el trabajo a futuro.

CAPÍTULO 2

ANTECEDENTES

En este capítulo se presentan los antecedentes del área de optimización aplicada a la simulación y la información teórica de diferentes técnicas integradas en la metodología propuesta en esta tesis.

2.1. Optimización aplicada a la simulación

El objetivo de los métodos de optimización aplicada a la simulación es proveer una estructura para determinar los valores de las variables controlables con el fin de optimizar una función de las salidas de un modelo de simulación (Swisher et al., 2000). Una rutina de optimización utiliza la salida del simulador como función objetivo dependiente de las entradas que fueron alimentadas al modelo y, en base a ésta y a evaluaciones anteriores, propone un nuevo conjunto de soluciones de entrada. Este procedimiento continúa hasta que se cumple un criterio de terminación (April et al., 2004). La **Figura 2.1** muestra un diagrama genérico de optimización aplicada a la simulación.

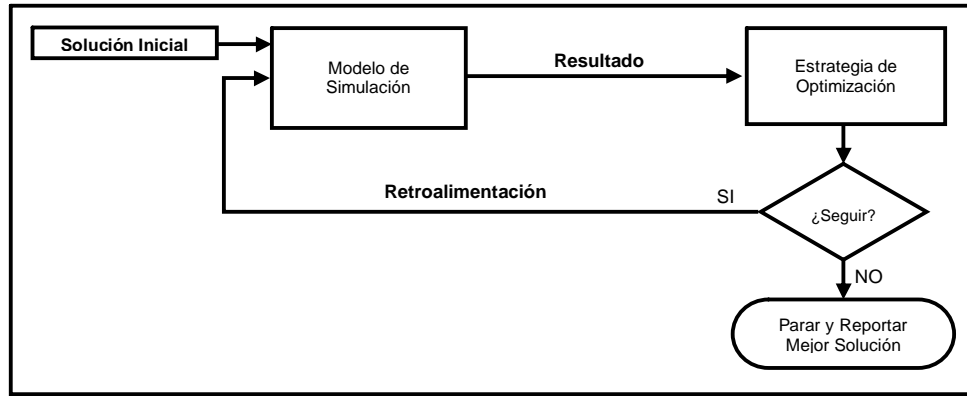


Figura 2.1: Diagrama de Optimización aplicada a la Simulación.

El problema de optimización de interés, en general, está dado por:

$$\begin{array}{ll}
 \text{Encontrar} & \mathbf{x} \quad \text{para} \\
 \\
 \text{Minimizar} & E[f(\mathbf{x})] \\
 \text{Sujeto a} & \mathbf{Ax}^T \leq \mathbf{b} \quad (\text{Restricciones}) \\
 & \mathbf{g}_l \leq \mathbf{G}(\mathbf{x}) \leq \mathbf{g}_u \quad (\text{Requerimientos}) \\
 & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \quad (\text{Cotas})
 \end{array} \tag{2.1}$$

Donde $\mathbf{x} = (x_1, x_2, \dots, x_v)$ es un vector que contiene los valores de las v variables controlables del modelo de simulación cuyos valores pueden ser continuos o discretos. El conjunto de restricciones son lineales, los valores de la matriz \mathbf{A} y del vector del lado derecho \mathbf{b} son conocidos. Los requerimientos son cotas inferiores y superiores impuestas por funciones que pueden ser lineales o no lineales; los valores de las cotas \mathbf{g}_l y \mathbf{g}_u son conocidos (\mathbf{g}_l y \mathbf{g}_u son vectores). Todas las variables (v) tienen cotas inferiores y superiores, \mathbf{l} y \mathbf{u} . Todo esto, para minimizar el valor esperado de una función objetivo $f(\mathbf{x})$ a través de un número de réplicas. Por supuesto, en cada caso específico de optimización las restricciones, requerimientos y cotas pueden también no estar presentes.

El requerimiento de la optimización matemática clásica de tener explícitamente ecuaciones tanto como función objetivo como restricciones, limita su aplicación directa

cuando se refiere a simulación. Un avance en este sentido proviene de las áreas estadísticas de modelación empírica (metamodelación) y diseño de experimentos (DOE), que proveen la capacidad de crear modelos estadísticos de predicción a partir de un número limitado de muestras medibles. Estas muestras bien pueden tomar la forma de instancias puntuales de un modelo de simulación. Los modelos estadísticos, dada su forma matemática explícita, pueden tomar entonces los lugares de las funciones objetivo y las restricciones en un problema de optimización clásica. La clave en este último caso está en saber seleccionar un número de muestras representativo que permita crear los modelos estadísticos de una manera adecuada. Un número muy grande de muestras implicaría una cantidad impráctica de simulaciones. Esto sería más grave en situaciones donde una simulación toma tiempos muy largos de computación, como en las simulaciones de fenómenos físicos con mallados muy finos de elementos finitos (González, 2005). Por otro lado, las muestras deberán ser suficientemente representativas para añadir conocimiento al modelo estadístico.

Otra opción que ha sido desarrollada para optimización de simulaciones es utilizar directamente las salidas de la simulación de manera iterativa hasta converger a una combinación final. Dentro de estos métodos podemos encontrar algoritmos de optimización que no utilizan derivadas y otros que utilizan estrategias de inteligencia artificial como los algoritmos genéticos. Los primeros métodos, utilizados para enfrentar problemas altamente no lineales, es común que requieran una secuencia de evaluaciones de la función objetivo, implicando aquí una ejecución de la simulación (además de réplicas si es requerido). Estos métodos comúnmente empiezan desde un punto determinado para después iniciar una nueva secuencia desde otro punto distinto y así encontrar varios puntos competidores. Las estrategias de inteligencia artificial de una u otra manera también requieren generar soluciones iniciales cuya diversidad les permita generar otras soluciones hasta llegar a una que sea atractiva. En ambos casos, el número requerido de simulaciones puede dispararse, impactando así negativamente el costo computacional del proceso de optimización. También en estos casos, no sería plausible poder optimizar una simulación que tome tiempos muy largos como por ejemplo, un mallado muy fino de elementos finitos. Una sola ejecución del modelo de simulación de

estos últimos puede alcanzar a tomar varias semanas de tiempo computacional de una estación de trabajo dedicada al 100% a este fin (González, 2005).

Existen varios métodos más de optimización aplicada a la simulación en la literatura. Estos métodos incluyen técnicas como programación no lineal, métodos de superficie de respuesta, aproximación estocástica, búsqueda aleatoria, ranking y selección, metaheurísticas, entre otras. Revisiones de literatura comprensivas acerca de técnicas de optimización aplicada a la simulación bajo diferentes enfoques se pueden encontrar en (Fu, 1994; Andradóttir, 1998; Azadivar, 1999; Swisher et al., 2000; Fu, 2001; Ólafsson y Kim, 2002; Fu et al., 2005).

De particular interés en este trabajo son los métodos de optimización aplicada a la simulación que utilizan metamodelos. Una revisión amplia del uso de metamodelos se puede encontrar en (Barton, 1998). La regresión lineal estándar aparece como una de las técnicas más populares para este fin. Más recientemente las Redes Neuronales Artificiales (RNAs) también han ganado popularidad (April et al., 2003). Buot et al. (2001) utilizaron una RNA dentro del problema de minimizar la ocupación de dos zonas del área de embarque de una empresa chilena. Altiparmak et al. (2002) construyeron una RNA basado en técnicas de inteligencia artificial para determinar el tamaño óptimo de los buffers en una línea de ensamble asincrónica. Martin y Simpson (2002) desarrollaron un método basado en metamodelos kriging adaptativo con el objeto de representar lo mejor posible toda la superficie de respuesta; Cheng y Currie (2004) utilizaron aproximación Bayesiana para construir una regresión cuadrática; Moniaci et al. (2005) utilizaron una RNA para representar la efectividad de una línea de producción. Dado el alto grado de éxito del uso de metamodelos, y a que se ha reportado la efectividad del uso de diseño de experimentos para optimización aplicada a la simulación de fenómenos físicos y químicos (Cabrera-Ríos et al., 2002a; Cabrera-Ríos et al., 2002b; Cabrera-Ríos et al., 2002c; Castro et al., 2004; Castro et al., 2005; Castro et al., 2006), se decidió seguir esta línea en este trabajo.

Muchas de las técnicas propuestas en la literatura, sin embargo, pueden resultar un reto para personas con conocimientos modestos de optimización y programación. Por esta razón se ha buscado automatizar este proceso. Actualmente varios paquetes comerciales de simulación tanto estocástica como de fenómenos físicos, contienen módulos de optimización utilizando, entre otros, estrategias metaheurísticas (April et al., 2003). Con este esquema en general se gana rapidez a un costo de perder certeza sobre la optimalidad de las soluciones encontradas. Algunos de estos módulos son: *OptQuest*, incluido en Arena, ProModel, SIMUL8, entre otros, que utiliza Scatter Search como estrategia de búsqueda, combinando además Tabu Search y Redes Neuronales Artificiales; y *AutoStat*, incluido en AutoMod, que utiliza principalmente algoritmos genéticos como estrategia de búsqueda (Fu et al., 2005). Estos módulos de optimización tienen un costo adicional a la licencia del software de simulación y, aunque aumentan grandemente la capacidad del usuario para optimizar, es común que no permitan la construcción a medida de funciones objetivo o de restricciones funcionales. Creemos que la flexibilidad y transparencia de la solución de los problemas de optimización pueden contribuir mucho al proceso creativo de diseño de procesos y sistemas y que, por tanto, son características deseables en los métodos de optimización aplicada a la simulación. De ahí que éstas se hayan incluido como cualidades a cuidar en el desarrollo del presente trabajo.

2.2. Diseño estadístico de experimentos

Las técnicas de diseño estadístico de experimentos nos ayudan a planear un experimento de tal forma que se recaben datos adecuados que puedan ser analizados con métodos estadísticos y así llegar a conclusiones objetivas y repetibles. Un experimento puede definirse como una prueba o serie de pruebas en las que se hacen cambios deliberados en las variables de entrada (factores) de un proceso o sistema (ver **Figura 2.2**) para observar e identificar las razones de los cambios que pudieran influir en la respuesta de salida (Montgomery, 2005). Un ejemplo puede ser un proceso de moldeo por inyección, donde una entrada corresponde a la materia prima y la salida corresponde a la pieza producida; dentro de las variables controlables de este proceso tenemos la

temperatura del molde y la presión de inyección, y una respuesta puede ser el tiempo que se requiere para fabricar la parte.

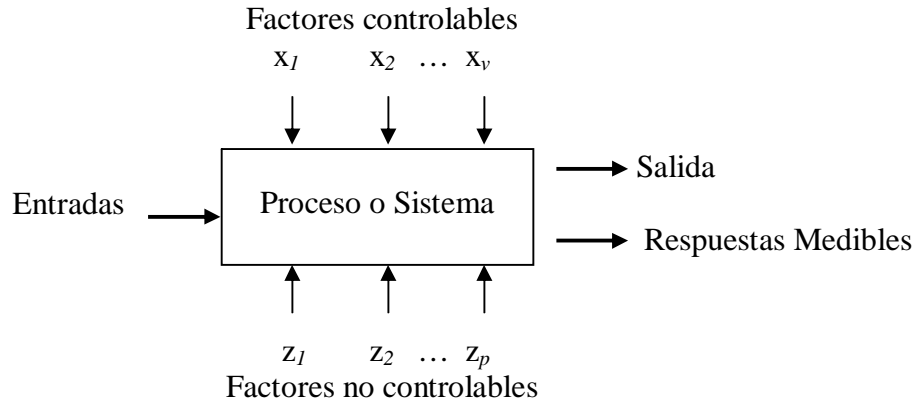


Figura 2.2: Modelo general de un proceso o sistema

Adicionalmente, dado que el diseño de experimentos es una técnica de muestreo, éste es fuertemente utilizado para obtener una representación estadística de la respuesta del proceso. A esta representación se le llama metamodelo o modelo empírico.

Una de las técnicas más populares de diseño de experimentos utilizada para ajustar metamodelos de segundo orden es el *diseño compósito central* (DCC). En general, un DCC para K factores (en este caso variables de decisión) consta de: un diseño factorial 2^K (o de un factorial fraccionado) con n_f corridas, $2K$ corridas axiales o estrella y n_c corridas centrales (Montgomery, 2005). Información relacionada con diseño factorial y factorial fraccionado puede consultarse en esta misma referencia.

Dentro de un DCC hay dos parámetros que deben especificarse: la distancia α de las corridas axiales al centro del diseño y el número de puntos centrales n_c . A continuación se analiza la selección de estos dos parámetros.

Corridas axiales

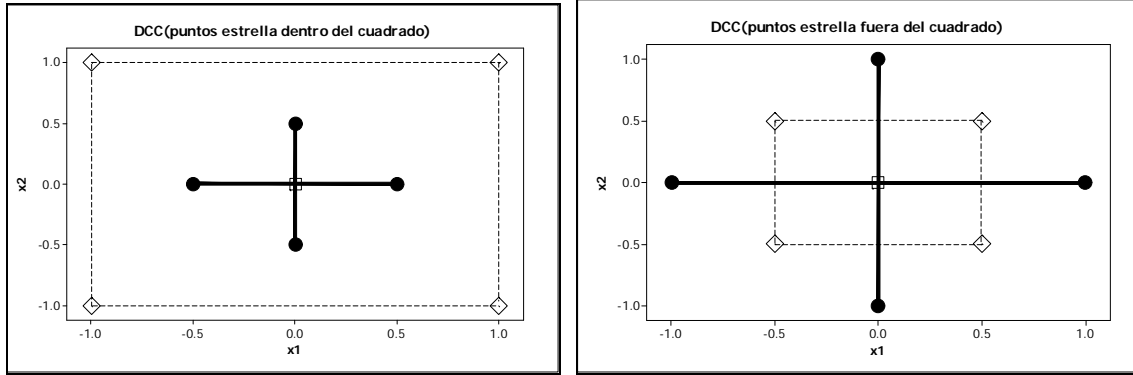
Es importante que un metamodelo proporcione predicciones adecuadas en toda la región de interés, puesto que la finalidad principal de éstos es la optimización y la localización del óptimo se desconoce antes de correr el experimento. Por tanto, tiene sentido utilizar un diseño que proporcione una precisión constante de estimación en todas las direcciones (Montgomery, 2005). Un *diseño rotatable* permite que la calidad de predicción se comporte de igual manera independientemente de la dirección en la que se explore a partir del centro del diseño. Esto quiere decir que la varianza de la predicción del metamodelo en un punto x ($\text{var}(\hat{f}(x))$) sólo depende de la distancia del punto x al centro del diseño y no de la dirección en la que se encuentre (Gutiérrez et al., 2004). Un DCC se hace rotar mediante la elección de α . El valor de α para la rotatibilidad depende del número de puntos en la porción factorial del diseño; $\alpha=(n_f)^{1/4}$ produce un DCC rotatable. Sin embargo cuando la región de interés es esférica, la mejor selección de α desde el punto de vista de la varianza de predicción es hacer $\alpha = \sqrt{K}$; este diseño coloca todos los puntos factoriales y axiales del diseño sobre la superficie de una esfera de radio \sqrt{K} (Montgomery, 2005). Las **Figuras 2.3a** y **2.3b** muestran un diseño compuesto central para dos variables utilizando un valor de α igual a 0.5 y 1 respectivamente, donde los puntos estrella están representados por los círculos sólidos. De igual manera, las **Figuras 2.4a** y **2.4b** muestran un diseño para tres variables.

Corridas centrales

La elección de α en un DCC está dictada principalmente por la región de interés. Cuando esta región es una esfera, el diseño debe incluir corridas centrales para proporcionar una varianza razonablemente estable de la respuesta predicha. En general se recomienda de tres a cinco corridas axiales (Montgomery, 2005).

Las **Figuras 2.3a** y **2.3b** muestran un DCC para $K = 2$, utilizando un valor de $\alpha=0.5$ y $\alpha=1$ respectivamente, y una corrida central. Del mismo modo, las **Figuras 2.4a** y **2.4b**

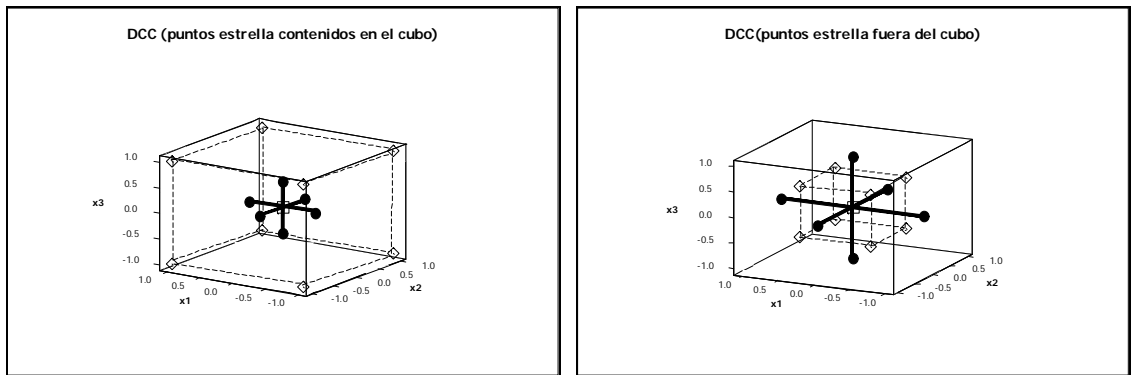
para $K = 3$. En las cuatro figuras los puntos circulares sólidos representan las corridas estrella, los rombos los puntos factoriales y el cuadrado los puntos centro.



(a)

(b)

Figura 2.3: Diseño Compósito Central para $K=2$



(a)

(b)

Figura 2.4: Diseño Compósito Central para $K=3$

Cabe mencionar que los casos mostrados anteriormente no son todas las formas de un DCC, pues el valor de $\alpha \in [0,1]$ y aunque los puntos del diseño factorial regularmente comprenden las esquinas del diseño, se pueden considerar puntos que no sean esquinas.

Además del DCC, otro diseño opcional para ajustar metamodelos es el *diseño D-Óptimo*. El diseño D-Óptimo proviene de un algoritmo computacional y es

particularmente utilizado cuando un diseño clásico no se puede aplicar ya sea por irregularidad geométrica de la zona de experimentación, o bien, por requerir un número específico de corridas no disponible por otras técnicas. La diferencia con los diseños clásicos tales como diseño factorial o factorial fraccionado es que la matriz de diseño en el D-Óptimo usualmente es no ortogonal, lo que significa que no todos los vectores columna de esta matriz son independientes entre sí, provocando que alguno(s) de los coeficientes del modelo ajustado estén correlacionados (Gutiérrez, 2004).

El diseño D-Óptimo incluye solamente un subconjunto de combinaciones de un conjunto inicial que usualmente está formado por todas las posibles combinaciones de las variables a varios niveles i.e. un diseño factorial completo, escogiéndolas de tal forma que se minimice la varianza de los estimadores de un modelo estadístico previamente especificado y de satisfacer las restricciones pertinentes. El algoritmo computacional generalmente utiliza un proceso de intercambio para seleccionar el conjunto de combinaciones (Montgomery, 2005). En general el modelo de optimización para definir un diseño D-Óptimo es como se muestra a continuación:

$$\left. \begin{array}{l} \max_{d \in D} |\mathbf{X}_d^T \mathbf{X}_d| \\ H(\mathbf{x}_d) \leq h \end{array} \right\} = |\mathbf{X}_{d^*}^T \mathbf{X}_{d^*}| \quad (2.2)$$

donde D representa el conjunto de todas las posibles combinaciones; d es del tamaño del subconjunto que se desea seleccionar; \mathbf{X}_d es una matriz posible de diseño y \mathbf{X}_{d^*} representa el diseño D-Óptimo (Jeff-Wu et al., 2000). H es una función que restringe a una combinación \mathbf{x}_d (ésta es opcional). \mathbf{X}^T corresponde a la matriz transpuesta de \mathbf{X} .

Como se explicó anteriormente, las principales situaciones en las que es conveniente utilizar un diseño D-Óptimo en lugar de un diseño clásico son: 1) cuando un diseño estándar como el factorial completo requiere muchas corridas y, por tanto, involucra

invertir mayores recursos (por ejemplo materia prima) y/o más tiempo de experimentación del disponible, y 2) cuando el espacio de soluciones del diseño contiene restricciones, por lo que muchas corridas de un diseño clásico no son factibles o resultan imposibles de correr. En este trabajo se puede utilizar el diseño D-Óptimo en las dos modalidades; sin embargo en la experimentación realizada solo se utilizó cuando el diseño tenía corridas innecesarias.

2.3. Metamodelación

Un modelo empírico o metamodelo es una ecuación derivada de los datos muestrales que expresa la relación entre la respuesta (salida) y los factores (variables de decisión). Los modelos empíricos suelen usarse en dos casos: 1) para predecir la salida de un proceso y/o 2) para encontrar los niveles de las variables de decisión que dan como resultado los mejores valores de la respuesta, esto para la optimización de proceso. En este trabajo se utilizó el segundo enfoque. Algunos detalles de optimización fueron presentados en la primera sección de este capítulo. Se profundizará más en este tema en la Sección 2.4.

Un enfoque general para ajustar modelos empíricos son los modelos de regresión. Un modelo de regresión se ajusta a un conjunto de datos muestrales donde las variables independientes representan las variables de decisión, la variable dependiente corresponde a la respuesta y los coeficientes de regresión (valores $\hat{\beta}$ en la ecuación 2.3) están relacionados con la estimación de los efectos de los variables en la respuesta. La **Ecuación 2.3** muestra un modelo de regresión cuadrático completo para dos variables de decisión (x_1 y x_2).

$$\hat{f}(x_1, x_2) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1^2 + \hat{\beta}_4 x_2^2 + \hat{\beta}_5 x_1 x_2 \quad (2.3)$$

Para la estimación de los coeficientes de regresión generalmente se utiliza el método de mínimos cuadrados (del inglés *least squares*) (Montgomery, 2005). Si las variables de decisión son independientes, se pueden estimar los coeficientes de regresión de forma analítica, esto es, despejando los valores de los coeficientes de regresión de tal modo que la suma de errores cuadrados (SSE) se minimice. El estimador resultante de esta operación se muestra de forma matricial en la **Ecuación 2.4**, donde la matriz \mathbf{X} representa las variables de decisión (independientes), el vector \mathbf{y} las respuestas del experimento (o la salida del simulador) y $\hat{\boldsymbol{\beta}}$ es el vector que contiene los coeficientes de regresión (Montgomery, 2005). Sin embargo, si las variables no son independientes el cálculo anterior no se puede efectuar, por lo cual es recomendado utilizar una matriz pseudoinversa *Moore-Penrose* para estimar los valores de los coeficientes de regresión que más se ajusten al modelo. Información relacionada con este tipo de matriz puede encontrar en Pukelsheim (1993) y Bernstein (2005).

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (2.4)$$

Con el objetivo de tener un estimado de la proporción de la variabilidad de los datos explicada por el metamodelo, se puede calcular el valor del coeficiente de determinación (R^2 , **Ecuación 2.5**). Este valor debe ser mayor a cero y menor de uno (100%), siendo más deseable un valor mayor (Montgomery, 2005).

$$R^2 = 1 - \frac{SSE}{SST} \quad (2.5)$$

En referencia a la **Ecuación 2.5**, SSE (suma de errores cuadrados, **Ecuación 2.6**), corresponde a la suma de la diferencia entre la respuesta real (y_i) y la respuesta del metamodelo (\hat{y}_i), al cuadrado, para todo punto i ; SST es la suma de cuadrados totales (**Ecuación 2.7**); y el vector \mathbf{y} contiene todos los valores de la respuesta medida en las n combinaciones utilizadas para construir el metamodelo.

$$SSE = \sum_i (y_i - \hat{y}_i)^2 \quad (2.6)$$

$$SST = \mathbf{y}^T \mathbf{y} - \frac{\left(\sum_{i=1}^n y_i \right)^2}{n} \quad (2.7)$$

Además de tener una representación matemática de la superficie de respuesta, es de interés para el trabajo presente poder optimizarla. La siguiente sección contiene algunos de los conceptos principales de optimización utilizados en este trabajo.

2.4. Optimización

La optimización es un proceso con el cual se intenta producir la mejor solución posible a un problema dado (óptimo viene del latín, *optimus*, que significa mejor). La optimización caracteriza la manera en que nos enfrentamos a situaciones cotidianas conflictivas en las cuales deseamos la opción más apropiada o benéfica, usualmente bajo un escenario en el que los medios para alcanzarlos son limitados (Barrios-García et al., 2005).

En ocasiones, es posible formalizar matemáticamente los problemas de optimización, traduciendo las acciones en variables, los objetivos en funciones y los medios disponibles en restricciones. El análisis y resolución de tales problemas matemáticos (modelo de optimización) es el objetivo de la disciplina “Optimización Matemática”, que comprende un amplio conjunto de técnicas encaminadas a la obtención de los valores más adecuados para las variables de decisión que permitan satisfacer el objetivo deseado (Barrios-García et al, 2005).

Un modelo de optimización a minimizar en general es representado como la formulación (2.8).

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.a.} \quad & g(\mathbf{x}) \leq 0 \\ & \mathbf{x} \in X \subseteq \mathbb{R}^n \end{aligned} \tag{2.8}$$

En pocas palabras, un modelo de optimización trata de encontrar valores a las variables de decisión (\mathbf{x}) (entre el conjunto de todos los valores posibles (X)), que optimicen (minimicen o maximicen) una función objetivo ($f(\mathbf{x})$) y además que satisfagan las restricciones dadas ($g(\mathbf{x})$) (Winston, 2006).

2.4.1. Clasificación de modelos de optimización

Las técnicas de optimización matemática admiten distintas clasificaciones dependiendo de las características del problema dado. Estas características se refieren, entre otras, a los valores que pueden tomar las variables y a los tipos de funciones que aparecen en el modelo. Una manera general de clasificar un modelo de optimización es problema de optimización *lineal* o *no lineal*, donde en los primeros tanto la función objetivo como las restricciones son funciones lineales, mientras que en los segundos al menos una de estas funciones es no lineal. Adicionalmente, dependiendo de la naturaleza de los valores de las variables de decisión podemos decir que un problema es *entero* o *entero mixto*, el primero cuando las variables (todas) sólo pueden tomar valores enteros y para los segundos cuando tenemos variables que toman valores enteros y continuos. Cuando no se especifica si las variables son enteras, se toman como variables continuas.

La función objetivo de los modelos de optimización a su vez puede clasificarse como función *convexa* (*cóncava*) o *no convexa*. Una función es *convexa* (**Figura 2.5a**) en un intervalo si la recta tangente a cualquier punto en la función en ese intervalo está por debajo de la función. Similarmente, se dice que una función es *cóncava* (**Figura 2.5b**) en un intervalo si la recta tangente a cualquier punto en la función está por encima de

ésta. Se dice que una función es *no convexa* (**Figura 2.6**) en un intervalo si en subintervalos la función es cóncava y en otros convexa. Una función lineal se considera cóncava y convexa.

Los problemas de optimización *no lineal* predominan en un gran número de aplicaciones ingenieriles. Adicionalmente, estos problemas suelen ser no convexos (ver **Figura 2.6**), en cuyo caso no se puede garantizar que la solución final sea una solución óptima global (Salazar, 2005). Por otro lado, es común que se requiera que las variables de decisión sean enteras (ver **Figura 2.7**). Todas estas características hacen que el problema sea difícil de resolver.

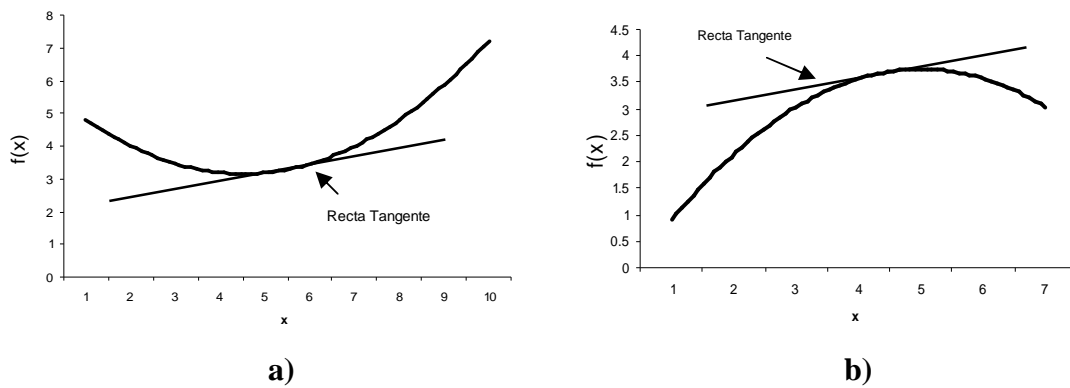


Figura 2.5: a) Función convexa y b) Función cóncava

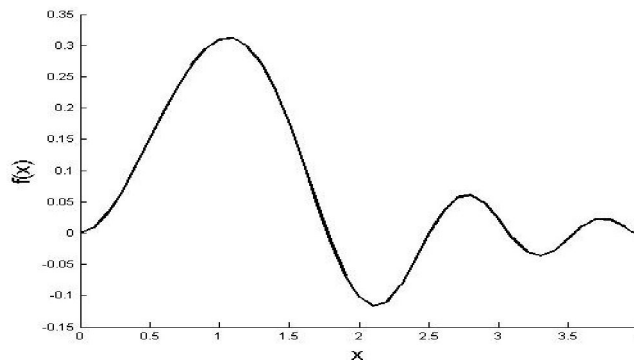


Figura 2.6: Función no convexa¹

¹ Gráficas tomadas de la referencia (Salazar, 2005).

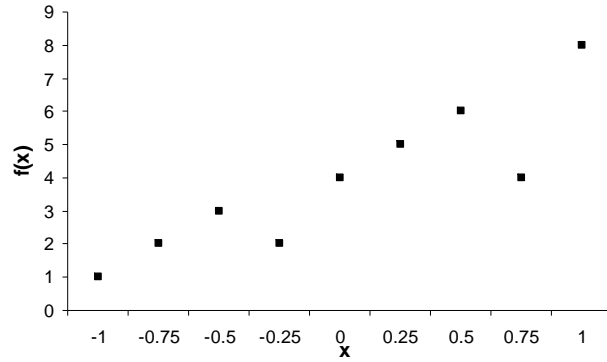


Figura 2.7: Función discreta

Los problemas no lineales no convexos presentan múltiples puntos estacionarios. La **Figura 2.8** muestra diferentes casos de puntos estacionarios (x_1 a x_6) que se pueden encontrar en una función $f(x)$, en este caso de una sola variable sobre el intervalo (a,b) . Se tienen **mínimos locales** (x_2 y x_4); **máximos locales** (x_1 , x_3 y x_6); x_6 corresponde a un **máximo global** ya que $f(x_6) = \max\{f(x_1), f(x_3), f(x_6)\}$; del mismo modo x_2 es un **mínimo global**. Adicionalmente, aunque x_1 es un punto máximo, difiere de los máximos locales restantes en que el valor de f correspondiente a por lo menos un punto en el entorno de x_1 es igual a $f(x_1)$, en este caso x_1 corresponde a un **máximo débil**. Por ejemplo, x_3 corresponde a un **máximo fuerte** y de manera similar x_4 corresponde a un **mínimo débil** y x_2 a un **mínimo fuerte**. Una observación importante acerca de los puntos x_1 a x_6 (ver **Figura 2.8**) es que la primera derivada de f (pendiente) se vuelve cero en estos puntos. Sin embargo, esta pendiente no es única de los puntos máximos o mínimos. Por ejemplo, la pendiente de $f(x)$ en x_5 , es cero. Este punto se conoce como punto de **inflexión** (o en casos especiales, **punto silla**) (Taha, 2005).

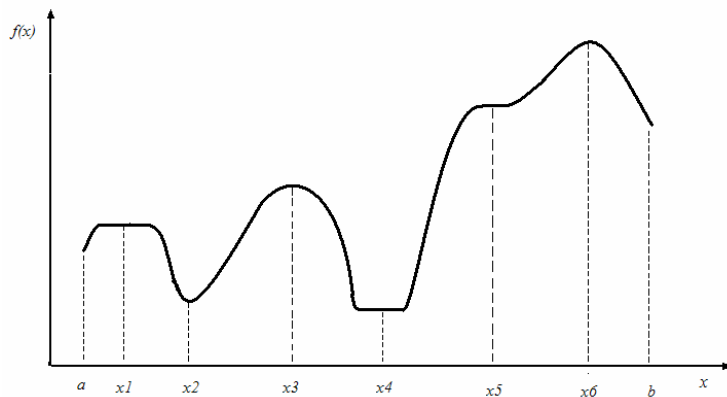


Figura 2.8: Función no lineal no convexa, con diferentes puntos estacionarios.

La mayoría de los métodos tradicionales de optimización no lineal están basados en gradientes (primera derivada). Por tanto, al desvanecerse las derivadas, lo más que se puede decir de una solución encontrada con estos métodos es que es un óptimo local; si los problemas son no convexos y/o no cuentan con propiedades estructurales, ni siquiera ésto se puede concluir. Solamente para el caso de los problemas convexos sabemos que la solución es la óptima global en presencia de un conjunto de soluciones factible convexo o en ausencia total de restricciones.

Para aumentar la probabilidad de encontrar una solución atractiva que se acerque a un óptimo global se puede utilizar un método heurístico denominado “múltiples comienzos”. Lo que este heurístico propone es que, cuando se utilice un método de optimización local, se ejecute varias veces empezando desde diferentes puntos cada vez. Con esta estrategia cada ocasión que el método inicia, converge al punto estacionario más cercano al punto inicial, que muchas veces será un máximo o un mínimo local. Finalmente se comparan estas soluciones y se reportar la mejor.

Dado que en buena parte de este trabajo se utilizó Microsoft Excel Solver, es necesario precisar cómo este optimizador resuelve los diferentes problemas de nuestro interés. Para problemas de optimización *no lineal* el Solver utiliza el algoritmo de Gradiente Reducido Generalizado (Generalized Reduced Gradient), que es un algoritmo de optimización local desarrollado por Leon Lasdon (University of Texas at Austin) y Allan Waren (Cleveland State University), (Fylstra et al., 1998; Soporte Técnico Microsoft, 2006). Para los problemas *no lineales* y *enteros* el Solver utiliza el método Ramificar y Acotar (Branch and Bound) encontrando las cotas por medio del método de Gradiente Reducido Generalizado. Estos algoritmos fueron implementados por John Watson y Dan Fylstra (Frontline Systems, Inc.) (Fylstra et al., 1998).

2.4.2 Criterios de parada

Cuando la forma de la función objetivo de un problema de optimización permite establecer la existencia de óptimos locales o globales, se puede hacer uso de algún algoritmo que explote este conocimiento para su análisis. La terminación de estos

algoritmos por lo general no representa mucho reto por tener disponible la caracterización de una solución óptima. En la gran mayoría de los casos, sin embargo, este conocimiento no está disponible y es, por tanto, difícil determinar cuándo un algoritmo de optimización debe terminar. En estos casos, o bien se define un criterio de terminación construido en base al problema específico o bien se recurre al uso de alguno(s) de los múltiples criterios de terminación ya disponibles en la literatura.

Uno de los criterios de terminación más común en la práctica es finalizar después de que un número preespecificado de evaluaciones se haya cumplido (Pohlheim, 2004). A diferencia de este criterio de parada, otros criterios tienen la ventaja de irse adaptando al estado de la corrida del optimizador, así un número considerable de corridas no será necesario evaluarlas. Desafortunadamente, es casi imposible definir un criterio de parada de este segundo tipo sin introducir uno o más parámetros que dependen del problema de optimización dado (Zielinski y Laur, 2007).

Dentro de los criterios más comunes se tiene: 1) *Tiempo computacional* (el método para cuando una cantidad de tiempo de CPU o un número máximo de evaluaciones -en este caso simulaciones- se ha cumplido), 2) *Una meta* (en este caso el algoritmo se detiene si el valor objetivo es mejor que algún valor meta preespecificado), 3) *Mínima mejora* (la diferencia absoluta entre el valor de la función objetivo encontrado en la última iteración y el mejor valor encontrado hasta el momento, es menor a un valor especificado – usualmente muy pequeño-), entre otros. Más información relacionada a criterios de parada pueden encontrarse en (Melin y Castillo, 2002; Zielinski et al., 2005).

En este trabajo se propone un método de optimización para simulaciones el cual se describe en el siguiente capítulo. Este método trabaja con un esquema similar al descrito por la **Figura 2.1**. El método utiliza técnicas tanto de diseño de experimentos como de metamodelación. Un metamodelo es obtenido cada iteración el cual, en combinación con una estrategia de optimización, se va adaptando para converger finalmente a una solución altamente atractiva.

Capítulo 3

MÉTODO PROPUESTO

En este trabajo se propone un método iterativo de optimización aplicada a la simulación basado en metamodelación adaptativa. El método comienza con un diseño de experimentos con el cual se determina una solución incumbente y un metamodelo inicial. En cada iteración, el metamodelo es utilizado para generar un punto atractivo en el cual se ejecuta una simulación. El valor simulado del nuevo punto se compara contra el incumbente para actualizarlo, y finalmente, se prueban varios criterios de parada. Si se decide seguir, el nuevo punto se añade al conjunto de puntos disponibles en una nueva iteración y un nuevo metamodelo es construido. La manera como hemos definido la inclusión de un nuevo punto e información disponible, la selección y construcción analítica de un metamodelo saturado, así mismo la selección del diseño experimental y los criterios de terminación forman un método con características originales y novedosas dentro de la literatura que, como se discute en los casos de estudios que se presentan más adelante, resulta en soluciones muy atractivas. Adicionalmente, el método es suficientemente sencillo y transparente para ser codificado con recursos computacionales modestos y utilizado con poco entrenamiento en optimización.

3.1. Descripción del método

El método propuesto incluye técnicas de diseño de experimentos, metamodelación, simulación y optimización, y se muestra esquemáticamente en la **Figura 3.1**. Éste consta de una fase de inicialización la cual se describe en los pasos 1 y 2, y de una iteración principal detallada desde el paso 3 al 9.

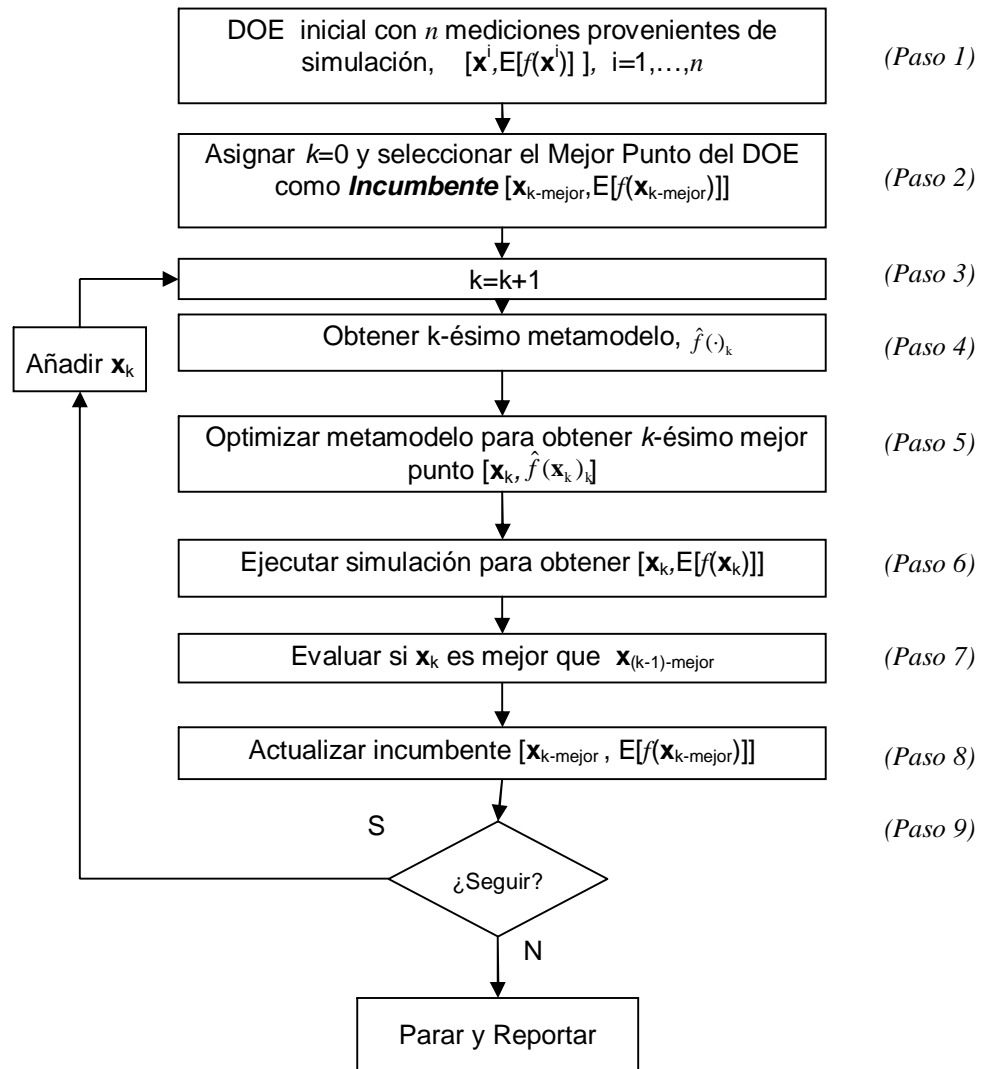


Figura 3.1: Método Propuesto

En referencia a la **Figura 3.1**, el método propuesto para problemas de optimización similares al modelo (2.1) formulado en el capítulo anterior es como sigue:

$$\begin{aligned}
 \min \quad & E[f(\mathbf{x})] \\
 \text{s.a.} \quad & A\mathbf{x}^T \leq b \\
 & g_l \leq G(\mathbf{x}) \leq g_u \\
 & l \leq \mathbf{x} \leq u
 \end{aligned} \tag{2.1}$$

INICIALIZACIÓN

1. DOE inicial

El método inicia con un Diseño de Experimentos (DOE por sus siglas en inglés). El DOE consistirá en n corridas representadas por combinaciones $\mathbf{x}^i = (x_1, x_2, x_3, \dots, x_v)^i$ de las v variables controlables a incluir en el problema de optimización, así como su evaluación $E[f(\mathbf{x}^i)]$, donde $i=1,2,\dots,n$. Si el DOE es replicado, entonces $E[f(\mathbf{x}^i)]$ será la media de las réplicas (Law y McComas, 2000). Dependiendo del número de variables de decisión es recomendable utilizar un DOE específico con el fin de no realizar corridas de simulación innecesarias. En este trabajo se realizó una clasificación de forma empírica para seleccionar este diseño, la cual se presenta en el Apéndice A.

El DOE inicial tiene como objetivo realizar un muestreo de forma estratégica en toda la superficie de respuesta.

2. Seleccionar solución incumbente

Tomando en cuenta el criterio detallado en la formulación (2.1), se selecciona la combinación del DOE con el mínimo valor de la función objetivo. Esta combinación se convierte en la k -ésima solución incumbente (inicialmente $k=0$) [$\mathbf{x}_{k\text{-mejor}}, E[f(\mathbf{x}_{k\text{-mejor}})]$]. También en este paso se inicializa un contador de iteraciones $k:=0$.

ITERACIÓN PRINCIPAL

3. Actualizar contador

En este paso se incrementa el valor de k en una unidad.

4. *Obtener el k-ésimo metamodelo*

Se utilizan los puntos obtenidos en el DOE inicial y los puntos encontrados en cada iteración del método para construir el k-ésimo metamodelo $\hat{f}(\cdot)_k$. El metamodelo elegido en este método es un modelo de regresión saturado, i.e. con $(n-1)+(k-1)$ coeficientes de regresión.

Adicionalmente, las variables de entrada deberán estar escaladas en el rango $[-1,1]$ para evitar problemas de dimensionalidad al obtener los coeficientes de regresión (April et al., 2004).

5. *Optimizar el k-ésimo metamodelo para obtener \mathbf{x}_k*

En este paso se desea obtener una nueva solución (\mathbf{x}_k) del modelo de simulación utilizando el metamodelo derivado del paso 4. Para esto se formula un problema de optimización donde el metamodelo es utilizado como función objetivo a ser minimizada (o maximizada) sujeto a las restricciones pertinentes. Este problema de optimización la mayoría de las veces es no lineal y las variables de decisión (x_i) son enteras, por lo que pierde factibilidad el poder garantizar un óptimo global al utilizar un optimizador local como el que se utilizó en la experimentación de prueba. Cabe señalar que en esta fase se puede utilizar cualquier optimizador. Información relevante a optimización se puede consultar el capítulo de antecedentes.

6. *Ejecutar una simulación en el punto nuevo*

En esta etapa se estima, vía simulación, el valor de $E[f(\mathbf{x}_k)]$ cuidando que si el modelo de simulación fue replicado, en esta estimación se tengan el mismo número de réplicas y se reporte la media.

7. *Evaluar si el punto nuevo es mejor que el incumbente*

En esta fase se quiere evaluar si el punto encontrado (\mathbf{x}_k) tiene un valor objetivo estrictamente menor al mejor punto encontrado hasta este momento ($\mathbf{x}_{(k-1)\text{-mejor}}$), esto es, si $E[f(\mathbf{x}_k)] < E[f(\mathbf{x}_{(k-1)\text{-mejor}})]$.

8. *Actualizar la solución incumbente*

En este punto se actualizará la solución incumbente de acuerdo a la evaluación del punto anterior. Si $E[f(\mathbf{x}_k)] < E[f(\mathbf{x}_{(k-1)\text{-mejor}})]$, entonces se asignará $[\mathbf{x}_{k\text{-mejor}}, E[f(\mathbf{x}_{k\text{-mejor}})]] := [\mathbf{x}_k, E[f(\mathbf{x}_k)]]$, de otra manera, la solución incumbente no cambiará.

9. *Evaluar los criterios de terminación del método*

Los criterios de terminación determinan cuándo el método tiene que finalizar. En este trabajo se utilizan tres criterios de terminación. El método deberá terminar si:

1. El nuevo punto (\mathbf{x}_k) forma parte de los puntos del DOE inicial o es igual a alguno de los puntos utilizados en previas iteraciones para construir el metamodelo. Este criterio es utilizado ya que un punto conocido no añade información para la construcción de un nuevo metamodelo.
2. El coeficiente de determinación (R^2 , **Ecuación 2.5**) del k-ésimo metamodelo es igual o mayor a un valor ϵ especificado.

La justificación para utilizar este criterio de terminación viene del hecho de que si se logra ajustar un metamodelo casi perfectamente a las observaciones, éste es ahora un buen representante de la superficie de respuesta del modelo de simulación que hereda su fidelidad al punto atractivo encontrado. A lo largo de esta tesis se utilizaron valores de 0.95 y 1.00 para ϵ .

3. El número máximo de iteraciones, definido por el usuario, se ha alcanzado.

La elección de este número dependerá de los recursos disponibles, como por ejemplo, tiempo y dinero, entre otros. Para las pruebas desarrolladas en este trabajo, se utilizó el número de combinaciones que se generan con un diseño de experimentos factorial completo.

Entonces, si alguno de los criterios de terminación se cumple, el método para y se toma la solución incumbente como la salida final. Si ninguno de los criterios se cumple, entonces se añade \mathbf{x}_k al conjunto de puntos disponibles (i.e. los puntos del diseño de experimentos inicial y los obtenidos mediante la optimización del metamodelo en cada iteración) para construir un nuevo metamodelo, y se regresa a la iteración principal.

El objetivo de ir construyendo un nuevo metamodelo en cada iteración es lograr la convergencia a una solución altamente atractiva en un número modesto de ejecuciones del modelo de simulación, esto último se mostrará de forma empírica con los resultados de prueba. Este método contrasta entonces con la manera tradicional de optimizar en base a metamodelos en donde se toma un número suficientemente grande de combinaciones (corridas de simulación) y se ajusta un único metamodelo para posteriormente optimizarlo. En este último caso, la idea es realizar una representación de toda la superficie de respuesta, mientras que el método propuesto realiza una buena representación sólo de manera local guiado por el criterio de optimalidad, previo a una inspección general de la superficie de respuesta. Este enfoque de aproximación local permite disminuir el número de ejecuciones del modelo de simulación, haciéndolo muy atractivo con respecto al tiempo de análisis y, por tanto, con respecto a los recursos computacionales.

Esencialmente, este método presenta variaciones sobre estrategias de metamodelación que ya han sido probadas, como la de Martin y Simpson (2002). Creemos, sin embargo, que la manera como hemos definido la inclusión de \mathbf{x}_k , la selección y construcción analítica de un metamodelo saturado, así como la selección del diseño experimental y la selección de los criterios de terminación forman una estrategia

con características propias cuyos resultados son muy prometedores. El siguiente capítulo muestra la evaluación que se realizó del método propuesto para llegar a la conclusión de que éste converge a una solución altamente atractiva con un número modesto de corridas de simulación. Adicionalmente se muestra la aplicación del método por medio de varios modelos de simulación sencillos. Esto último se hace con el fin de hacer visible la simplicidad del funcionamiento del método, pues es ésta otra de las cualidades que lo hacen muy atractivo para su instauración y utilización por parte de personas con conocimientos modestos en el área de optimización.

Capítulo 4

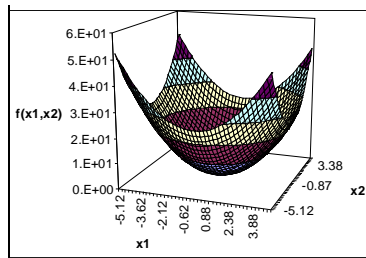
EVALUACIÓN DEL MÉTODO PROPUESTO

En este capítulo se presentan las diferentes evaluaciones realizadas del método propuesto, con el objetivo de probar empíricamente la efectividad del mismo. Los escenarios de evaluación utilizados fueron: 1) Funciones de prueba para optimización global 2) Diferentes redes neuronales artificiales y 3) Modelos de simulación sencillos.

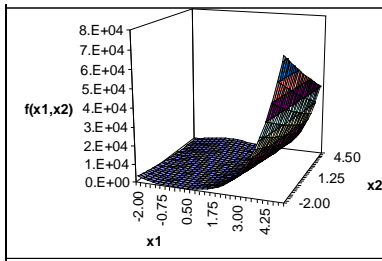
4.1 Funciones de prueba para optimización global

En esta sección se presentan los resultados de la evaluación de siete funciones de prueba conocidas para optimización global: a) Sphere, b) Rosenbrock, c) Rastringin, d) Griewank, e) Goldstein-Price, f) Easom, g) Schwefel. La **Figura 4.1** muestra gráficamente estas funciones para el caso de 2 variables. Las funciones a, b y f son claramente unimodales. Especialmente la función c exhibe múltiples mínimos locales; la función f tiene además la dificultad de que el valor de la función objetivo es constante en la mayor parte del espacio de búsqueda. Más información de estas funciones puede consultarse en

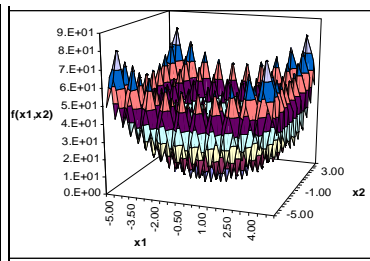
www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm y (Pohlheim, 2004). Los resultados que se presentan en esta sección incluyen inicialmente las siete funciones de prueba para optimización global en un espacio de 2 dimensiones. Posteriormente se incluye a las funciones Sphere y Griewank evaluadas en un espacio 9-dimensional. Finalmente se presentan los resultados de evaluar 3 funciones polinomiales no lineales (Figura 4.2) en un espacio de 2 dimensiones.



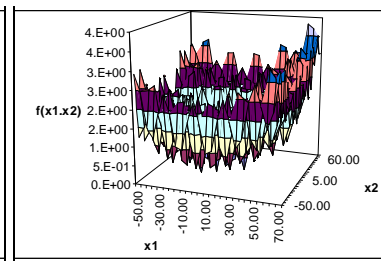
a) Sphere



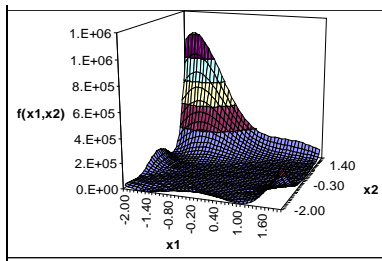
b) Rosenbrock



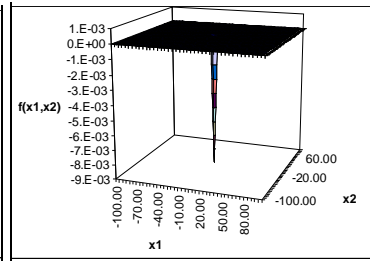
c) Rastrigin



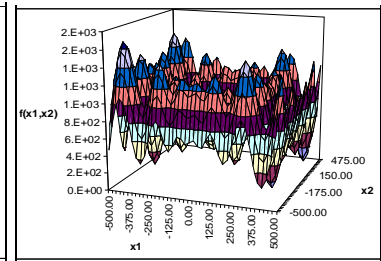
d) Griewank



e) Goldstein-Price

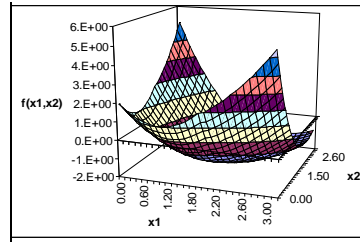


f) Easom

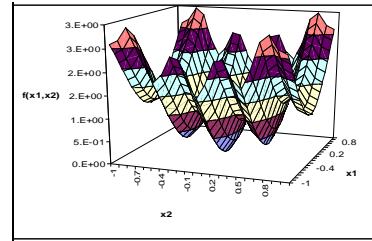
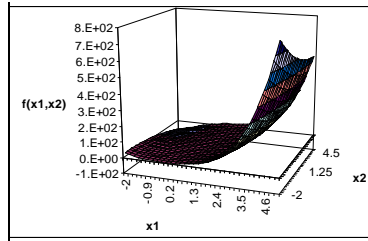


g) Schwefel

Figura 4.1: Funciones de prueba para optimización global irrestrictas en 2 dimensiones



1. $f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2 - x_1 x_2$



2. $f(x_1, x_2) = x_1^4 - 2x_1^2 x_2 + x_1^2 + x_1 x_2^2 - 2x_1 + 4$ 3. $f(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3 \cos(3x_1) - 0.4 \cos(4\pi x_2) + 0.7$

Figura 4.2: Funciones no lineales polinomiales en 2 dimensiones utilizadas para validar el método.

Para cada uno de los 12 casos se aplicó el método propuesto utilizando cada función como si ésta fuera un modelo de simulación, donde el valor de la función representa la salida del modelo de simulación y las variables independientes de la función representan las variables de decisión correspondientes. Las restricciones son cotas inferiores y superiores únicamente. En todos los casos el objetivo fue minimizar el valor de la función objetivo. El diseño de experimentos utilizado en cada caso se tomó en base al número de variables de decisión (ver Apéndice A), para el caso de 2 variables se utilizó un diseño compósito central (DCC) y para el caso de 9 variables se recurrió al diseño D-Óptimo. Como se mencionó en el Capítulo 2, al utilizar un DCC es necesario seleccionar el valor del parámetro α . En estas pruebas se realizaron dos casos con respecto de este parámetro con el objetivo de definir cuál diseño es mejor en términos de disminuir el número de evaluaciones del modelo de simulación hasta converger a la solución final. Se utilizaron valores de $\alpha=0.5$ y $\alpha=1$ en cada caso respectivamente como se mostró en las **Figuras 2.3a** y **2.3b**.

Las **Tablas 4.1 y 4.2** contienen los resultados de la evaluación de las siete funciones de optimización global en un espacio de 2 dimensiones, la primera utilizando un DCC con $\alpha = 1$ y la segunda tomando $\alpha = 0.5$. La **Tabla 4.3** muestra la evaluación de las funciones Sphere y Griewank en un espacio de 9 dimensiones. Las **Tablas 4.4 y 4.5** corresponden a los resultados de evaluar las tres funciones no lineales en dos dimensiones utilizando un DCC con $\alpha = 1$ y $\alpha = 0.5$ respectivamente. Las cinco tablas están organizadas de la siguiente manera: la primer columna de izquierda a derecha indica la función analizada; sigue después el número de ejecuciones de simulaciones (evaluaciones de la función objetivo) que se realizaron utilizando el método propuesto hasta que algún criterio de parada fue cumplido; posteriormente se encuentra el valor objetivo óptimo, el mejor valor encontrado y el orden del rango de los valores de la función objetivo en una zona acotada de búsqueda. Cabe señalar que la zona de búsqueda puede ser mayor a la zona del experimento inicial, esto principalmente debido a que la zona de búsqueda puede tener restricciones funcionales que un diseño clásico no considera.

Función	No. de corridas de simulación	Valor objetivo óptimo	Mejor valor objetivo encontrado	Orden del rango de valores de la función objetivo	Zona de búsqueda
Sphere	9	0	0.00	10^1	$x_q \in [-5.12, 5.12] \forall q$
Rosenbrock	13	0	0.00	10^4	$x_q \in [-2.00, 5.00] \forall q$
Rastrigin	13	0	0.00	10^1	$x_q \in [-5.00, 5.00] \forall q$
Griewank	9	0	0.00	10^0	$x_q \in [-50, 70] \forall q$
Goldstein-Price	17	3	5.58	10^6	$x_q \in [-2.00, 2.00] \forall q$
Easom	9	-1	0.00	10^0	$x_q \in [-100, 100] \forall q$
Schwefel	10	0	464.78	10^3	$x_q \in [-500, 500] \forall q$

Tabla 4.1: Solución a funciones de prueba, en un espacio de 2 dimensiones, utilizando un DCC con $\alpha = 1$.

Función	No. de corridas de simulación	Valor objetivo óptimo	Mejor valor objetivo encontrado	Orden del rango de valores de la función objetivo	Zona de búsqueda
Sphere	9	0	0	10^1	$x_q \in [-5.12, 5.12] \forall q$
Rosenbrock	13	0	0	10^4	$x_q \in [-2.00, 5.00] \forall q$
Rastrigin	9	0	0	10^1	$x_q \in [-5.00, 5.00] \forall q$
Griewank	9	0	0	10^0	$x_q \in [-50, 70] \forall q$
Goldstein-Price	15	3	3	10^6	$x_q \in [-2.00, 2.00] \forall q$
Easom	9	-1	0	10^0	$x_q \in [-100, 100] \forall q$
Schwefel	9	0	464.78	10^3	$x_q \in [-500, 500] \forall q$

Tabla 4.2: Solución a funciones de prueba, en un espacio de 2 dimensiones, utilizando un DCC con $\alpha = 0.5$.

Función	No. de corridas de simulación	Valor objetivo óptimo	Mejor valor objetivo encontrado	Orden del rango de valores de la función objetivo	Zona de búsqueda
Sphere	74	0	0	10^2	$x_q \in [-5.12, 5.12] \forall q$
Griewank	74	0	0.16	10^1	$x_q \in [-50, 70] \forall q$

Tabla 4.3: Solución a funciones de prueba, en un espacio de 9 dimensiones, utilizando un diseño D-Óptimo con 74 combinaciones.

Función	No. de corridas de simulación	Valor objetivo óptimo	Mejor valor objetivo encontrado	Orden del rango de valores de la función objetivo	Zona de búsqueda
No lineal 1	10	-2	-2	10^0	$x_q \in [0, 3] \forall q$
No lineal 2	12	-62	-62	10^4	$x_q \in [-2, 5] \forall q$
No lineal 3	9	0	0	10^0	$x_q \in [-1, 1] \forall q$

Tabla 4.4: Solución a funciones no lineales en un espacio de 2 dimensiones, utilizando un DCC con $\alpha = 1$.

Función	No. de corridas de simulación	Valor objetivo óptimo	Mejor valor objetivo encontrado	Orden del rango de valores de la función objetivo	Zona de búsqueda
No lineal 1	9	-2	-2	10^0	$x_q \in [0, 3] \forall q$
No lineal 2	9	-62	-62	10^4	$x_q \in [-2, 5] \forall q$
No lineal 3	9	0	0	10^0	$x_q \in [-1, 1] \forall q$

Tabla 4.5: Solución a funciones no lineales en un espacio de 2 dimensiones, utilizando un DCC con $\alpha = 0.5$.

Observando los resultados mostrados en las **Tablas 4.1 y 4.2** podemos notar que para esta serie de casos, al utilizar un DCC con valor $\alpha=0.5$, se reduce el número de corridas de simulación realizadas hasta que el método paró. Adicionalmente, con un valor $\alpha=0.5$ en cinco de los siete casos se encontró la solución óptima mientras que con $\alpha=1$ sólo en cuatro ocasiones se encontró esta solución; en dos ocasiones se encontró un valor muy cercano a ésta y solamente en una ocasión se quedó muy lejos, i.e. con la función de Schwefel. En la **Tabla 4.3** podemos ver que, con las funciones 9-dimensionales, en el primer caso se encontró la solución óptima y en el segundo una solución muy cercana a la óptima. Para el caso de las funciones no lineales el método encontró la solución óptima en las tres ocasiones, sin embargo, cuando se utilizó un DCC con $\alpha=0.5$ se redujo el número de evaluaciones de la función objetivo como se puede corroborar en las **Tablas 4.4 y 4.5**.

Consideramos que el desempeño del método es bastante competitivo, especialmente dada la complejidad de las funciones. Creemos también que en las situaciones reales las funciones objetivo distarán de tener comportamientos tan irregulares como los de estas funciones de prueba. Además de que el método encontró soluciones muy atractivas, lo hizo en un número muy pequeño de corridas de simulación.

De estos mismos resultados podemos observar que para los casos donde se utilizó un diseño de experimentos compósito central con $\alpha=0.5$ (i.e. se incluyen los puntos esquinas de la superficie de repuesta (ver **Figuras 2.3a y 2.4a**)), el método converge más rápido a la solución final. Bajo esta observación al utilizar un DCC inicial con $\alpha=1$, el mejor punto seleccionado mediante los metamodelos en las iteraciones iniciales es un punto que se encuentra en alguna de las esquinas de la región factible. El método no selecciona todas, pero tiende a seleccionar la mayoría de éstas en las primeras iteraciones. Entonces, si en el diseño de experimentos inicial se pueden incluir las esquinas (ver **Figuras 2.3a y 2.4a**), el método ya no necesitará inspeccionarlas, ahorrándose estas evaluaciones. La tendencia de métodos similares a explorar las esquinas de una zona factible había sido documentada por Martin y Simpson (2002), sin

una explicación posible para tal comportamiento. Aunque aquí proveemos un poco más sobre este fenómeno en términos de la selección del diseño de experimentos, un estudio dirigido específicamente a explicar este fenómeno se deja como extensión a futuro.

El **Apéndice B** contiene los resultados de cada iteración para todos los casos presentados en esta sección. La siguiente sección contiene más evaluaciones del método propuesto.

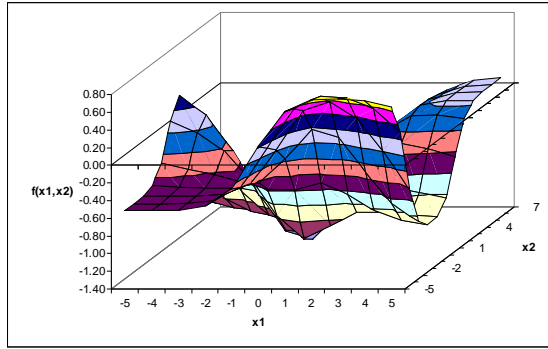
4.2 Redes Neuronales Artificiales (RNAs)

Algunas arquitecturas específicas de RNAs son conocidas por ser aproximadores universales. Por esta razón se consideró conveniente utilizar RNAs para generar más funciones no lineales y no polinomiales de prueba. En específico, se utilizaron RNAs multicapa y de avance con 2 y 3 neuronas en la capa de entrada (número de variables de decisión (v)), 3 neuronas en la capa oculta (m) y una neurona en la capa de salida (predicción de la función objetivo). Información relacionada a RNAs puede consultarse en (Hagan et. al, 1995). La **Ecuación 4.1** muestra matricialmente lo que correspondería a una de estas funciones, en particular a la gráfica (a) de la **Figura 4.3**. Los parámetros de las funciones restantes se muestran en el **Apéndice C**. La **Figura 4.3** muestra gráficamente las primeras 5 funciones contenidas en la **Tabla 4.6** (funciones con dos variables de decisión).

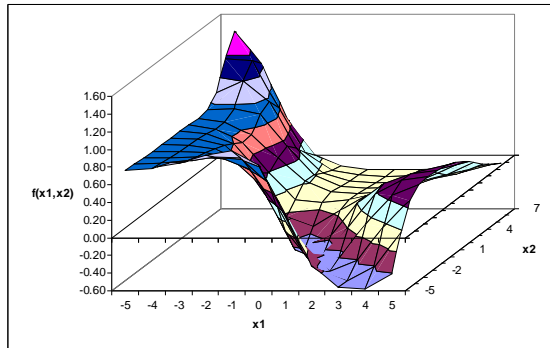
$$f(x_1, x_2) = f_2 \left(\begin{pmatrix} 0.90 \\ 0.54 \\ 0.65 \end{pmatrix}^T * f_1 \left(\begin{pmatrix} -0.51 & -0.19 & 0.85 \\ -0.86 & 0.86 & -0.14 \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} -0.39 \\ -0.72 \\ -0.56 \end{pmatrix} \right) + (-0.23) \right) \quad (4.1)$$

Donde

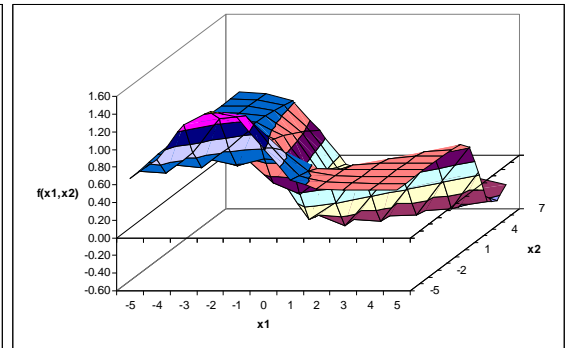
$$f_1(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} \qquad f_2(s) = s$$



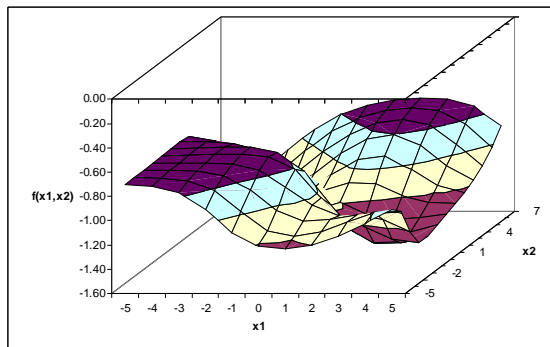
a) RNA(2,3,1)



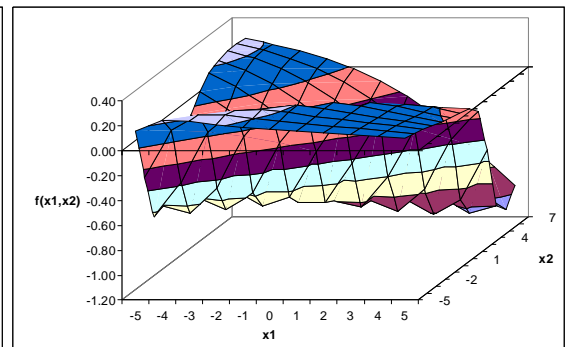
b) RNA(2,3,1)



c) RNA(2,3,1)



d) RNA(2,3,1)



e) RNA(2,3,1)

Figura 4.3: RNAs con dos neuronas en la capa de entrada utilizadas para validar el método de optimización.

La **Tabla 4.6** muestra los resultados al aplicar el método propuesto a todas estas funciones. El diseño de experimentos inicial fue un DCC con parámetro DCC $\alpha=1$. La **Tabla 4.6** está organizada de la siguiente manera: la primera columna lista el tipo de RNA utilizada, donde v y m indican el número de neuronas en la capa de entrada y oculta respectivamente, teniendo todas una sola salida; la segunda columna contiene el

número de corridas de simulación realizadas hasta que algún criterio de parada se cumplió; la tercer columna muestra la mejor solución conocida (para el caso de de 2 variables se compararon 121 soluciones y en el caso de 3 variables 1331 soluciones); la cuarta columna contiene la mejor solución encontrada a través del método propuesto. Los valores de cada una de las variables de decisión se generaron en un intervalo de -5 a 5 para x_1 y de -5 a 7 para x_2 , en los caso de funciones con dos variables; para los casos de tres variables todas fueron generadas en un rango de -5 a -5.

RNA($v,m,1$)	No. de corridas de simulación	Mejor solución conocida (meta)	Mejor valor objetivo encontrado
a) RNA(2,3,1)	16	-1.30	-1.23
b) RNA(2,3,1)	13	-0.58	-0.41
c) RNA(2,3,1)	13	-0.43	-0.34
d) RNA(2,3,1)	13	-1.47	-1.44
e) RNA(2,3,1)	14	-1.09	-0.99
f) RNA(3,3,1)	23	-2.63	-2.52
g) RNA(3,3,1)	24	-2.18	-2.16

Tabla 4.6: Solución a diferentes RNAs utilizadas para prueba.

Para el caso de las RNAs utilizadas como funciones de prueba, el método propuesto encontró soluciones muy cercanas a la mejor solución conocida en un número muy pequeño de corridas de simulación. Junto con los resultados mostrados en la sección anterior, el método acusa un buen desempeño para optimizar funciones no lineales tanto polinomiales como no polinomiales. Al ser estos tipos de funciones comunes en la práctica, estos resultados apuntan a que la solución final del método propuesto será altamente atractiva y que la encontrará en un número pequeño de corridas de simulación.

En el **Apéndice C** se muestran de forma detallada las funciones que representan cada una de las RNAs utilizadas en esta sección, así como los resultados presentados en la **Tabla 4.6**. La siguiente sección muestra la aplicación del método a tres ejemplos de simulaciones sencillas: dos de ellos corresponden a simulaciones de sistemas que además son instancias de simulación de eventos discretos, y el otro a un proceso de moldeo por inyección, que es una simulación de fenómenos continuos.

4.3 Ejemplos de simulación sencillos

En esta sección se muestra la aplicación de la metodología de solución propuesta para problemas de optimización de simulaciones. Para realizar la ejemplificación se consideraron dos modelos de demostración incluidos en el software de simulación SIMUL8 (simulador de eventos discretos): el primero corresponde a una sala de hospital y el segundo a una línea de embotellado; adicionalmente se utilizó un modelo de demostración incluido en el software de simulación MoldFolwTM (simulador de fenómenos continuos), el cual simula la inyección a presión de un polímero dentro de un molde, en este caso para producir la carátula de un teléfono celular. Los siguientes apartados muestran detalladamente cada una de las aplicaciones.

4.3.1. Sala de Hospital

Como se mencionó anteriormente el modelo de simulación a presentarse en esta sección fue tomado de las simulaciones muestra incluidas en el software de simulación SIMUL8 (edición SIMUL8 2006 Professional, versión 13). La **Figura 4.4** muestra una foto de la pantalla de este modelo, el cual consiste en una sala de hospital a la que los pacientes llegan de acuerdo a una distribución *Exponencial* ($\lambda=5$ min.) y esperan en una fila con capacidad finita (si la fila esta llena y llega un paciente, éste se irá, no esperará). Los pacientes esperarán en la fila hasta ser atendidos por una recepcionista quien tarda en atender a una persona un tiempo proveniente de una distribución *Log Normal* ($\mu=1.5$ min. y $\sigma=2.5$ min.). Una vez atendidos por la recepcionista los pacientes esperarán nuevamente, ahora en una sala de espera con capacidad finita donde, si ésta se llega a llenar, bloqueará a la recepcionista. Los pacientes esperarán hasta que alguno de los dos doctores disponibles esté libre. El tiempo que le toma a un doctor atender a un paciente es modelado mediante una distribución *Normal* ($\mu=10$ min. y $\sigma=2.5$ min.). Al terminar su consulta, los pacientes tendrán una probabilidad de 80% de salir del hospital y una probabilidad de 20% de ir a enfermería. Si llegase a ir a enfermería, el paciente necesitará esperar nuevamente en una fila (infinita) hasta ser revisado por una enfermera, el tiempo que le toma atender a un paciente es modelado mediante una

distribución *Normal* ($\mu=10$ min. y $\sigma=2.5$ min.). Posterior a su atención los pacientes dejarán el hospital.

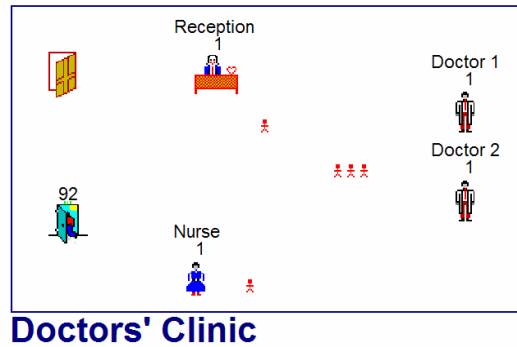


Figura 4.4: Modelo que simula una sala de hospital (modelo de demostración de SIMUL8).

En este sistema se consideraron dos variables de decisión: 1) la capacidad máxima de la fila justo antes de ser atendidos por la recepcionista (x_1) con un rango de 5 a 9 pacientes, y 2) la capacidad máxima de la sala de espera (x_2) en un rango de 10 a 14 pacientes. El objetivo fue maximizar la cantidad de pacientes esperados a atender en un período de 5 días ($E[f(\mathbf{x})]$).

Dado que varios datos del modelo de simulación son generados por distribuciones de probabilidad, un número de réplicas para un mismo conjunto de parámetros es necesaria (Kelton et al., 2002). La definición del número de réplicas de un modelo de simulación depende de la variabilidad del mismo. Pruebas de hipótesis tales como *diferencia de medias* o la prueba no-paramétrica *Kruskal-Wallis*, nos indican si las medias o las medianas, respectivamente, de una población son todas iguales o si al menos una es diferente. Con las anteriores podemos definir si existe o no una diferencia estadísticamente significativa entre la salida de una sola corrida del modelo de simulación y el promedio de las salidas de varias corridas independientes para un mismo conjunto de parámetros. Si no existe una diferencia entonces no es necesario realizar réplicas del modelo. En este caso para definir el número de réplicas se realizó la prueba *Kruskal-Wallis*, donde se prueba H_0 : todas las medianas de la población son iguales vs.

H_1 : al menos una mediana es diferente. H_0 se rechaza si el nivel de significancia (α) es mayor al valor p. Esta prueba fue realizada con apoyo del software estadístico Minitab. La **Tabla 4.7** muestra los resultados de la prueba, donde la primera columna indica el número de réplicas del modelo de simulación realizadas, la segunda contiene la mediana de cada caso, la tercera columna contiene una jerarquía para cada caso (donde el número más pequeño corresponde al caso con menor mediana). El valor H corresponde al estadístico de prueba donde si $H > \chi^2_{\alpha, g.l.}$, se rechaza la hipótesis nula, así mismo, también se puede utilizar el enfoque del valor p descrito previamente. Información de esta prueba puede consultarse en (Montgomery, 2005).

No. de Réplicas	Mediana	Jerarquía Promedio
1	454.0	1
5	458.0	6
10	456.3	4
15	456.7	5
20	455.1	2
25	455.1	3
H = 5.00 g.l. = 5 valor-p = 0.416		

Tabla 4.7: Prueba Kruskal-Wallis sobre el promedio de las salidas de las replicas

Considerando un nivel de significancia $\alpha = 0.05$, no podemos rechazar la hipótesis nula, dado que el valor p es de 0.416. Entonces podemos concluir que no existe una diferencia estadísticamente significativa entre la mediana de las salidas de una réplica o el promedio de 5, 10, 15, 20, 25 réplicas. Por lo tanto, una sola réplica será suficiente para representar adecuadamente la salida del modelo de simulación. En este ejemplo en particular, sin embargo, se utilizaron tres réplicas para realizar una comparación justa con el optimizador OptQuest (incluido en SIMUL8) que utiliza este número de réplicas por defecto.

Siguiendo el método propuesto para resolver problemas de optimización de simulaciones, se realizó lo siguiente:

INICIALIZACIÓN

Paso 1:

Iniciar con un diseño de experimentos, en este caso un Diseño Compósito Central ($\alpha=1$) (**Figura 4.5**) y calcular $E[f(\mathbf{x}^i)]$ ($i=1, \dots, 9$) evaluando el modelo de simulación en los nueve puntos experimentales y reportando la cantidad de pacientes atendidos en 5 días. Cada uno de los puntos experimentales fue replicado 3 veces (de forma independiente) y se obtuvo el promedio de cada conjunto de réplicas (**Tabla 4.8**).

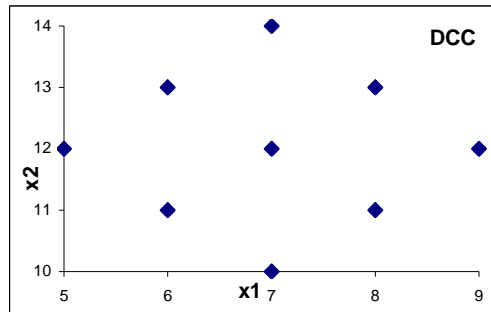


Figura 4.5: Diseño de Experimentos Compósito Central (\mathbf{x}^i , $i=1,2,\dots,9$)

\mathbf{x}^i	x_1 (pacientes)	x_2 (pacientes)	$E[f(x_1, x_2)]$ (pacientes)
\mathbf{x}^1	5	12	461.33
\mathbf{x}^2	6	11	461.33
\mathbf{x}^3	6	13	462.66
\mathbf{x}^4	7	10	461.33
\mathbf{x}^5	7	12	462.00
\mathbf{x}^6	7	14	463.66
\mathbf{x}^7	8	11	462.00
\mathbf{x}^8	8	13	463.66
\mathbf{x}^9	9	12	463.66

Tabla 4.8: $E[f(\mathbf{x}^i)]$, $i=1, \dots, 9$

Paso 2:

Definir la *solución incumbente*. En este paso se selecciona la solución con el máximo valor objetivo. En este caso las combinaciones (7,14), (8,13), y (9,12) exhiben el mismo valor en la función objetivo, por lo tanto cualquiera de ellas puede ser considerada como la solución incumbente. Adicionalmente se inicializa el contador de iteraciones $k:=0$.

PRIMERA ITERACIÓN

Paso 3:

En este paso se actualiza el contador de iteraciones $k = k+1 = 1$

Paso 4:

Ahora con los puntos del diseño de experimentos se obtiene un metamodelo ($\hat{f}(\cdot)_1$). Como se describió en el capítulo anterior se construye una regresión lineal múltiple utilizando el máximo número de grados de libertad, en éste caso 8 grados de libertad. Pevio a la construcción del metamodelo los datos fueron escalados en un rango de [-1,1]. Así mismo se calcula el valor del coeficiente de determinación (R^2) del metamodelo. A continuación se muestra la ecuación del metamodelo obtenido, el valor de R^2 y la gráfica del mismo (**Figura. 4.6**).

$$\hat{f}(x_1, x_2)_1 = -0.30 + 0.62x_1 + 1.38x_2 + 0.33x_1^2 + 0.33x_2^2 + 0.28x_1x_2 + 0.38x_1^3 - 0.38x_2^3$$

$$R^2 = 0.9944$$

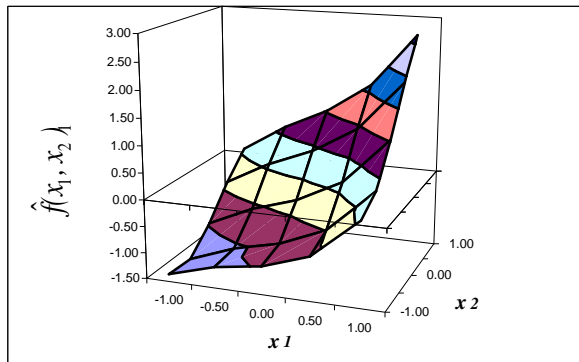


Figura. 4.6: Primer metamodelo

Paso 5:

Después de la obtención del metamodelo se construye un modelo de optimización, donde el metamodelo representa la función objetivo a ser maximizada, sujeta a las limitaciones de x_1 y x_2 . La optimización se realiza con el objetivo de encontrar una solución atractiva al sistema. En este caso se utilizó la herramienta de optimización incluida en Microsoft Excel (*Solver*) combinada con la técnica de múltiples comienzos para este fin. La mejor solución encontrada fue $\hat{f}(1,1)_1 = 2.65$ correspondiente a la combinación **(9,14)**.

Paso 6:

Correr el modelo de simulación ajustando los parámetros con el punto obtenido por el metamodelo (9,14) y calcular $E[f(9,14)]$ (replicando tres veces y reportando el promedio). Ésto da como resultado $E[f(9,14)] = 463.66$.

Paso 7:

En este paso se evalúa si el punto encontrado por el metamodelo es mejor que el punto incumbente, esto es:

$$E[f(9,14)] > E[f(7,14)]$$

$$463.66 > 463.66 \quad (\text{NO})$$

Paso 8:

Dado que el *paso 7* resultó falso, la solución incumbente no se actualiza, por lo tanto:

$$[x_{1\text{-mejor}}, E[f(x_{1\text{-mejor}})]] := [(7,14), 463.66]$$

Paso 9:

En seguida se evalúan los criterios de terminación del método (**Tabla 4.9**), con los cuales determinaremos si parar o continuar con una nueva iteración. En este paso es necesario introducir dos parámetros, el valor de ϵ , el cual sirve de cota inferior a R^2 , y el máximo número de evaluaciones del modelo de simulación aceptadas por el usuario; en este caso el valor de ϵ fue fijado en 1 dado que el valor de R^2 del metamodelo inicial es muy alto, y el número de evaluaciones de simulación fue fijado en 25 ya que es el número máximo de combinaciones en una enumeración exhaustiva.

Criterio	Valor	¿Para?
La combinación (9,14) ya había sido simulada.	(9,14) no había sido simulado	NO
El valor de $R^2 \geq \epsilon$ ($\epsilon=1$)	$R^2=0.9944$	NO
El número máximo de simulaciones se ha alcanzado (25 simulaciones).	10	NO

Tabla 4.9: Evaluación de criterios de terminación.

Dado que ninguno de los criterios de terminación se cumplió, entonces el punto obtenido mediante el modelo de optimización ((9,14)) junto con los puntos del diseño de experimentos es incluido en una nueva iteración.

SEGUNDA ITERACIÓN

Paso 3:

Actualizar el contador de iteraciones (k=2)

Paso 4:

Ahora utilizando todos los puntos disponibles se construye un nuevo metamodelo ($\hat{f}(\cdot)_2$) y se reporta el valor de R^2 , resultando en:

$$\hat{f}(x_1, x_2)_2 = -0.30 + 0.62x_1 + 1.93x_2 + 0.33x_1^2 + 0.33x_2^2 + 0.28x_1x_2 + 0.38x_1^3 - 0.93x_2^3 - 1.65x_1^2x_2$$

$$R^2=0.9952$$

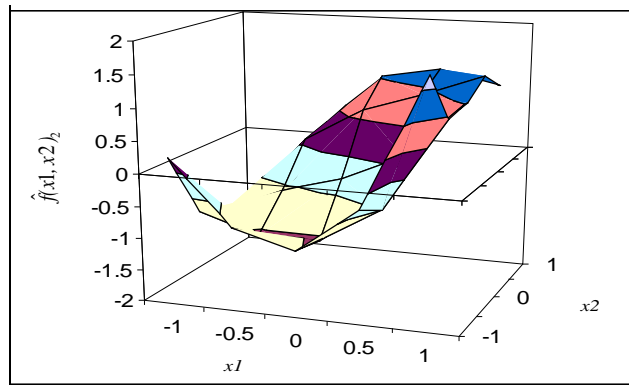


Figura 4.7: Segundo metamodelo

Paso 5:

Al optimizar el metamodelo como se hizo en la primera iteración, la mejor solución fue $\hat{f}(1,-1)_2 = 1.72$, correspondiente a la combinación **(9,10)**.

Paso 6:

Ejecutar el modelo de simulación en el punto (9,10) tres veces y calcular el promedio de las salidas. La solución fue $E[f(9,10)] = 462.00$.

Paso 7:

Evaluar si la nueva solución es mejor que la incumbente:

$$E[f(9,10)] > E[f(7,14)]$$

$$462.00 > 463.66 \quad (\text{NO})$$

Paso 8:

Actualizar la solución incumbente. En este caso como la nueva solución no es mejor que la incumbente, ésta no cambia, por lo tanto:

$$[x_{2\text{-mejor}}, E[f(x_{2\text{-mejor}})]] := [(7,14), 463.66]$$

Paso 9:

Evaluar los criterios de terminación, la **Tabla 4.10** muestra los resultados.

Criterio	Valor	¿Para?
La combinación (9,10) ya había sido simulada.	(9,10) no había sido simulado	NO
El valor de $R^2 \geq \epsilon$ ($\epsilon=1$)	$R^2=0.9952$	NO
El número máximo de simulaciones se ha alcanzado (25 simulaciones).	11	NO

Tabla 4.10: Evaluación de criterios de parada

Como ninguno de los criterios de parada se cumplió, entonces añadimos el punto (9,10) con los puntos disponibles e iniciamos una nueva iteración.

Dado que para que alguno de los criterios de parada se cumpla se requieren varias iteraciones, para efectos de esta ilustración sólo se mostrará la evaluación final donde el método paró (**Tabla 4.11**), esto ocurrió dos iteraciones adelante.

Criterio	Valor	¿Para?
La combinación (8,14) ya había sido simulada.	(8,14) no había sido simulado	SI
El valor de $R^2=1$	$R^2=0.9914$	NO
El número máximo de simulaciones se ha alcanzado (25 simulaciones).	13	NO

Tabla 4.11: Última evaluación de los criterios de parada (iteración 4)

Ahora el método para y se reporta la mejor solución, en este caso particular las mejores soluciones encontradas por el método propuesto fueron (7,14), (8,13), (8,14), (9,12), (9,14), con idénticos valores objetivo. Para este sistema, tomando en cuenta la primera solución de la lista se recomienda ajustar la capacidad máxima de la fila antes de la recepcionista a 7 personas y la capacidad máxima de la sala de espera a 14 personas, esperando atender 463 pacientes en un período de 5 días. En este caso particular se puede utilizar cualquiera de las soluciones encontradas esperando la misma cantidad de pacientes a atender. La **Figura 4.8** muestra cómo el método fue cambiando de una solución a otra iterativamente. Con respecto de esta gráfica, los puntos representados por un triángulo son las combinaciones del diseño de experimentos inicial, y los puntos en círculo son las soluciones de los metamodelos obtenidos en cada iteración; las flechas van indicando el orden como se fueron generando, iniciando en la esquina (9,14) y finalizando en (8,14).

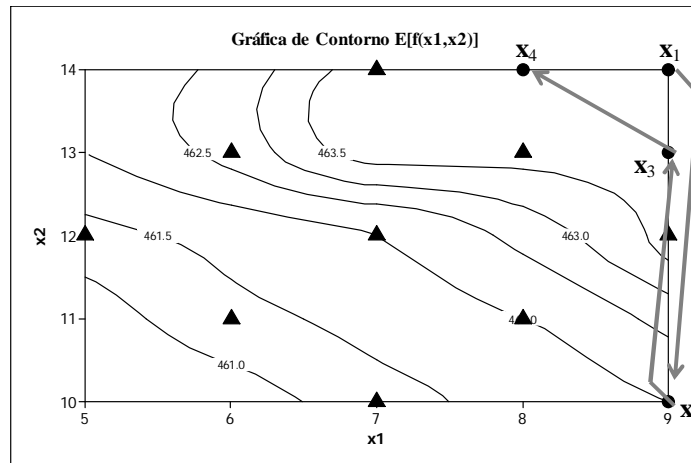


Figura 4.8: Soluciones encontradas por el método propuesto

Adicionalmente se realizó un análisis para este mismo sistema cambiando el valor de α en el diseño de experimentos inicial, en este caso con $\alpha = 0.50$. La Tabla 4.12 muestra los resultados de cada iteración. Esta tabla está organizada de la siguiente manera: la primera columna indica la iteración; la segunda el mejor punto encontrado al optimizar el metamodelo de tal iteración; la tercera la evaluación del punto encontrado en el modelo de simulación (considerando el promedio de 3 réplicas); la cuarta la solución

incubente hasta esa iteración y, finalmente, la quinta columna contiene el valor de R^2 del metamodelo. La **Figura 4.9** muestra el movimiento de las soluciones encontradas por iteración, iniciando en el punto (8,14) y finalizando en (8,13).

Iteración	Mejor solución del metamodelo (x_1, x_2) (pacientes, pacientes)	Evaluación en la Simulación $E[f(x_1, x_2)]$ (pacientes)	Solución Incubente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
Inicialización DOE	(7,13) (9,14)	463.66 463.66	[(7,13), 463.66]	--
1	(8,14)	463.66	[(7,13), 463.66]	0.9774
2	(8,13)	463.66	[(7,13), 463.66]	0.9822
3	(9,13)	463.66	[(7,13), 463.66]	0.9849
4	(7,10)	461.66	[(7,13), 463.66]	0.9888
5	(8,13)	463.66	[(7,13), 463.66]	0.9895

Tabla 4.12: Resultados de cada iteración para el modelo que simula una sala de hospital, iniciando con un DCC ($\alpha = 0.50$).

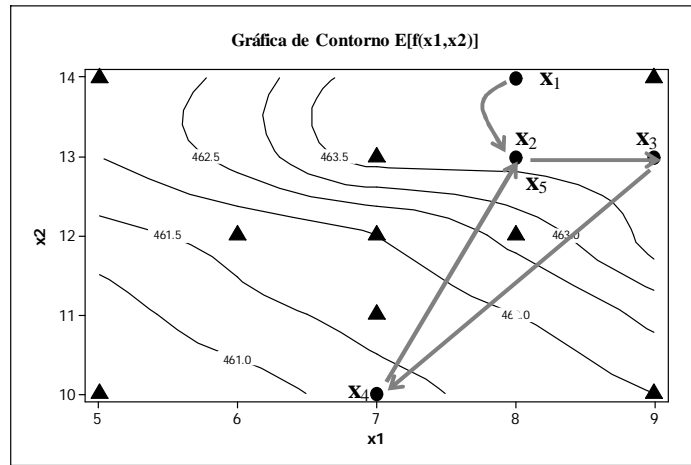


Figura 4.9: Puntos del diseño de experimentos inicial (triángulos) y soluciones encontradas en cada iteración por el método propuesto (círculos).

Posteriormente se realizó una evaluación exhaustiva de todas las posibles soluciones del sistema para comparar con el método propuesto. La **Tabla 4.13** y **Figura 4.10** muestran estas soluciones, donde los puntos marcados en negritas corresponden a las mejores soluciones.

x_1 (capacidad máxima de la fila de la recepcionista)	x_2 (capacidad máxima de la sala de espera)	$E[f(x_1, x_2)]$ (pacientes a atender en 5 días)	x_1 (capacidad máxima de la fila de la recepcionista)	x_2 (capacidad máxima de la sala de espera)	$E[f(x_1, x_2)]$ (pacientes a atender en 5 días)
5	10	461.00	7	13	463.67
5	11	460.67	7	14	463.67
5	12	461.33	8	10	461.67
5	13	462.00	8	11	462.00
5	14	462.00	8	12	462.67
6	10	460.67	8	13	463.67
6	11	461.33	8	14	463.67
6	12	461.67	9	10	462.00
6	13	462.67	9	11	462.67
6	14	462.67	9	12	463.67
7	10	461.33	9	13	463.67
7	11	461.67	9	14	463.67
7	12	462.00			

Tabla 4.13: Número esperado de pacientes a atender en un período de 5 días, variando la capacidad máxima de dos filas de espera.

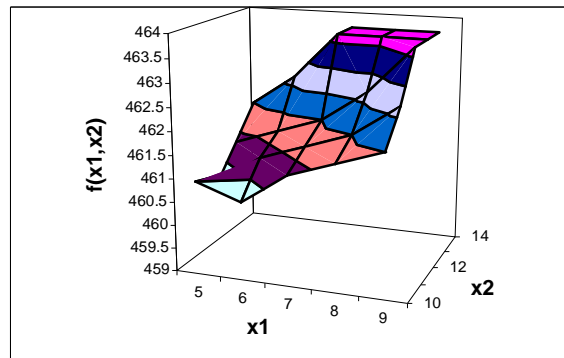


Figura 4.10: Número esperado de pacientes a atender en un periodo de 5 días.

Comparando los resultados obtenidos por el método propuesto y los presentados en la tabla anterior, podemos observar que la mejor solución al sistema es la misma a la encontrada por el método propuesto. El método, en este caso, fue capaz de encontrar el óptimo global.

Adicionalmente se utilizó el optimizador OptQuest para encontrar una solución a este sistema. Información relacionada con este optimizador se puede encontrar en el capítulo de antecedentes. Este optimizador requiere que el usuario ligue el modelo de

simulación con esta herramienta. Posteriormente se especifican las restricciones, la función objetivo y un criterio de parada basado en tiempo (minutos) o número de simulaciones (mínimo 10). Una vez ajustado el modelo, se corrió OptQuest y la mejor solución encontrada fue (7,14) donde $E[f(7,14)] = 463.66$.

Para este ejemplo simple, el método propuesto encontró una solución al problema del hospital en 13 corridas de simulación, que además resultó ser la mejor; así mismo OptQuest encontró una solución con el mismo valor objetivo, sin embargo éste se detiene en 25 corridas -pues son todas las posibles combinaciones que se pueden realizar con la variables de entrada- y selecciona la mejor encontrada. El método propuesto, en cambio, se detiene automáticamente en la décimo tercera simulación.

4.3.2. Línea de embotellado

La segunda aplicación corresponde a una línea de embotellado. Este modelo, al igual que el anterior, fue tomado de los modelos de simulación muestra incluidos en SIMUL8 (edición SIMUL8 2006 Professional, versión 13). La **Figura 4.11** muestra la pantalla de este modelo. El modelo trabaja como sigue: botellas de tres tipos provenientes de una línea de lavado y preprocesado entran al sistema, con una tasa fija de una cada 0.25 min., así mismo hay una probabilidad de 1/3 de ser tipo 1 o tipo 2 o tipo 3; al entrar al sistema las botellas son incorporadas a un banda transportadora (conveyor) de longitud 3.5 pies y velocidad 2.5 pies/min., al salir del conveyor un trabajador separa las botellas de acuerdo a su tipo y las manda a la sublínea correspondiente (3 sublíneas), hacer ésto le toma 0.25 min. Cada sublínea trabaja como sigue: las botellas esperarán en una fila hasta que la máquina de llenado esté desocupada. Esta máquina requiere 0.5 min. para llenar una botella. De ahí son incorporadas a un nuevo conveyor (de longitud 4 pies y velocidad 1 pie/min.). Posteriormente otra máquina pondrá una etiqueta correspondiente a la botella, esta operación toma 0.5 min., de ahí las botellas serán incorporadas a otro conveyor con las mismas características que el anterior. Al salir del conveyor un trabajador tomará una botella de cada tipo y formará un grupo, lo cual le toma 0.25 min. Se envían las botellas agrupadas por un conveyor (de longitud 5 pies y velocidad 0.80

pies/min.) para ser puestas en una caja, donde el trabajador que las empaqueta toma 0.5 min. Enseguida la caja es incorporada a otro conveyor (longitud 3 pies y velocidad 1 pie/min.), de ahí otra máquina etiquetará la caja tomándole 0.5 min. en realizar esta operación. Finalmente las cajas serán pasadas a un conveyor de longitud 3 pies y velocidad 1 pie/min. el cual las transporta a la salida. En este caso únicamente el tipo de las botellas que entran al sistema es una variable estocástica, tanto la velocidad de las máquinas como el tiempo que requiere un trabajador en realizar su operación son variables determinísticas.

Alguna información adicional del sistema es el diámetro de la base de la botella (0.3 pies), la longitud de la base de la caja (0.7 pies), así como que la línea trabaja 8 horas continuas por día (no hay probabilidad de falla). Además, si algún conveyor llegara a estar lleno, éste bloqueará la máquina previa a él.

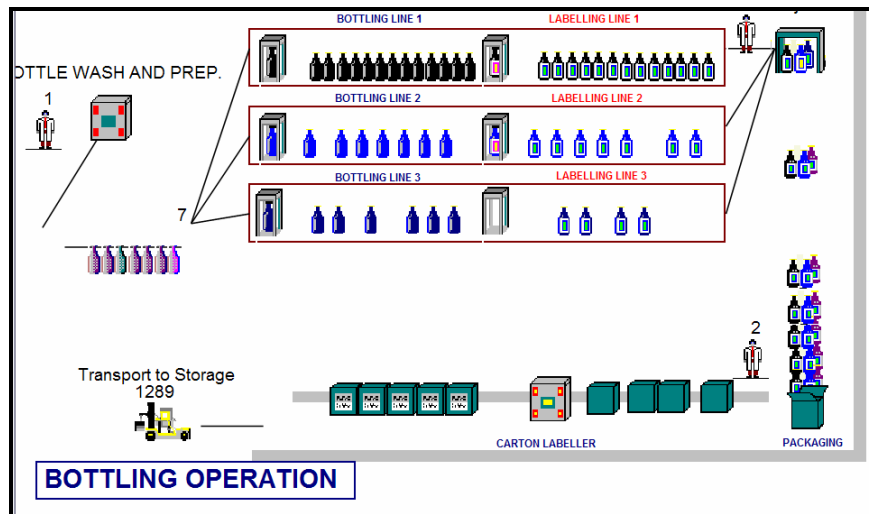


Figura 4.11: Modelo de simulación de una línea de embotellado (modelo tomado de los ejemplos muestra de SIMUL8)

El objetivo en este sistema fue maximizar la cantidad de cajas producidas en 999 min. (un poco más de 2 días de trabajo). Las variables controlables fueron la velocidad de nueve conveyors (3 de embotellado, 3 de etiquetado, el de empaquetado, el de etiquetado de la caja y el de la salida). Tanto la velocidad de las máquinas como la de los

trabajadores quedaron fijas al valor como trabaja en sistema actual (descrito en le inicio de esta sección). El rango en el cual se decidió variar las velocidades fue de [0.50, 1.50] pies/min.

Identificado el objetivo y las variables de decisión se aplicó la metodología propuesta. En la iteración inicial se construyó un diseño de experimentos compósito central con 51 combinaciones (18 puntos estrella, 1 punto centro y 32 puntos de un diseño factorial fraccionado ($2^{9.4}$) resolución IV, i.e. las interacciones de dos factores son alias entre sí) y se seleccionó la solución incumbente. Adicionalmente los criterios de parada fueron ajustados en $R^2 \geq 95\%$ y el máximo número de iteraciones a 512 (combinaciones de un diseño factorial 2^9). La **Tabla 4.14** muestra los resultados más importantes de cada iteración, como lo son el mejor punto encontrado del metamodelo, su valor en el modelo de simulación, la solución incumbente por iteración y el valor R^2 de cada metamodelo.

Iteración	Mejor solución del metamodelo (x_1, x_2, \dots, x_9) (pies/min)	Evaluación en la Simulación $E[f(x_1, x_2, \dots, x_9)]$ (cajas producidas en 999min)	Solución Incumbente $[(x_1, x_2, \dots, x_9), E[f(x_1, x_2, \dots, x_9)]]$	R^2 del metamodelo
Inicialización DOE	(1.00,1.00,1.00,1.00,1.0 0,1.00,1.00,1.00,1.50)	1295	[(1.00,1.00,1.00,1.00,1.00,1 .00,1.00,1.00,1.50), 1295]	--
1	(1.50,1.50,1.50,1.50,1.5 0,1.50,1.13,1.10,1.13)	1262	[(1.00,1.00,1.00,1.00,1.00,1 .00,1.00,1.00,1.50), 1295]	0.9470
2	(1.50,1.50,1.50,1.50,1.5 0,0.50,1.50,1.10,1.13)	1064	[(1.00,1.00,1.00,1.00,1.00,1 .00,1.00,1.00,1.50), 1295]	0.9480
3	(0.50,0.50,1.50,1.50,0.5 0,0.50,1.50,1.10,1.11)	1222	[(1.00,1.00,1.00,1.00,1.00,1 .00,1.00,1.00,1.50), 1295]	0.9480
4	(1.50,1.50,1.50,1.50,1.5 0,0.50,0.50,1.50,1.13)	1064	[(1.00,1.00,1.00,1.00,1.00,1 .00,1.00,1.00,1.50), 1295]	0.9490
5	(0.50,1.50,0.50,1.50,0.5 0,0.50,1.50,1.50,0.50)	963	[(1.00,1.00,1.00,1.00,1.00,1 .00,1.00,1.00,1.50), 1295]	0.9490
6	(0.50,1.50,1.50,0.50,0.5 0,1.50,1.50,1.50,0.50)	962	[(1.00,1.00,1.00,1.00,1.00,1 .00,1.00,1.00,1.50), 1295]	0.9500

Tabla 4.14: Resultados de cada iteración para la simulación de la línea de embotellado. Los resultados se encuentran en su escala natural.

En la tabla anterior podemos observar que en la iteración 6 uno de los criterios de parada se cumplió ($R^2 \geq 95\%$), por lo tanto el método paró y la mejor configuración al sistema resultó en ajustar la velocidad de los conveyors de la siguiente manera:

- Conveyor línea 1 de embotellado = 1 pie / min.
- Conveyor línea 2 de embotellado = 1 pie / min.
- Conveyor línea 3 de embotellado = 1 pie / min.
- Conveyor línea 1 de etiquetado = 1 pie / min.
- Conveyor línea 2 de etiquetado = 1 pie / min.
- Conveyor línea 3 de etiquetado = 1 pie / min.
- Conveyor salida de ensamble = 1 pie / min.
- Conveyor salida de empackado = 1 pie / min.
- Conveyor salida de etiquetado de la caja = 1.5 pies / min.

Con los ajustes anteriores la cantidad esperada de cajas producidas en dos días es de **1295** cajas, comparado con **1258** cajas producidas con el sistema actual.

Adicionalmente se utilizó el optimizador OptQuest para encontrar una solución al sistema. Después de ajustar los parámetros requeridos por el software (restricciones, función objetivo y criterio de terminación (512 iteraciones)), la mejor solución encontrada por el optimizador fue (1.50, 1.50, 1.50, 1.50, 1.50, 1.50, 1.50, 1.50, 1.50) con valor objetivo igual a **1285** cajas. Comparando la solución encontrada por el método aquí propuesto, podemos ver que nuestro método encontró una solución ligeramente mayor a la solución de OptQuest, sin embargo, nuestro método lo hizo en solo 57 evaluaciones del modelo de simulación comparado con 512 realizadas por OptQuest. Lo anterior nos da certeza de que el método propuesto genera soluciones atractivas en un número considerablemente pequeño. Así mismo, se capitaliza en el hecho de que los paquetes de optimización comercial llevan detrás muchas pruebas que demuestran su buen funcionamiento, por lo que arrojar soluciones tan buenas o mejores en menor tiempo, es un punto a favor de nuestro método.

El **Apéndice D** contiene los puntos del diseño de experimentos inicial, y el valor de la evaluación de estos puntos en el modelo de simulación, para este caso.

4.3.3. Moldeo por Inyección de la Carátula de un teléfono celular

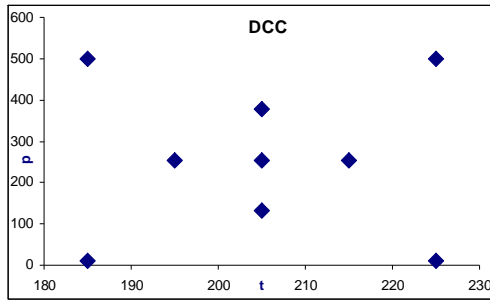
El caso que se muestra a continuación involucra una pieza de demostración incluida en el software de simulación MoldFlow™ (Moldflow Part Adviser 7.2) el cual tiene la capacidad de simular el flujo de un polímero en un molde en un proceso de moldeo por inyección; el polímero aquí utilizado fue Petrothene GA584-000: Mellenium Petrochemical Inc y el molde está diseñado para reproducir la carátula de un teléfono celular (**Figura 4.12**). Vale la pena subrayar que, a diferencia de los ejemplos anteriores, esta simulación involucra fenómenos continuos y no eventos discretos, por lo que sirve para ilustrar la versatilidad del método.



Figura 4.12: Carátula del teléfono celular.

El objetivo en este caso es minimizar el tiempo de ciclo de inyección (TC) (respuesta), donde las variables controlables (entradas) son la temperatura del molde (t) variado en un rango de 185°C a 225°C y la presión de inyección en la máquina (p) con un rango de 10MPa a 500MPa (éstos son rangos sugeridos por la base de datos de materiales de Moldflow). Las demás condiciones del proceso tales como la temperatura del molde quedan fijas en el valor recomendado por defecto.

Con el fin de optimizar este proceso se utilizó la metodología de optimización de simulaciones aquí propuesta. La **Figura 4.13** y la **Tabla 4.15** muestran el diseño de experimentos inicial (diseño compósito central con $\alpha = 0.5$).



t (°C)	p (MPa)	$E[TC(t,p)]$ (seg.)
185	10	4.06
185	500	3.87
195	255	3.76
205	132.5	4.02
205	255	4.02
205	377.5	4.02
215	255	4.16
225	10	4.20
225	500	4.20

Figura 4.13: x^i , $i = 1, 2, \dots, 9$ obtenidos con el Diseño de Experimentos Compósito Central

Tabla 4.15: $[x^i, E[TC(x^i)]]$ $i = 1, 2, \dots, 9$

Cabe señalar que para este ejemplo no es necesario realizar réplicas pues al no contener la simulación variables estocásticas, se obtendría la misma respuesta todas las veces. La **Tabla 4.16** contiene los resultados de todas las iteraciones. Ésta está organizada de manera similar a las tablas de resultados anteriores. Adicionalmente la **Figura 4.14** muestra los puntos del diseño de experimentos inicial (triángulos) y los puntos encontrados cada iteración mediante el metamodelo (iniciando en (190.38, 269.36) y terminando en (194.37, 10.00)).

Iteración	Mejor solución del metamodelo (t,p) (°C, MPa)	Evaluación en la Simulación $E[TC(t,p)]$ (segundos)	Solución Incumbente $[(t,p), E[TC(t,p)]]$	R^2 del metamodelo
Inicialización DOE	(195.00, 255.00)	3.76	[(195, 255), 3.76]	--
1	(190.38, 269.36)	3.76	[(195, 255), 3.76]	0.9920
2	(196.94, 500.00)	4.01	[(195, 255), 3.76]	0.9942
3	(191.04, 362.02)	3.76	[(195, 255), 3.76]	0.9942
4	(190.21, 341.35)	3.76	[(195, 255), 3.76]	0.9932
5	(225.00, 170.29)	4.41	[(195, 255), 3.76]	0.9991
6	(194.37, 10.00)	3.89	[(195, 255), 3.76]	1.0000

Tabla 4.16: Resultados de cada iteración del método de optimización para la simulación del flujo de un polímero dentro del molde de una carátula celular.

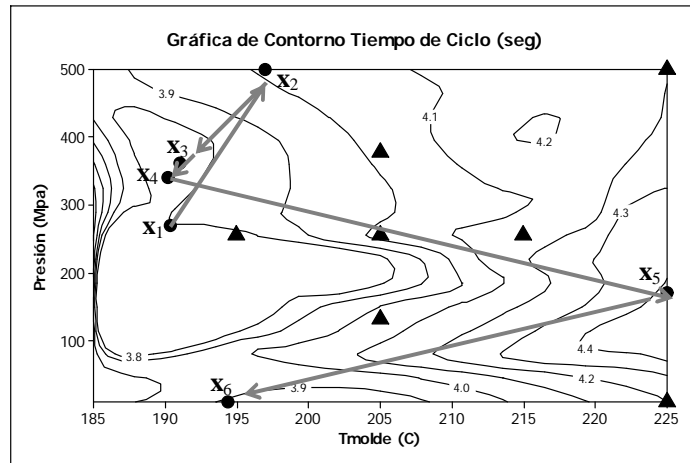


Figura 4.14: Gráfica de contorno para Tiempo de Ciclo en función de temperatura del molde y presión de inyección en máquina. Los puntos en triángulo representan los puntos del DOE inicial y los círculos son las combinaciones encontradas mediante los metamodelos.

Como se puede observar en la **Tabla 4.16** el metamodelo construido en la sexta iteración muestra un valor de $R^2 = 1$, por lo cual el método para ahí. Ahora el método para y se reporta la mejor solución. En este caso particular se encontraron cuatro puntos con el mismo valor en la función objetivo: (195.00, 255.00), (190.38, 269.36), (191.04, 362.02), (190.21, 341.35). El tiempo esperado que se requerirá para llenar el molde ajustando la temperatura del molde a 195°C y la presión de inyección a 255 MPa (o cualquiera de las otras tres configuraciones) será 3.76 segundos de acuerdo con la simulación.

Con el objetivo de tener una idea de qué tan buena fue la predicción del método de optimización en este caso, se realizó una comparación con los resultados presentados por Urbano-Vázquez y Cabrera-Ríos (2006) para el mismo modelo de simulación. En dicho trabajo se reporta como mejor valor objetivo (mínimo tiempo de ciclo) 3.76 segundos, para llegar a este resultado se realizó un diseño de experimentos factorial completo con 81 combinaciones (evaluaciones de simulación). Cabe destacar que aunque los dos trabajos reportan la misma solución, el método propuesto en este trabajo encuentra esta solución después de 15 corridas de simulación vs. 81 evaluaciones realizadas por

Urbano-Vázquez y Cabrera-Ríos. Además, considerando que cada corrida del modelo de simulación toma aproximadamente 20 minutos, el método propuesto tuvo un ahorro estimado de 1,320 minutos en tiempo computacional total. Lo anterior utilizando una computadora de escritorio con sistema operativo Windows XP Professional (Versión 2002), procesador Intel Pentium 4 de 2.8 GHz y 1 GB de memoria RAM.

4.4. Conclusiones

En este capítulo se presentaron las diferentes evaluaciones realizadas al método propuesto para resolver problemas de optimización de simulaciones. La evaluación se llevó a cabo con funciones de prueba para optimización global, funciones no lineales polinomiales y no polinomiales. En 17 de las 19 aplicaciones realizadas el método encontró la solución óptima o una muy cercana a ésta, solamente en una ocasión el método reportó una solución lejana a la óptima global. Adicionalmente se mostró cómo el método puede ser utilizado tanto en simulaciones discretas como continuas mediante la aplicación a tres modelos de simulación sencillos.

Con las evaluaciones realizadas, se mostró empíricamente que el método encuentra soluciones altamente atractivas en un número modesto de simulaciones. Aunado a que el método es transparente y se presta para una codificación muy conveniente. Todos estos casos se realizaron utilizando MS Excel (2002).

El siguiente capítulo muestra la aplicación del método de optimización a un caso real correspondiente a una compañía mexicana dedicada a la fabricación de partes de acero automotrices. Así mismo el Capítulo 6 muestra la aplicación del método a un modelo de simulación correspondiente a un proceso de moldeo por inyección.

Capítulo 5

CASO DE ESTUDIO 1

SIMULACIÓN Y OPTIMIZACIÓN DE UNA LÍNEA DE PINTADO DE CHASIS

El caso descrito en este capítulo sirvió para detectar la necesidad de tener un algoritmo de optimización aplicado a la simulación, que es la propuesta principal de esta tesis. Se presenta aquí la caracterización y el modelo de simulación correspondiente a una de las líneas de pintura de una compañía de manufactura de partes automotrices de acero de la ciudad de Monterrey. Se presenta también la optimización del modelo de simulación, con el objetivo de definir la mejor configuración de la línea de pintura para incrementar la productividad. La optimización se realizó utilizando el método propuesto en el Capítulo 3 y validado en el Capítulo 4. La compañía aprobó la precisión del

modelo de simulación, por lo cual el desarrollo de este trabajo ha ayudado a evaluar cambios en el proceso sin necesidad de realizar modificaciones físicas al sistema.

Adicionalmente, el caso aquí presentado es la validación a escala real del método propuesto en simulaciones de eventos discretos. La aplicación a simulación de fenómenos continuos se abordará en el siguiente capítulo.

5.1 Descripción del sistema

La primera fase en la construcción de un modelo de simulación consiste en definir el sistema de interés, determinar las interacciones con otros sistemas, las restricciones y las variables que interactúan dentro del sistema, sus interrelaciones y las medidas de eficiencia que se van a utilizar para definir y estudiar el sistema.

A continuación se presenta la descripción del sistema en estudio, una línea de pintado de chasis. La **Figura 5.1** muestra el diagrama de flujo de la línea y posterior a ésta se muestra la descripción de sus componentes.

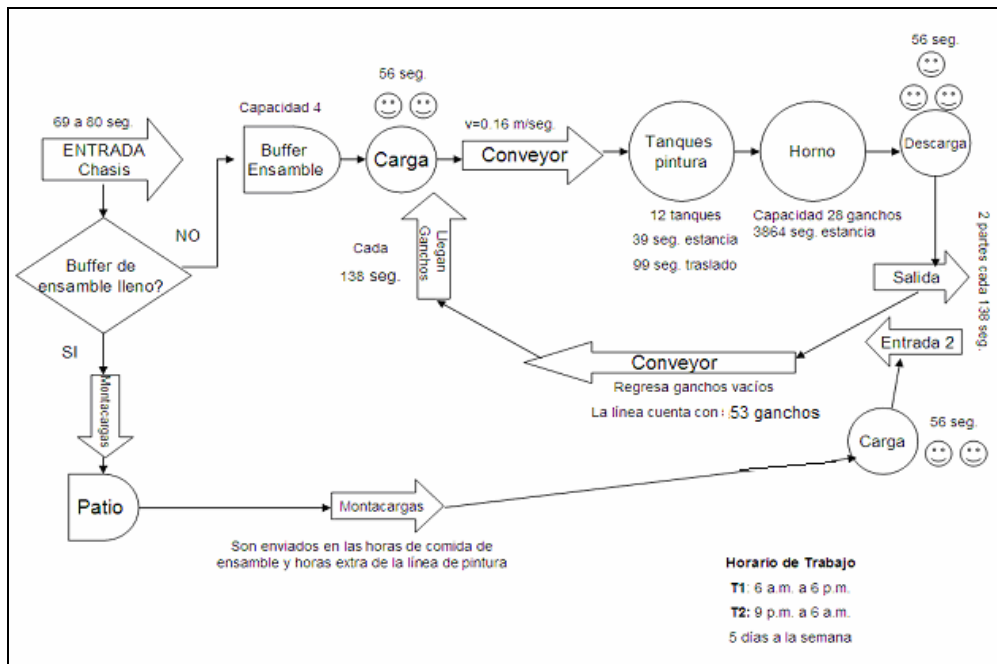


Figura 5.1: Diagrama de flujo de la línea de pintura a simular.

La línea a modelar tiene la siguiente dinámica entre sus elementos:

1. Dos entradas alimentan al sistema:
 - a) La entrada principal está enlazada directamente con las salidas de la línea de ensamble. La línea de ensamble alimenta partes con una frecuencia que oscila entre 69 y 80 segundos por unidad. Dado que la empresa solo pudo proporcionar esta información, se decidió modelar el tiempo entre llegadas de acuerdo a una distribución uniforme con valor mínimo 69 y máximo 80, ésto es recomendable cuando no se cuenta con más información que los rangos (Kelton, 2002).

Ésta operación se realiza de 6 a.m. a 6 p.m. y de 10 p.m. a 6 a.m., descontando 2 medias horas de comida (una cada turno).
 - b) Una entrada secundaria introduce partes que se encuentran en el patio. Esto sucede solamente en las horas de comida de la línea de ensamble y en una hora extra de la línea de pintura (de 9 p.m. a 10 p.m.).
2. Provenientes de la entrada principal, las partes que entran en el sistema esperan en un buffer que tiene una capacidad máxima de 4 unidades. Cuando éste está lleno, las partes son enviadas a un almacén situado en el patio.
3. Posteriormente las partes son cargadas (en pares) al conveyor principal, el cual los transporta en ganchos desde la entrada hasta los tanques de pintura. El proceso de cargar las partes requiere de dos personas que tardan aproximadamente 56 segundos en realizar esta operación (las dos unidades). Los trabajadores cargarán los chasis si al momento que llega un gancho al área de carga el buffer tiene al menos dos piezas, de otro modo el gancho se irá vacío. La línea cuenta con 53 ganchos actualmente. Estos ganchos están instalados en el conveyor principal el cual tiene una longitud total de 131 metros y se mueve a una velocidad de 0.16 m/seg. La distancia desde el área de carga a los tanques de pintura es de 93 metros.
4. Al llegar al área de pintado es necesario que cada par de partes sea sumergido en 12 tanques con diferentes tratamientos que incluyen limpiadores, pinturas y selladores. En cada uno de los tanques permanecerán 39 seg., con un tiempo de traslado entre

tanques de 99 segundos. Los tanques 2, 5 y 9 tienen capacidad para 2 ganchos por lo cual en estos tanques permanecerán 177 segundos.

5. Al salir de los tanques de pintura, los ganchos son transportados mediante un conveyor secundario hasta el horno. Esta operación tarda 138 segundos.
6. Al entrar al horno, requieren permanecer un tiempo aproximado de 3864 segundos para realizar el curado de las partes. El horno tiene una capacidad de 28 ganchos.
7. Al salir del horno, los ganchos son incorporados nuevamente al conveyor principal el cual lo transporta hasta la descarga, esto ocurre cada 138 seg..
8. La descarga se realiza al salir del horno, para esto se requiere de 3 personas a las cuales les toma 55 segundos en realizar la descarga.
9. Al momento de ser descargadas las partes son estibadas en grupos de 8 y transportadas mediante montacargas al área de embarque.

Otro aspecto importante de la línea de pintura es el horario de trabajo. El proceso se desarrolla en dos turnos: el primer turno de 6 a.m. a 6 p.m. y el segundo de 9 p.m. a 6 a.m. De 5 p.m. a 5:45 p.m. no se alimentan (entran) partes a la línea con el objetivo de que al finalizar el turno no se encuentren ganchos llenos en ninguno de los tanques de pintura. Si se llegase a quedar alguna pieza en esa sección por más de 2 horas, ésta ya no servirá dado que empezara a oxidarse. Cabe también señalar que el cliente de la línea no acepta retrabajos y el desperdicio de una parte como ésta llega a costar cientos de dólares. Por otro lado, si al finalizar un turno se encuentran partes dentro de los tanques de pintura, se corren los tiempos de ciclo necesarios para que las partes alcancen a llegar por lo menos hasta el horno.

La línea de pintura cuenta con 8 acumuladores distribuidos a lo largo del conveyor principal, su función es parar los ganchos en diferentes puntos estratégicos. Tres de ellos corresponden a un punto de carga o descarga, los cinco restantes permiten que los ganchos estén lo suficientemente separados para que no se produzca ningún choque entre ellos. Cabe mencionar que el conveyor principal siempre está en movimiento salvo

que ocurra un paro. Los acumuladores están distribuidos de la siguiente manera: uno en la entrada principal (carga), dos en el transcurso para llegar a los tanques de pintura, uno justo antes de entrar a los tanques de pintura, uno en la descarga, el siguiente en la entrada secundaria (carga), y dos más en el trayecto a la entrada principal.

Adicionalmente se sabe que la línea presenta una falla (paro) en promedio cada 2202 segundos y el tiempo de reparación promedio es de 409 segundos. Los paros más frecuentes ocurren cuando un gancho se atora dentro de un tanque de pintura o en algún segmento de los conveyors (principalmente en el conveyor secundario), o bien cuando los trabajadores no alcanzan a cargar o descargar las piezas, entre otros. Cabe señalar que cuando un paro ocurre toda la línea para.

La siguiente sección muestra el modelo de simulación del sistema actual de la línea de pintura aquí descrita.

5.2 Modelo de simulación

Después de identificar las características del sistema, haber analizado y entendido el funcionamiento del mismo (comúnmente con la ayuda de gente de la planta especializada en los varios aspectos de la línea), se prosiguió a recolectar la información necesaria para construir el modelo de simulación. Parte de esta información fue proporcionada por la empresa y parte hubo que adquirirla personalmente en la planta.

En esta etapa se adquirió un software de simulación de eventos discretos, SIMUL8 en su edición Professional 2006 (Versión 13.0). La elección de este software tuvo que ver con la disponibilidad de una versión del sistema de simulación de libre distribución que permite al usuario ver, correr y analizar modelos de simulación creados en SIMUL8 Professional. Esta versión libre de SIMUL8 se llama *SIMUL8 Viewer*. En esta versión, sin embargo, no se pueden realizar modificaciones del modelo, solamente introducir

algunos parámetros por medio de cajas de diálogo. La existencia de esta versión libre fue muy atractiva para la empresa, puesto que no necesitan realizar una inversión en la adquisición de software de simulación.

En el software antes mencionado se construyó el modelo de simulación de la línea de pintura. Se utilizó Visual Logic (lenguaje de simulación incluido en SIMUL8) (Concannon et al., 2004), para realizar la programación de todas las rutinas descritas en la sección anterior. La **Figura 5.2** muestra una foto de la pantalla del modelo.

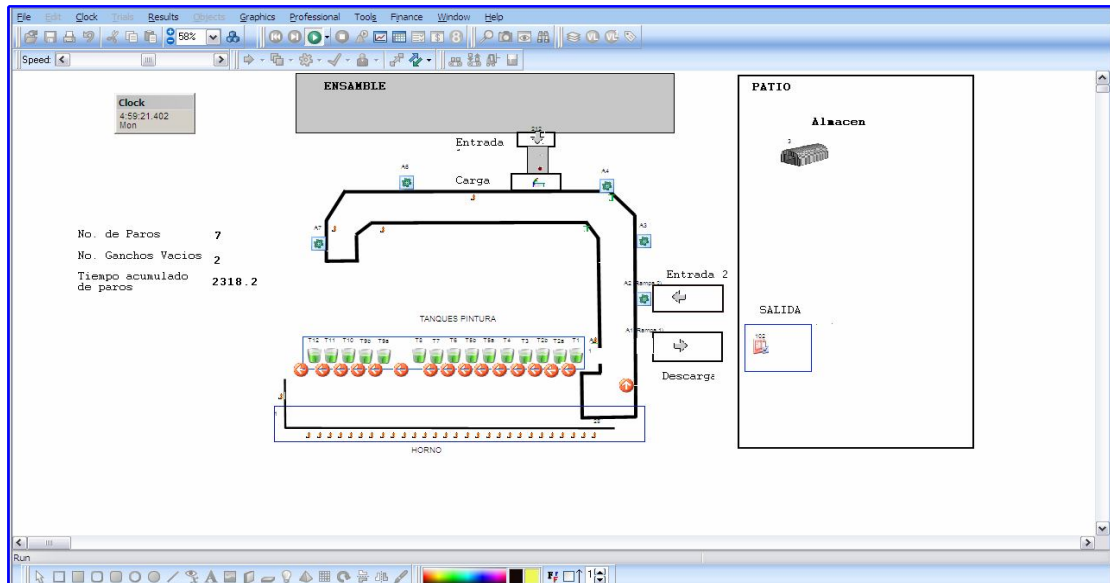


Figura 5.2: Modelo de simulación de la línea de puntado de chasis.

Como se había mencionado en capítulos previos, cuando las variables de un modelo de simulación son generadas por distribuciones de probabilidad, al momento que se desea estimar valores de las salidas del modelo, es recomendado realizar un número determinado de réplicas para estimar estos valores de forma adecuada. Para determinar este número se cuenta con diferentes pruebas estadísticas (ver Capítulo 4).

En este caso, el tiempo que tarda en llegar una unidad proveniente de la línea de ensamble así como la ocurrencia de un paro y el tiempo de reparación son generados por distribuciones de probabilidad. Por estas características es necesario realizar una prueba para definir cuantas réplicas son necesarias para poder tener un buen estimador de la salida. En este caso se consideraron como salidas el número de unidades producidas en un día de trabajo (dos turnos), así como la cantidad de ganchos que quedaron vacíos en los dos turnos. Se decidió realizar 1, 5, 10, 15, 20, 25 y 100 réplicas independientes del modelo de simulación y calcular el promedio de las salidas de cada caso. Adicionalmente para la cantidad de piezas producidas se calculó un intervalo de confianza del 95%. La **Tabla 5.1** muestra estos resultados.

Número de Réplicas	Promedio de la Cantidad de Piezas Producidas (chasis)	Intervalo de Confianza del 95% para la cantidad de piezas producidas	Promedio de la Cantidad de Ganchos Que Quedan Vacíos
1	726.00	---	28
5	738.80	(726.81,750.79)	28
10	739.40	(732.65,746.15)	26
15	739.33	(733.96,744.71)	28
20	738.70	(734.51,742.89)	28
25	739.44	(735.88,743.00)	27
100	740.16	(738.91,741.41)	27

Tabla 5.1: Salida del modelo de simulación considerando el promedio de un conjunto de réplicas.

Después de realizar las réplicas y calcular los promedios se realizó la prueba Kruskal-Wallis, donde se prueba H_0 : todas las medianas de la población son iguales vs. H_1 : al menos una mediana es diferente. H_0 se rechaza si el nivel de significancia (α) es mayor al valor de p. Esta prueba fue realizada en el software estadístico Minitab (Versión 14). Las siguientes tablas muestran los resultados para tres casos: a) La cantidad de piezas producidas (**Tabla 5.2**), b) El rango del intervalo de confianza para el caso anterior (**Tabla 5.3**) y c) El cantidad de ganchos vacíos (**Tabla 5.4**). Las tablas están organizadas de la siguiente manera, la primera columna indica el número de réplicas realizadas del modelo de simulación; la segunda columna muestra la mediana de las réplicas dentro del experimento (esto es, si se corrió varias veces el mismo número de replicas y se reportó la media, en este caso particular solo se realizó una réplica para

casa caso) y la tercera columna contiene un jerarquía por caso (donde el número más pequeño corresponde al caso donde la mediana fue menor). El valor H corresponde al estadístico de prueba que se compara con un valor de tabla, en este caso la de la distribución χ^2 . En el capítulo anterior se mostraron más detalles de esta prueba.

No. de Réplicas	Mediana	Jerarquía Promedio
1	726.0	1
5	738.8	3
10	739.4	5
15	739.3	4
20	738.7	2
25	739.4	6
100	740.2	7
H = 6.00 g.l. = 6 valor-p = 0.423		

Tabla 5.2: Prueba Kruskal-Wallis sobre el promedio de la cantidad de piezas producidas considerando diferentes números de réplicas.

No. de Réplicas	Mediana	Jerarquía Promedio
5	23.98	6
10	13.50	5
15	10.75	4
20	8.38	3
25	7.12	2
100	2.50	1
H = 5.00 g.l. = 5 valor-p = 0.416		

Tabla 5.3: Prueba Kruskal-Wallis sobre el rango del IC para la cantidad de piezas producidas.

No. de Réplicas	Mediana	Jerarquía Promedio
1	28	5.5
5	28	5.5
10	26	1
15	28	5.5
20	28	5.5
25	27	2.5
100	27	2.5
H = 6.00 g.l. = 6 valor-p = 0.423		

Tabla 5.4: Prueba Kruskal-Wallis sobre la cantidad de ganchos que salen vacíos en un día laboral.

Considerando un nivel de significancia de $\alpha = 0.05$, no podemos rechazar la hipótesis nula en ninguno de los tres casos, dado que todos los valores de p son mayores a 0.05, por lo que podemos concluir que no existe una diferencia estadísticamente

significativa entre la mediana de una réplica o 5, 10, 15, 20, 25, 100 réplicas. Por lo tanto, una sola réplica será suficiente para representar adecuadamente las salidas del modelo de simulación.

Finalizado el modelo de simulación viene la fase de validación. La siguiente sección presenta esta fase al igual que algunos resultados del sistema actual de interés para la compañía.

5.3 Validación y resultados de interés del modelo de simulación

Cuando se construye un modelo de simulación se recomienda validarlo para evaluar qué tan bien esta representando al sistema real, así como para identificar deficiencias en la formulación del modelo. Las formas más comunes de validar un modelo de simulación son: a) la opinión de los ingenieros en la planta respecto a los resultados de la simulación, b) la exactitud con que se predicen datos históricos, c) la comprobación de falla del modelo de simulación al utilizar datos que hacen fallar al sistema real, y d) ligado al punto (a) está la aceptación y confianza en el modelo de la persona que hará uso de los resultados que arroje el modelo de simulación (Kelton et al., 2002).

La validación de este modelo se llevó acabo mediante la comparación con datos históricos proporcionados por la empresa así como con la aprobación de los ingenieros de la línea en la planta.

Dentro de los datos históricos con los que se contó se tuvo la cantidad de piezas producidas por día y los ganchos que salen vacíos en este mismo tiempo. Para el primer caso el modelo de simulación predijo 726 piezas y 28 ganchos vacíos, mientras que los valores proporcionados por la empresa fueron 700 piezas y 30 ganchos respectivamente. De lo anterior podemos observar que la predicción de estos dos valores es muy cercana al valor real estimado. El personal de la empresa consideró factible esta variación. Como

se mencionó anteriormente, el modelo de simulación fue mostrado al personal encargado de la línea de pintura, quienes avalaron la lógica con la que se construyó el modelo.

Dado que el modelo de simulación ha sido aprobado por los ingenieros en la planta y la precisión de las predicciones es competitiva, éste se utilizó para calcular diferentes datos de interés para el personal de la empresa, tales como: a) el porcentaje de tiempo que la línea está en operación (up-time), que fue de **84%**, b) el porcentaje de tiempo que la línea está parada (down-time), igual a **16%**, c) la cantidad de piezas producidas en promedio por hora, igual a **40 piezas**, d) la cantidad estimada de piezas a producir en un día, igual a **726 piezas**, e) la cantidad de ganchos que salen vacíos por día, igual a **28 ganchos**, y f) la tasa de productividad definida como la proporción del total de piezas que se puede producir en una unidad de tiempo (en este caso un día de trabajo (sin incluir el tiempo de paros)), correspondiente a las piezas que actualmente se están produciendo en la misma unidad de tiempo, que resultó igual a **0.86**.

La empresa en la cual se desarrolló esta aplicación cuenta con una planta adicional en la ciudad de Saltillo, la cual frecuentemente reporta incrementos de demanda. Estos incrementos no pueden ser cubiertos en su totalidad por dicha planta, por cual recurren a la ayuda de la planta ubicada en Monterrey para pintar los chasis. La planta de Monterrey, sin embargo, no sabía si tenía capacidad para ayudarles y sí podía, no podía cuantificarlo. De los resultados mostrados anteriormente vemos que la tasa de productividad es de 0.86, lo cual nos indica que ellos pueden producir más unidades, en un orden de aproximadamente 118 piezas por día.

Dado que la planta en Monterrey es capaz de producir más unidades, de las que se están produciendo actualmente, se decidió inspeccionar por qué no se estaba reportando esta producción. Una primera idea fue que la línea de ensamble no producía estas unidades, por lo cual la línea de pintura nunca las podría producir, sin embargo la línea de ensamble produce más unidades de las que la línea de pintura está pintando en un día: las unidades no pintadas se encuentran en el almacén del patio. Debido a lo anterior, se

recurrió a la opinión de los ingenieros de la planta para que sugirieran las posibles causas detrás de esto.

De acuerdo con la opinión de las personas encargadas de la línea, se plantearon como áreas de oportunidad para aumentar la tasa de productividad el tratar que los ganchos que salen vacíos del sistema sean llenados en algún lado del la línea diferente al área de carga, para lo cual se sugiere instalar un buffer de capacidad limitada; adicionalmente se creyó que el instalar un conjunto de acumuladores en el conveyer principal en el trayecto de la carga hasta justo antes de llegar a los tanques de pintura podría ayudar. Sin embargo, realizar estas modificaciones físicas en el sistema real es costoso y requiere que la línea pare un determinado número de días. Por ejemplo el costo de instalar un acumulador es de aproximadamente 4,800 dólares. En estas situaciones es donde se vuelve de vital importancia el contar con un modelo de simulación que permita predecir si los cambios propuestos tendrán un beneficio o no, sin necesidad de realizar las modificaciones físicas previamente en el sistema.

Dados los escenarios sugeridos se decidió aplicar el método de optimización para simulaciones propuesto. Con éste se pudo determinar cual es la mejor configuración del sistema para maximizar la cantidad de piezas producidas en un día de trabajo (lo cual directamente incrementa la tasa de productividad). La siguiente sección muestra esta aplicación.

5.4 Optimización del sistema

El sistema actual puede cumplir con la demanda pronosticada, sin embargo como se mencionó, la planta que se encuentra en Saltillo no. También, como se vio en los resultados de la sección anterior, la planta de Monterrey tiene capacidad para producir más piezas de las que actualmente está produciendo, sin embargo es necesario realizar algunos ajustes o modificaciones al sistema actual. El personal de la empresa ya tiene idea de qué cambios pueden ser benéficos. El objetivo de esta sección es mostrar si estos

cambios en realidad ayudan a aumentar la producción o no. Los cambios a evaluar incluyen: a) instalar de 0 a 9 acumuladores en el trayecto de la entrada principal a los tanques de pintura y b) añadir un buffer de tamaño adecuado para recibir unidades al momento de que el buffer de entrada sea saturado en lugar de ser enviadas al patio (la capacidad máxima de éste por cuestiones de espacio físico es de 18 unidades, impactando el tamaño en el costo).

Las variables de decisión definidas para este caso fueron:

- x_j variable binaria que toma el valor de 1 si se decide instalar el acumulador j ó 0 en caso contrario, para $j=1, \dots, 9$
- $x_{10} \in [0, 18]$, variable entera que indica la capacidad máxima del buffer adicional, donde 0 nos indica que éste no es requerido.

La medida de desempeño utilizada fue la cantidad de piezas producidas en un día, $E[f(\mathbf{x}^i)]$, donde $\mathbf{x}^i = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})^i$, corresponde a la i -ésima solución.

Definido ésto, se realizaron los ajustes pertinentes al modelo de simulación para poder calcular la salida de los escenarios bajo análisis. Siguiendo la metodología de optimización propuesta se realizó lo siguiente:

INICIALIZACIÓN

Paso 1:

Iniciar con un diseño de experimentos. En los casos de prueba mostrados en el capítulo anterior se utilizó un diseño compósito central. Éste es un diseño de experimentos de 5 niveles por variable, sin embargo 9 de las 10 variables de decisión consideradas en este estudio sólo pueden tomar 2 niveles. Debe recordarse, sin embargo, que un diseño compósito central está conformado por un diseño factorial 2^j

(o factorial fraccionado 2^{j-r}), junto con puntos estrella y puntos centro. Por esta conformación, el diseño a utilizar aquí podría ser un DCC no completo. Tomando este análisis en consideración, las 9 variables binarias se usaron para construir un diseño factorial fraccionado (2^{j-5}), con el cual se generaron 16 corridas; y para la décima variable se utilizó un compósito central con 2 puntos estrella ($\alpha=1$) y un punto centro. Finalmente, se combinaron cada una de las 16 corridas del diseño factorial fraccionado con las tres del DCC para generar las 48 corridas iniciales (la **Tabla 5.5** muestra el diseño construido ya con resultados). Para cada corrida se calculó el $E[f(\mathbf{x}^i)]$ ($i=1,\dots,48$) ejecutando el modelo de simulación una sola vez (en la sección anterior se presentó una prueba para determinar el número de réplicas requerido para obtener buenos estimadores). Estos resultados se muestran en la **Tabla 5.5**.

No. de corrida (i)	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	Cantidad de piezas esperadas para producir en un día $E[f(\mathbf{x}^i)]$
1	0	0	0	0	0	0	0	0	1	0	748
2	0	0	0	1	0	1	1	1	0	0	720
3	0	0	1	0	1	1	1	0	0	0	720
4	0	0	1	1	1	0	0	1	1	0	710
5	0	1	0	0	1	1	0	1	0	0	718
6	0	1	0	1	1	0	1	0	1	0	710
7	0	1	1	0	0	0	1	1	1	0	710
8	0	1	1	1	0	1	0	0	0	0	718
9	1	0	0	0	1	0	1	1	0	0	720
10	1	0	0	1	1	1	0	0	1	0	710
11	1	0	1	0	0	1	0	1	1	0	710
12	1	0	1	1	0	0	1	0	0	0	720
13	1	1	0	0	0	1	1	0	1	0	710
14	1	1	0	1	0	0	0	1	0	0	720
15	1	1	1	0	1	0	0	0	0	0	720
16	1	1	1	1	1	1	1	1	1	0	662
17	0	0	0	0	0	0	0	0	1	9	782
18	0	0	0	1	0	1	1	1	0	9	754
19	0	0	1	0	1	1	1	0	0	9	756
20	0	0	1	1	1	0	0	1	1	9	740
21	0	1	0	0	1	1	0	1	0	9	754
22	0	1	0	1	1	0	1	0	1	9	740
23	0	1	1	0	0	0	1	1	1	9	740
24	0	1	1	1	0	1	0	0	0	9	754

Tabla 5.5: Diseño de experimentos inicial con su respectiva evaluación en el modelo de simulación. Los datos se encuentran en su escala natural.

Continuación Tabla 5.5

No. de corrida (i)	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	Cantidad de piezas esperadas para producir en un día $E[f(\mathbf{x}^i)]$
25	1	0	0	0	1	0	1	1	0	9	756
26	1	0	0	1	1	1	0	0	1	9	740
29	1	1	0	0	0	1	1	0	1	9	740
30	1	1	0	1	0	0	0	1	0	9	756
31	1	1	1	0	1	0	0	0	0	9	756
32	1	1	1	1	1	1	1	1	1	9	710
33	0	0	0	0	0	0	0	0	1	18	790
34	0	0	0	1	0	1	1	1	0	18	758
35	0	0	1	0	1	1	1	0	0	18	760
36	0	0	1	1	1	0	0	1	1	18	746
37	0	1	0	0	1	1	0	1	0	18	758
38	0	1	0	1	1	0	1	0	1	18	746
39	0	1	1	0	0	0	1	1	1	18	746
40	0	1	1	1	0	1	0	0	0	18	758
41	1	0	0	0	1	0	1	1	0	18	760
42	1	0	0	1	1	1	0	0	1	18	746
43	1	0	1	0	0	1	0	1	1	18	746
44	1	0	1	1	0	0	1	0	0	18	760
45	1	1	0	0	0	1	1	0	1	18	746
46	1	1	0	1	0	0	0	1	0	18	760
47	1	1	1	0	1	0	0	0	0	18	760
48	1	1	1	1	1	1	1	1	1	18	714

Paso 2:

Definir la *solución incumbente*, en ese paso se selecciona la solución con el máximo valor objetivo, en este caso corresponde a la combinación 33, i.e. (0, 0, 0, 0, 0, 0, 0, 0, 0, 1,18), con una producción estimada de 790 partes. Adicionalmente se inicializa el contador de iteraciones $k:=0$.

PRIMERA ITERACIÓN

Paso 3:

En este paso se actualiza el contador de iteraciones $k = k+1 = 1$

Paso 4:

Con los puntos del diseño de experimentos se obtiene un metamodelo ($\hat{f}(\cdot)_1$). Como se describió en el Capítulo 3 se construye una regresión lineal múltiple utilizando el máximo número de grados de libertad, en éste caso 47. Previo a la construcción del

metamodelo los datos fueron escalados en un rango de [-1,1]. Se calculó también el valor del coeficiente de determinación (R^2) del metamodelo. A continuación se muestra la ecuación del metamodelo obtenido y el valor de R^2 .

$$\begin{aligned} \hat{f}(x_1, x_2, \dots, x_{10})_1 = & 0.3500 - 0.0710x_1 - 0.0788x_2 - 0.0749x_3 - 0.0775x_4 - 0.0749x_5 - 0.0814x_6 \\ & - 0.0736x_7 - 0.0775x_8 - 0.0944x_9 + 0.307x_{10} + 0x_1^2 + 0x_2^2 + 0x_3^2 + 0x_4^2 + 0x_5^2 \\ & + 0x_6^2 + 0x_7^2 + 0x_8^2 + 0x_9^2 - 0.2290x_{10}^2 + 0.0072x_1x_2 + 0.0033x_1x_3 + 0.0059x_1x_4 \\ & + 0.0033x_1x_5 + 0x_1x_6 + 0.0020x_1x_7 + 0.0059x_1x_8 + 0x_1x_9 + 0.0059x_1x_{10} + 0x_2x_3 \\ & + 0x_2x_4 + 0x_2x_5 + 0x_2x_6 + 0x_2x_7 + 0x_2x_8 + 0x_2x_9 + 0.0059x_2x_{10} + 0x_3x_4 + 0x_3x_5 \\ & + 0x_3x_6 + 0x_3x_7 + 0x_3x_8 + 0x_3x_9 + 0.0059x_3x_{10} + 0x_4x_5 + 0x_4x_6 + 0x_4x_7. \end{aligned}$$

$$R^2 = 0.9930$$

Paso 5:

Posteriormente se construye un modelo de optimización (**Modelo 5.1**) y se optimiza. La función objetivo del modelo de optimización corresponde al metamodelo, las primeras 9 restricciones a la instalación de los acumuladores y la décima a la capacidad máxima del buffer. Para la optimización se utilizó la herramienta de optimización incluida en Microsoft Excel (*Solver*) combinada con la técnica de múltiples comienzos. La mejor solución encontrada fue $\hat{f}(-1, -1, -1, -1, -1, -1, -1, -1, -1, 0.67)_1 = 1.1720$, correspondiente a la combinación $\mathbf{x}_1 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 15)$.

$$\begin{aligned} & \text{Encontrar } x_j^* \quad \forall j \text{ para} & & \text{donde} \\ \max & \quad \hat{f}(\cdot)_1 & & \\ \text{s.a.} & \quad x_j^* \in \{0, 1\} \quad j = 1, 2, \dots, 9 & & x_j = 2x_j^* - 1 \quad j = 1, 2, \dots, 9 \\ & \quad 0 \leq x_{10}^* \leq 18 & & x_{10} = 0.11x_{10}^* - 1 \\ & \quad x_{10}^* \in \mathbf{Z}^+ & & \end{aligned} \tag{5.1}$$

Paso 6:

Correr el modelo de simulación ajustando los parámetros con el punto obtenido en el *paso 5*, lo cual da como resultado $E[f(0,0,0,0,0,0,0,0,0,15)] = 792$ chasis, normalizado corresponde a la solución $[(-1, -1, -1, -1, -1, -1, -1, -1, -1, 0.67), 1.0313]$.

Paso 7:

Evaluar si el punto encontrado mediante el modelo de optimización es mejor que el punto incumbente, esto es:

$$E[f(0,0,0,0,0,0,0,0,0,15)] > E[f(0,0,0,0,0,0,0,0,1,18)]$$

$$792 > 790 \quad (SI)$$

Paso 8:

Dado que el *paso 7* resultó verdadero, la solución incumbente se actualiza, esto es:

$$[x_{1\text{-mejor}}, E[f(x_{1\text{-mejor}})]] := [(0,0,0,0,0,0,0,0,15),792]$$

Paso 9:

Enseguida se evalúan los criterios de terminación del método. En este paso es necesario definir dos parámetros, el valor de ϵ , el cual acota el valor de R^2 y el máximo número de evaluaciones de modelo de simulación aceptadas; en este caso el valor de ϵ fue fijado en 1 dado que el valor R^2 del metamodelo inicial es muy alto, y el número de evaluaciones de simulación fue fijado en 1,536 corridas (correspondiente a las corridas de un diseño factorial completo a dos niveles para el caso de x_j , $j=1,\dots,9$ y de tres niveles para x_{10}). La **Tabla 5.6** muestra estas evaluaciones.

Criterio	Valor	¿Para?
La combinación (0,0,0,0,0,0,0,0,15) ya había sido simulada.	(0,0,0,0,0,0,0,0,15) no ha sido simulado	NO
El valor de $R^2 \geq \epsilon$, ($\epsilon=1$)	$R^2=0.9930$	NO
El número máximo de simulaciones se ha alcanzado (1,536 simulaciones).	49	NO

Tabla 5.6: Evaluación de criterios de terminación.

Dado que ninguno de los criterios de terminación se cumplió, entonces el punto obtenido en el *paso 5* ($[(0,0,0,0,0,0,0,0,15),792]$) junto con los puntos del diseño de experimento son incluidos a una nueva iteración (iteración principal).

SEGUNDA ITERACIÓN***Paso 3:***

Actualizar el contador de iteraciones ($k=2$)

Paso 4:

Utilizando todos los puntos disponibles se construye un nuevo metamodelo ($\hat{f}(\cdot)_2$) y se reporta el valor de R^2 , a continuación se muestran estos resultados:

$$\begin{aligned} \hat{f}(x_1, x_2, \dots, x_{10})_2 = & 0.3500 - 0.0710x_1 - 0.0788x_2 - 0.0749x_3 - 0.0775x_4 - 0.0749x_5 - 0.0814x_6 \\ & - 0.0736x_7 - 0.0775x_8 - 0.0240x_9 + 0.307x_{10} + 0x_1^2 + 0x_2^2 + 0x_3^2 + 0x_4^2 + 0x_5^2 \\ & + 0x_6^2 + 0x_7^2 + 0x_8^2 + 0x_9^2 - 0.2290x_{10}^2 + 0.0072x_1x_2 + 0.0033x_1x_3 + 0.0059x_1x_4 \\ & + 0.0033x_1x_5 + 0.0704x_1x_6 + 0.0020x_1x_7 + 0.0059x_1x_8 + 0x_1x_9 + 0.0059x_1x_{10} \\ & + 0x_2x_3 + 0x_2x_4 + 0x_2x_5 + 0x_2x_6 + 0x_2x_7 + 0x_2x_8 + 0x_2x_9 + 0.0059x_2x_{10} + 0x_3x_4 \\ & + 0x_3x_5 + 0x_3x_6 + 0x_3x_7 + 0x_3x_8 + 0x_3x_9 + 0.0059x_3x_{10} + 0x_4x_5 + 0x_4x_6 + 0x_4x_7 \end{aligned}$$

$$R^2 = 0.9937$$

Paso 5:

Aquí se genera y optimiza un modelo de optimización igual al (5.1), solamente reemplazando $\hat{f}(\cdot)_1$ por $\hat{f}(\cdot)_2$, la mejor solución encontrada a través de la optimización fue: $\hat{f}(-1, -1, -1, -1, -1, -1, -1, -1, -1, 0.67)_2 = 1.0312$, correspondiente a la solución $\mathbf{x}_2 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 15)$.

Paso 6, 7 y 8:

En esta etapa se evalúa el punto generado por el paso anterior en el modelo de simulación, sin embargo en este caso éste ya había sido simulado. La comparación y actualización de la solución incumbente no es necesaria, pues esta no cambiará, entonces $[\mathbf{x}_{2\text{-mejor}}, \mathbf{E}[f(\mathbf{x}_{2\text{-mejor}})]] := [(0, 0, 0, 0, 0, 0, 0, 0, 0, 15), 792]$.

Paso 9:

Evaluar los criterios de parada, la **Tabla 5.7** muestra estos resultados. Ahora uno de los criterios de terminación se cumplió, por lo cual el método para y se reporta la mejor solución. En este caso la mejor solución encontrada por el método propuesto fue

(0,0,0,0,0,0,0,0,0,15), donde el valor esperado de partes a producir en un día es de **792 partes**.

Criterio	Valor	¿Para?
La combinación (0,0,0,0,0,0,0,0,0,15) ya había sido simulada.	(0,0,0,0,0,0,0,0,0,15) ya había sido simulado (simulación 49)	SI
El valor de $R^2 \geq \epsilon$, ($\epsilon=1$)	$R^2=0.9937$	NO
El número máximo de simulaciones se ha alcanzado (1,536 simulaciones).	50	NO

Tabla 5.7: Evaluación de criterios de parada

Para evaluar qué tan buena fue la solución encontrada por el método de optimización propuesto, se utilizó el optimizador OptQuest para encontrar una solución a este sistema. El optimizador requiere que el usuario ligue éste con el modelo de simulación. Posteriormente se especifican las variables de decisión, restricciones, la función objetivo y un criterio de parada. Una vez ajustado el modelo, y seleccionando un criterio de parada de 100 simulaciones, se corrió OptQuest y la mejor solución encontrada fue **[(0,0,0,0,0,0,0,0,0,10),792.22]**. Cabe señalar que OptQuest realiza por lo menos 3 réplicas para cada conjunto de parámetros por lo cual se observa un número con decimales en la salida.

Con el objeto de realizar una comparación justa contra la solución obtenida por el método propuesto, se decidió correr la solución de OptQuest sólo una ocasión y reportar la salida. Finalmente la solución encontrada fue de 788 partes. Analizando el historial de las iteraciones, se notó que otras 3 soluciones mostraban el mismo valor de función objetivo. Se tomaron éstas y se evaluaron en el modelo de simulación. La **Tabla 5.8** muestra las soluciones en el orden que fueron encontradas por OptQuest, notando que la primera es la que se mostró en el reporte final.

Solución Número	Solución (x_{mejor})	Valor Objetivo en base a una réplica ($E[f(x_{mejor})]$), (piezas)
1	(0,0,0,0,0,0,0,0,10)	788
2	(0,0,0,0,0,0,0,0,14)	792
3	(0,0,0,0,0,0,0,0,15)	792
4	(0,0,0,0,0,0,0,0,17)	790

Tabla 5.8: Soluciones encontradas utilizando OptQuest.

Si observamos la **Tabla 5.8**, la solución 3 encontrada por OptQuest es igual a la encontrada con el método aquí propuesto, por lo cual se tiene mayor certeza que esta es una solución buena. Cabe destacar que el método propuesto realiza únicamente 50 evaluaciones de simulación, parando ahí automáticamente, mientras que el OptQuest realizó 100 evaluaciones, además que este número máximo de evaluaciones es un parámetro que el usuario debe determinar. Para determinar este último es recomendable realizar pruebas con diferentes valores en los criterios de parada y seleccionar la mejor (www.simul8.com) lo que implica aumentar el número de evaluaciones del modelo de simulación. Dentro de algunos de los problemas de optimización conocidos resueltos mediante OptQuest está: 1) el problema de selección de proyectos (10 artículos) tomándole alrededor de 100 evaluaciones de la función objetivo (Laguna, 1997a); 2) un problema de diseño de redes de tuberías de agua de 20 nodos necesitó aproximadamente 10,000 evaluaciones (Laguna, 1997a); 3) un problema de optimización no lineal con variables enteras, continuas y binarias (20 variables) tomó 6,000 evaluaciones (Laguna, 1997). Adicionalmente se tuvo la oportunidad de platicar con una de las personas que desarrollaron esta herramienta de optimización, indicando que éste necesita de un número considerable de evaluaciones del modelo de simulación para asegurar que la solución final sea buena. Aunque es cierto que en ocasiones la ejecución del modelo de simulación tarda segundos o un número pequeño de minutos, en otras ocasiones éstos llegan a tardar horas o días, por lo cual es crítico reducir el número de evaluaciones del modelo de simulación.

En este caso en particular, una ejecución del modelo de simulación tarda aproximadamente 60 segundos, por lo tanto el método propuesto requirió

aproximadamente 3,000 seg. de análisis computacional más el tiempo de la optimización (que es insignificante respecto al tiempo de simulación), mientras que OptQuest requirió aproximadamente 180,000 segundos de tiempo de simulación más el tiempo de la optimización, esto considerando 3 réplicas para cada solución evaluada (una por iteración). De lo anterior tenemos un ahorro de 177,000 segundos en tiempo de simulación. Tanto la optimización como las simulaciones se realizaron en una computadora portátil con sistema operativo Windows XP Home Edition (Versión 2002), procesador Intel Cetrino de 1.86 GHz y 1 GB de memoria RAM.

A continuación se presenta la decisión final propuesta en base a los resultados del análisis anterior.

5.5 Decisión Final Propuesta

De acuerdo con los resultados obtenidos del análisis previo, se ha recomendado instalar un buffer con una capacidad máxima de 14 o 15 unidades, así como no instalar ningún acumulador. Estos ajustes repercutirán en un incremento de aproximadamente 65 unidades diarias. Así mismo cabe mencionar que el costo de instalación de un acumulador es de aproximadamente 4,800 dólares, e inicialmente el personal de la planta planeaba instalar 3 unidades, por lo que al determinar que ésto no es requerido para el aumento de productividad, se logró un ahorro en esta inversión.

Actualmente la empresa ha iniciado trabajos para adecuar un buffer en el área determinada, sin embargo para los tiempos de este trabajo no se logró recolectar información para corroborar la solución del nuevo sistema comparada con la solución arrojada por el modelo de simulación, por lo que se deja como trabajo a futuro.

5.6 Conclusiones

En este capítulo se presentó el modelo de simulación de una línea de pintura de partes automotrices de una empresa de la ciudad de Monterrey. Con dicho modelo se pudo determinar el estado actual de la línea. Además se determinó la mejor configuración del sistema tratando de maximizar la producción. Para esto último se utilizó el método de optimización aplicado a la simulación aquí propuesto. Así mismo se pudo determinar si los cambios propuestos tendrían beneficios o no en términos de incrementar la productividad sin necesidad de realizar las modificaciones físicas del sistema.

Adicionalmente se comparó la mejor solución encontrada por el método propuesto con la solución obtenida por el optimizador OptQuest, con resultados muy competitivos y un ahorro significativo con respecto al número de evaluaciones del modelo de simulación.

El siguiente capítulo muestra una aplicación del método de optimización propuesto, en un modelo de simulación de fenómenos continuos en el proceso de moldeo por inyección.

Capítulo 6

CASO DE ESTUDIO 2

OPTIMIZACIÓN DE UN PROCESO DE MOLDEO POR INYECCIÓN DE TERMOPLÁSTICOS

Los resultados presentados en este capítulo forman parte de los trabajos desarrollados en una estancia de investigación en The Ohio State University, en la cual se trabajó en conjunto con el Grupo de Procesamiento de Polímeros, a cargo del Dr. José M. Castro.

En un proceso de moldeo por inyección de termoplásticos, al igual que en cualquier proceso de manufactura, es de vital importancia la calidad de las partes. Hay muchas maneras de cuantificar la calidad de una parte. En particular en este proceso, es de interés que la parte no sufra encogimiento, esto es, que no se desvíe de las dimensiones

requeridas y que conserve la geometría de diseño. Estas medidas se vuelven más importantes cuando la parte que se va a producir formará parte de un ensamble. Adicionalmente es de interés que el tiempo que se requiere para fabricar una pieza (tiempo de ciclo) sea el mínimo. La definición del tiempo de ciclo para este proceso se puede consultar en el **Apéndice E**. Por tanto la optimización de este tipo de procesos es, entonces, muy necesaria.

El modelamiento del proceso de moldeo por inyección se puede realizar con base en los fenómenos físicos y químicos involucrados. Para geometrías muy sencillas, se pueden generar estimaciones de los aspectos de interés de forma analítica. Sin embargo, la interrelación de fenómenos en el procesamiento de polímeros y la geometría de las partes suelen ser muy complejos. Es por esta razón que se recurre a la modelación del proceso vía simulación. Por consiguiente, la aplicación de un método de optimización para simulaciones en estos casos se vuelve también de vital importancia.

En este capítulo se muestra el modelo de simulación de un proceso de moldeo por inyección de termoplásticos desarrollado en el software MoldFlowTM por Mulyana (2006). El modelo es posteriormente optimizado con el objetivo de encontrar la configuración del proceso que minimice el tiempo de ciclo y maximice la calidad de la parte a producir.

6.1 Descripción y modelo de simulación del proceso

Como se mencionó previamente, el modelo que se presenta a continuación fue desarrollado por Mulyana (2006) en el software de simulación MoldFlowTM (Moldflow Plastics Insight 6.0). El modelo, que es un mallado de elementos finitos, se utiliza para simular la transferencia de masa y calor durante el proceso de inyección de un polímero hacia la cavidad rectangular de un molde. La parte resultante es, en este caso, una placa rectangular. El molde tiene catorce puertos de inyección paralelos fijados en un sólo lado de la cavidad. Las dimensiones de diseño de la placa son 12cm x 6cm x 0.3cm

(largo x ancho x grosor) y un volumen total de 21.6 cm^3 . La **Figura 6.1** muestra el mallado de la parte.

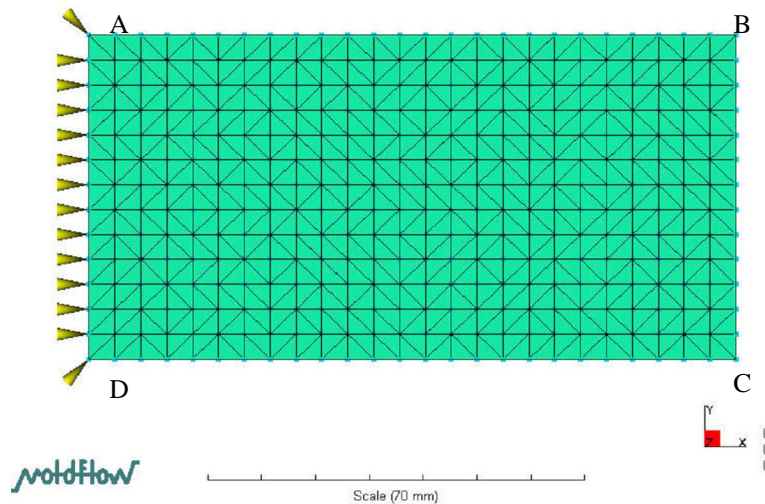


Figura 6.1: Mallado de la placa rectangular con un conjunto de puertos de inyección

Existen diferentes parámetros de la configuración del proceso que se pueden ajustar con diferentes fines. Dos objetivos muy comunes son la maximización de la calidad final de la parte y la minimización del tiempo de ciclo. La siguiente sección muestra la optimización de este proceso involucrando estos objetivos.

6.2 Optimización del proceso

Dadas las consideraciones descritas al inicio de este capítulo, el objetivo en este caso fue minimizar el encogimiento de la parte medido a través de la suma de las diferencias absolutas entre el tamaño original y el final de la parte para los cuatro lados de la pieza AB, BC, CD y AD (considerando únicamente largo y ancho), como se muestra en la **Figura 6.1**. Mulyana (2006) realizó un análisis para determinar cuáles de las variables controlables del proceso afectan el encogimiento de la parte. Las variables controlables en este proceso, seleccionadas de acuerdo con el estudio de Mulyana, fueron la temperatura del polímero (t_p) en un rango de 176°C a 260°C y la máxima presión de empacado (p_e) en un rango de 40 a 80 MPa. Estos rangos fueron tomados de acuerdo a la

sugerencia del software de simulación, los cuales dependen del material a utilizar. El material utilizado en este caso fue *Dow Chemical USA Styron 615 APR (poliestireno)*. Los demás parámetros del proceso quedaron fijos en los valores sugeridos por el simulador.

Asimismo cabe mencionar que si bien el encogimiento es la contracción del polímero debido a los cambios de temperatura entre la temperatura de polímero y la temperatura ambiente, en ocasiones el tamaño de la parte final es mayor al tamaño de diseño, resultando ésto en un valor negativo de encogimiento.

Al igual que en el caso mostrado en el Capítulo 4 desarrollado en el simulador MoldFlowTM, este modelo de simulación no contiene variables estocásticas. Esto implica que el modelo siempre generará la misma salida y, por tanto, no es necesario realizar réplicas del modelo.

Posterior a la configuración del modelo de simulación, se aplicó el método de optimización aquí propuesto. Se inició con un diseño de experimentos compuesto central, el cual se muestra en la **Tabla 6.1**, donde AB_E indica el valor del encogimiento del lado AB (tamaño de diseño menos tamaño final) y $|AB_E|$ representa el valor absoluto de AB_E , de mismo modo se pueden interpretar los valores para los lados restantes. La **Tabla 6.2** muestra los resultados de cada iteración del método de optimización, donde la primera columna indica el número de la iteración; la segunda la mejor solución encontrada utilizando cada metamodelo; la tercera columna contiene la evaluación de la solución en el modelo de simulación; la cuarta muestra la solución incumbente hasta esa iteración y la columna cinco muestra el valor de R^2 del metamodelo. La **Figura 6.2** muestra los puntos del diseño de experimentos inicial (triángulos) y los puntos encontrados mediante los metamodelos en cada iteración (círculos) (iniciando en (185.0°C, 80.0MPa) y terminando en (235.0°C, 40.0MPa)).

Punto	t_p (°C)	p_e (MPa)	Encogimiento (mm)				Suma de Encogimientos $E[f(t_p, p_e)]$ (mm)
			$[AB]_E$	$[BC]_E$	$[CD]_E$	$[DA]_E$	
1	176	40	0.23	0.16	0.23	0.17	0.81
2	176	80	0.06	0.03	0.06	0.17	0.34
3	197	60	0.12	0.04	0.12	0.19	0.49
4	218	50	0.23	0.08	0.23	0.20	0.75
5	218	60	0.21	0.07	0.21	0.19	0.70
6	218	70	0.20	0.06	0.20	0.19	0.67
7	239	60	0.30	0.15	0.30	0.21	0.98
8	260	40	0.35	0.18	0.35	0.12	1.03
9	260	80	0.38	0.20	0.38	0.22	1.19

Tabla 6.1: Puntos del diseño de experimentos inicial y evaluación en el modelo de simulación (encogimiento de los lados AB, BC, CD, DA).

Iteración	Mejor solución del metamodelo (t_p, p_e) (C, MPa)	Evaluación en la Simulación $E[f(t_p, p_e)]$ (mm)	Solución Incumbente $[(t_p, p_e), E[f(t_p, p_e)]]$	R^2 del metamodelo
Inicialización DOE	(176.0, 80.0)	0.34	[(176.0, 80.0), 0.34]	--
1	(185.0, 80.0)	0.26	[(185.0, 80.0), 0.26]	0.9998
2	(189.0, 80.0)	0.22	[(189.0, 80.0), 0.22]	0.9999
3	(176.0, 53.4)	0.53	[(189.0, 80.0), 0.22]	0.9999
4	(193.0, 80.0)	0.26	[(189.0, 80.0), 0.22]	0.9997
5	(260.0, 55.4)	1.14	[(189.0, 80.0), 0.22]	0.9994
6	(188.5, 80.0)	0.23	[(189.0, 80.0), 0.22]	0.9994
7	(188.0, 79.0)	0.22	[(189.0, 80.0), 0.22]	0.9994
8	(235.0, 40.0)	0.80	[(189.0, 80.0), 0.22]	1.000

Tabla 6.2: Resultados de cada iteración del método de optimización para la simulación del flujo de un polímero dentro del molde de una placa rectangular, minimizando el encogimiento de la parte.

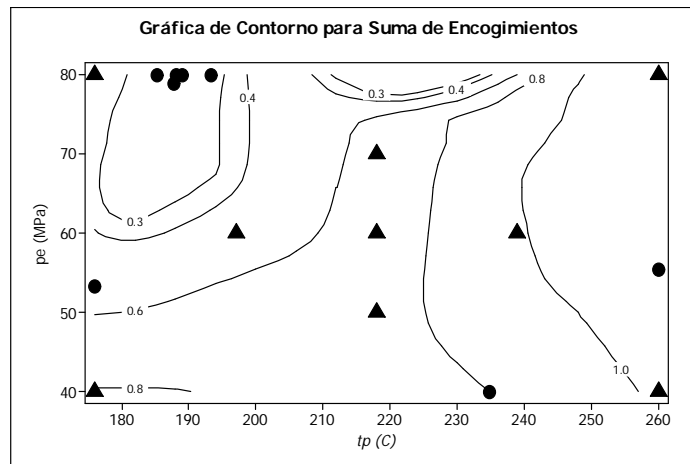


Figura 6.2: Gráfica de contorno para la Suma de Encogimientos en función de la temperatura del polímero y máxima presión de empaçado.

En la **Tabla 6.2** podemos observar que la mejor solución encontrada por el método de optimización es: temperatura del polímero igual a 189°C y máxima presión de empacado igual a 80 MPa, con los cuales la suma de los encogimientos de la parte (sin considerar el grosor) será de 0.22 mm. Se encontró también que la combinación (188.0°C, 79.0MPa) tiene el mismo valor objetivo. Cabe mencionar que ninguno de estos puntos estaba contenido en el DOE inicial.

Tomando este caso como base y utilizando el mismo modelo de simulación, se realizaron diferentes análisis donde se varió el material a utilizar y la medida de desempeño (funciones objetivo). Los diferentes casos en términos de la función objetivo utilizada fueron: a) Minimizar el Tiempo de Ciclo (**TC**), b) Minimizar la diferencia absoluta entre el encogimiento de los lados DA y BC de la parte ($|DA_E - BC_E|$), c) Minimizar la suma de todos los encogimientos (**SE**) ($|AB_E| + |BC_E| + |CD_E| + |DA_E|$), d) Minimizar una función conjunta entre los casos (a) y (b) ($0.5 * \mathbf{TC} + 0.5 * |DA_E - BC_E|$), e) Minimizar una función conjunta entre el caso (a) y (c) ($0.5 * \mathbf{TC} + 0.5 * \mathbf{SE}$), f) Minimizar una función conjunta entre los casos (b) y (c) ($0.5 * |DA_E - BC_E| + 0.5 * \mathbf{SE}$), g) Minimizar una función conjunta entre los casos (a), (b) y (c) ($1/3 * \mathbf{TC} + 1/3 * |DA_E - BC_E| + 1/3 * \mathbf{SE}$). Todos los casos se desarrollaron utilizando dos materiales: el primero *Dow Chemical USA Styron 615 APR (poliestireno)* mostrado en la **Tabla 6.3** y el segundo *Dow Chemical USA LDPE 4012 (polietileno)* mostrado en la **Tabla 6.4**. Las variables controlables en todos los casos fueron la temperatura del polímero (t_p) y la máxima presión de empacado (p_e). Para el primer material las variables controlables varían en un rango de 176°C a 260°C para la temperatura del polímero y la máxima presión de empacado entre 40 y 80 MPa. Para el segundo caso, la temperatura del polímero varía entre 180°C y 280°C y la presión de empacado entre 20 MPa y 60 MPa. Los valores anteriores fueron tomados de acuerdo a lo sugerido por el software de simulación. Los demás parámetros del proceso se dejaron en los valores por defecto.

Medida de Desempeño (minimizar)	No. de corridas de simulación	Mejor solución(s) encontrada(s) (t_p, p_e)	Mejor valor objetivo encontrado $E[f(t_p, p_e)]$	R ² Metamodelo Final
Tiempo de Ciclo	10	(176°C, 40 MPa) (176°C, 60 MPa) (176°C, 80 MPa)	TC = 14.68 seg.	1.000
$ DA_E - BC_E $	11	(176°C, 40 MPa)	$ DA_E - BC_E = 0.01$ mm	0.9997
$ AB_E + BC_E + CD_E + DA_E $	17	(188°C, 79 MPa)	SE = 0.22 mm.	1.000
$0.5 * TC + 0.5 * DA_E - BC_E $	9	(176°C, 40 MPa)	TC = 14.68 seg. $ DA_E - BC_E = 0.01$ mm	0.9993
$0.5 * TC + 0.5 * (AB_E + BC_E + CD_E + DA_E)$	9	(176°C, 80 MPa)	TC = 14.68 seg. SE = 0.34 mm	0.9994
$0.5 * DA_E - BC_E + 0.5 * (AB_E + BC_E + CD_E + DA_E)$	9	(176°C, 40 MPa)	$ DA_E - BC_E = 0.01$ mm SE = 0.81 mm	0.9994
$1/3 * TC + 1/3 * DA_E - BC_E + 1/3 * (AB_E + BC_E + CD_E + DA_E)$	9	(176°C, 40 MPa)	TC = 14.68 seg. $ DA_E - BC_E = 0.01$ mm SE = 0.81 mm	0.9995

Tabla 6.3: Resultados al aplicar el método de optimización para la simulación de la placa rectangular, utilizando el polímero *Dow Chemical USA Styron 615 APR*, evaluando diferentes medidas de desempeño a minimizar.

Medida de Desempeño (minimizar)	No. de corridas de simulación	Mejor solución(s) encontrada(s) (t_p, p_e)	Mejor valor objetivo encontrado $E[f(t_p, p_e)]$	R ² Metamodelo Final
Tiempo de Ciclo	10	(180°C, 20 MPa) (180°C, 40 MPa) (180°C, 60 MPa)	TC = 14.44 seg.	1.000
$ DA_E - BC_E $	12	(280°C, 20 MPa)	$ DA_E - BC_E = 0.00$ mm	0.9999
$ AB_E + BC_E + CD_E + DA_E $	9	(280°C, 60 MPa)	SE = 2.27 mm	1.000
$0.5 * TC + 0.5 * DA_E - BC_E $	9	(180°C, 20 MPa)	TC = 14.44 seg. $ DA_E - BC_E = 0.01$ mm.	0.9994
$0.5 * TC + 0.5 * (AB_E + BC_E + CD_E + DA_E)$	9	(180°C, 60 MPa)	TC = 14.44 seg. SE = 2.62 mm.	1.0000
$0.5 * DA_E - BC_E + 0.5 * (AB_E + BC_E + CD_E + DA_E)$	9	(280°C, 60 MPa)	$ DA_E - BC_E = 0.01$ mm. SE = 2.27 mm.	0.9994
$1/3 * TC + 1/3 * DA_E - BC_E + 1/3 * (AB_E + BC_E + CD_E + DA_E)$	12	(180°C, 20 MPa)	TC = 14.44 seg. $ DA_E - BC_E = 0.01$ mm. SE = 3.05 mm.	0.9960

Tabla 6.4: Resultados al aplicar el método de optimización para la simulación de la placa rectangular, utilizando el polímero *Dow Chemical USA LDPE 4012*, evaluando diferentes medidas de desempeño a minimizar.

Las **Figuras 6.3** y **6.4** muestran los resultados descritos en las tabas anteriores (**Tabla 6.3** y **6.4**, respectivamente), sin embargo, en este caso los puntos representan las

soluciones en términos de las medidas de desempeño, considerando los siete casos optimizados.

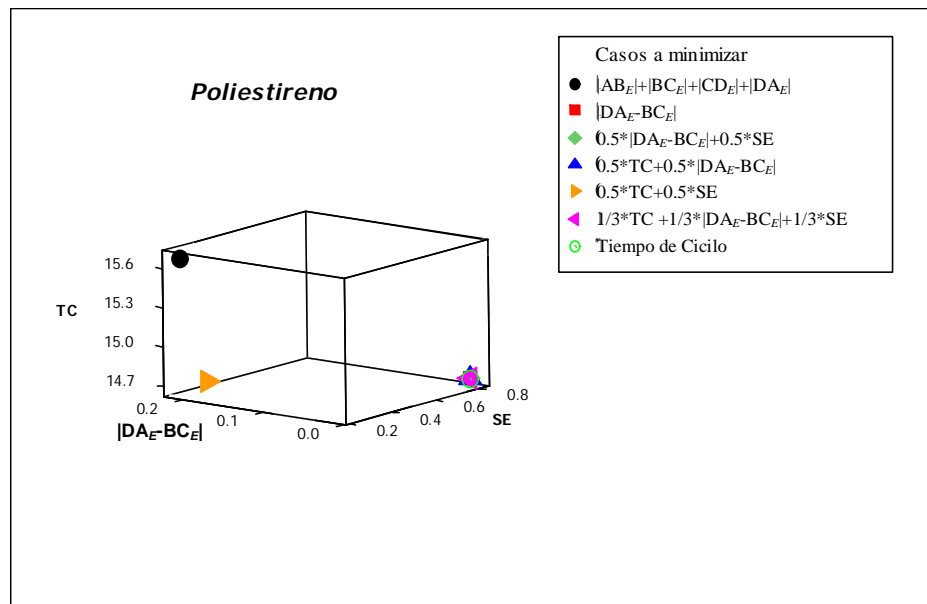


Figura 6.3: Solución en términos de las medidas de desempeño para todos los casos mostrados en la **Tabla 6.3**.

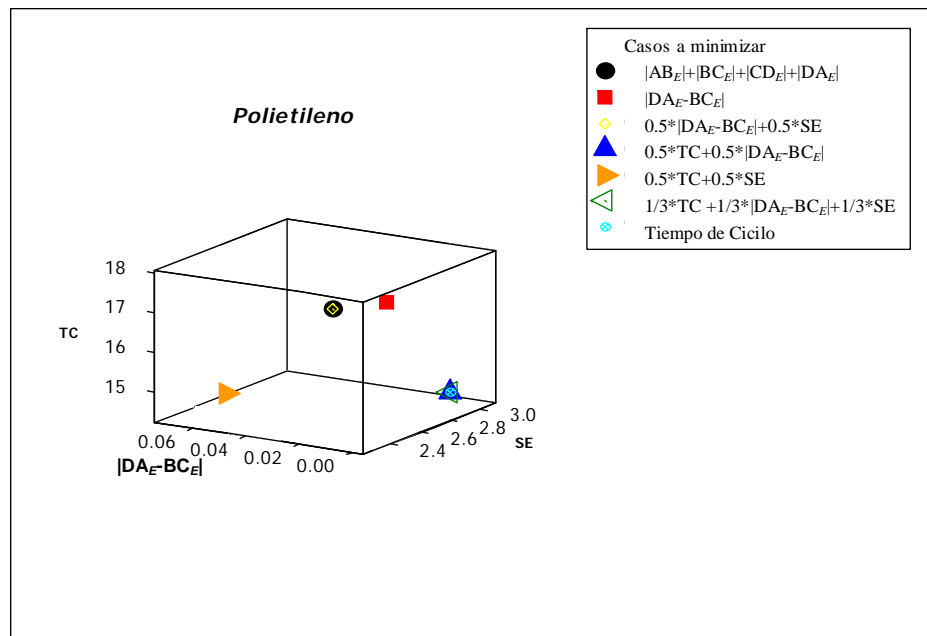


Figura 6.4: Solución en términos de las medidas de desempeño para todos los casos mostrados en la **Tabla 6.4**.

Finalmente, para saber qué tan bueno fue el desempeño del método de optimización en estos casos, se platicó con las personas de grupo de trabajo anteriormente mencionado, los cuales son expertos en el área de procesamiento de polímeros. Ellos pudieron validar estos resultados y, a su criterio y por las bases físicas y químicas de proceso, confirmaron que estos resultados tienen sentido, pues por ejemplo, cuando se desea reducir el tiempo de ciclo, el cual está dominado por el tiempo de enfriamiento de la parte dentro del molde, y se incrementa si la temperatura del polímero es mayor (ver Apéndice E), por lo tanto, con el objetivo de minimizar esta medida de desempeño, la temperatura del polímero deberá encontrarse en el nivel más bajo posible, este mismo resultado fue encontrado por medio del método de optimización aquí utilizado.

6.3 Consideraciones para optimización de problemas multicriterio

En este caso se pudo observar que al tratar de minimizar una medida de desempeño, por ejemplo el tiempo de ciclo, se maximiza el encogimiento total de la parte, lo que quiere decir que estos dos objetivos (medidas de desempeño (MDs)) se encuentran en conflicto. La consideración de solamente una medida de desempeño para su optimización puede resultar en combinaciones de variables de decisión que determinen niveles muy pobres en otras MDs, dependiendo del nivel de conflicto entre éstas. De ahí que la consideración simultánea de las MDs de interés sea indispensable en presencia de conflicto. La resolución de tales problemas, al contrario de los problemas donde solamente se considera una MD, no busca una solución única, sino un abanico de soluciones conocidas como *soluciones eficientes*. El conjunto de estas soluciones define una *frontera eficiente*.

Lo anterior nos lleva a un problema de optimización multicriterio (o multiobjetivo). Una forma de resolver este tipo de problemas es construir una función objetivo conjunta asignando pesos o preferencias a cada una de las medidas de desempeño, tal como se realizó en este ejemplo. Sin embargo, el realizar este procedimiento implica seleccionar de forma subjetiva los valores de los pesos. Por ejemplo si se quiere minimizar el tiempo que se requiere para producirse una parte y su encogimiento total, dado el conflicto

observado en el caso anterior, para la persona encargada de producción tendrá mayor importancia el tiempo de ciclo, mientras que para alguien del departamento de calidad, será preferible minimizar el encogimiento.

Una de las técnicas que ha sido utilizada por Cabrera-Ríos et al. (2002, 2002a, 2002b, 2002c, 2003, 2004) y, Castro et al. (2003, 2005), para resolver este tipo de problemas en el procesamiento de polímeros de una forma objetiva es el *Análisis Envoltente de Datos*, la cual se basa en programación lineal para encontrar la frontera eficiente. Esta técnica fue desarrollada por Charnes, Cooper y Rhodes (Charnes et al., 1978), (Charnes et al., 1993).

En el siguiente capítulo correspondiente a conclusiones y trabajo a futuro, se muestra el bosquejo de una idea para extender el método propuesto para resolver problemas multiobjetivo de forma objetiva.

Capítulo 7

CONCLUSIONES Y TRABAJO A FUTURO

En este capítulo se presentan las conclusiones de este trabajo, las contribuciones y el trabajo a futuro.

7.1 Conclusiones

En este trabajo de tesis se propuso un método para resolver problemas de optimización aplicada a la simulación, el cual incluye técnicas de diseño de experimentos y metamodelación. Este método puede ser utilizado tanto en simulaciones discretas como continuas, donde la primera tiene una aplicación común en sistemas de manufactura y la segunda, involucra la modelación de fenómenos físicos y químicos, en procesos de manufactura.

Esencialmente, este método presenta variaciones sobre estrategias de metamodelación que ya han sido probadas, como la de Martin y Simpson (2002). Creemos, sin embargo, que la manera como hemos definido (i) la inclusión iterativa de soluciones atractivas, (ii) la selección y construcción analítica de un metamodelo saturado, (iii) la selección del diseño experimental y (iv) la selección de los criterios de terminación, forma una estrategia con características propias que converge a soluciones altamente atractivas en un número modesto de corridas del modelo de simulación.

Las evaluaciones que se realizaron del método propuesto incluyeron funciones de prueba para optimización global y funciones no lineales tanto polinomiales como no polinomiales. En 17 de las 19 aplicaciones de evaluación realizadas el método encontró la solución óptima o una muy cercana a ésta. Solamente en una ocasión el método reportó una solución lejana a la óptima global. Adicionalmente se realizó la aplicación del método propuesto a diferentes modelos de simulación sencillos que son fáciles de replicar. Estos últimos se presentaron explícitamente en varias iteraciones de la evolución del método con el fin de mostrar la simplicidad de su funcionamiento, pues es ésta otra de las cualidades que lo hacen muy atractivo para su instauración y utilización por parte de personas con conocimientos modestos en el área de optimización.

También se presentó la construcción del modelo de simulación de una línea de pintura de partes automotrices de una empresa de la ciudad de Monterrey. Con dicho modelo se pudo determinar el estado actual de la línea para posteriormente determinar la mejor configuración del sistema tratando de maximizar la producción. Esto último se pudo lograr gracias a la aplicación del método aquí propuesto, demostrando así su capacidad de utilización en un sistema a escala real en el ámbito de simulaciones de eventos discretos. El método fue aplicado también a un mallado de elementos finitos que simula el transporte de masa y calor en una parte destinada a producción en moldeo por inyección, haciendo la contraparte a escala real de una simulación de fenómenos continuos.

Para el caso de las simulaciones de eventos discretos realizadas se comparó la mejor solución encontrada por el método propuesto con la solución obtenida por el optimizador OptQuest con resultados muy competitivos y un ahorro significativo con respecto al número de evaluaciones del modelo de simulación. Para el caso de las simulaciones de fenómenos continuos, se realizaron comparaciones con diseños de experimentos y se contó con la validación de expertos en el área de procesamiento de polímeros.

El método propuesto, además de presentar la versatilidad de poder aplicarse a los dos tipos de simulación ya descritos, es transparente y se presta para una codificación muy conveniente. De hecho, en este trabajo todos los casos se realizaron utilizando Microsoft Excel.

7.2 Contribuciones

La principal contribución de este trabajo es el desarrollo de un método de optimización para simulaciones que: (a) permite obtener resultados muy competitivos de optimización, (b) es aplicable a casos de simulación de eventos discretos o fenómenos continuos, (c) es fácil de utilizar, (d) no requiere de un amplio conocimiento en el área de optimización, (e) es económico en términos monetarios y computacionales, y, (f) es atractivo para las compañías mexicanas, puesto que no requerirán de la adquisición de un software especial ni de entrenamiento altamente especializado.

Adicionalmente, se considera que la construcción del modelo de simulación de la línea de pintura, así como la optimización del sistema es en sí una contribución a la práctica de la investigación de operaciones en compañías mexicanas. La compañía cuenta ahora con el modelo de simulación de la línea para poder evaluar modificaciones sin necesidad de realizar cambios físicos.

7.3 Trabajo a Futuro

Como extensión de este trabajo se tratará de identificar qué tipo de diseño de experimentos es el más adecuado para iniciar el método dependiendo de las características de cada problema que incluyen el número de variables de decisión, la naturaleza de las variables (discretas o continuas), así como de la naturaleza de las restricciones del problema. Como se puntualizó en los Capítulos 2 y 4, un diseño de experimentos clásico realiza un muestreo a diferentes niveles de las variables en una región limitada usualmente por restricciones laterales. Así mismo se vio que el tamaño de las muestras se va incrementando considerablemente cuando el número de variables es grande. Una de las técnicas de diseños de experimentos que se ha propuesto para los casos en los que la región de búsqueda involucra restricciones funcionales o en los que el número de puntos muestrales es limitado, es el uso de diseños generados por computadora, como los diseños óptimos y los de mínima aberración. Se plantea explorar estos diseños en un futuro. Aunque una alternativa ya en uso es tomar la muestra mediante un diseño clásico en una subregión del espacio de búsqueda que tome la forma de la superficie del diseño de experimentos, creemos que con esta opción se dejan espacios sin cubrir que podrían contener soluciones de alta calidad.

En el Capítulo 4 se mencionó que el método propuesto tiende a inspeccionar en las primeras iteraciones los puntos esquina de la superficie de respuesta. En vista de este comportamiento se buscará una explicación a través de la exploración de la estructura matemática del problema. Con esto se buscará también identificar cualidades que puedan hacer más eficiente el método de optimización.

Como se vio en el capítulo anterior, es de nuestro interés expandir el método propuesto a problemas multiobjetivo, los cuales son muy comunes en la industria. En estos casos es deseable que la selección de los compromisos de cada medida de desempeño se realice de manera objetiva; como se mencionó previamente, una de las

técnicas que nos ayudan a resolver estos problemas es el *análisis envolvente de datos*, la cual reporta un conjunto de soluciones eficientes (frontera eficiente) que representan los mejores compromisos entre todas las medidas de desempeño. La idea preliminar para adaptar el método a este tipo de problemas se muestra esquemáticamente en la **Figura 7.1**. Esencialmente la diferencia con el problema de optimización uniobjetivo será la solución del modelo de optimización, utilizando aquí la técnica de análisis envolvente de datos.

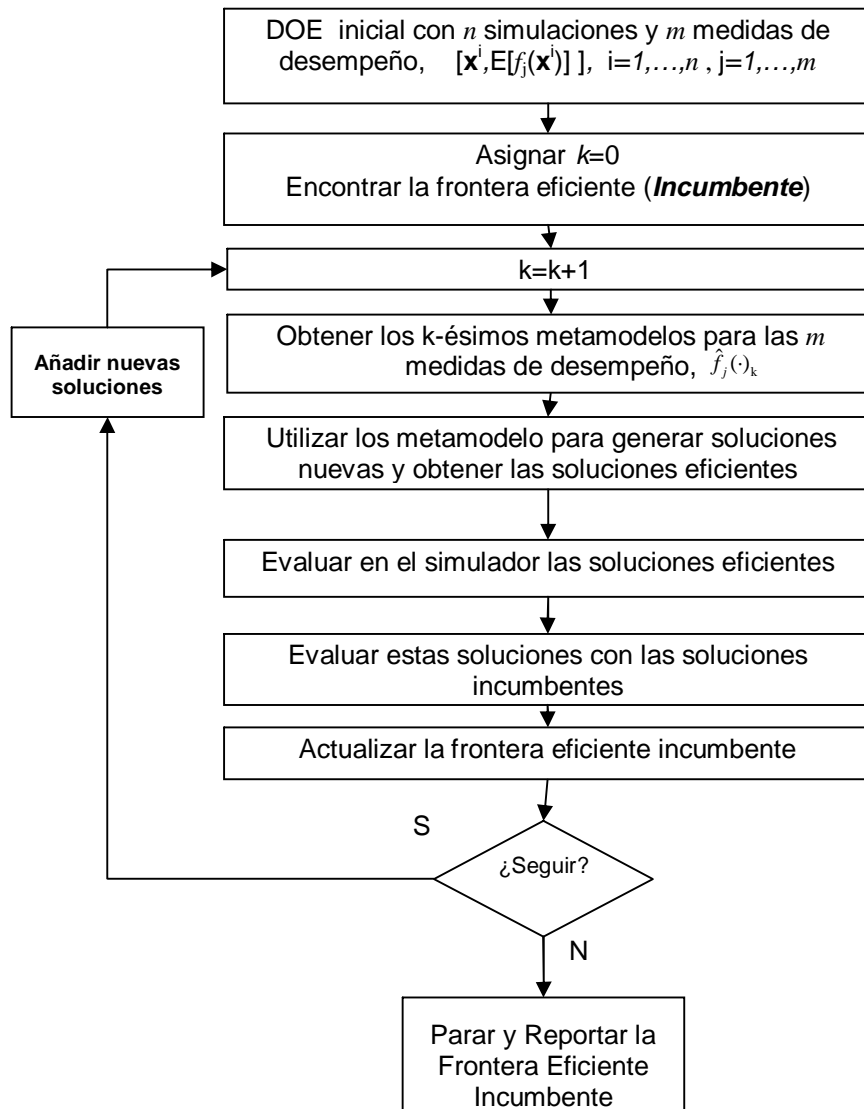


Figura 7.1: Idea preliminar del método de optimización para simulaciones adaptado para resolver problemas multiobjetivo.

Finalmente, y como consecución al problema de obtener la frontera eficiente en problemas multiobjetivo, se planea la construcción de ventanas de proceso multicriterio. Una ventana de proceso multicriterio es una herramienta gráfica en la cual se muestran los niveles a los cuales dos factores (variables controlables) se pueden ajustar para obtener los mejores compromisos entre dos o más medidas de desempeño (Urbano-Vázquez, 2007). La clave en la definición de la ventana de proceso multicriterio estará en poder resolver tal problema, el de optimización de múltiples criterios. Para tal desarrollo se tomará como base el trabajo desarrollado por Urbano-Vázquez (2007).

APÉNDICE A

En este apéndice se muestra, en base a una prueba simple, cuántos puntos debe contener en diseño de experimentos inicial para obtener el primer metamodelo con el mismo resultado que si se tuviera un diseño de experimentos más grande, lo anterior con el objetivo de lograr un ahorro en corridas de simulación.

Como se mencionó en el Capítulo 3, al momento de construir el metamodelo que representa la superficie de respuesta de la salida de un modelo de simulación, se utiliza un modelo de regresión saturado, i.e., utilizando el máximo número de grados de libertad. Para la construcción de este metamodelo es necesario un conjunto de puntos iniciales. En este caso se utilizan los puntos del diseño de experimentos inicialmente, sin embargo, en ocasiones las columnas de la matriz de diseño están correlacionadas, lo que significa que para cada una de estas correlaciones sólo un coeficiente de regresión es estimado. Esto implica que se tienen grados de libertad adicionales disponibles. Con esto en mente se planteó determinar en qué ocasiones, iniciando con un DCC, las columnas de la matriz de diseño son linealmente dependientes y cuántas de ellas lo son. Identificado este número se busca reducir el número de puntos del diseño experimental inicial. Para realizar esta prueba se utilizó un DCC a diferentes niveles, i.e., si la parte factorial del diseño se consideró completa o fraccionada. En la **Tabla A.1** se especifica cada uno de ellos tomando casos de 2 hasta 9 variables controlables, todas en un rango de -1 a 1. En todos los casos se consideró un parámetro $\alpha=0.5$ y un sólo punto centro. La matriz de diseño se formó considerando un modelo de regresión saturado.

La **Tabla A.1** muestra cuántos puntos se generan al utilizar un diseño compósito central a diferentes tamaños y número de variables controlables. Así mismo se muestra el número de columnas de la matriz de diseño que están linealmente correlacionadas. Finalmente se muestra el número reducido de corridas que se necesitan para generar el modelo de regresión. Estos resultados fueron generados con la ayuda de Minitab. Posterior a la tabla se muestra un ejemplo de cómo se identifican las columnas de la matriz que son linealmente dependientes.

No. de Variables Controlables	Tipo de DCC	Número de puntos generados por el DCC	Número de columnas de la matriz de diseño correlacionadas	Número de corridas necesarias
2	Completo	9	0	9
3	Completo	15	1	14
4	Completo	25	5	20
5	Medio	27	0	27
6	Medio	45	11	34
7	Medio	79	38	41
8	Cuarto	81	27	54
9	Cuarto	146	72	74

Tabla A.1. Número de Corridas necesarias para generar el metamodelo inicial, dependiendo el número de variables de decisión.

Dados los resultados de la tabla anterior, con el objetivo de ahorrar tiempo en corridas de simulación, se recomienda utilizar un diseño de experimentos compósito central (al tamaño indicado) para los casos que tengan de 2 a 5 variables de decisión, para el caso de 6 a 9 variables se puede utilizar un diseño compósito central más pequeño o tratar de implementar un diseño D-Óptimo indicando esta cantidad de puntos deseados, basado en un DCC completo.

Identificación de vectores linealmente dependientes

Se dice que n vectores $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ en un espacio vectorial V , son *linealmente dependientes* si existen n escalares c_1, c_2, \dots, c_n no todos cero tales que:

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \dots + c_n\mathbf{v}_n = \mathbf{0}$$

Si los vectores no son linealmente dependientes se dice que son *linealmente independientes* (Grossman, 2002).

Utilizando esta definición veremos si las columnas de la matriz de diseño (**A1**) para el caso de 2 variables con cinco corridas arbitrarias son linealmente dependientes. Con este diseño se puede ajustar la salida del experimento a un modelo lineal e interacciones (utilizando los 4 grados de libertad).

$$\begin{matrix} & \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 \\ \begin{pmatrix} 1 & 2 & -6 & -12 \\ 1 & -1 & 3 & -3 \\ 1 & 0 & 0 & 0 \\ 1 & 3 & -9 & -27 \\ 1 & 1 & -3 & -3 \end{pmatrix} & & & & \end{matrix} \tag{A1}$$

Supongamos que:

$$c_1 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + c_2 \begin{pmatrix} 2 \\ -1 \\ 0 \\ 3 \\ 1 \end{pmatrix} + c_3 \begin{pmatrix} -6 \\ 3 \\ 0 \\ -9 \\ -3 \end{pmatrix} + c_4 \begin{pmatrix} -12 \\ -3 \\ 0 \\ -27 \\ -3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Igual a:

$$\begin{pmatrix} c_1 + 2c_2 - 6c_3 - 12c_4 \\ c_1 - c_2 + 3c_3 - 3c_4 \\ c_1 \\ c_1 + 3c_2 - 9c_3 - 27c_4 \\ c_1 + c_2 - 3c_3 - 3c_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Si este sistema de ecuaciones tiene soluciones no triviales (diferentes de cero) decimos que los vectores son linealmente dependientes. Si escribimos este sistema utilizando una matriz y la reducimos por renglones tenemos:

$$\begin{pmatrix} 1 & 2 & -6 & -12 & 0 \\ 1 & -1 & 3 & -3 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 3 & -9 & -27 & 0 \\ 1 & 1 & -3 & -3 & 0 \end{pmatrix} \text{ es } \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

lo que quiere decir que $c_1=0$, $c_2=3c_3$, $c_3=(1/3)c_2$ y $c_4=0$, por lo cual este conjunto de vectores son linealmente dependientes, dado que por ejemplo:

$$0 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + 3 \begin{pmatrix} 2 \\ -1 \\ 0 \\ 3 \\ 1 \end{pmatrix} + 1 \begin{pmatrix} -6 \\ 3 \\ 0 \\ -9 \\ -3 \end{pmatrix} + 0 \begin{pmatrix} -12 \\ -3 \\ 0 \\ -27 \\ -3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Ahora que sabemos que este conjunto de vectores son linealmente dependientes, deseamos saber cuántos de ellos son independientes, lo cual es el objetivo final de esta prueba. Para saber cuántos vectores de este conjunto son linealmente independientes podemos utilizar la definición de rango de una matriz, la cual es equivalente al número de columnas en las que al menos uno de sus componentes es diferente de cero al hacer una reducción por columnas o por renglones (Grossman, 2002).

$$\begin{pmatrix} 1 & 2 & -6 & -12 \\ 1 & -1 & 3 & -3 \\ 1 & 0 & 0 & 0 \\ 1 & 3 & -9 & -27 \\ 1 & 1 & -3 & -3 \end{pmatrix} \text{ esto es } \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{5}{3} & \frac{5}{3} & 0 & -3 \\ \frac{5}{3} & \frac{1}{3} & 0 & -1 \end{pmatrix}$$

Por lo tanto en este caso tres de los cuatro vectores son linealmente independientes.

Ahora, retomado el objetivo de la prueba para este trabajo que es el construir un metamodelo el cual utilice todos los grados de libertad disponibles, en el caso anterior

sólo se utilizaran 3 de los 4 disponibles, esto después de detectar los vectores que son linealmente dependientes. Por lo tanto, podemos construir el mismo metamodelo teniendo solamente 4 filas en la matriz de diseño lo que equivale a descartar una fila que a su vez corresponde a disminuir una corrida experimental (o simulación). Para determinar cuál(s) es(son) la(s) corrida(s) que hay que descartar se recomienda utilizar un algoritmo que seleccione las mejores combinaciones de acuerdo algún criterio, como por ejemplo minimizar la varianza del metamodelo a ajustar. Detalles sobre estos algoritmos fueron citados en el Capítulo 4.

APÉNDICE B

Este apéndice contiene la expresión matemática de cada una de las funciones de prueba para optimización global utilizadas en la Sección 4.1 del Capítulo 4 para probar el método de optimización para simulaciones propuesto; asimismo contiene los resultados de cada iteración al aplicar el método de optimización a cada una de estas funciones en diferentes casos.

A continuación se muestra cada una de las funciones de prueba para optimización global citadas en el Capítulo 4. La expresión matemática de dichas funciones se hace en base a v variables. Seguido de cada función se indica el rango en el cual varían cada una de las variables. Posteriormente se muestra la solución óptima (\mathbf{x}^*) resultado de la minimización de cada una de las funciones sujetas a los rangos de los valores que puede tomar cada variable, asimismo se muestra el valor de la función objetivo en este punto.

a. Función *Sphere* (v variables)

$$f(\mathbf{x}) = \sum_{q=1}^v x_q^2; \quad -5.12 \leq x_q \leq 5.12; \quad \mathbf{x}^* = \bar{\mathbf{0}}; \quad f(\mathbf{x}^*) = 0$$

b. Función *Rosenbrock* (v variables)

$$f(\mathbf{x}) = \sum_{q=1}^{v-1} [100 * (x_q^2 - x_{q+1})^2 + (x_q - 1)^2]; \quad -2 \leq x_q \leq 5; \quad \mathbf{x}^* = \bar{\mathbf{1}}; \quad f(\mathbf{x}^*) = 0$$

c. Función *Rastrigin* (v variables)

$$f(\mathbf{x}) = 10v + \sum_{q=1}^v (x_q^2 - 10 \cos(2\pi x_q)); \quad -5 \leq x_q \leq 5; \quad \mathbf{x}^* = \bar{\mathbf{0}}; \quad f(\mathbf{x}^*) = 0$$

d. Función *Griewank* (v variables)

$$f(\mathbf{x}) = \sum_{q=1}^v \frac{x_q^2}{100} - \prod_{q=1}^v \cos\left(\frac{x_q}{\sqrt{q}}\right) + 1; \quad -50 \leq x_q \leq 70; \quad \mathbf{x}^* = \bar{\mathbf{0}}; \quad f(\mathbf{x}^*) = 0$$

e. Función *Goldstein-Price* (2 variables)

$$f(\mathbf{x}) = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 13x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) \quad x_1, x_2 \in [-2, 2] \\ * (30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 - 48x_2 - 36x_1x_2 + 27x_2^2)); \quad \mathbf{x}^* = (0, 1); \quad f(\mathbf{x}^*) = 3$$

f. Función *Easom* (2 variables)

$$f(\mathbf{x}) = -\cos(x_1)\cos(x_2)e^{-(x_1-\pi)^2-(x_2-\pi)^2}; \quad x_1, x_2 \in [-100, 100]; \quad \mathbf{x}^* = (\pi, \pi); \quad f(\mathbf{x}^*) = -1$$

g. Función *Schwefel* (v variables)

$$f(\mathbf{x}) = 418.9829v - \sum_{q=1}^v (x_q \operatorname{sen} \sqrt{|x_q|}); \quad -500 \leq x_q \leq 500; \quad \mathbf{x}^* = \bar{\mathbf{1}}; \quad f(\mathbf{x}^*) = 0$$

Las siguientes tablas muestran los resultados de cada iteración al aplicar el método de optimización para simulaciones, a cada uno de los casos mostrados en la Sección 4.1. Todas las tablas están organizadas de la siguiente manera: la primera columna indica la iteración; la segunda muestra el mejor punto encontrado al optimizar el metamodelo de tal iteración; la tercera la evaluación del punto encontrado en el modelo de simulación; la cuarta la solución incumbente hasta esa iteración y, finalmente, la quinta columna contiene el valor de R^2 del metamodelo. Todas las tablas indican a si así mismo el criterio de parada que se cumplió, para todos los casos se utilizó un valor $\varepsilon = 1$ y el máximo número de iteraciones fue fijado arbitrariamente. Cabe mencionar que todos los resultados que se muestran a continuación se encuentran en la escala de $[-1, 1]$.

Las **Tablas B.1 a B.7** muestran de forma iterativa los resultados presentados en la **Tabla 4.1**. El diseño de experimentos inicial fue un diseño compuesto central con parámetro $\alpha = 1$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0,0)	-1	[(0,0), -1]	--
<i>1</i>	(0,0)	-1	[(0,0), -1]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (0,0) ya había sido simulado $R^2 \geq \epsilon (\epsilon = 1)$ 			

Tabla B.1: Función *Sphere* (2 dimensiones), x_1 y $x_2 \in [-5.12, 5.12]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0.00,0.00)	-1.0000	[(0.00,0.00), -1.0000]	--
<i>1</i>	(-1.00,-1.00)	-0.8697	[(0.00,0.00), -1.0000]	0.9945
<i>2</i>	(-0.01,0.14)	-1.0003	[(-0.01,0.14), -1.0003]	0.9946
<i>3</i>	(-1.00,1.00)	-0.9965	[(-0.01,0.14), -1.0003]	0.9947
<i>4</i>	(-0.14,-0.14)	-1.0004	[(-0.14,-0.14), -1.0004]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \epsilon (\epsilon = 1)$ 			

Tabla B.2: Función *Rosenbrock* (2 dimensiones), x_1 y $x_2 \in [-2.00, 5.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0.00,0.00)	-1.0000	[(0.00,0.00), -1.0000]	--
<i>1</i>	(-1.00,-1.00)	0.9048	[(0.00,0.00), -1.0000]	0.0010
<i>2</i>	(1.00,1.00)	0.9048	[(0.00,0.00), -1.0000]	0.0720
<i>3</i>	(1.00,-1.00)	0.9048	[(0.00,0.00), -1.0000]	0.1077
<i>4</i>	(0.00,0.00)	-1.0000	[(0.00,0.00), -1.0000]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (0,0) ya había sido simulado $R^2 \geq \epsilon (\epsilon = 1)$ 			

Tabla B.3: Función *Rastrigin* (2 dimensiones), x_1 y $x_2 \in [-5.00, 5.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0,0)	-1	[(0,0), -1]	--
<i>1</i>	(0,0)	-1	[(0,0), -1]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (0,0) ya había sido simulado $R^2 \geq \epsilon (\epsilon = 1)$ 			

Tabla B.4: Función *Griewank* (2 dimensiones), x_1 y $x_2 \in [-50.00, 70.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R ² del metamodelo
<i>Inicialización DOE</i>	(0.00,0.00)	-1.0000	[(0.00,0.00), -1.0000]	--
1	(0.21,-0.02)	-0.7877	[(0.00,0.00), -1.0000]	0.9710
2	(-1.00,-1.00)	1.8212	[(0.00,0.00), -1.0000]	0.9726
3	(1.00,-1.00)	0.9048	[(0.00,0.00), -1.0000]	0.9730
4	(0.71,0.11)	-1.0007	[(0.71,0.11), -1.0007]	0.9983
5	(-0.10,-0.62)	-1.0047	[(-0.10,-0.62), -1.0047]	0.9996
6	(1.00,1.00)	-0.3203	[(-0.10,-0.62), -1.0047]	0.9999
7	(0.63,1.00)	0.1664	[(-0.10,-0.62), -1.0047]	0.9987
8	(-0.04,-0.49)	-1.0053	[(-0.04,-0.49), -1.0053]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \varepsilon (\varepsilon = 1)$ 			

Tabla B.5: Función *Goldstein-Price* (2 dimensiones), x_1 y $x_2 \in [-2.00, 2.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R ² del metamodelo
<i>Inicialización DOE</i>	(0,0)	-1	[(0,0), -1]	--
1	(0,0)	-1	[(0,0), -1]	0.5000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (0,0) ya había sido simulado 			

Tabla B.6: Función *Easom* (2 dimensiones), x_1 y $x_2 \in [-1.00, 1.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R ² del metamodelo
<i>Inicialización DOE</i>	(0.00,-1.00)	-1.0000	[(0.00,-1.00), -1.0000]	--
1	(-1.00,-1.00)	-2.000	[(-1.00,-1.00), -2.0000]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \varepsilon (\varepsilon = 1)$ 			

Tabla B.7: Función *Schwefel* (2 dimensiones), x_1 y $x_2 \in [-500, 500]$.

De modo similar a las tablas anteriores las **Tablas B.8 a B.14** muestran los resultados de la **Tablas 4.2**, solamente que para estos casos el diseño de experimentos inicial fue un diseño compósito central con parámetro $\alpha = 0.5$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0,0)	-1	[(0,0), -1]	--
<i>1</i>	(0,0)	-1	[(0,0), -1]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (0,0) ya había sido simulado $R^2 \geq \varepsilon (\varepsilon = 1)$ 			

Tabla B.8: Función *Sphere* (2 dimensiones), x_1 y $x_2 \in [-5.12, 5.12]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0.00,0.00)	-1.0000	[(0.00,0.00), -1.0000]	--
<i>1</i>	(-1.00,-0.36)	-0.9627	[(0.00,0.00), -1.0000]	0.9996
<i>2</i>	(-0.11,-1.00)	-0.9728	[(0.00,0.00), -1.0000]	0.9996
<i>3</i>	(-1.00,0.05)	-0.9997	[(0.00,0.00), -1.0000]	0.9996
<i>4</i>	(-0.14,-0.14)	-1.0016	[(-0.14,-0.14), -1.0000]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \varepsilon (\varepsilon = 1)$ 			

Tabla B.9: Función *Rosenbrock* (2 dimensiones), x_1 y $x_2 \in [-2.00, 5.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0.00,0.00)	-1.0000	[(0.00,0.00), -1.0000]	--
<i>1</i>	(0.00,0.00)	-1.0000	[(0.00,0.00), -1.0000]	0.8379
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (0,0) ya había sido simulado 			

Tabla B.10: Función *Rastrigin* (2 dimensiones), x_1 y $x_2 \in [-5.00, 5.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0,0)	-1	[(0,0), -1]	--
<i>1</i>	(0,0)	-1	[(0,0), -1]	1.000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (0,0) ya había sido simulado $R^2 \geq \varepsilon (\varepsilon = 1)$ 			

Tabla B.11: Función *Griewank* (2 dimensiones), x_1 y $x_2 \in [-50.00, 70.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0.00,-0.50)	-1.0000	[(0.00,-0.50), -1.0000]	--
1	(1.00,0.45)	-0.9960	[(0.00,-0.50), -1.0000]	0.9984
2	(-0.41,-0.39)	-0.9987	[(0.00,-0.50), -1.0000]	0.9984
3	(-0.23,-1.00)	-0.8729	[(0.00,-0.50), -1.0000]	0.9985
4	(0.26,0.19)	-0.9979	[(0.00,-0.50), -1.0000]	0.9980
5	(-1.00,-0.54)	-0.9594	[(0.00,-0.50), -1.0000]	0.9988
6	(0.39,-0.84)	-0.9948	[(0.00,-0.50), -1.0000]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \epsilon (\epsilon = 1)$ 			

Tabla B.12: Función *Goldstein-Price* (2 dimensiones), x_1 y $x_2 \in [-2.00, 2.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0,0)	-1	[(0,0), -1]	--
1	(0,0)	-1	[(0,0), -1]	0.1552
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (0,0) ya había sido simulado 			

Tabla B.13: Función *Easom* (2 dimensiones), x_1 y $x_2 \in [-1.00, 1.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(-1.00,-1.00)	-1.0000	[(-1.00,-1.00), -1.0000]	--
1	(-1.00,-1.00)	-1.0000	[(-1.00,-1.00), -1.0000]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (-1,-1) ya había sido simulado $R^2 \geq \epsilon (\epsilon = 1)$ 			

Tabla B.14: Función *Schwefel* (2 dimensiones), x_1 y $x_2 \in [-500, 500]$.

Las **Tablas B.15** y **B.16** muestran los resultados de la **Tablas 4.3**. El diseño de experimentos inicial fue un diseño D-Óptimo con 74 combinaciones iniciales.

Las **Tablas B.17** a **B.19** muestran los resultados presentados en la **Tabla 4.4**. El diseño de experimentos inicial fue un diseño compósito central con parámetro $\alpha = 1$. De igual modo las **Tablas B.20** a la **B.22** muestran los resultados de la **Tablas 4.5**, solamente que para estos casos se utilizó un DCC con parámetro $\alpha = 0.5$.

Iteración	Mejores soluciones del metamodelo (x_1, x_2, \dots, x_9)	Evaluación en la Simulación $E[f(x_1, x_2, \dots, x_9)]$	Solución Incumbente [(x_1, x_2, \dots, x_9), $E[f(x_1, x_2, \dots, x_9)]$]	R ² del metamodelo
<i>Inicialización DOE</i>	(0,0,0,0,0,0,0,-0.5,0) (0,0,0,0,-0.5,0,0,0,0) (0,0,0,0,0,0,0,0,-0.5) (0,0,0,0,0,0.5,0,0,0) (0,0,0,0,0,0,0.5,0,0) (0,0,-0.5,0,0,0,0,0,0) (0,0,0,-0.5,0,0,0,0,0) (-0.5,0,0,0,0,0,0,0,0) (0.5,0,0,0,0,0,0,0,0) (0,0.5,0,0,0,0,0,0,0)	-1.0000	[(0,0,0,0,0,0,0,-0.5,0), -1.0000]	--
<i>1</i>	(0,0,0,0,0,0,0,0,0)	-1.0571	[(0,0,0,0,0,0,0,0,0), -1.0571]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \varepsilon (\varepsilon = 1)$ 			

Tabla B.15: Función *Shpere* (9 dimensiones), x_1 y $x_2 \in [-5.12, 5.12]$.

Iteración	Mejores soluciones del metamodelo (x_1, x_2, \dots, x_9)	Evaluación en la Simulación $E[f(x_1, x_2, \dots, x_9)]$	Solución Incumbente [(x_1, x_2, \dots, x_9), $E[f(x_1, x_2, \dots, x_9)]$]	R ² del metamodelo
<i>Inicialización DOE</i>	(0,0,0,-0.5,0,0,0,0,0)	-1.0000	[(0,0,0,-0.5,0,0,0,0,0), -1.0000]	--
<i>1</i>	(-1,1,0,1,1,0,0,0,0)	-0.1590	[(0,0,0,-0.5,0,0,0,0,0), -1.0000]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \varepsilon (\varepsilon = 1)$ 			

Tabla B.16: Función *Griewank* (9 dimensiones), x_1 y $x_2 \in [50, 70]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente [(x_1, x_2), $E[f(x_1, x_2)]$]	R ² del metamodelo
<i>Inicialización DOE</i>	(0.50,0.50)	-1.0000	[(0.50,0.50), -1.0000]	--
<i>1</i>	(0.33,0.33)	-1.0392	[(0.33,0.33), -1.0392]	1.000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \varepsilon (\varepsilon = 1)$ 			

Tabla B.17: Función *No Lineal 1* (2 dimensiones), x_1 y $x_2 \in [0, 3]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R ² del metamodelo
<i>Inicialización DOE</i>	(-0.50,0.50)	-1.0000	$[(-0.50,0.50), -1.0000]$	--
1	(-0.05,0.27)	-0.9862	$[(-0.50,0.50), -1.0000]$	0.9596
2	(-1.00,1.00)	-1.2195	$[(-1.00,1.00), -1.2195]$	0.9557
3	(0.91,1.00)	0.3590	$[(-1.00,1.00), -1.2195]$	0.9540
4	(-1.00,1.00)	-1.2195	$[(-1.00,1.00), -1.2195]$	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (-1.00,1.00) ya había sido simulado $R^2 \geq \epsilon$ ($\epsilon = 1$) 			

Tabla B.18: Función *No Lineal 2* (2 dimensiones), x_1 y $x_2 \in [-2.00, 5.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R ² del metamodelo
<i>Inicialización DOE</i>	(0.00,0.00)	-1.0000	$[(0.00,0.00), -1.0000]$	--
1	(0.00,0.00)	-1.0000	$[(0.00,0.00), -1.0000]$	0.9908
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (0.00,0.00) ya había sido simulado 			

Tabla B.19: Función *No Lineal 3* (2 dimensiones), x_1 y $x_2 \in [-1.00, 1.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R ² del metamodelo
<i>Inicialización DOE</i>	(0.00,0.00)	-1.0000	$[(0.00,0.00), -1.0000]$	--
1	(0.33,0.33)	-1.0741	$[(0.33,0.33), -1.0741]$	1.000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \epsilon$ ($\epsilon = 1$) 			

Tabla B.20: Función *No Lineal 1* (2 dimensiones), x_1 y $x_2 \in [0, 3]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R ² del metamodelo
<i>Inicialización DOE</i>	(-1.00,1.00)	-1.0000	$[(-1.00,1.00), -1.0000]$	--
1	(-1.00,1.00)	-1.0000	$[(-1.00,1.00), -1.0000]$	0.9986
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (-1.00,1.00) ya había sido simulado 			

Tabla B.21: Función *No Lineal 2* (2 dimensiones), x_1 y $x_2 \in [-2.00, 5.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R ² del metamodelo
<i>Inicialización DOE</i>	(0.00,0.00)	-1.0000	$[(0.00,0.00), -1.0000]$	--
1	(0.00,0.00)	-1.0000	$[(0.00,0.00), -1.0000]$	0.9996
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (0.00,0.00) ya había sido simulado 			

Tabla B.22: Función *No Lineal 3* (2 dimensiones), x_1 y $x_2 \in [-1.00, 1.00]$.

APÉNDICE C

La primera parte de este apéndice muestra los parámetros utilizados para formar cada una de las funciones (RNAs) de prueba utilizadas en la Sección 4.2 del Capítulo 4. La segunda sección muestra los resultados al optimizar cada una de las funciones mediante el método de optimización para simulaciones aquí propuesto.

C.1. Parámetros de RNAs

Como se mencionó en el Capítulo 4, algunas Redes Neuronales Artificiales son conocidas por ser aproximadores universales. De manera general, una RNA multicapa de avance (*Feedforward*) está compuesta de muchas capas formadas por conjuntos de neuronas que poseen características similares (**Figura C.1**). La primera capa se denomina capa de entrada y es donde se recibe la información externa. La última capa es conocida como capa de salida y es donde se obtiene la solución final del problema que se esté tratando. La capa de entrada y la capa de salida están separadas por una o más capas denominadas capas ocultas, estas capas son las encargadas de brindarle flexibilidad a la RNA para aproximar relaciones complejas entre los datos. Las neuronas en las capas adyacentes están completamente conectadas. La **Figura C.1**, muestra una RNA como la utilizada para generar las funciones de prueba mostradas en el Capítulo 4, ésta contiene una capa de entrada, una capa oculta y una capa de salida. La cantidad de neuronas en cada capa varía dependiendo del caso.

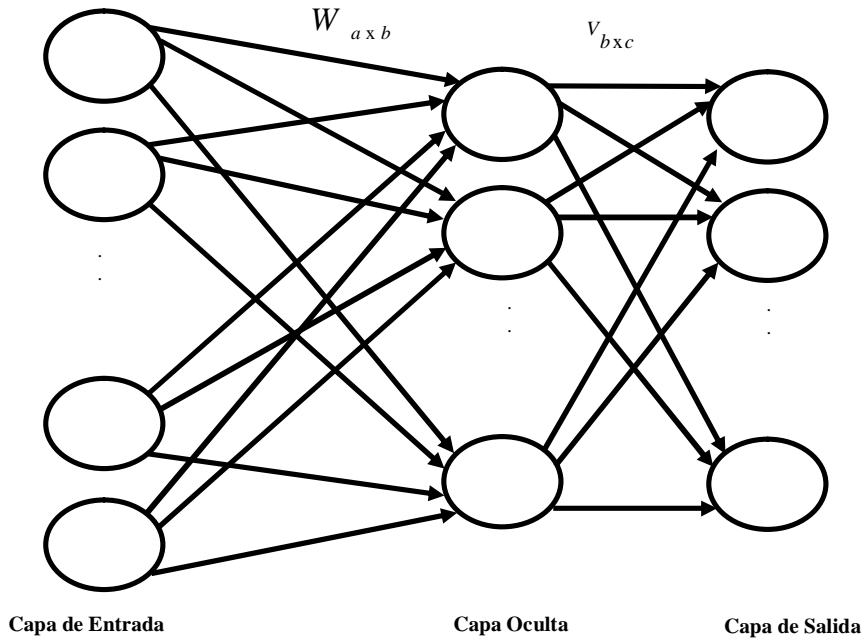


Figura C.1. RNA multicapa de Avance.

La forma matricial de la salida ($f(\cdot)$) de la RNA esta dada por:

$$f(\cdot) = f_2(\mathbf{V}_{b \times c}^T * f_1(\mathbf{W}_{a \times b}^T \mathbf{x}_a + \mathbf{b}_1) + \mathbf{b}_2)$$

Donde f_1 (función tangente hiperbólica, **C.1**) y f_2 (función identidad, **C.2**) son las funciones de transferencia para la capa oculta y la capa de salida respectivamente, \mathbf{x}_a representa el vector de entradas (a), $\mathbf{W}_{a \times b}$ es la matriz de pesos que van de las neuronas de la capa de entrada (a) a las neuronas de la capa oculta (b), $\mathbf{V}_{b \times c}$ es la matriz de pesos que van de las neuronas de la capa de oculta (b) a las neuronas de la capa de salida (c), en el caso de que haya sólo una salida este se reduce a un vector de dimensión c, \mathbf{b}_1 y \mathbf{b}_2 son los vectores de pesos que salen de las neuronas de sesgo de la capa de entrada y la capa oculta respectivamente (Cabrera-Ríos et al., 2005).

$$f_1(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} \tag{C.1}$$

$$f_2(s) = s \quad (\text{C.2})$$

A continuación se muestran las matrices \mathbf{W}_{axb} y \mathbf{V}_{bxc} así como los vectores \mathbf{b}_1 y \mathbf{b}_2 para cada una de las funciones mostradas en la Tabla 4.6.

a) RNA(2,3,1)

$$\mathbf{W}_{\text{axb}} = \begin{pmatrix} -0.51 & -0.19 & 0.85 \\ -0.86 & 0.86 & -0.14 \end{pmatrix} \quad \mathbf{V}_{\text{bxc}} = \begin{pmatrix} 0.90 \\ 0.54 \\ 0.65 \end{pmatrix} \quad \mathbf{b}_1 = \begin{pmatrix} -0.39 \\ -0.72 \\ -0.56 \end{pmatrix} \quad \mathbf{b}_2 = (-0.23)$$

b) RNA(2,3,1)

$$\mathbf{W}_{\text{axb}} = \begin{pmatrix} -0.49 & 0.78 & -0.72 \\ 0.34 & 0.77 & -0.28 \end{pmatrix} \quad \mathbf{V}_{\text{bxc}} = \begin{pmatrix} -0.29 \\ 0.54 \\ 1.00 \end{pmatrix} \quad \mathbf{b}_1 = \begin{pmatrix} 0.98 \\ -0.64 \\ -0.68 \end{pmatrix} \quad \mathbf{b}_2 = (0.58)$$

c) RNA(2,3,1)

$$\mathbf{W}_{\text{axb}} = \begin{pmatrix} -0.99 & 0.91 & 0.79 \\ -0.16 & -0.87 & -0.68 \end{pmatrix} \quad \mathbf{V}_{\text{bxc}} = \begin{pmatrix} 0.47 \\ 0.62 \\ -0.23 \end{pmatrix} \quad \mathbf{b}_1 = \begin{pmatrix} -0.78 \\ -0.89 \\ 0.27 \end{pmatrix} \quad \mathbf{b}_2 = (0.63)$$

d) RNA(2,3,1)

$$\mathbf{W}_{\text{axb}} = \begin{pmatrix} 0.57 & 0.70 & -0.35 \\ 0.73 & -0.39 & -0.26 \end{pmatrix} \quad \mathbf{V}_{\text{bxc}} = \begin{pmatrix} -0.52 \\ -0.30 \\ -0.55 \end{pmatrix} \quad \mathbf{b}_1 = \begin{pmatrix} -0.36 \\ -0.86 \\ 0.60 \end{pmatrix} \quad \mathbf{b}_2 = (-0.96)$$

e) RNA(2,3,1)

$$\mathbf{W}_{\text{axb}} = \begin{pmatrix} -0.15 & -0.22 & 0.77 \\ -0.18 & 0.25 & -0.93 \end{pmatrix} \quad \mathbf{V}_{\text{bxc}} = \begin{pmatrix} 0.20 \\ 0.85 \\ 0.73 \end{pmatrix} \quad \mathbf{b}_1 = \begin{pmatrix} 0.74 \\ -0.83 \\ 0.12 \end{pmatrix} \quad \mathbf{b}_2 = (0.08)$$

f) RNA(3,3,1)

$$\mathbf{W}_{\text{axb}} = \begin{pmatrix} 0.76 & 0.66 & -0.75 \\ 0.24 & 0.00 & 0.98 \\ 0.47 & 1.00 & 0.22 \end{pmatrix} \quad \mathbf{V}_{\text{bxc}} = \begin{pmatrix} -0.37 \\ 0.39 \\ 0.92 \end{pmatrix} \quad \mathbf{b}_1 = \begin{pmatrix} -0.11 \\ -0.57 \\ -0.19 \end{pmatrix} \quad \mathbf{b}_2 = (-0.98)$$

g) RNA(3,3,1)

$$\mathbf{W}_{\text{axb}} = \begin{pmatrix} 0.02 & -0.09 & 0.56 \\ -0.63 & -0.02 & -0.17 \\ 0.05 & 0.56 & 0.51 \end{pmatrix} \quad \mathbf{V}_{\text{bxc}} = \begin{pmatrix} 0.27 \\ -0.66 \\ 0.30 \end{pmatrix} \quad \mathbf{b}_1 = \begin{pmatrix} -0.68 \\ 0.07 \\ 0.11 \end{pmatrix} \quad \mathbf{b}_2 = (-0.98)$$

C.2. Resultados de Optimización

Las **Tablas C.1 a C.7** muestran los resultados de aplicar el método de optimización para simulaciones a cada una de las funciones mostradas en la **Tabla 4.6** del Capítulo 4, de forma iterativa. Estas tablas muestran la mejor solución encontrada por iteración, la correspondiente evaluación en el modelo de simulación, así como la solución incumbente y el valor R^2 del metamodelo por iteración.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(-0.50,0.50)	-1.0000	[(-0.50,0.50), -1.0000]	--
1	(-0.49,1.00)	-1.0129	[(-0.49,1.00), -1.0129]	0.9784
2	(1.00,-1.00)	1.4258	[(-0.49,1.00), -1.0129]	0.9821
3	(1.00,1.00)	0.6516	[(-0.49,1.00), -1.0129]	0.9874
4	(-0.40,1.00)	-1.0129	[(-0.49,1.00), -1.0129]	0.9935
5	(-0.23,0.74)	-1.0000	[(-0.49,1.00), -1.0129]	0.9995
6	(-0.29,0.42)	-0.9871	[(-0.49,1.00), -1.0129]	0.9957
7	(-1.00,1.00)	-1.0129	[(-0.49,1.00), -1.0129]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \varepsilon (\varepsilon = 1)$ 			

Tabla C.1. Función *a) RNA(2,3,1)*, donde $x_1 \in [-5.00, 5.00]$ y $x_2 \in [-5.00, 7.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0.00,0.00)	-1.0000	[(0.00,0.00), -1.0000]	--
1	(0.42,-0.21)	-0.3846	[(0.00,0.00), -1.0000]	0.9738
2	(1.00,-1.00)	-1.1197	[(1.00,-1.00), -1.1197]	0.9745
3	(1.00,1.00)	-0.5556	[(1.00,-1.00), -1.1197]	0.9787
4	(1.00,-1.00)	-1.1197	[(1.00,-1.00), -1.1197]	0.9669
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (1.00,-1.00) ya había sido simulado 			

Tabla C.2. Función *b*) $RNA(2,3,1)$, donde $x_1 \in [-5.00, 5.00]$ y $x_2 \in [-5.00, 7.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0.50,0.50)	-1.0000	[(0.50,0.50), -1.0000]	--
1	(0.50,0.59)	-0.9265	[(0.50,0.50), -1.0000]	0.9745
2	(-1.00,-1.00)	0.4853	[(0.50,0.50), -1.0000]	0.9801
3	(-1.00,1.00)	0.5588	[(0.50,0.50), -1.0000]	0.9810
4	(0.32,0.30)	-1.0000	[(0.50,0.50), -1.0000]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \epsilon (\epsilon = 1)$ 			

Tabla C.3. Función *c*) $RNA(2,3,1)$, donde $x_1 \in [-5.00, 5.00]$ y $x_2 \in [-5.00, 7.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0.50,0.50)	-1.0000	[(0.50,0.50), -1.0000]	--
1	(1.00,-1.00)	0.3016	[(0.50,0.50), -1.0000]	0.9984
2	(-1.00,1.00)	-1.3810	[(-1.00,1.00), -1.3810]	0.9984
3	(-1.00,-1.00)	0.9683	[(-1.00,1.00), -1.3810]	0.9058
4	(-1.00,1.00)	-1.3810	[(-1.00,1.00), -1.3810]	0.9219
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \epsilon (\epsilon = 1)$ 			

Tabla C.4. Función *d*) $RNA(2,3,1)$, donde $x_1 \in [-5.00, 5.00]$ y $x_2 \in [-5.00, 7.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2)	Evaluación en la Simulación $E[f(x_1, x_2)]$	Solución Incumbente $[(x_1, x_2), E[f(x_1, x_2)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0.50,0.50)	-1.0000	[(0.50,0.50), -1.0000]	--
1	(0.60,0.62)	-1.0000	[(0.50,0.50), -1.0000]	0.9984
2	(-1.00,-1.00)	0.9825	[(0.50,0.50), -1.0000]	0.9910
3	(-1.00,1.00)	1.1053	[(0.50,0.50), -1.0000]	0.9910
4	(1.00,0.96)	-1.0351	[(1.00,0.96), -1.0351]	0.9990
5	(0.66,0.14)	0.5439	[(1.00,0.96), -1.0351]	1.0000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> $R^2 \geq \epsilon (\epsilon = 1)$ 			

Tabla C.5. Función *e*) $RNA(2,3,1)$, donde $x_1 \in [-5.00, 5.00]$ y $x_2 \in [-5.00, 7.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2, x_3)	Evaluación en la Simulación $E[f(x_1, x_2, x_3)]$	Solución Incumbente $[(x_1, x_2, x_3), E[f(x_1, x_2, x_3)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(0.50,-0.50,-0.50)	-1.0000	[(0.50,-0.50,-0.50),-1.0000]	--
1	(1.00,1.00,-1.00)	-0.6598	[(0.50,-0.50,-0.50),-1.0000]	0.9515
2	(1.00,1.00,1.00)	0.8257	[(0.50,-0.50,-0.50),-1.0000]	0.9540
3	(1.00,-1.00,-0.83)	-1.1494	[(1.00,-1.00,-0.83),-1.1494]	0.9584
4	(1.00,-0.54,-0.96)	-1.2158	[(1.00,-0.54,-0.96),-1.2158]	0.9639
5	(-0.27,1.00,-1.00)	0.8008	[(1.00,-0.54,-0.96),-1.2158]	0.9681
6	(-0.14,-1.00,-1.00)	-0.7178	[(1.00,-0.54,-0.96),-1.2158]	0.9705
7	(-1.00,-1.00,1.00)	0.4606	[(1.00,-0.54,-0.96),-1.2158]	0.9997
8	(-1.00,-1.00,1.00)	0.4606	[(1.00,-0.54,-0.96),-1.2158]	1.000
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (-1.00,-1.00,1.00) ya había sido simulado $R^2 \geq \varepsilon$ ($\varepsilon = 1$) 			

Tabla C.6. Función f) $RNA(3,3,1)$, donde x_1, x_2 y $x_3 \in [-5.00, 5.00]$.

Iteración	Mejor solución del metamodelo (x_1, x_2, x_3)	Evaluación en la Simulación $E[f(x_1, x_2, x_3)]$	Solución Incumbente $[(x_1, x_2, x_3), E[f(x_1, x_2, x_3)]]$	R^2 del metamodelo
<i>Inicialización DOE</i>	(-0.50,0.50,0.50)	-1.0000	[(-0.50,0.50,0.50),-1.0000]	--
1	(-1.00,-1.00,0.66)	-0.4051	[(-0.50,0.50,0.50),-1.0000]	0.9819
2	(-1.00,1.00,1.00)	-1.1538	[(-1.00,1.00,1.00),-1.1538]	0.9824
3	(1.00,-1.00,1.00)	-0.0462	[(-1.00,1.00,1.00),-1.1538]	0.9861
4	(-1.00,0.02,1.00)	-0.8359	[(-1.00,1.00,1.00),-1.1538]	0.9862
5	(1.00,0.10,1.00)	-0.4872	[(-1.00,1.00,1.00),-1.1538]	0.9873
6	(0.51,1.00,1.00)	-0.6103	[(-1.00,1.00,1.00),-1.1538]	0.9875
7	(-1.00,0.55,0.77)	-1.1641	[(-1.00,0.55,0.77),-1.1641]	0.9932
8	(-0.22,0.95,1.00)	-0.6615	[(-1.00,0.55,0.77),-1.1641]	0.9979
9	(-0.86,1.00,0.66)	-1.1846	[(-0.86,1.00,0.66),-1.1846]	0.9995
10	(-0.86,1.00,0.66)	-1.1846	[(-0.86,1.00,0.66),-1.1846]	0.9899
<i>Criterio de parada que fue cumplido</i>	<ul style="list-style-type: none"> El punto (-0.86,1.00,0.66) ya había sido simulado 			

Tabla C.7. Función g) $RNA(3,3,1)$, donde x_1, x_2 y $x_3 \in [-5.00, 5.00]$.

APÉNDICE D

Este apéndice contiene los puntos del diseño de experimentos inicial utilizado al aplicar el método de optimización de simulaciones al modelo que simula el funcionamiento de una línea de embotellado, dicho modelo se presentó en el apartado 4.3.2 del Capítulo 4. Adicionalmente se presenta el valor de la evaluación de estos puntos en el modelo de simulación.

Corrida	x_1 (pies/min)	x_2 (pies/min)	x_3 (pies/min)	x_4 (pies/min)	x_5 (pies/min)	x_6 (pies/min)	x_7 (pies/min)	x_8 (pies/min)	x_9 (pies/min)	$E[f(x_1, x_2, \dots, x_9)]$ (cajas producidas en 999min)
1	0.50	0.50	0.50	0.50	0.50	1.50	1.50	1.50	1.50	1253
2	0.50	0.50	0.50	0.50	1.50	0.50	0.50	0.50	0.50	945
3	0.50	0.50	0.50	1.50	0.50	0.50	0.50	0.50	1.50	957
4	0.50	0.50	0.50	1.50	1.50	1.50	1.50	1.50	0.50	968
5	0.50	0.50	1.50	0.50	0.50	0.50	0.50	1.50	0.50	956
6	0.50	0.50	1.50	0.50	1.50	1.50	1.50	0.50	1.50	963
7	0.50	0.50	1.50	1.50	0.50	1.50	1.50	0.50	0.50	959
8	0.50	0.50	1.50	1.50	1.50	0.50	0.50	1.50	1.50	1061
9	0.50	1.50	0.50	0.50	0.50	0.50	1.50	0.50	0.50	960
10	0.50	1.50	0.50	0.50	1.50	1.50	0.50	1.50	1.50	1056
11	0.50	1.50	0.50	1.50	0.50	1.50	0.50	1.50	0.50	961
12	0.50	1.50	0.50	1.50	1.50	0.50	1.50	0.50	1.50	966
13	0.50	1.50	1.50	0.50	0.50	1.50	0.50	0.50	1.50	957
14	0.50	1.50	1.50	0.50	1.50	0.50	1.50	1.50	0.50	959
15	0.50	1.50	1.50	1.50	0.50	0.50	1.50	1.50	1.50	1245
16	0.50	1.50	1.50	1.50	1.50	1.50	0.50	0.50	0.50	958
17	1.50	0.50	0.50	0.50	0.50	1.50	0.50	0.50	0.50	952
18	1.50	0.50	0.50	0.50	1.50	0.50	1.50	1.50	1.50	1275
19	1.50	0.50	0.50	1.50	0.50	0.50	1.50	1.50	0.50	962
20	1.50	0.50	0.50	1.50	1.50	1.50	0.50	0.50	1.50	960
21	1.50	0.50	1.50	0.50	0.50	0.50	1.50	0.50	1.50	962
22	1.50	0.50	1.50	0.50	1.50	1.50	0.50	1.50	0.50	959

Tabla D.1: Diseño compósito central con 51 corridas, correspondientes al modelo de simulación de una línea de embotellado. Los datos se encuentran en su escala natural.

APÉNDICE E

En este apéndice se presenta la definición de tiempo de ciclo de un proceso de molde por inyección de termoplásticos.

El tiempo de ciclo en un proceso de molde por inyección de termoplásticos inicia cuando la máquina inyectora cierra el molde, seguido por la inyección del polímero dentro de la cavidad del molde, es aquí cuando la cavidad se llena y una presión de sostenimiento es mantenida para compensar el encogimiento del material, de ahí viene la fase de enfriamiento y cuando la parte está lo suficientemente fría el molde abre y la parte es expulsada. Entonces el tiempo de ciclo se puede calcular usando $t_{ciclo} = t_{cerrado} + t_{llenado} + t_{enfriamiento} + t_{expulsión}$, el tiempo de ciclo esta dominado por el tiempo de enfriamiento de la parte dentro de la cavidad del molde, el tiempo que se requiere para cerrar el molde y el tiempo de expulsión, pueden ser fracciones de segundo o pocos segundos dependiendo el tamaño del molde y la máquina. El tiempo que se requiere para llenar el molde depende principalmente del tamaño del mismo.

El tiempo de enfriamiento de un placa rectangular, como la mostrada en el Capítulo 6 (**Figura 6.1**), de grosor h puede estimarse mediante la **Ecuación E.1**, donde el parámetro α representa la difusividad térmica, T_M corresponde a la temperatura del polímero, T_W representa la temperatura del molde y T_D es la temperatura promedio de la parte al momento de ser expulsada (Osswald, 1998).

$$t_{enfriamiento} = \frac{h^2}{\pi\alpha} \ln \left(\frac{8}{\pi^2} \frac{T_M - T_W}{T_D - T_W} \right) \quad (\text{E.1})$$

ÍNDICE DE TABLAS

Tabla 4.1: Solución a funciones de prueba, en un espacio de 2 dimensiones, utilizando un DCC con $\alpha = 1$	33
Tabla 4.2: Solución a funciones de prueba, en un espacio de 2 dimensiones, utilizando un DCC con $\alpha = 0.5$	34
Tabla 4.3: Solución a funciones de prueba, en un espacio de 9 dimensiones, utilizando un diseño D-Óptimo con 74 combinaciones	34
Tabla 4.4: Solución a funciones no lineales en un espacio de 2 dimensiones, utilizando un DCC con $\alpha = 1$	34
Tabla 4.5: Solución a funciones no lineales en un espacio de 2 dimensiones, utilizando un DCC con $\alpha = 0.5$	34
Tabla 4.6: Solución a diferentes RNAs utilizadas para prueba	38
Tabla 4.7: Prueba Kruskal-Wallis sobre el promedio de las salidas de las replicas	41
Tabla 4.8: $E[f(\mathbf{x}^i)]$, $i=1, \dots, 9$	42
Tabla 4.9: Evaluación de criterios de terminación	44
Tabla 4.10: Evaluación de criterios de parada	46
Tabla 4.11: Última evaluación de los criterios de parada (iteración 4)	46
Tabla 4.12: Resultados de cada iteración para el modelo que simula una sala de hospital, iniciando con un DCC ($\alpha = 0.50$)	48
Tabla 4.13: Número esperado de pacientes a atender en un período de 5 días, variando la capacidad máxima de dos filas de espera	49
Tabla 4.14: Resultados de cada iteración para la simulación de la línea de embotellado	52
Tabla 4.15: $[\mathbf{x}^i, E[TC(\mathbf{x}^i)]]$ $i=1,2,\dots,9$, modelo de carátula de teléfono celular	55
Tabla 4.16: Resultados de cada iteración del método de optimización para la simulación del flujo de un polímero dentro del molde de una carátula celular	55
Tabla 5.1: Salida del modelo de simulación considerando el promedio de un conjunto de réplicas	64

Tabla 5.2: Prueba Kruskal-Wallis sobre el promedio de la cantidad de piezas producidas considerando diferentes números de réplicas	65
Tabla 5.3: Prueba Kruskal-Wallis sobre el rango del IC para la cantidad de piezas producidas	65
Tabla 5.4: Prueba Kruskal-Wallis sobre la cantidad de ganchos que salen vacíos en un día laboral	65
Tabla 5.5: Diseño de experimentos inicial con su respectiva evaluación en el modelo de simulación. Los datos se encuentran en su escala natural	70-71
Tabla 5.6: Evaluación de criterios de terminación, optimización línea de pintura	73
Tabla 5.7: Evaluación de criterios de parada, optimización línea de pintura	75
Tabla 5.8: Soluciones encontradas utilizando OptQuest, optimización línea de pintura	76
Tabla 6.1: Puntos del diseño de experimentos inicial y evaluación en el modelo de simulación (encogimiento de los lados AB, BC, CD, DA)	83
Tabla 6.2: Resultados de cada iteración del método de optimización para la simulación del flujo de un polímero dentro del molde de una placa rectangular, minimizando el encogimiento de la parte	83
Tabla 6.3: Resultados al aplicar el método de optimización para la simulación de la placa rectangular, utilizando el polímero <i>Dow Chemical USA Styron 615 AP</i> , evaluando diferentes medidas de desempeño a minimizar	85
Tabla 6.4: Resultados al aplicar el método de optimización para la simulación de la placa rectangular, utilizando el polímero <i>Dow Chemical USA LDPE 4012</i> , evaluando diferentes medidas de desempeño a minimizar	85
Tabla A.1. Número de Corridas necesarias para generar el metamodelo inicial, dependiendo el número de variables de decisión	96
Tabla B.1. Función <i>Sphere</i> (2 dimensiones), x_1 y $x_2 \in [-5.12, 5.12]$	102
Tabla B.2. Función <i>Rosenbrock</i> (2 dimensiones), x_1 y $x_2 \in [-2.00, 5.00]$	102
Tabla B.3. Función <i>Rastrigin</i> (2 dimensiones), x_1 y $x_2 \in [-5.00, 5.00]$	102
Tabla B.4. Función <i>Griewank</i> (2 dimensiones), x_1 y $x_2 \in [-50.00, 70.00]$	102
Tabla B.5. Función <i>Goldstein-Price</i> (2 dimensiones), x_1 y $x_2 \in [-2.00, 2.00]$	103
Tabla B.6. Función <i>Easom</i> (2 dimensiones), x_1 y $x_2 \in [-1.00, 1.00]$	103
Tabla B.7. Función <i>Schweffel</i> (2 dimensiones), x_1 y $x_2 \in [-500, 500]$	103
Tabla B.8. Función <i>Sphere</i> (2 dimensiones), x_1 y $x_2 \in [-5.12, 5.12]$	104
Tabla B.9. Función <i>Rosenbrock</i> (2 dimensiones), x_1 y $x_2 \in [-2.00, 5.00]$	104

Tabla B.10. Función <i>Rastrigin</i> (2 dimensiones), x_1 y $x_2 \in [-5.00, 5.00]$	104
Tabla B.11. Función <i>Griewank</i> (2 dimensiones), x_1 y $x_2 \in [-50.00, 70.00]$	104
Tabla B.12. Función <i>Goldstein-Price</i> (2 dimensiones), x_1 y $x_2 \in [-2.00, 2.00]$	105
Tabla B.13. Función <i>Easom</i> (2 dimensiones), x_1 y $x_2 \in [-1.00, 1.00]$	105
Tabla B.14. Función <i>Schwefel</i> (2 dimensiones), x_1 y $x_2 \in [-500, 500]$	105
Tabla B.15. Función <i>Shpere</i> (9 dimensiones), x_1 y $x_2 \in [-5.12, 5.12]$	106
Tabla B.16. Función <i>Griewank</i> (9 dimensiones), x_1 y $x_2 \in [50, 70]$	106
Tabla B.17. Función <i>No Lineal 1</i> (2 dimensiones), x_1 y $x_2 \in [0, 3]$	106
Tabla B.18. Función <i>No Lineal 2</i> (2 dimensiones), x_1 y $x_2 \in [-2.00, 5.00]$	107
Tabla B.19. Función <i>No Lineal 3</i> (2 dimensiones), x_1 y $x_2 \in [-1.00, 1.00]$	107
Tabla B.20. Función <i>No Lineal 1</i> (2 dimensiones), x_1 y $x_2 \in [0, 3]$	107
Tabla B.21. Función <i>No Lineal 2</i> (2 dimensiones), x_1 y $x_2 \in [-2.00, 5.00]$	107
Tabla B.22. Función <i>No Lineal 3</i> (2 dimensiones), x_1 y $x_2 \in [-1.00, 1.00]$	107
Tabla C.1. Función <i>l) RNA(2,3,1)</i> , donde $x_1 \in [-5.00, 5.00]$ y $x_2 \in [-5.00, 7.00]$	111
Tabla C.2. Función <i>b) RNA(2,3,1)</i> , donde $x_1 \in [-5.00, 5.00]$ y $x_2 \in [-5.00, 7.00]$	112
Tabla C.3. Función <i>c) RNA(2,3,1)</i> , donde $x_1 \in [-5.00, 5.00]$ y $x_2 \in [-5.00, 7.00]$	112
Tabla C.4. Función <i>d) RNA(2,3,1)</i> , donde $x_1 \in [-5.00, 5.00]$ y $x_2 \in [-5.00, 7.00]$	112
Tabla C.5. Función <i>e) RNA(2,3,1)</i> , donde $x_1 \in [-5.00, 5.00]$ y $x_2 \in [-5.00, 7.00]$	112
Tabla C.6. Función <i>f) RNA(3,3,1)</i> , donde x_1, x_2 y $x_3 \in [-5.00, 5.00]$	113
Tabla C.7. Función <i>g) RNA(3,3,1)</i> , donde x_1, x_2 y $x_3 \in [-5.00, 5.00]$	113
Tabla D.1. Diseño compósito central con 51 corridas, correspondientes al modelo de simulación de una línea de embotellado. Los datos se encuentran en su escala natural	114-115

ÍNDICE DE FIGURAS

Figura 2.1: Diagrama de Optimización aplicada a la Simulación	7
Figura 2.2: Modelo general de un proceso o sistema	11
Figura 2.3: Diseño Compósito Central para $K=2$	13
Figura 2.4: Diseño Compósito Central para $K=3$	13
Figura 2.5: a) Función convexa y b) Función cóncava	19
Figura 2.6: Función no convexa	19
Figura 2.7: Función discreta	20
Figura 2.8: Función no lineal no convexa, con diferentes puntos estacionarios	20
Figura 3.1: Método Propuesto	24
Figura 4.1. Funciones de prueba para optimización global irrestrictas en 2 dimensiones	31
Figura 4.2. Funciones no lineales polinomiales en 2 dimensiones utilizadas para validar el método	32
Figura 4.3: RNAs con dos neuronas en la capa de entrada utilizadas para validar el método de optimización	37
Figura 4.4: Modelo que simula una sala de hospital	40
Figura 4.5: Diseño de Experimentos Compósito Central (\mathbf{x}^i , $i=1,2,\dots,9$)	42
Figura 4.6: Primer metamodelo	43
Figura 4.7: Segundo metamodelo	45
Figura 4.8: Soluciones encontradas por el método propuesto	47
Figura 4.9: Puntos del diseño de experimentos inicial (triángulos) y soluciones encontradas en cada iteración por el método propuesto (círculos)	48
Figura 4.10: Número esperado de pacientes a atender en un periodo de 5 días	49
Figura 4.11: Modelo se simulación de una línea de embotellado	51
Figura 4.12: Carátula del teléfono celular	54

Figura 4.13: x^i , $i = 1,2,\dots,9$ obtenidos con el Diseño de Experimentos Compósito Central	55
Figura 4.14: Gráfica de contorno para Tiempo de Ciclo en función de temperatura del molde y presión de inyección en máquina. Los puntos en triángulo representan los puntos del DOE inicial y los círculos son las combinaciones encontradas mediante los metamodelos	56
Figura 5.1: Diagrama de flujo de la línea de pintura a simular	59
Figura 5.2: Modelo de simulación de la línea de puntado de chasis	63
Figura 6.1: Mallado de la placa rectangular con un conjunto de puertos de inyección .	81
Figura 6.2: Gráfica de contorno para la Suma de Encogimientos en función de la temperatura del polímero y máxima presión de empaçado	83
Figura 6.3: Solución en términos de las medidas de desempeño para todos los casos mostrados en la Tabla 6.3	86
Figura 6.4: Solución en términos de las medidas de desempeño para todos los casos mostrados en la Tabla 6.4	86
Figura 7.1: Idea preliminar del método de optimización para simulaciones adaptado para resolver problemas multiobjetivo	93
Figura C.1. RNA multicapa de Avance	109

BIBLIOGRAFÍA

Altıparmak, F., Dengiz, B. y Bulgak, A. (2002). “Optimization of buffer sizes in assembly system using intelligent techniques”. *Proceedings of the 2002 Winter Simulation Conference*, pp. 1157-1162.

Andradóttir, S. (1998). “A review of Simulation Optimization techniques”. *Proceedings of the 1998 Winter Simulation Conference*. pp. 151-158.

April, J., Glover, F., Kelly, J.P. y Laguna, M. (2003). “Practical introduction to Simulation Optimization”. *Proceedings of the 2003 Winter Simulation Conference*, pp. 71-78.

April, J., Better, M., Glover, F. y Kelly, J. (2004). “New Advances and applications for marrying Simulation and Optimization”. *Proceedings of the 2004 Winter Simulation Conference*.

Azadivar, F. (1999). “Simulation Optimization Methodologies”. *Proceedings of the 1999 Winter Simulation Conference*, pp. 93-100.

Barrios-Garcia, J., Carrillo, M., Gil, M.C., González, C. y Pestano, C. (2005). *Análisis de funciones en economía y empresa: un enfoque interdisciplinario*. Ediciones Díaz de Santos.

Barton, R.R. (1998). “Simulation Metamodels”. *Proceedings of the 1998 Winter Simulation Conference*, pp. 167-174.

Bernstein, D.S. (2005). *Matrix mathematics: theory, facts, and formulas with application to linear systems theory*. Princeton University Press.

Buot, G., Henríquez, F., Baesler F. y Ramírez, M. (2001). “Optimización en simulación vía Inteligencia Artificial: Aplicación a un problema de manejo de material”. *Teoría Universidad de Bío-Bío*, Vol.10, pp. 25-32.

Cabrera-Rios, M. (2002). *Multiple Criteria Optimization Studies in Reactive In-Mold Coating*. Tesis de Doctorado, The Ohio State University, Ohio, USA.

Cabrera-Ríos, M, Castro, J.M. y Mount-Campbell, C.A. (2002a). “Multiple Quality Criteria Optimization In In-Mold Coating (IMC) With A Data Envelopment Analysis Approach”. *Journal of Polymer Engineering*, Vol.22, No.5, pp. 305-340.

Cabrera-Ríos, M., Mount-Campbell, C.A. y Irani, S.A. (2002b). “An Approach to the Design of a Manufacturing Cell under Economic Considerations”. *International Journal of Production Economics*, Vol.78, No.3, pp. 223-237.

Cabrera-Ríos, M., Zuyev, K.S., Chen, X., Castro, J.M. y Straus, E.J. (2002c). “Optimizing Injection Gate Location and Cycle Time for the In-Mold Coating (IMC) Process”. *Polymer Composites*, Vol.23, No. 5, pp.723-738.

Cabrera-Ríos, M. y Castro, J.M. (2003). “The Balance Between Durability, Reliability, and Affordability in Structural Composites Manufacturing: Preliminary Results”, Technical Paper 2003-01-0459, Reliability and Robust Design in Automotive Engineering, 2003 SAE World Congress Book SP-1736.

Cabrera-Ríos, M., Castro, J.M. y Mount-Campbell, C.A. (2004). “Multiple Quality Criteria Optimization In Reactive In-Mold Coating With A Data Envelopment Analysis Approach II: A Case With More Than Three Performance Measures”. *Journal of Polymer Engineering*, Vol.24, No.4, pp. 435-450.

Cabrera-Ríos, M., Zuyev, K.S., Chen, X. y Castro, J.M. (2005). “Optimizing injection gate location and cycle time for the In-Mold Coating (IMC) process”, *Polymer Composites*, Vol.23, No.5, pp. 723-737.

Castro, C.E., Cabrera-Ríos, M., Lilly, B., Castro, J.M. y Mount-Campbell, C.A. (2003). “Identifying The Best Compromises Between Multiple Performance Measures In Injection Molding (IM) Using Data Envelopment Analysis (DEA)”. *Journal of Integrated Design & Process Science*, No.7, pp. 77-87.

Castro, J.M., Cabrera-Ríos, M. y Mount-Campbell, C.A. (2004). “Modelling and Simulation in Reactive Polymer Processing”. *Modelling and Simulation in Materials Science and Engineering*, Vol.12, pp. S121-S149.

Castro, C.E., Cabrera-Ríos, M., Lilly, B. y Castro, J.M. (2005). “Simultaneous Optimization of Mold Design and Processing Conditions in Injection Molding”. *Journal of Polymer Engineering* , Vol.25, No.6, pp.459-486.

Castro, C.E., Cabrera-Ríos, M., Lilly, B. y Castro, J.M. (2006). “Optimization and Analysis of Variability in Injection Molding”. Aceptado para publicación en *Journal of Polymer Engineering and Science*.

Charnes, A., Cooper, W.W. y Rhodes, E. (1978). “Measuring the efficiency of decision making units”. *European Journal of Operation Research*, Vol. 2, No.6, pp. 429-444.

Charnes, A., Cooper, W.W., Lewin, A.Y. y Seiford, L.M. (1993). *Data Envelopment Analysis: Theory, Methodology, and Applications*. Kluwer Academic Publishers.

Cheng, R.C.H. y Currie, C.S.M. (2004). “Optimization by Simulation Metamodeling Methods”. *Proceedings of the 2004 Winter Simulation Conference*, pp. 485-490.

Concanno, K., Elder, M., Hunter, K., Tremble, J. y Tse, S. 2004. *Simulation Modeling with SIMUL8*. VISULA8 Simulation Solutions.

Fu, M.C. (1994). “A tutorial review of techniques for Simulation Optimization”. *Proceedings of the 1994 Winter Simulation Conference*, pp. 149-156.

Fu, M.C. (2001). “Simulation Optimization”. *Proceedings of the 2001 Winter Simulation Conference*, pp. 53-61.

Fu, M.C., Glover, F. y April, J. (2005). “Simulation Optimization: A review, new developments, and application”. *Proceedings of the 2005 Winter Simulation Conference*, pp. 83-95.

Fylstra, D., Lasdon, L., Watson, J. y Waren, A. (1998). “Design and Use of the Microsoft Excel Solver”. *Interfaces*, No.28, pp. 29-55.

González-Espinosa, C.P. (2005). *Simulación de la deflexión de un molde de inyección por el método de elementos finitos*. Tesis de maestría, Universidad Autónoma de Nuevo León, Nuevo León, México.

Grossman, S.I. (2002). *Álgebra lineal*. 5ª ed., Mc Graw Hill.

Gutiérrez-Pulido, H. y De la Vara-Salazar, R. (2004). *Análisis y diseño de experimentos*. 2ª ed, Mc Graw Hill.

Hagan, M.T. y Demut, H.B. (1995). *Neural Network Design*. 1ª ed., PWS Publishing Company.

Jeff-Wu, C.F. y Hamada, M. (2000). *Experiments: Planning, Analysis, and Parameter Design Optimization*. John Wiley & Sons, Inc.

Kelton, W.D., Sadowski, R.P. y Sandwiski, D.A. (2002). *Simulation with Arena*. 2ª ed., Mc Graw Hill.

Laguna, M. (1997). “Metaheuristic Optimization6 with Evolver, Genocop and OptQuest”. <http://www.decisioneering.com/optquest/metaheuristic.pdf>

Laguna, M. (1997a). “Optimization of complex systems with OptQuest”. <http://www.decisioneering.com/optquest/optimization.pdf>

- Law, A.M. and McComas, M.G. (2000).** “Simulation-Based Optimization”. *Proceedings of the 2000 Winter Simulation Conference*, pp. 46-49.
- Martin, J.D. y Simpson, T.W. (2002).** “Use of adaptive metamodeling for design optimization”. *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 1-9.
- Melin, P. y Castillo, O. (2002).** *Modeling, Simulation and control of non-linear dynamical systems: An intelligent approach using soft computing and fractal theory*. CRC Press.
- Moniaci, W.E., Pasero, E. y Carmellino, P. (2005).** “Application of neural networks in production system simulation”. *Proceedings of the International Conference on Systems, Man and Cybernetics*, Vol.1, pp. 827-831.
- Montgomery, D.C. (2005).** *Diseño y Análisis de Experimentos*. 2^{da}. ed., Editorial Limusa.
- Ólafsson, S. y Kim, J. (2002).** “Simulation Optimization”. *Proceedings of the 2002 Winter Simulation Conference*, pp. 79-84.
- Osswald, T.A. (1998).** *Polymer Processing Fundamentals*. 1^a ed., Hanser Publishers.
- Pohlheim, H. (2004).** *GEAtbx: Genetic and Evolutionary Algorithms Toolbox for use with MATLAB Documentation*, Version 3.5.
- Pukelsheim, F. (1993).** *Optimal Design of Experiments*. 2^a ed., Wiley.
- Rachmat, M. (2006).** *Design of Experiments and Data Envelopment Analysis Based Optimization in Injection Holding*. Tesis de Maestría, The Ohio State University, Ohio, USA.
- Salazar, M.A. (2005).** *Pronóstico de demanda por medio de Redes Neuronales Artificiales (RNAs) en la industria de telecomunicaciones*. Tesis de Maestría, Universidad Autónoma de Nuevo León, Nuevo León, México.
- Soporte Técnico Microsoft (2006).** “Solver Uses Generalized Reduced Gradient Algorithm”. No. 82890, Revisión 1.4. <http://support.microsoft.com/kb/82890/en-us#top>
- Swisher, J.R., Hyden, P.D., Jacobson, S.H. y Schruben, L.E. (2000).** “A survey of simulation optimization techniques and procedures”. *Proceedings of the 2000 Winter Simulation Conference*, pp. 119-128.
- Taha, H.A. (1995).** *Investigación de Operaciones*. 5^a. ed., Alfaomega Grupo Editorial.

Urbano-Vázquez, M. A. y Cabrera-Ríos, M. (2006). “Comparación de diseños experimentales para la preedición de líneas de costura en el moldeo por inyección”. *Memorias del IEEE 4º Congreso Internacional en Innovación y Desarrollo Tecnológico*, pp.1-8.

Urbano-Vázquez, M. A. (2007). *Aplicaciones ingenieriles de análisis y diseño de experimentos: Ventanas de Proceso Multicriterio y modelos de confiabilidad para bulbos automotrices*. Tesis de Maestría, Universidad Autónoma de Nuevo León, Nuevo León, México.

Winston, W.L. (2006). *Investigación de Operaciones: Aplicaciones y Algoritmos*. 4^a ed., Thomson Learning Ibero.

Zielinski, K., Peters, D. y Laur, R. (2005). “Stopping Criteria for Single-Objective Optimization”. *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems*, pp.1-6.

Zielinski, K. y Laur, R. (2007). “Stopping Criteria for a Constrained Single-Objective Particle Swarm Optimization Algorithm”. *Informatica*, No. 31, pp.51-59.

AUTOBIOGRAFÍA

MARÍA GUADALUPE VILLARREAL MARROQUÍN

Candidata para el grado de maestro en Ciencias en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis

**“MODELACIÓN Y OPTIMIZACIÓN APLICADA A LA SIMULACIÓN DE
PROCESOS Y SISTEMAS DE MANUFACTURA”**

Nacida el 12 de Diciembre de 1984 en la Ciudad de Monterrey, Nuevo León. Hija del Sr. Rogelio Villarreal Silva y la Sra. María H. Marroquín Salazar. Graduada de la Universidad Autónoma de Nuevo León como Licenciada en Matemáticas (2005). Participó en el VII Verano de Investigación Científica y Tecnológica (PROVERICYT-UANL), durante este período trabajó en la FIME de la UANL en el PISIS bajo la tutela del Dr. Mauricio Cabrera Ríos. Inició sus estudios de Maestría en Ciencias en Ingeniería de Sistemas en Enero del 2006 con el apoyo del PISIS, la FIME, la UANL y una beca de manutención otorgada por el CONACYT. Durante el verano del 2007 con apoyo de una beca MIXTA CONACYT realizó una estancia de investigación en The Ohio State University en colaboración con el Dr. Mauricio Cabrera Ríos y el Dr. José M. Castro.

PUBLICACIONES

PUBLICACIONES RELACIONADAS CON ESTE TRABAJO DE TESIS

-Memorias In Extenso

- [2] M.G. Villarreal Marroquín y M. Cabrera-Ríos, **Un Método de Optimización vía Simulación: Aplicación a un Proceso de Moldeo por Inyección de Termoplásticos**, Congreso Nacional 2007 de la Asociación de México de Control Automático, Monterrey, Nuevo León, México, Octubre 24-26, 2007. (Ponente)
- [1] M.G. Villarreal Marroquín y M. Cabrera-Ríos, **Un método adaptativo basado en metamodelación para optimización aplicada a la simulación**, 5° Congreso Internacional de Innovación y Desarrollo Tecnológico 2007, Cuernavaca, Morelos, México, Octubre 10-12, 2007. (Ponente), (Mención honorífica al mejor artículo en el área de Sistemas Computacionales)

-Memorias

- [2] M.G. Villarreal Marroquín, M. Cabrera-Ríos, **An Adaptive Metamodeling-Based Method For Simulation Optimization**, Winter Simulation Conference 2007, Washington, D.C., USA, Diciembre 9-12, 2007. (Ponente)
- [1] M.G. Villarreal Marroquín y M. Cabrera-Ríos, **Modeling and simulation of a Chassis Painting**, INFORMS International Meeting, Río Grande, Puerto Rico, USA, Julio 8-11, 2007. (Ponente)

OTRAS PUBLICACIONES

-Artículos en Revistas Científicas

- [1] M.G. Villarreal Marroquín, M.L. Sánchez Peña, C.E. Castro, J.M. Castro, and M. Cabrera-Ríos, **Use of Data Envelopment Analysis and Clustering in Multiple Criteria Optimization**, Aceptado para su publicación en *Intelligent Data Analysis – An International Journal*, 2007.

-Artículos en Revistas de Divulgación

- [1] M.G. Villarreal Marroquín y M. Cabrera-Ríos, **Agrupamiento de Datos para la Solución de Problemas de Optimización Multicriterio**, *Ciencia-UANL*, Vol.10, No.2, pp. 137-142, 2007.

-Memorias In Extenso

- [4] M.L. Sánchez Peña, M.G. Villarreal Marroquín y M. Cabrera-Ríos, **Agrupamiento de datos para la solución del problema de optimización multicriterio por medio de Análisis Envoltante de Datos: Segunda Parte**, 5° Congreso Internacional de Innovación y Desarrollo Tecnológico 2007, Cuernavaca, Morelos, México, Octubre 10-12, 2007.
- [3] M.G. Villarreal Marroquín y M. Cabrera-Ríos, **Agrupamiento de datos para la solución del problema de optimización multicriterio por medio de Análisis Envoltante de Datos: Resultados Preliminares**, 4° Congreso Internacional de Innovación y Desarrollo Tecnológico 2006, Cuernavaca, Morelos, México, Octubre 11-13, 2006. (Ponente)

- [2] M.G. Villarreal Marroquín y M. Cabrera-Ríos, **Series de tiempo: Caracterización Empírica y Selección de Técnicas de Pronóstico basada en Redes Neuronales Artificiales**, Encuentro Nacional de Mujeres en la Ciencia 2006, León, Guanajuato, México, Mayo 2006. (Ponente)
- [1] M.G. Villarreal Marroquín y M. Cabrera-Ríos, **Series de tiempo: Caracterización Empírica y Selección de Técnicas de Pronóstico basada en Redes Neuronales Artificiales**, XVI Escuela Nacional de Optimización y Análisis Numérico, Apizaco, Tlaxcala, Mexico, Marzo 2006. (Ponente)

-Memorias

- [3] M. Cabrera-Ríos, M.L. Sánchez, M.G. Villarreal Marroquín, **Data Clustering Applied to Multiple Criteria Optimization through Data Envelopment Analysis**, INFORMS International Meeting, Río Grande, Puerto Rico, USA, Julio 8-11, 2007. (Ponente)
- [2] M. Cabrera-Ríos y M.G. Villarreal Marroquín, **Grouping Schemes for Data Envelopment Analysis to Solve Multiple Criteria Optimization Problems**, INFORMS Annual Meeting, Pittsburgh, PA, USA, Noviembre 5-8, 2006.
- [1] M.G. Villarreal Marroquín y M. Cabrera-Ríos, **Series de tiempo: Caracterización Empírica y Selección de Técnicas de Pronóstico Basada en Redes Neuronales Artificiales**, XXXIX Congreso Nacional de la Sociedad Matemática Mexicana, Villahermosa, Tabasco, México, Octubre 2006. (Ponente)