



UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA DE SISTEMAS
DEPARTAMENTO DE INVESTIGACION DE OPERACIONES

**DESARROLLO DE UNA HERRAMIENTA DE OPTIMIZACION
BASADA EN ANALISIS DE SUPERFICIES DE RESPUESTA
PARA MODELOS DE SIMULACION**

Por:

Andy Elinxon Bolívar Mendoza

Proyecto presentado ante la ilustre Universidad de Los Andes
Como requisito final para optar al título de Ingeniero de Sistemas

Tutor:

Prof. Sebastián Medina

Mérida, Enero 2005

A Dios por el amor y apoyo que me ha dado y por estar siempre a mi lado...

A mis Padres por darme la vida y criarme hasta hoy día.

www.bdigital.ula.ve

Reconocimiento-No comercial-Compartir igual

Agradecimientos

Existen muchas personas a las cuales me gustaría agradecer por contribuir directa o indirectamente en este trabajo.

Primero, me gustaría dar las gracias a Dios por darme la habilidad, fuerza y privilegio de haber estudiado una carrera tan retadora y maravillosa.

Al profesor Sebastián Medina, por sus palabras y consejos que no permitieron rendirme, por su dedicación y apoyo para este trabajo. Estoy eternamente agradecido por la confianza que puso en mí.

Gracias en especial a Ernesto Pérez (Q.E.P.D), quién me ayudó a utilizar nuevas herramientas de programación y por haber sido tan buen amigo, nunca olvidare sus enseñanzas.

A todas aquellas personas que me aconsejaron para realizar este trabajo, muchísimas gracias.

Resumen

El área donde se integran la simulación y la optimización, recientemente se han experimentado cambios muy notables. Nuevos avances están haciendo disponibles aplicaciones de simulación que previamente habían sido considerados no factibles o más allá del alcance de la tecnología actual.

Este trabajo describe una herramienta recientemente desarrollada que integra con eficacia la simulación y la optimización. La herramienta emplea la Metodología de Superficies de Respuesta que forma parte de los Diseños Experimentales, los cuales tienen el propósito de determinar las condiciones óptimas de operación. Es por ello que se aplicará esta metodología con la finalidad de encontrar una manera más sencilla de determinar las condiciones bajo las cuales debe operar un modelo de simulación generado en el software ARENA.

En este trabajo se desarrolla la herramienta RSOptim, que implementa esta metodología y se aplica a un ejemplo práctico de un modelo de simulación, mostrando buenos resultados en la optimización de los mismos.

PALABRAS CLAVES:

Simulación, Modelos Matemáticos

Optimización

Programación Visual

COTA:

T57.62

B65

Indice general

Introducción	1
1. Identificación del Proyecto	5
1.1. Antecedentes	5
1.2. Objetivos	6
1.2.1. Objetivo General	6
1.2.2. Objetivos Específicos	6
1.3. Delimitación del Proyecto	7
2. Marco Teórico	9
2.1. Introducción	9
2.2. Optimización	9
2.3. Búsqueda Tabú	11
2.3.1. Algoritmo Usado para la Búsqueda Tabú	12
2.4. Metodología de Superficies de Respuesta	17
2.5. Modelo	18
2.6. Simulación	18
2.7. Optimización para Modelos de Simulación	19
2.8. Software Utilizado	20
2.8.1. ARENA	20
2.8.2. Herramientas de Programación	21
3. Aplicación de Superficies de Respuesta	23
3.1. Descripción del método	23
3.1.1. Planteamiento Factorial 2^k	24
3.1.2. Variables codificadas	25
3.1.3. Aproximación de Funciones De Respuesta	27
3.1.4. Estimación por Mínimos Cuadrados	28
3.1.5. Método de Máxima Pendiente en Ascenso	28

3.1.6. Máxima Pendiente en Ascenso Sujeta a Una Restricción Lineal	31
3.2. Ejemplos de aplicación de Superficies de Respuestas	34
3.2.1. Ejemplos Cualitativos	34
3.2.2. Ejemplo Cuantitativo	40
4. Desarrollo de la Herramienta RSOptim	47
4.1. Optimización de Modelos en ARENA	47
4.2. Interfaz RSOptim	50
4.2.1. Estructura de RSOptim	52
4.3. Metodología Aplicada en RSOptim	55
5. Aplicación de RSOptim	59
5.1. Aplicación en el Modelo	59
6. Conclusiones y Recomendaciones	65
6.1. Conclusiones	65
6.2. Recomendaciones	66
Bibliografía	68

www.bdigital.ula.ve

Indice de figuras

2.1.	Iteración en Búsqueda Tabú	13
2.2.	Lista Tabú	13
2.3.	Algoritmo Búsqueda Tabú	14
2.4.	Arbol de Expansión Mínima	15
2.5.	Arbol de Expansión Mínima, 1ra Iteración	15
2.6.	Arbol de Expansión Mínima, 2da Iteración	16
2.7.	Arbol de Expansión Mínima, 3ra Iteración	16
2.8.	Arbol de Expansión Mínima, 4ta Iteración	17
3.1.	Máxima Pendiente en Ascenso con una Restricción Lineal.	32
3.2.	y en función de a	35
3.3.	Dirección de Máxima Pendiente en Ascenso.	36
3.4.	Movimiento de Máxima Pendiente en Ascenso.	36
3.5.	Gráfica de Contornos.	37
3.6.	Vector de Máxima pendiente en Ascenso.	38
3.7.	Movimiento de Máxima Pendiente en Ascenso en dos Dimensiones	39
3.8.	Nueva Dirección de Máxima Pendiente en Ascenso.	40
3.9.	Informe de respuestas generado por SOLVER.	41
3.10.	Porción del análisis de regresión en EXCEL.	42
4.1.	Referencias en Microsoft Visual Basic.	50
4.2.	Acceso a ARENA desde Microsoft Visual Basic.	51
4.3.	Ventana Principal en RSOptim.	53
4.4.	Ventana Restricciones en RSOptim.	53
4.5.	Ventana Opciones en RSOptim.	54
4.6.	Ventana Controles en RSOptim.	55
4.7.	Ventana Objetivo en RSOptim.	56
5.1.	Modelo Estudiado en ARENA	60
5.2.	Selección de Entradas: Recursos y Variables	61

5.3. Restricciones	61
5.4. Función Objetivo	62
5.5. Ventana de Salida	63
5.6. Ventana de Salida de OptQuest	63
5.7. Salida de las Variables de RSOptim	64
5.8. Salida Función Objetivo de RSOptim	64

www.bdigital.ula.ve

Indice de tablas

3.1. Tabla de Resultados, valores originales y codificados	42
3.2. Tabla de movimiento en trayectoria máximo ascenso y modificada	45

www.bdigital.ula.ve

Introducción

En la presente tesis se pretende desarrollar una nueva herramienta que pueda optimizar cualquier modelo de simulación descrito en el programa ARENA, mediante un método moderno y de rápida ejecución que aporte buenos resultados. Para un mejor entendimiento se introducirán los temas bases de este proyecto que son: Simulación, Optimización y la "Metodología de Superficies de Respuesta".

La Simulación es una técnica de análisis basada en la experimentación sobre un modelo de un sistema real, para evaluar su comportamiento bajo ciertas condiciones. El propósito es probar hipótesis de trabajo antes de su desarrollo e implementación, permitiendo aspirar a grandes economías de dinero, tiempo y esfuerzo.

El uso de la simulación como herramienta para el estudio y el diseño de ciertos tipos de sistemas, es cada vez mayor. En la literatura es común encontrar referencias de aplicaciones en ingeniería, economía, medicina (fisiología), arquitectura (básicamente simulación gráfica), física, química, finanzas, administración de negocios y aún en aspectos del comportamiento humano tales como sociología y psicología. El auge de esta técnica se fundamenta en el hecho de que el modelo de simulación se convierte en una especie de laboratorio sobre el cual podemos, sin consecuencias dañinas para el sistema, realizar todo tipo de experimentos conducentes a mejoras en la estructura y el manejo del mismo. Muy a menudo nos vemos ante la necesidad de conocer las consecuencias que podrían tener ciertas acciones cuando nos enfrentamos a una toma de decisiones; o conocer los impactos que la variación de ciertos parámetros puede tener sobre un fenómeno dado; o simplemente

conocer el comportamiento de un sistema a través del tiempo, con la intención, quizás, de analizar un posible estado de equilibrio o el proceso que lleva a este estado.

En su definición estricta la optimización está restringida al empleo de técnicas matemáticas para representar por medio de un modelo y poder llegar a soluciones de maximización o minimización que se consideren ni igualables ni superables por cualquier otra solución factible.

Los problemas de optimización comprenden la determinación de tres elementos fundamentales como son los criterios de decisión, expresados por ejemplo en términos de una función objetivo a cumplir o a satisfacer, las opciones de decisión y las restricciones del problema. Más allá de las técnicas para la solución de problemas de optimización el factor determinante para su resolución es poder realizar una adecuada formulación del problema a resolver. Esto requiere desarrollar capacidades analíticas y de conceptualización para la adecuada representación en términos matemáticos de los elementos alrededor de un problema de optimización.

La Metodología de Superficies de Respuestas es un método que se utiliza en la etapa de optimización propiamente dicha. Su aplicación permite seleccionar la combinación de niveles óptimos en la obtención de la mejor respuesta para una situación dada.

Desarrollar lo mencionado en los párrafos anteriores implica la utilización de diversas áreas que se estudian en la Ingeniería de Sistemas como son la Investigación de Operaciones, los Diseños de Experimentos, la Simulación, Análisis Numérico, Álgebra Lineal y la Programación Orientada a Objetos.

Este informe está estructurado en 6 capítulos, en los cuales se presenta lo siguiente:

En el Capítulo 1, se presentan los antecedentes, objetivos y delimitaciones de este proyecto.

En el Capítulo 2, se muestra el Marco Teórico en el cual se describen algunos conceptos y describen metodologías de Optimización y Simulación, además se hace una pequeña introducción de la Metodología de Superficies de Respuesta y una breve descripción del

software a usarse: ARENA.

En el Capítulo 3, está dedicado en su totalidad a la Metodología de Superficies de Respuesta, se presenta la descripción de la metodología, la descripción de todo el proceso que se realiza en la metodología y finalmente la implementación del método a un problema de programación lineal.

El Capítulo 4, se presenta la descripción de la herramienta de optimización RSOptim, parte de su construcción y una breve descripción de la integración con el software ARENA.

En el Capítulo 5, se realiza la validación de la herramienta RSOptim aplicándola a un modelo de simulación en ARENA, se muestra paso a paso la utilización de la interfaz para este ejemplo, los resultados obtenidos y el análisis de los resultados obtenidos en la ejecución del programa.

En el Capítulo 6, se presentan las Conclusiones y Recomendaciones del proyecto.

www.bdigital.ula.ve

Capítulo 1

Identificación del Proyecto

En este capítulo se identifican los antecedentes que motivaron su realización, objetivos generales y específicos y la delimitación del estudio.

www.bdigital.ula.ve

1.1. Antecedentes

La Formación y el posterior ejercicio profesional en la Investigación de Operaciones están fuertemente asociados a la simulación de diferentes sistemas, estudiando, analizando y determinando el mejor curso de acción para la resolución de éstos, que generalmente están enmarcados por la restricción en cuanto a los recursos disponibles. Entonces es necesario además de simular emplear técnicas de optimización a un modelo en estudio.

El software de simulación ARENA tiene una herramienta llama OptQuest, la cual decide como cambiar las entradas del modelo que son seleccionadas para luego correr una secuencia de simulaciones en búsqueda de una combinación de estas entradas que optimicen (maximice o minimice) una medida de funcionamiento de la salida que se designe.

La herramienta OptQuest emplea como método de optimización la "búsqueda Tabú"(Tabu

Search), por otro lado el Profesor Sebastián Medina [14] propone para este proyecto la aplicación de la Metodología de Superficies de Respuesta en la creación de una nueva herramienta de optimización para el software de simulación ARENA.

1.2. Objetivos

1.2.1. Objetivo General

Aplicar la Metodología de Superficies de Respuesta para la optimización de modelos de simulación en el programa de simulación ARENA.

1.2.2. Objetivos Específicos

- Estudiar los diferentes métodos de optimización, para determinar su idoneidad en la aplicación a estudiar.
- Investigar el funcionamiento de la herramienta Optquest de ARENA.
- Estudiar las herramientas computacionales necesarias para implementar una interfaz entre los modelos de simulación en ARENA y el programa de optimización basado en los métodos bajo estudio.
- Implementar las metodologías de superficies de respuesta y redes neuronales planteadas usando la interfaz construida en el paso anterior.
- Construir la interfaz.
- Probar las herramientas de optimización diseñadas aplicándolas a problemas de optimización de modelos de simulación.

1.3. Delimitación del Proyecto

Se han hecho algunas delimitaciones y simplificaciones en la creación de la herramienta de optimización y en la aplicación de la metodología para la misma:

- La Metodología de Superficies de Respuesta es aplicada hasta su primera fase en la cual se hace un ajuste de primer orden.
- En la aplicación de la metodología no se realizarán pruebas de significancia de los coeficientes estimados en el modelo ajustado, se asume que el usuario de la herramienta conoce esta información antes de usar la herramienta.
- La herramienta de optimización será aplicada a un sólo modelo de simulación creado en ARENA, ya que en este proyecto sólo se requiere demostrar que la metodología es aplicable.
- Se tomará como modelo base para la creación de la interfaz a la herramienta OptQuest, sin copiar todas sus propiedades.

Capítulo 2

Marco Teórico

2.1. Introducción

Al optimizar vía simulación, se trata de estimar los parámetros del modelo que optimicen una salida específica del modelo de simulación. En este proceso de optimización, el modelo de simulación es considerado a veces como un modelo de caja-negra [16] donde la salida del modelo de simulación puede ser mirada como una función de los parámetros del modelo. En este capítulo se muestran los conceptos necesarios para comprender los desarrollos posteriores.

2.2. Optimización

Por optimización con frecuencia queremos indicar al proceso mediante el cual obtenemos la “mejor solución” en algún sentido a un determinado problema. El campo de optimización matemática trata del estudio de la minimización o maximización de funciones, restringidas o no a un determinado conjunto. La optimización tiene amplias aplicaciones en ingeniería, economía, finanzas, medicina, estadística y toda ciencia aplicada en general; es por ello que

constituye un área de gran interés para la comunidad científica internacional. Los problemas de optimización se componen generalmente de estos tres elementos:

Función Objetivo Es la medida cuantitativa del funcionamiento del sistema que se desea optimizar (maximizar o minimizar).

Variables. Representan las decisiones que se pueden tomar para afectar el valor de la función objetivo. Desde un punto de vista funcional se pueden clasificar en: variables independientes o de control, y variables dependientes o auxiliares, aunque matemáticamente todas son iguales.

Restricciones Representan el conjunto de relaciones (expresadas mediante ecuaciones o inecuaciones) que ciertas variables están obligadas a satisfacer.

Resolver un problema de optimización consiste en encontrar el valor que deben tomar las variables para hacer óptima la función objetivo, satisfaciendo el conjunto de restricciones.

Los métodos de optimización los podemos clasificar en: métodos clásicos y métodos metaheurísticos. Dentro de los primeros se encuentran la optimización lineal, lineal entera mixta, no lineal, estocástica, dinámica, y otras. En el segundo grupo se encuentran los algoritmos evolutivos, el método de recocido simulado y las búsquedas heurísticas como Búsqueda Tabú y la Metodología de Superficies de Respuesta.

Los procedimientos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que métodos heurísticos clásicos no son ni efectivos ni eficientes. Los metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de inteligencia artificial, evolución biológica y mecanismos estadísticos.

En particular en este proyecto se describe el método de Búsqueda Tabú por estar presente en la herramienta de optimización OptQuest del software de simulación ARENA, para obtener

una idea más clara de como implementar la Metodología de Superficies de Respuesta a la nueva herramienta de optimización.

2.3. Búsqueda Tabú

Búsqueda Tabú es un procedimiento iterativo para resolver problemas discretos de optimización combinatoria. La idea básica del método es la de explorar el espacio de búsqueda de todas las soluciones factibles por una secuencia de movimientos. Sin embargo, para escapar de un óptimo local y para prevenir los ciclos, algunos movimientos, en una iteración en particular, son clasificados como prohibidos o tabú. Los movimientos en tabú están basados en la historia de la secuencia de movimientos a corto y largo término. Una implementación simple, por ejemplo, puede clasificar un movimiento como tabú si el movimiento contrario ha sido hecho recientemente o frecuentemente. Algunas veces, cuando es favorable, un movimiento tabú puede ser realizado a pesar de todo.

La Búsqueda Tabú, fue iniciada por Hansen y Glover en los años 1987 y 1989 respectivamente [8], es un método que no tiene una base teórica que lo respalde y su principal objetivo es encontrar soluciones factibles y óptimas en problemas con alto grado combinatorio; mediante una definición adecuada de criterios de selección y parámetros es una eficiente herramienta de resolución de problemas de optimización de gran tamaño, permitiendo obtener soluciones de buena calidad con un bajo consumo de recursos computacionales.

Puesto que la Búsqueda Tabú quiere de hecho incluir en sus propias reglas de funcionamiento algunas técnicas heurísticas, sería más apropiado caracterizar a la Búsqueda Tabú como metaheurística. Su papel es a menudo dirigir y orientar en lo posible la acción de otro procedimiento de búsqueda (más local).

Algunas de las diversas aplicaciones de Búsqueda Tabú de hoy día son; Telecomunicaciones: enrutamiento de llamadas, diseño de redes para servicios, redes

ópticas síncronos; Inteligencia lógica y artificial: lógica probabilística, reconocimiento de patrones, integridad de datos; Diseño: diseño de redes de transporte; Tecnología: Inversión sísmica, distribución de energía eléctrica, construcción de estación espacial; Optimización de estructuras: estructuras proteicas, secuencia ADN, estructuras electromagnéticas,

Con Búsqueda Tabú, la complejidad no está solamente presente en los problemas sino también en la misma técnica.

Supóngase que f es la función objetivo, evaluada en los reales, en un espacio de búsqueda S y se requiere encontrar un $c \in S$ tal que $f(c)$ tiene su valor máximo. Para problemas combinatoriamente difíciles, este requerimiento debe ser relajado a encontrar un $c \in S$ tal que $f(c)$ esta cerca del valor máximo (un valor sub-óptimo). Esto es debido a que cualquier algoritmo conocido para determinar una solución máxima requiere un tiempo que es exponencial en el tamaño del problema. Problemas sub-óptimos pueden ser resueltos parando el algoritmo cuando se alcanza un cierto umbral para una solución aceptable o cuando se han completado un cierto número de iteraciones.

Una caracterización del espacio de búsqueda S para el cual una Búsqueda Tabú puede ser aplicada es que haya un conjunto de k movimientos $M = \{m_1, m_2, \dots, m_k\}$ y la aplicación de los movimientos a una solución factible $s \in S$ lleva a k soluciones, normalmente distintas, $M(s) = \{m_1(s), \dots, m_k(s)\}$. El subconjunto $N(s) \subset M(s)$ de soluciones factibles es conocido como el vecindario de s .

2.3.1. Algoritmo Usado para la Búsqueda Tabú

Esta sección se ha tomado casi en su totalidad de Rhone y Prasetio [18]. Una iteración en Búsqueda Tabú se muestra en la figura 2.1, y la lista Tabú que es otro elemento importante se muestra gráficamente en la figura 2.2.

El algoritmo para una Búsqueda Tabú se presenta mediante un diagrama de flujo, tal y como se muestra en la figura 2.3.

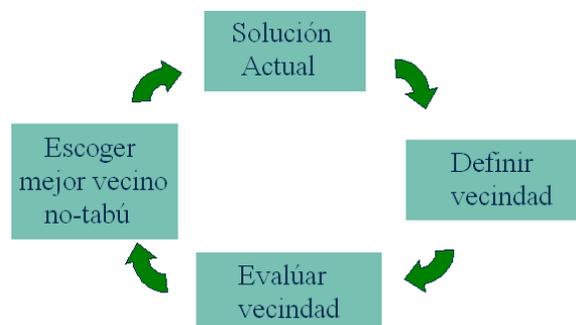
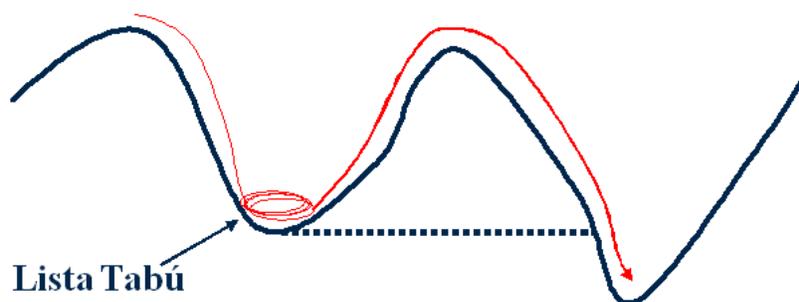


Figura 2.1: Iteración en Búsqueda Tabú

**Lista Tabú**

Técnica de memoria que ayuda a indentificar ciclos

Figura 2.2: Lista Tabú

A continuación se muestra un ejemplo donde se aplica la Búsqueda Tabú a un árbol de expansión mínima, la meta de éste es encontrar el conjunto de arcos que conecte todos los nodos dando un costo mínimo.

Los arcos están etiquetados por x_1 hasta x_7 . Los valores numéricos a lado de las etiquetas de los arcos son los costos de cada arco. Si $x_i = 1$, entonces el arco i se encuentra en el árbol y se indica con una línea sólida. Si $x_i = 0$, entonces el arco i no está en el árbol y se indica con una línea rayada. Aquí, se tienen dos restricciones. La primera restricción asume que por lo menos uno de los tres arcos, x_1 , x_2 , y x_6 se les permite estar en el árbol ($x_1 + x_2 + x_6 \leq 1$). La segunda restricción asume que el arco x_1 es permitido en el árbol sólo si el arco x_3 se encuentra también en el árbol ($x_1 \leq x_3$). Alguna violación de las restricciones ocasionará una penalidad de 50 unidades. En la figura 2.4 se puede apreciar el árbol en estudio.

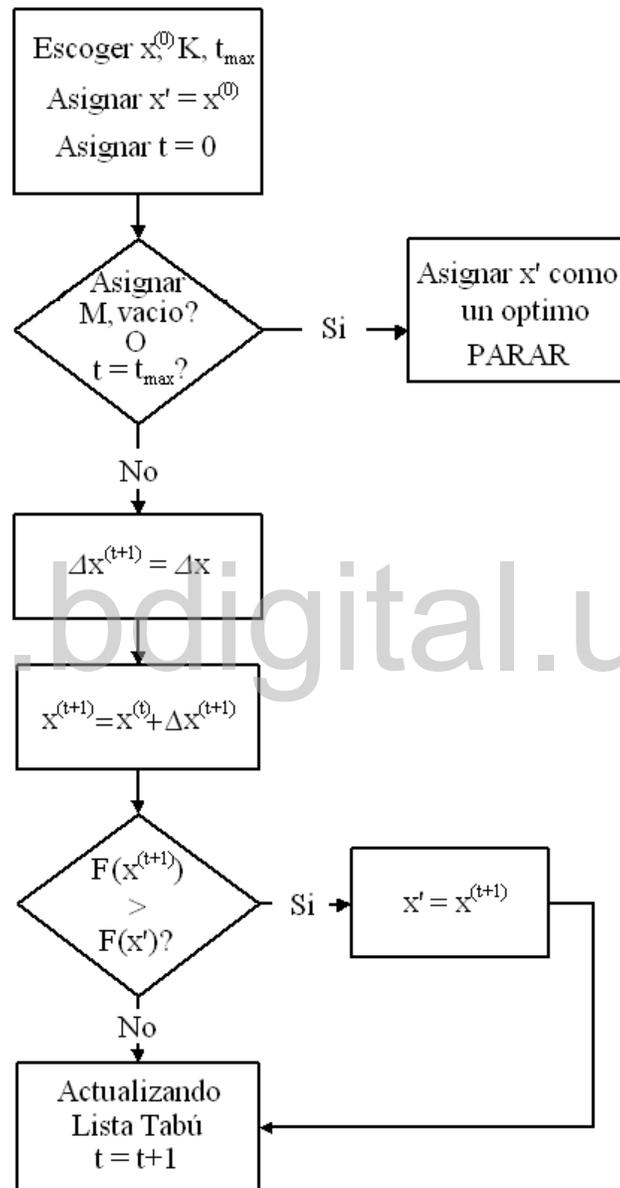


Figura 2.3: Algoritmo Búsqueda Tabú

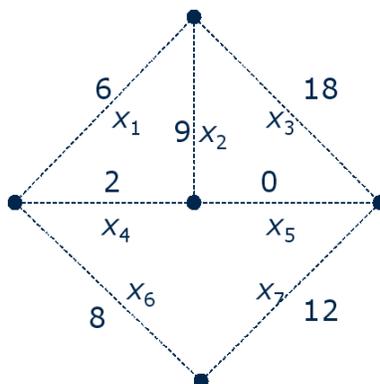


Figura 2.4: Arbol de Expansión Mínima

El movimiento considerado es el estándar “cambio de arco”, que consiste en añadir un arco descartando otro para transformar el árbol actual a uno nuevo.

Un arco se añade a la lista tabú sí fue agregado entre las dos últimas iteraciones. Sí el arco está en la lista tabú entonces no se permite descartarlo.

Primera Iteración: El árbol inicial de la figura 2.5 representa el costo mínimo sin considerar la penalidad. Dado que ambas restricciones son violadas, se agregan 100 unidades de penalidad al costo ($F(x) = 16 + 100$).

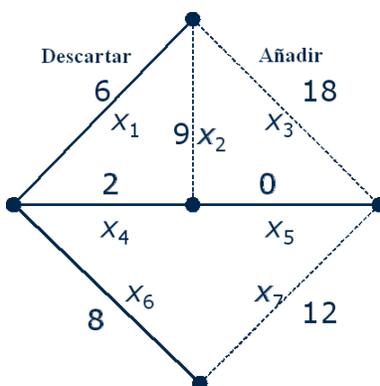


Figura 2.5: Arbol de Expansión Mínima, 1ra Iteración

Entre todos los movimientos posibles, añadir x_3 y descartar x_1 produce el mejor árbol al eliminar ambas violaciones. Nótese que ninguno de los arcos se encuentran en la lista tabú ya que no existen movimientos previos.

Segunda Iteración: Luego con sólo añadir x_3 al árbol, el arco agregado a la lista tabú. En éste caso, el estatus tabú de x_3 no afecta el proceso de búsqueda ya que éste no proporcionará un mejor árbol. Esto muestra que la restricción tabú no siempre afecta la escogencia. El resultado al evaluar la función con el árbol que se muestra en la figura 2.6 es $F(x) = 28$.

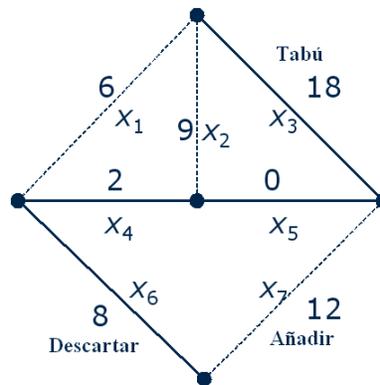


Figura 2.6: Arbol de Expansión Mínima, 2da Iteración

Tercera Iteración: El arco x_7 se une a x_3 para estar en la lista tabú. El movimiento descartando el arco x_3 usualmente no es permitido dado que éste se encuentra en la lista tabú. Debido a el hecho que se produce un árbol que satisface el criterio de aspiración, un nuevo árbol que es el mejor de los árboles hasta ahora, se permite éste movimiento. En ésta iteración el árbol mostrado en la figura 2.7 da como resultado $F(x) = 32$

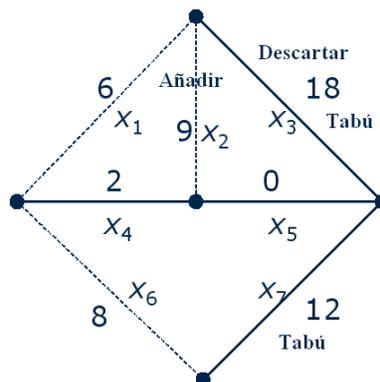


Figura 2.7: Arbol de Expansión Mínima, 3ra Iteración

Cuarta Iteración: La búsqueda continua hasta que la última iteración deseada se alcanza. La figura 2.8 muestra el árbol de esta iteración con un valor $F(x) = 23$.

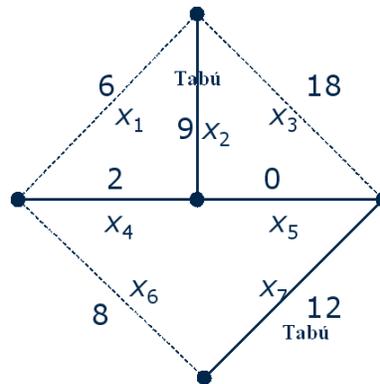


Figura 2.8: Arbol de Expansión Mínima, 4ta Iteración

2.4. Metodología de Superficies de Respuesta

“La Metodología de Superficies de Respuesta (MSR) es un conjunto de técnicas matemáticas y estadísticas utilizadas para desarrollar, mejorar y optimizar procesos. También tiene importante aplicación en el diseño, desarrollo y formulación de nuevos productos, así como en la mejora de diseños de productos existentes” [15, p. 1].

Estas técnicas se usan para modelar y analizar problemas en los que una variable de interés es influenciada por otras. El objetivo es optimizar la variable de interés. Esto se logra al determinar las condiciones óptimas de operación del sistema.

MSR esta basada en la aproximación de la función objetivo por un polinomio de bajo orden en una sub región pequeña del dominio. Los coeficientes del polinomio son estimados por regresión lineal (mínimos cuadrados) aplicado a un número de observaciones de la función objetivo estocástica. Para este fin, la función objetivo se evalúa en un arreglo de puntos referido como un diseño experimental [10]. De acuerdo con el polinomio ajustado, se deriva el mejor punto local, el cuál es usado como el estimador actual del óptimo y como el punto central de la nueva región de interés [7], donde de nuevo la función objetivo estocástica es aproximada por un polinomio de bajo orden.

Esto es solo una pequeña introducción donde se da el concepto del Método de Superficies de Respuesta, el capítulo 3 trata en un contexto mucho más amplio toda la metodología.

2.5. Modelo

Es una simplificación y una abstracción de la realidad, que a través de supuestos, argumentos y conclusiones explica un aspecto de un fenómeno.

Un modelo es un prototipo o sustituto de una situación compleja. Un modelo puede ser físico, tal como un modelo arquitectónico de un diseño urbano, o un modelo matemático de interacciones de muchas variables. Es usado en simulaciones para relacionar varios componentes o puede ser una herramienta única de soporte para evaluar diferentes acercamientos usando diferentes suposiciones. Los avances en computación, tanto en hardware como en software, han posibilitado el desarrollo de herramientas de simulación y modelado que pueden tratar sistemas muy complejos con relativa facilidad.

2.6. Simulación

Existen diversas definiciones y tipos de simulación. Simular, según el diccionario de la real academia, es “reproducción de un proceso o fenómeno mediante otro más sencillo, o más cómodo de manejar, que evolucione de manera homóloga al primero”.

La simulación se refiere a una amplia colección de métodos y aplicaciones para imitar la conducta de sistemas reales, normalmente en una computadora con el software apropiado [19]. La simulación es una imitación de las operaciones de un sistema o proceso real que involucra la generación de una historia artificial del comportamiento del sistema y a partir de dicha historia se efectúan inferencias relativas a las características operacionales del sistema real que representa; permitiendo de esta manera describir y analizar el comportamiento del sistema real, y responder ciertas interrogantes para apoyar la toma de decisiones, ya sea en el diseño de un sistema antes que éste sea construido o probando políticas de operación, antes que éstas sean implantadas [12].

El modelo de simulación puede incluir una rutina de búsqueda, para obtener una solución

óptima o cercana a la óptima. Por esto la optimización en la simulación puede ser una aproximación a la optimidad que ocurre en la programación matemática.

2.7. Optimización para Modelos de Simulación

El empleo de modelos de simulación ha sido una técnica muy utilizada para aproximarnos a problemas muy complicados de tratar analíticamente. Sin embargo, determinar los valores óptimos de los parámetros de una simulación puede requerir un tiempo de computación excesivo por dos razones: el número de combinaciones posibles puede ser muy alto y por otra parte, habitualmente, no se puede obtener la función objetivo en función de los parámetros de entrada de forma explícita, lo que obliga a ejecutar la simulación para obtener este valor.

Cuando el modelo matemático de un sistema es estudiado usando simulación, se le conoce como modelo de simulación. El comportamiento de un sistema en valores específicos de las variables de entrada se evalúan al correr el modelo de simulación en un periodo de tiempo fijo. Un experimento de simulación puede ser definido como una prueba o serie de pruebas en las cuales se hacen cambios significativos a las variables de entrada del modelo de simulación de manera que se puede observar e identificar las razones de cambios en la variable o variables de salida. Cuando el número de variables de entrada es grande y el modelo de simulación complejo, el experimento de simulación puede pasar a ser prohibitivo en términos computacionales. Además del alto costo computacional, se incurre en un costo aún más alto cuando se eligen valores de las variables de entrada sub-óptimas. El proceso de encontrar el mejor valor de las variables de entrada entre todas las posibilidades sin evaluar explícitamente cada posibilidad es optimización de simulación. El objetivo de optimización de simulación es minimizar el gasto de recursos mientras se maximiza la información obtenida en el experimento de simulación [6].

Muchos investigadores en los últimos años han desarrollado diferentes métodos para optimizar simulaciones, es decir buscar valores óptimos o cercanos al óptimo de los

parámetros de entrada. Algunas recientes recopilaciones de estos métodos se pueden encontrar en Azadivar [4], Andradóttir [3], Carson y Maria [6]. En los últimos años, la tendencia es aplicar las técnicas de búsqueda metaheurística. Incluso existen varios programas informáticos para optimizar simulaciones que usan estas estrategias; por ejemplo: AutoStat que usa una variante de las estrategias evolutivas de Bäck [5]; OptQuest usa Scatter Search y Búsqueda Tabú [9]; SimRunner2 usa algoritmos genéticos [17] y WITNESS usa Temple Simulado y Búsqueda Tabú [11].

Se necesitan métodos que puedan ser usados para determinar los valores de los parámetros del sistema que produzcan un funcionamiento óptimo del mismo. Se pueden considerar los casos cuando los valores de los parámetros del sistema pueden tomar un rango continuo y los casos cuando los valores de los parámetros deben estar en un conjunto discreto. Un enfoque podría estar basado en técnicas del gradiente para parámetros continuos en la optimización y métodos de búsqueda aleatoria para parámetros discretos. Sin embargo una clase de métodos importante para optimización de simulaciones es la Metodología de Superficies de Respuesta [3].

2.8. Software Utilizado

A continuación se introducen breves descripciones de los software's usados para la comprensión y desarrollo de la herramienta de optimización en este trabajo.

2.8.1. ARENA

ARENA, es una herramienta computacional de fácil uso, que permite crear modelos de simulación animadas y conducir experimentos sobre el comportamiento del sistema. Posee una interfaz gráfica que resulta amena al usuario, permitiendo observar el funcionamiento del sistema a través de la animación, y a su vez ofrece un reporte completo de los resultados de la simulación [12].

Este software es una versátil aplicación, adecuada para simular sistemas complejos, operacionales y procesos en el ámbito de los servicios, la fabricación, el transporte, la logística y cadenas de suministro entre otros. Ayudando a representar y analizar los procesos de negocios mediante la simulación de diferentes alternativas. La generación de modelos de flujos de datos y la animación gráfica, permitiendo visualizar cómo funcionan los procesos actuales o futuros en una empresa.

OptQuest

OptQuest para ARENA es una herramienta de optimización transformada para requisitos particulares y para analizar los resultados de funcionamiento de la simulación creada por ARENA. OptQuest incluye técnicas de muestreo y control de error avanzado para encontrar mejores y más rápidas respuestas, e incorpora los algoritmos avanzados basados en Búsqueda Tabú, búsqueda de la dispersión, y programación de números enteros.

2.8.2. Herramientas de Programación

Para el desarrollo de la interfaz, que es el principal propósito de este trabajo, se programa con ayuda de los paquetes de programación Microsoft Visual Basic [1] y Microsoft Visual C++ [2], los cuales desempeñan un papel vital dado que el primero es el que ayuda a interactuar con el software ARENA, y el segundo es usado para los cálculos del Método de Superficies de Respuesta.

Capítulo 3

Aplicación de Superficies de Respuesta

3.1. Descripción del método

En el Método de Análisis de Superficies de Respuesta se realizan planteamientos factoriales, y los resultados se ajustan usando modelos matemáticos. Estas etapas se conocen como etapa de modelamiento y desplazamiento, las cuales se repiten varias veces, mapeando las superficies de respuesta obtenidas en la dirección de la región del punto óptimo deseado. El modelamiento normalmente se hace ajustando modelos simples, como el lineal y el cuadrático. Al mismo tiempo, el planteamiento factorial, consiste de un número pequeño y predeterminado de experimentos, que se escogen a través del ajuste conseguido para el modelo que fue aplicado en la etapa inmediatamente anterior. Otro detalle importante es el uso de variables en forma codificada, de forma que su tamaño no interfiera en el desarrollo del proceso de optimización. Comúnmente el modelamiento se inicia utilizando el modelo lineal. Para eso se realiza un planteamiento factorial de primer orden.

Suponga que $y = y(a, b, c, \dots)$, donde y es la variable de interés o **respuesta** la cual se va a optimizar, y existen n parámetros, a, b, c, \dots los cuales pueden ser variados. Se asume que el proceso de optimización es una maximización. El objetivo de MSR es encontrar

eficientemente los valores óptimos de a, b, c, \dots tales que el valor y sea máximo.

MSR trabaja mediante el **Método de Máxima Pendiente en Ascenso**. Los parámetros se varían en la dirección del incremento máximo de la respuesta hasta que la misma no se incremente más.

3.1.1. Planteamiento Factorial 2^k

Los diseños factoriales son ampliamente usados en experimentos que involucran varios factores donde es necesario investigar el efecto común de éstos sobre una respuesta variable. El efecto común de los factores significa efectos principales e interacciones. Un caso especial muy importante de los diseños factoriales es donde cada uno de los k factores de interés tienen sólo dos niveles. Como cada replica de tales diseños tienen exactamente 2^k pruebas experimentales de funcionamiento, estos diseños usualmente se les llama **diseños factoriales 2^k** [15, p. 79].

La clase de diseños factoriales 2^k es muy importante en el trabajo de superficies de respuesta. Específicamente, ésta encuentra aplicaciones en tres áreas [15, p. 79]:

1. Un diseño 2^k es de gran ayuda al comienzo de un estudio de superficies de respuesta donde deben realizarse experimentos de investigación para identificar procesos importantes o sistemas variables.
2. Un diseño 2^k se usa a menudo para ajustar el modelo de superficie de respuesta a un primer orden y para generar los efectos estimados de los factores para realizar el Método de Máxima Pendiente en Ascenso.
3. El diseño 2^k es un bloque de construcción básico usado para crear otros diseños de superficies de respuesta.

3.1.2. Variables codificadas

En general, las unidades de los parámetros a, b, c , etc, difieren unas de otras. Incluso si algunos de los parámetros tienen las mismas unidades, no todos ellos serán sobre el mismo rango. Por ejemplo, el parámetro a puede ser una longitud, con rango de 5 mts., el parámetro b puede ser otra longitud, con rango de 0.1 mts., el parámetro c puede ser una temperatura, con rango de 20 °C.

Dado que los parámetros tienen diferentes unidades y/o diferentes rangos en el conjunto de datos experimental, un análisis de regresión no se debe realizar en crudo, con los mismos parámetros físicos. En lugar de eso, se debe primero **normalizar** los parámetros antes de realizar un análisis de regresión. Las variables normalizadas se llaman **variables codificadas**.

En otras palabras, en vez de usar los valores de a, b, c, \dots directamente en el análisis de regresión, se usan las variables codificadas, x_1, x_2, x_3, \dots como las variables independientes en el análisis de regresión. Una variable codificada debe ser definida para cada una de las variables actuales, es decir:

- x_1 se define para el parámetro a
- x_2 se define para el parámetro b
- x_3 se define para el parámetro c
- ...

Cada una de las variables codificadas se fuerza a un rango de -1 hasta 1, de modo que todos afecten la respuesta y más uniformemente, y a su vez las unidades de los parámetros sean irrelevantes.

Para convertir el parámetro a a su variable codificada x_1 , se aplica la siguiente fórmula a

cada valor de a en el conjunto de datos:

$$x_1 = \frac{a - [\text{máx}(a) + \text{mín}(a)] / 2}{[\text{máx}(a) - \text{mín}(a)] / 2} \quad (3.1)$$

Donde $\text{max}(a)$ y $\text{min}(a)$ son el máximo y mínimo valor de a respectivamente en el conjunto de datos.

Se puede verificar en la ecuación 3.1 que $\text{min}(a)$ equivale a la variable codificada $x_1 = -1$, $[\text{máx}(a) + \text{mín}(a)]/2$ equivale a la variable codificada $x_1 = 0$, y $\text{max}(a)$ equivale a la variable codificada $x_1 = 1$, para cualquier conjunto de valores del parámetro a , sin importar el rango o unidad del parámetro a . Una ecuación similar se aplica a los otros parámetros en el problema.

Luego, el análisis de regresión lineal realizado es sobre y en función de x_1, x_2, x_3, \dots , y las pendientes con respecto a estas variables codificadas son usadas para determinar la dirección de máxima pendiente en ascenso.

Cuando se usan variables codificadas, el vector de máxima pendiente en ascenso debe entonces convertirse de nuevo a los parámetros físicos originales (no codificados), usando la inversa de las ecuaciones antes descritas. En otras palabras, el incremento de máxima pendiente en ascenso en x_1 se convierte al incremento correspondiente en a , el incremento de máxima pendiente en ascenso en x_2 se convierte al incremento correspondiente en b , etc., de modo que el proceso actual pueda realizarse en variables físicas.

Por ejemplo, una vez que se elige el incremento en la variable codificada x_1 (Δx_1), se debe convertir de regreso a un incremento del parámetro a (no codificado) (Δa), usando la siguiente ecuación:

$$\Delta a = \frac{\text{máx}(a) - \text{mín}(a)}{2} \Delta x_1 \quad (3.2)$$

con ecuaciones similares a la ecuación 3.2 para los incrementos en los restantes parámetros.

3.1.3. Aproximación de Funciones De Respuesta

Ya que la verdadera forma de la función real y de la respuesta es desconocida, debemos aproximarla. De hecho, el uso acertado de MSR es críticamente dependiente sobre la capacidad del experimentador de desarrollar una aproximación conveniente para y . Generalmente, un polinomio de orden bajo en una cierta región relativamente pequeña del espacio de las variables independientes es apropiado. En muchos casos, se utiliza un modelo de primer orden o segundo orden. Para el caso general de variables independientes, el modelo de primer orden, en términos de las variables codificadas, para la función de respuesta es [15, p. 3]:

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \dots + \beta_k \cdot x_k + \varepsilon$$

Obsérvese que hay una conexión cercana entre MSR y el análisis de regresión lineal. Los β 's son un sistema de parámetros desconocidos. Para estimar los valores de estos parámetros, debemos recoger datos sobre el sistema que estamos estudiando. El análisis de regresión es una rama de la creación de modelos estadísticos que utiliza estos datos para estimar los β 's. dado que, en general, los modelos polinómicos son funciones lineales de los β 's desconocidos, nos referiremos a la técnica como análisis de regresión lineal [15, p. 7].

En los modelos de regresión lineal, la variable dependiente se asume como una función lineal de una o más variables independientes más un error introducido para añadir los demás factores:

$$y_i = b_0 + b_1 \cdot x_{1i} + b_2 \cdot x_{2i} + \dots + b_k \cdot x_{ki} + u_i$$

En la ecuación de regresión anterior, y_i es la variable dependiente, x_{i1}, \dots, x_{ik} son las variables independientes, y u_i es el término del error. La meta del análisis de regresión es obtener los estimados b_0, b_1, \dots, b_k de los parámetros desconocidos $\beta_0, \beta_1, \dots, \beta_k$ los cuales indican cómo un cambio en una de las variables independientes afecta los valores tomados por la variable dependiente.

3.1.4. Estimación por Mínimos Cuadrados

El método usual para estimar un modelo de regresión lineal es el de **mínimos cuadrados**. Asuma que b_1, \dots, b_k denotan los estimadores de mínimos cuadrados para β_1, \dots, β_k . El valor predicho de la función de respuesta y_i es:

$$\hat{y}_i = b_0 + b_1 \cdot x_{1i} + b_2 \cdot x_{2i} + \dots + b_k \cdot x_{ki}$$

El error en mínimos cuadrados para la predicción de y , llamado residuo, es:

$$\varepsilon = y_i - \hat{y}_i$$

La idea básica de la estimación por mínimos cuadrados es elegir estimadores de β_1, \dots, β_k que minimicen la sumatoria de los cuadrados residuales:

$$\sum_{i=1}^n \varepsilon_i^2$$

www.bdigital.ula.ve

Se puede demostrar que:

$$b = (X' \cdot X)^{-1} \cdot X' \cdot y \quad (3.3)$$

donde X es una matriz $n \times k + 1$ con elementos $(i, k + 1) x_{ki}$, y es un vector $n \times 1$ con elemento típico y_i , y b es un vector $k + 1 \times 1$ con elemento típico b_k .

3.1.5. Método de Máxima Pendiente en Ascenso

Frecuentemente la estimación inicial de las condiciones de operación óptimas está alejada del óptimo real, en este caso se desea moverse rápidamente a la vecindad del óptimo. El Método de Máxima Pendiente en Ascenso es un procedimiento para recorrer secuencialmente la trayectoria de la máxima pendiente, que nos lleva en dirección del máximo aumento de la respuesta.

De acuerdo a Montgomery [15], la dirección de máximo ascenso es en la que \hat{y} aumenta más rápido, ésta es paralela a la normal de la superficie de respuesta ajustada. Los incrementos a lo largo de la trayectoria son proporcionales a los coeficientes de regresión b_0, b_1, b_2, \dots

Los experimentos se llevan a cabo hasta que deje de observarse un incremento en la respuesta, entonces se ajusta un nuevo modelo de primer orden con el que se determina una nueva trayectoria y se continua con el procedimiento. Finalmente, se consigue llegar a la cercanía del óptimo.

Dado que los estimadores b_0, b_1, \dots, b_k dependen de la codificación de los factores, la dirección de máxima pendiente en ascenso también depende de la escala.

En lo que sigue w_i representan las variables no codificadas y las x_i representan las codificadas.

Cálculo de la Trayectoria de Máxima Pendiente en Ascenso

Esta sección y la sección 3.1.6 se han tomado casi totalmente de Montgomery [15].

Considérese el modelo de regresión de primer orden:

$$\hat{y} = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_k \cdot x_k$$

La trayectoria de máxima pendiente en ascenso (MPA) es aquella que produce una respuesta máxima estimada con la restricción $\sum_{i=1}^k x_i^2 = r^2$. En otras palabras, de todos los puntos que están a una distancia fija del centro (distancia = r) del diseño, se busca el punto $x_1, x_2, x_3, \dots, x_k$ para el cual \hat{y} es maximizado. Se debe recordar que en la métrica de las variables codificadas del diseño el centro es $(0, 0, \dots, 0)$. Como resultado, la restricción dada por $\sum_{i=1}^k x_i^2 = r^2$ es una esfera de radio r .

El proceso de maximización involucra el uso de multiplicadores de Lagrange. La maximización requiere las derivadas parciales de la ecuación 3.4 con respecto a x_j ($j =$

$1, 2, \dots, k$).

$$L = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_k \cdot x_k - \lambda \left(\sum_{i=1}^k x_i^2 = r^2 \right) \quad (3.4)$$

La derivada con respecto a x_j está claramente dada por:

$$\frac{\partial L}{\partial x_j} = b_j - 2\lambda \cdot x_j \quad (j = 1, 2, \dots, k)$$

Al permitir que $\partial L / \partial x_j = 0$ se obtiene la siguiente coordenada de x_j de la trayectoria de MPA:

$$x_j = \frac{b_j}{2\lambda} \quad (j = 1, 2, \dots, k)$$

Ahora, la constante $1/2\lambda$ puede ser vista como una constante de proporcionalidad. De esta forma las coordenadas están dadas por:

$$x_1 = \rho \cdot b_1, x_2 = \rho \cdot b_2, \dots, x_k = \rho \cdot b_k \quad (3.5)$$

donde, para la MPA, la constante ρ es positiva. Ahora la elección de ρ , el cual esta relacionado a λ , simplemente determina la distancia desde el centro del diseño que residirá el punto resultante. Como resultado la constante ρ es determinada por el experimentador.

Cálculo del Ajuste de los Factores en la Dirección de MPA

El problema para encontrar el ajuste de los factores en la dirección de MPA que se encuentran a una distancia ρ del origen es dado por el problema de optimización,

$$\text{Maximizar } b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_k \cdot x_k$$

$$\text{Sujeto a: } \sum_{i=1}^k x_i^2 \leq \rho^2$$

Para resolverlo, se usa una aproximación por multiplicadores de Lagrange. Primero, se añade una penalidad λ para las soluciones que no satisfacen la restricción (dado que se quiere la dirección de MPA, se debe maximizar, y por lo tanto la penalidad es negativa)

$$\text{Maximizar } L = b'x - \lambda(x'x - \rho^2)$$

Se calculan las derivadas parciales y se igualan a cero

$$\frac{\partial L}{\partial x} = b - 2\lambda x = 0$$

$$\frac{\partial L}{\partial \lambda} = -(x'x - \rho^2) = 0$$

Estas ecuaciones tienen dos incógnitas (el vector x y el escalar λ) y así pueden ser resuelto produciendo la solución deseada:

$$X^* = \rho \frac{b}{\|b\|}$$

o en notación no vectorial:

$$X_i^* = \rho \frac{b_i}{\sqrt{\sum_{j=1}^k b_j^2}} \quad i = 1, 2, \dots, k \quad (3.6)$$

De la ecuación 3.6 se puede observar que para cualquier múltiplo ρ de la dirección del gradiente (dado por $b/\|b\|$) conducirá a los puntos en la dirección de MPA.

3.1.6. Máxima Pendiente en Ascenso Sujeta a Una Restricción Lineal

Si en un experimento se necesitan movimientos considerables, siempre existe la probabilidad de que el punto obtenido esté en un área del espacio donde una o más de las variables no estén en el rango permitido por el problema. Como resultado, en muchos casos se hace necesario construir la dirección de MPA con una restricción impuesta a las variables de diseño. Suponga que la restricción es vista como una forma de límite. Entonces, se habla

de un problema limitado por:

$$c_0 + c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_k \cdot x_k = 0 \quad (3.7)$$

Este límite no necesariamente involucra a todas las k variables. Se debe tener en cuenta que en la práctica las restricciones deben ser formuladas en términos de las variables naturales (no codificadas) y luego escritas en forma de variables codificadas para manipulación.

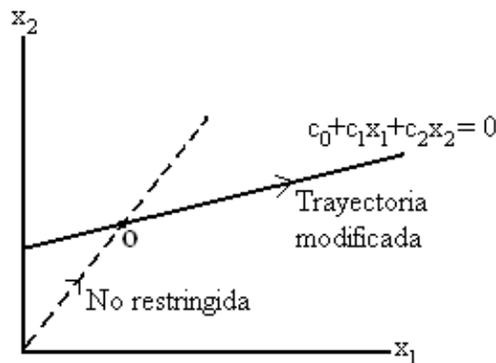


Figura 3.1: Máxima Pendiente en Ascenso con una Restricción Lineal.

La figura 3.1 muestra el procedimiento de MPA necesario cuando se va a aplicar una restricción lineal para un problema con $k = 2$. El procedimiento es el siguiente:

1. Se realiza el movimiento a través de la dirección usual de MPA hasta que se hace contacto con la línea de restricción lineal (o plano en el caso de $k > 2$). Se llama a este punto de contacto punto **O**.
2. Comenzando en el punto O, se debe mover a través de la trayectoria ajustada o modificada.
3. Se toman experimentos a lo largo de la trayectoria modificada, con los criterios de parada usuales.

Qué es la trayectoria modificada?

La figura 3.1 claramente muestra la trayectoria modificada para $k = 2$. En general, sin embargo, el vector apropiado de dirección es aquel que satisface la restricción y aún realiza grandes progresos posibles hacia la maximización de la respuesta.

Resulta (y es intuitivamente razonable) que la trayectoria modificada esta dada por la dirección del vector:

$$b_i - d \cdot c_i \quad (i = 1, 2, \dots, k)$$

para el cual

$$\sum_{i=1}^k (b_i - d \cdot c_i)^2$$

es minimizado. Donde d es un número real, que se debe calcular.

Esto significa, que la dirección a través de la línea de restricción (o plano) es cercana a la trayectoria original. Una vez más, se puede considerar esto como una situación simple de regresión, en la cual los b_i están siendo regresados contra los c_i . Como resultado, la cantidad d juega un papel importante de la pendiente de regresión y la minimización de los “residuos” de la suma de cuadrados produce el valor:

$$d = \frac{\sum_{i=1}^k b_i \cdot c_i}{\sum_{i=1}^k c_i^2} \quad (3.8)$$

De esta forma, la trayectoria modificada comienza en el punto O y procede usando la dirección del vector $b_1 - d \cdot c_1, b_2 - d \cdot c_2, \dots, b_k - d \cdot c_k$.

Sólo queda por determinar el punto O, que es el punto que pertenece a la trayectoria original pero que también está en la línea de restricción (o plano). De la ecuación 3.5 se sabe que $x_j = \rho b_j$ para $j = 1, 2, \dots, k$. Pero también se sabe que para el punto O, la ecuación de restricción (ecuación 3.7) se debe mantener. Como resultado la colisión entre la trayectoria original de MPA y el plano restringido debe ocurrir para $\rho = \rho_0$, satisfaciendo

$$c_0 + (c_1 \cdot b_1 + c_2 \cdot b_2 + \dots + c_k \cdot b_k) \cdot \rho_0 = 0$$

Así,

$$\rho_0 = \frac{-c_0}{\sum_{i=1}^k c_i \cdot b_i} \quad (3.9)$$

Esta ecuación en términos de variables no codificadas queda,

$$\rho_0 = \frac{-c_0 - \left(\sum_{i=1}^k c_i \cdot w_i\right)}{\sum_{i=1}^k c_i \cdot b_i} \quad (3.10)$$

Como resultado, la trayectoria modificada comienza en $x_{j,0} = \rho_0 b_j$ (para $j = 1, 2, \dots, k$) y se mueve a través de ella, estando definida por

$$x_{j,m} = x_{j,0} + \lambda (b_j - d \cdot c_j) \quad (j = 1, 2, \dots, k) \quad (3.11)$$

donde d esta dada por la ecuación 3.8 y λ es simplemente una constante de proporcionalidad. Nótese que la trayectoria modificada es parecida a la trayectoria estándar excepto que no comienza en el origen y el coeficiente de regresión b_j es reemplazado por $b_j - d \cdot c_j$ de manera que la restricción sea acomodada.

3.2. Ejemplos de aplicación de Superficies de Respuestas

MSR se puede ilustrar mejor con ejemplos. Primero, algunos ejemplos cualitativos simples serán dados con uno y dos parámetros ($n = 1$ y $n = 2$), seguido por un ejemplo cuantitativo con tres parámetros ($n = 3$).

3.2.1. Ejemplos Cualitativos

Ejemplo para $n = 1$

Dado: Respuesta $y = y(a)$. Aquí, $n = 1$ ya que solo existe un parámetro en el problema. El experimento comienza en algún punto de operación arbitrario, por ejemplo en algún valor

de a , $a = a_0$.

Hacer: Aplique MSR para encontrar el valor óptimo de y , por ejemplo, encuentre el valor del parámetro a donde y es un máximo. *

Solución: Imagine una gráfica de y en función de a , con condición de operación inicial a_0 indicado por el punto rojo:

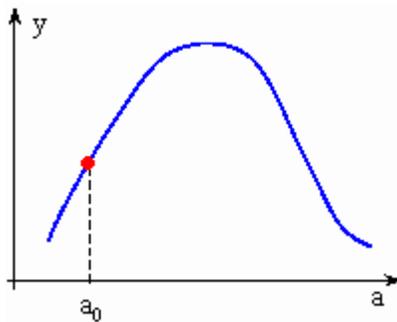


Figura 3.2: y en función de a .

En un experimento real, tal gráfica no estaría disponible, pero se usa aquí para propósitos ilustrativos.

Con solo un parámetro ($n = 1$) como en este ejemplo simple, todo lo que se necesita saber es si buscar a la izquierda o a la derecha. Una de estas es la **dirección de MPA**. Para encontrar esta dirección de MPA, se debe medir y en un par de puntos de referencia en la vecindad a_0 , tal como se ilustra en la gráfica 3.3 por los puntos azules.

Otros dos puntos de referencia, además de estos, en $a = a_0$, son suficientes para este caso simple. Si existe mucha dispersión en los datos, se puede requerir la repetición del proceso.

Un **análisis de regresión lineal** es realizado en los datos, encontrando la intercepción y la pendiente de y , según como se muestra en la figura 3.3.

- Si la pendiente es positiva (y se incrementa con a), la dirección de MPA es a la derecha (hacia el aumento de a).

*Note que esto es un ejemplo cualitativo solamente.

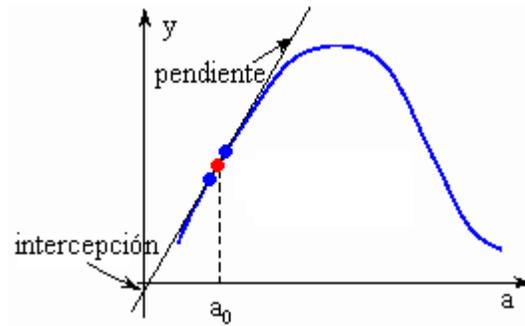


Figura 3.3: Dirección de Máxima Pendiente en Ascenso.

- Si la pendiente es negativa (y decrece con a), la dirección de MPA es a la izquierda (hacia la disminución de a).

Aquí, la pendiente es positiva. Tal y como se muestra en la figura 3.3, la tangente es muy cercana a la curva $a = a_0$, pero llega a ser absolutamente inexacta cuando el parámetro a está lejos de a_0 , (en el punto máximo, la pendiente es cero). Sin embargo, se asume que la dirección de MPA señala hacia la meta deseada. Entonces, se debe mover en esa dirección (con algún incremento razonable del parámetro a) hasta que no siga incrementándose, es decir, se comienza a disminuir, según lo demostrado en la figura 3.4

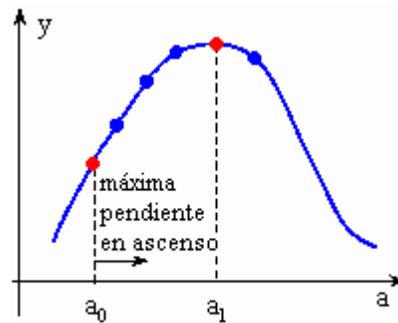


Figura 3.4: Movimiento de Máxima Pendiente en Ascenso.

Después de pocas corridas experimentales, definidas por los puntos azules en la figura 3.4, se encuentra que y alcanzó un máximo (el segundo punto rojo) en $a = a_1$. Este punto de operación (en $a = a_1$) puede que no sea el verdadero óptimo, pero está más cercano de lo que estaba el punto de operación original (en $a = a_0$). Se puede repetir el procedimiento de MSR, usando $a = a_1$ como el nuevo punto de arranque, para encontrar eficientemente el

punto de operación óptimo.

Este ejemplo es muy sencillo, pero el principio se aplica a un problema de optimización con cualquier número de parámetros.

Ejemplo para $n = 2$

Dado: Respuesta $y = y(a, b)$. Aquí, $n = 2$ ya que existen dos parámetros en el problema. El experimento comienza en algún punto de operación arbitrario, por ejemplo en algún valor $a = a_0$ y $b = b_0$.

Hacer: Aplique MSR para encontrar el valor óptimo de y , es decir, encuentre los valores de los parámetros a y b donde y sea máximo.

Solución: Imagine una **gráfica de contornos** de y en función de a y b (el valor máximo de y está en el centro de los contornos). La condición de operación inicial en (a_0, b_0) está indicada por el punto rojo como se observa en la figura 3.5.

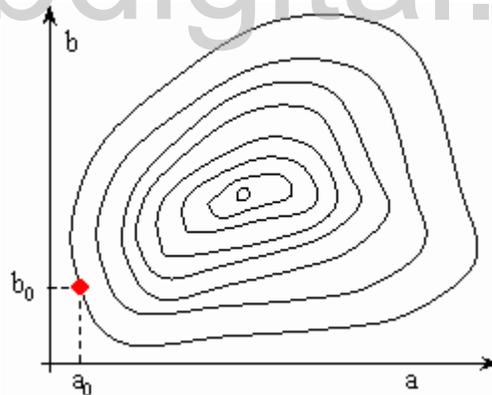


Figura 3.5: Gráfica de Contornos.

Con dos parámetros ($n = 2$) al igual que en el ejemplo simple, se necesita encontrar la **dirección de MPA**, el cual es un **vector**, en el plano a, b . Para encontrar esta dirección de MPA, se debe medir y en algunos puntos de referencia en la vecindad de (a_0, b_0) , como se ilustra por los puntos azules en la figura 3.6:

Por lo menos tres puntos de referencia, además del centro (a_0, b_0) , son requeridos para una

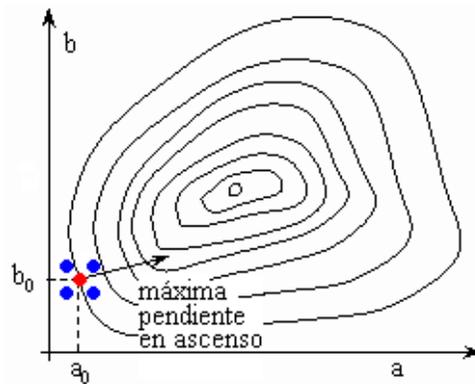


Figura 3.6: Vector de Máxima pendiente en Ascenso.

regresión lineal razonable cuando $n = 2$. Cuatro se demuestran en la figura 3.6. Si existe mucha dispersión en los datos.

Un análisis de regresión lineal multivariable es realizado con los datos. Se encuentran la intercepción de y y dos pendientes, una con respecto al parámetro a y la otra respecto al parámetro b . El vector producido por estas pendientes se muestra en la figura 3.6.

- Si la pendiente con respecto al parámetro a es positiva (y se incrementa con a), la dirección de MPA es a la derecha (hacia el aumento de a).
- Si la pendiente con respecto al parámetro a es negativa (y decrece con a), la dirección de MPA es a la izquierda (hacia la disminución de a).
- Si la pendiente con respecto al parámetro b es positiva (y se incrementa con b), la dirección de MPA es para arriba (hacia el aumento de b).
- Si la pendiente con respecto al parámetro b es negativa (y decrece con b), la dirección de MPA es a para abajo (hacia el disminución de b).

Con $n = 2$ parámetros, sin embargo, no sólo se sabe simplemente la dirección (derecha o izquierda, arriba o abajo), se conoce también el **vector de dirección de MPA**, según lo mostrado en la figura 3.6. Aquí la dirección de MPA es a la derecha y hacia arriba, en un cierto ángulo pequeño sobre el horizonte.

Como se ve en la figura 3.6, en las cercanías de (a_0, b_0) la dirección es perpendicular a las curvas de nivel (en un punto máximo, ambas pendientes son cero). Sin embargo, se asume que la dirección de MPA apunta hacia la meta deseada.

Luego se escogen algunos incrementos razonables para ambos parámetros. En la práctica, se escoge el incremento para uno de los parámetros, y el otro o los otros parámetros se calculan basados en la dirección de MPA. Típicamente, se escoge el incremento para el parámetro con la magnitud de pendiente más grande. En este problema, se asume que es el parámetro a . Así, el incremento Δa es seleccionado, ni muy pequeño ni muy grande, sino un valor razonable, de acuerdo con el experimento.

- Si la pendiente de y respecto al parámetro a es positivo, un valor positivo de Δa es escogido.
- Si la pendiente de y con respecto al parámetro a es negativa, un valor negativo de Δa es escogido.

Cuando el incremento en el parámetro a es escogido, el incremento en el parámetro b es calculado, de forma tal que el vector $(\Delta a, \Delta b)$ apunte en la dirección de MPA. Ahora se debe mover en la dirección de MPA hasta que y deje de incrementarse, es decir comience a decrecer, según lo demostrado en la figura 3.7:

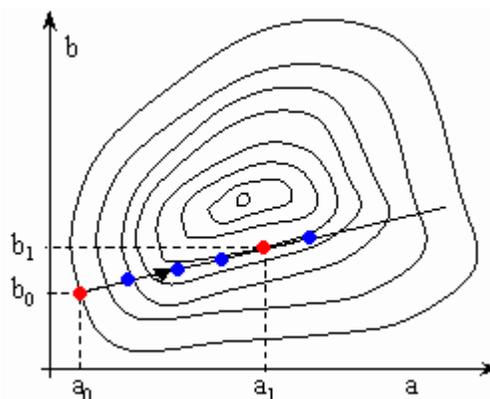


Figura 3.7: Movimiento de Máxima Pendiente en Ascenso en dos Dimensiones

Después de pocas corridas experimentales, definidas por los puntos azules en la gráfica, se

encuentra que y ha alcanzado un máximo (el segundo punto rojo) en (a_1, b_1) . (El próximo punto azul tiene un valor bajo de y , como puede ser visto en las líneas del contorno.) Este punto de operación (a_1, b_1) no es el **verdadero óptimo**, pero está mucho más cercano que el punto de operación original (a_0, b_0) . Se puede repetir el procedimiento de MSR, usando (a_1, b_1) como el nuevo punto de arranque, para encontrar eficientemente el punto de operación óptimo.

En el segundo paso, una nueva dirección de MPA es calculado tomando medidas en los puntos de referencias en la vecindad de (a_1, b_1) , según lo bosquejado abajo. Aquí, cuatro medidas son mostradas por los puntos azules alrededor del punto rojo en (a_1, b_1) :

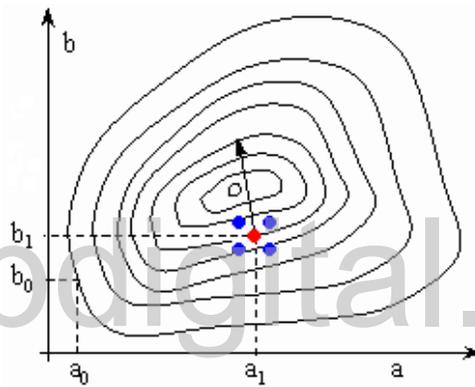


Figura 3.8: Nueva Dirección de Máxima Pendiente en Ascenso.

En la figura 3.8, la nueva dirección de MPA se indica por la flecha. Ahora se debe mover en esta nueva dirección hasta que la respuesta y comience a disminuir. Según lo visto en la gráfica, el verdadero punto óptimo todavía no se ha encontrado en este segundo paso, pero seguramente está más cerca.

3.2.2. Ejemplo Cuantitativo

Para esta parte se usa un problema de programación lineal del libro de Investigación de Operaciones [20, p. 151], con cambios de los parámetros x_1, x_2, x_3 por a, b, c .

Dado: El problema lineal,

Maximice $y = 10a + 12b - 2c$

Sujeto a las restricciones:

$$a + 2b - c \leq 320$$

$$3a + b \leq 350$$

$$c \leq 10$$

$$a, b, c \geq 0$$

Se decide tomar como punto de operación $(a_0, b_0, c_0) = (7, 14, 2)$. En este punto de operación, el valor de y es 188,4 (en unidades que representan alguna medida de calidad).

Hacer: Usar MSR para optimizar. Para los cálculos en este problema, se usó el programa Microsoft EXCEL. El valor máximo de la función es 2360, la cual ocurre en $(a, b, c) = (70, 140, 10)$, tal y como lo demuestra la solución mediante la herramienta SOLVER de EXCEL en la figura 3.9. Para este ejemplo, se pretende que esta información es desconocida.

Microsoft Excel 11.0 Informe de respuestas
Hoja de cálculo: [prueba.xls]Hoja1

Celda objetivo (Máximo)

Celda	Nombre	Valor original	Valor final
\$G\$2	max	0	2360

Celdas cambiantes

Celda	Nombre	Valor original	Valor final
\$C\$2	a	0	70
\$D\$2	b	0	140
\$E\$2	c	0	10

Figura 3.9: Informe de respuestas generado por SOLVER.

Solución: Aquí, $n = 3$ (3 parámetros), es decir $y = y(a, b, c)$. Siguiendo el procedimiento MSR, un análisis de regresión lineal se realiza en un parte de los puntos de referencia en la vecindad del punto de operación inicial. Se construye un diseño experimental de 2-niveles, 3-parámetros. Este requiere 4 puntos de datos en la vecindad del punto de operación inicial.

Un incremento de 20% es escogido entre los niveles bajo y medio, medio y alto para todos los parámetros a, b, c . En este caso los niveles del parámetro a serán 5,6 y 8,4; del parámetro b 11,2 y 16,8; y para el parámetro c 1,6 y 2,4. También se calcularon las variables

codificadas x_1 , x_2 , y x_3 tal y como lo muestra la tabla 3.1.

	a	b	c	y	x_1	x_2	x_3
	5,6	11,2	1,6	187,2	-1	-1	-1
	5,6	11,2	2,4	185,6	-1	-1	1
	5,6	16,8	1,6	254,4	-1	1	-1
	5,6	16,8	2,4	252,8	-1	1	1
	8,4	11,2	1,6	215,2	1	-1	-1
	8,4	11,2	2,4	213,6	1	-1	1
	8,4	16,8	1,6	282,4	1	1	-1
	8,4	16,8	2,4	280,8	1	1	1
máximo	8,4	16,8	2,4				
mínimo	5,6	11,2	1,6				
rango	2,8	5,6	0,8				
valor medio	7	14	2				

Tabla 3.1: Tabla de Resultados, valores originales y codificados

Un análisis de regresión se realiza con los ocho valores de y que aparecen en la tabla, y con los valores de x_1 , x_2 y x_3 correspondientes. En EXCEL se marcan las celdas que contienen estos datos para realizar el análisis de regresión lineal. Una porción de la salida del análisis de regresión de EXCEL se muestra en la figura 3.10.

	<i>Coefficientes</i>
Intercepción	234
Variable X 1	14
Variable X 2	33,6
Variable X 3	-0,8

Figura 3.10: Porción del análisis de regresión en EXCEL.

Los resultados de interés son las pendientes de y con respecto a las variables codificadas x_1 , x_2 y x_3 , las cuales son 14; 33,6 y -0,8 respectivamente. Matemáticamente, éstas son las **derivadas parciales** de y con respecto a x_1 , x_2 y x_3 .

$$\frac{\partial y}{\partial x_1} = 14; \quad \frac{\partial y}{\partial x_2} = 33,6; \quad \frac{\partial y}{\partial x_3} = -0,8$$

Estos tres valores representan la dirección de MPA en términos de las variables codificadas;

es decir la dirección de MPA es en la dirección del vector

$$\left(\frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \frac{\partial y}{\partial x_3} \right)$$

Para este conjunto de datos, la dirección de MPA es en la dirección del vector (14; 33,6; -0,8). La magnitud del vector no es importante, es la dirección lo que importa en realidad. Se calculan los ajustes de los factores en la dirección de MPA con $\rho = 1$, dando como resultado los incrementos $\Delta x_1 = 0,384$; $\Delta x_2 = 0,923$; $\Delta x_3 = -0,022$ aproximadamente. Con estos resultados se asegura que la dirección de movimiento es la misma dirección de MPA.

Para moverse en la dirección de MPA, se debe primero transformar los incrementos en las variables físicas, a , b y c . Usando las conversiones definidas anteriormente, los incrementos en los parámetros a , b y c se calculan, los cuales son 0,538; 2,584 y -0,009 respectivamente.

La dirección de MPA en las variables originales (dimensionales, no-codificadas, físicas) han sido encontradas, siendo la dirección en la cual los parámetros a , b y c son incrementados por $\Delta a = 0,538$; $\Delta b = 2,584$; $\Delta c = -0,009$. Teniendo en cuenta las restricciones:

$$\begin{aligned} a + 2 \cdot b - c &\leq 320 \\ 3 \cdot a + b &\leq 350 \\ c &\leq 10 \\ a &\geq 0 \\ b &\geq 0 \\ c &\geq 0 \end{aligned}$$

se deben transformar primero los parámetros contenidos en ellas a variables codificadas, esto nos da como resultado:

$$\begin{aligned}
 1,4 \cdot x_1 + 5,6 \cdot x_2 - 0,4 \cdot x_3 &\leq 287 \\
 4,2 \cdot x_1 + 2,8 \cdot x_2 &\leq 315 \\
 &0,4 \cdot x_3 \leq 8 \\
 1,4 \cdot x_1 &\geq 7 \\
 &2,8 \cdot x_2 \geq 14 \\
 &0,4 \cdot x_3 \geq 2
 \end{aligned}$$

Con estas nuevas restricciones codificadas, se procede a calcular ρ_0 para cada restricción aplicando la ecuación 3.9, luego se escoge el ρ_0 positivo de menor valor de la restricción para luego continuar con el calculo de d según la ecuación 3.8, con esto obtenemos por medio de la ecuación 3.11 el comienzo de la trayectoria modificada por esta restricción y el movimiento a través de esta trayectoria.

En este problema de la restricción $1,4 \cdot x_1 + 5,6 \cdot x_2 - 0,4 \cdot x_3 \leq 287$ se obtiene $\rho_0 = 1,379$ siendo la de menor valor positivo entre las demás restricciones, escogiéndose así esta de la cual se obtiene $d = 6,215$,

$$\begin{aligned}
 x_{1,m} &= 1,379 \cdot (14) + \lambda ((14) - (6,215) \cdot (1,4)) = 19,310 + \lambda \cdot 5,299 \\
 x_{2,m} &= 1,379 \cdot (33,6) + \lambda ((33,6) - (6,215) \cdot (5,6)) = 46,344 + \lambda \cdot -1,204 \\
 x_{3,m} &= 1,379 \cdot (-0,8) + \lambda ((-0,8) - (6,215) \cdot (-0,4)) = -1,103 + \lambda \cdot 1,686
 \end{aligned}$$

se asume $\lambda = 1$.

Ahora los tres parámetros son incrementados simultáneamente, y y es medido. Este proceso de movimiento se continua hasta que y comience a decrecer o se tope con la trayectoria modificada, en la tabla 3.2 se puede observar esto.

Nótese que el punto de inicio es $(7, 14, 2)$, que son las condiciones de operación original. El parámetro a es incrementado por 0,538, el parámetro b por 2,584, y el parámetro c por -0,009. El movimiento es hecho con las tres variables simultáneamente, en la dirección de MPA.

	x1	x2	x3	a	b	c	Resultado	
base	0	0	0	7	14	2	234	
Δ	0,384	0,923	-0,022	0,538	2,584	-0,009		
1 Δ	0,384	0,923	-0,022	7,538	16,584	1,991	270,409	
2 Δ	0,769	1,846	-0,044	8,077	19,168	1,982	306,818	
3 Δ	1,154	2,769	-0,066	8,615	21,752	1,973	343,226	
4 Δ	1,538	3,691	-0,088	9,153	24,336	1,965	379,635	
50 Δ	19,226	46,146	-1,099	33,917	143,200	1,560	2054,439	
51 Δ	19,611	47,066	-1,121	34,455	145,784	1,552	2090,848	<i>sepasa</i>
$x_{j,0}$	19,310	46,344	-1,103	34,034	143,762	1,559	2062,370	
1 · $x_{j,m}$	24,609	45,139	0,583	41,452	140,390	2,233	2094,741	
2 · $x_{j,m}$	29,908	43,935	2,269	48,871	137,018	2,907	2127,113	
3 · $x_{j,m}$	35,207	42,731	3,955	56,289	133,646	3,582	2159,485	
4 · $x_{j,m}$	40,506	41,526	5,641	63,708	130,274	4,256	2191,856	
5 · $x_{j,m}$	45,804	40,322	7,327	71,126	126,902	4,931	2224,228	OPTIMO

Tabla 3.2: Tabla de movimiento en trayectoria máximo ascenso y modificada

La respuesta y aumenta con cada incremento hasta que se encuentra con la trayectoria modificada, en ese momento debe comenzar a moverse de acuerdo a los valores de ésta.

Nótese que un siguiente movimiento deja de ser factible al no cumplir con la segunda restricción, un simple cálculo con los parámetros lo demuestra. Sin embargo el proceso de la Metodología de Superficies de Respuesta (MSR) ha incrementado la respuesta y satisfactoriamente desde un valor inicial de 234 hasta un nuevo valor de 2224,228.

Dado que la respuesta obtenida fue hecha en la trayectoria modificada, significa que en el proceso de optimización se tuvo contacto con la restricción y se procedió a moverse a través de esta, significa entonces que se debe observar si esta trayectoria modificada tiene contacto con otra.

Capítulo 4

Desarrollo de la Herramienta RSOptim

Este Capítulo presenta la herramienta de optimización creada, la cual se hizo en forma conjunta con el bachiller Freddy Marcano [13] en su proyecto de grado, aplicando metodologías diferentes pero siendo común su función de comunicación con el software ARENA.

4.1. Optimización de Modelos en ARENA

Supóngase que el funcionamiento de un sistema de interés depende de los valores de parámetros escogidos en el sistema, y dicho sistema es de una gran complejidad en la cual es necesario usar simulación para estimar el funcionamiento de éste en cada conjunto de valores de parámetros.

En ARENA se consideran como entradas a todos los recursos y las variables creadas por el usuario en el modelo, de las cuales se pueden elegir todas o algunas. Como salidas se consideran tres tipos generadas en el programa que son: DSTATS, TALLIES y OUTPUTS. La tecnología interna de ARENA es el lenguaje de simulación SIMAN. Los módulos contenidos en las plantillas de ARENA fueron creados usando bloques de modelado de

SIMAN como sus componentes. A través de los módulos de SIMAN se puede tener acceso a los valores de las entradas y a los valores de salida en el modelo que está analizando. A continuación se introducen en un breve resumen las entradas y salidas.

Entradas:

Los elementos RESOURCES definen las características de los recursos, incluyendo los nombres de recurso y las capacidades iniciales. Cada tipo diferente de recurso tiene un Número opcional, un Nombre único, y una Capacidad. La Capacidad representa el número de unidades idénticas del nombre del recurso que están disponibles, las cuales son representados por Capacidad u Horario.

Los elementos VARIABLES especifican el número total de variables globales, sus nombres y, si se desea, valores iniciales. Se hace referencia a las variables tanto usando sus nombres como usando el arreglo de variable $V(k)$ donde k es calculado como el índice dentro del elemento VARIABLES.

Salidas:

Estadísticas Tiempo-Persistente (DSTATS) y de Cuenta (TALLY): Para múltiples réplicas, seis valores son mostrados:

- Average: es el promedio de los promedios de cada réplica,
- Half Width: calculado a través de réplicas,
- Minimum Average: el promedio más pequeño a través de todas las réplicas,
- Maximum Average: el promedio más grande a través de todas las réplicas,
- Minimum Value: el valor más pequeño observado a través de todas las réplicas,
- Maximum Value: el valor más grande observado a través de todas las réplicas.

Para una sola réplica, cuatro valores son mostrados:

- Average: el promedio de la réplica,

- Half Width: calculado dentro de la réplica,
- Minimum Value: el valor más pequeño observado,
- Maximum Value: el valor más grande observado.

Los elementos DSTATS son usados para obtener las estadísticas de tiempo-persistente en un sistema discreto. Estas estadísticas pueden ser obtenidas para cualquier variable de usuario, variable de sistema, o expresión que contenga variables del sistema o del usuario. Las expresiones que contengan atributos no son permitidas.

Los elementos TALLIES definen toda la información necesaria para los bloques TALLY en un modelo SIMAN. Los bloques TALLY hacen referencia a cuentas individuales a través del número o nombre de tally. El nombre de tally también aparece en el Reporte Sumario SIMAN como el identificador tally.

Contadores y Salidas (Outputs): Para múltiples réplicas, cuatro valores son mostrados:

- Average: el promedio del final de la réplica/valores actuales,
- Half Width: calculado a través de réplicas,
- Minimum Average: el valor más pequeño a través de todas las réplicas,
- Maximum Average: el valor más grande a través de todas las réplicas.

Para una sola réplica, un valor es mostrado:

- Valor: valor final/valor actual de la réplica.

Los elementos OUTPUTS definen expresiones de SIMAN cuyos valores deben ser mostrados en el Reporte Sumario SIMAN y opcionalmente grabados en los archivos o reportes de salida al final de cada réplica de una simulación.

Todos estos elementos son fácilmente manipulados a través de SIMAN, por medio del acceso a una librería dinámica que provee ARENA para su comunicación con Visual Basic.

4.2. Interfaz RSOptim

Para la construcción de la herramienta de optimización para modelos de simulación creados en ARENA, se utilizó el programa Microsoft Visual Basic en la programación de la interfaz gráfica la cual tiene la capacidad de la obtención de datos desde ARENA y manipulación de los mismos. El programa de simulación ARENA provee una librería de enlace dinámico (DLL) que es usada en el programa Visual Basic, esta librería permite un manejo casi total del programa ARENA.

En el programa Visual Basic se puede tener uso de esta DLL activándola desde Proyecto→Referencias, allí al tener instalado el programa ARENA en el computador debe aparecer la DLL bajo el nombre de “ARENA 5.0 Type Library”, la cual se selecciona para poder comenzar a usar todas la herramientas de programación. En la figura 4.1 se ilustra lo anterior, nótese que en la parte inferior se detalla la ubicación del archivo de la librería bajo el nombre de “SMSYS.DLL”.

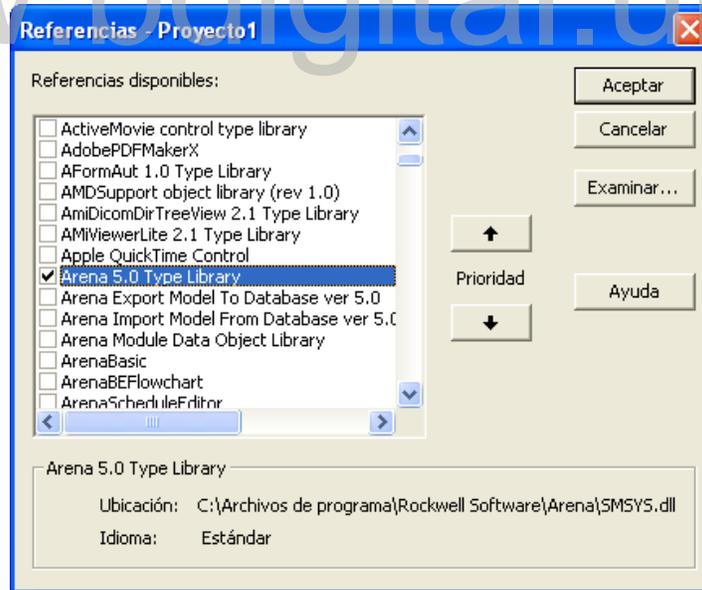


Figura 4.1: Referencias en Microsoft Visual Basic.

A continuación se muestran algunos ejemplos de uso de la librería desde Visual Basic, aquí sólo se muestran algunas sentencias usadas para la construcción de la interfaz, pero obviamente se usaron muchas más, esto sólo sirve como referencia dado que la forma de

uso de cualquier clase en la librería es la misma, sólo se debe conocer su funcionamiento.

La sentencia “Application.ActiveModel” permite utilizar un conjunto de clases y miembros de clases dentro de Visual Basic que se encargan de realizar diferentes funciones en ARENA aplicadas al modelo de simulación que se encuentre activo (ActiveModel). Visual Basic permite visualizar y utilizar todas estas clases una vez se escriba esta sentencia como se puede ver en la figura 4.2, la librería que se usa de ARENA tiene dos clases que son esenciales en la interfaz, de éstas se usan varios miembros que permiten lectura, escritura y control de ciertos procesos en ARENA, una forma más detallada se presenta a continuación.

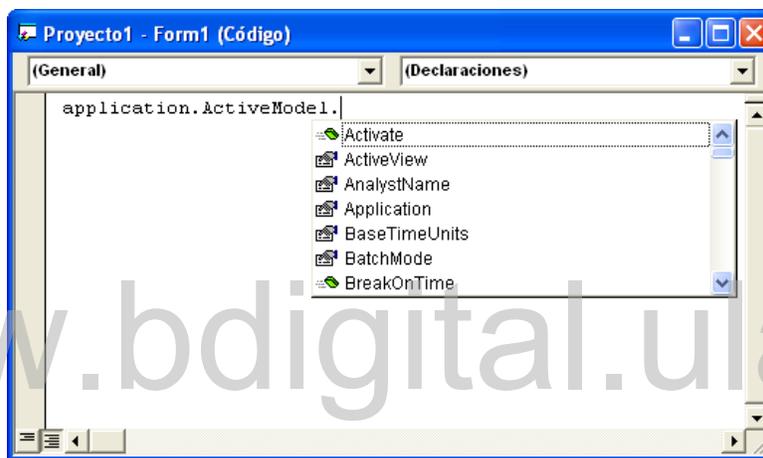


Figura 4.2: Acceso a ARENA desde Microsoft Visual Basic.

Clase “Model”

Esta clase ya está activa, dado que ActiveModel obtiene todas las características de la clase Model como se ve en Visual Basic (Property ActiveModel As Model), en esta se pueden citar los siguientes miembros:

StartOver: Es una subrutina que chequea e inicializa el modelo de simulación, pero no comienza a correr la simulación, es esencial para poder tener el modelo en un estado de inicio que permita usar otras funciones como las incluidas en la clase SIMAN.

QuietMode: Es una propiedad que acepta un booleano, si se le asigna el valor “True” (verdadero) se activa la propiedad que evita que aparezca cualquier tipo de ventana

de mensajes durante la ejecución de ARENA. Muy importante para evitar mensajes inesperados desde ARENA durante la ejecución de la interfaz.

BatchMode: Es una propiedad que acepta un booleano, si se le asigna el valor “True” (verdadero) se desactiva la propiedad de animación en la simulación. Necesario para que la interfaz se ejecute y mantenga a la simulación de ARENA en un segundo plano.

Clase “SIMAN”

Se llama a través de la línea de código “`Application.ActiveModel.SIMAN`”, es la clase que permite el uso de las propiedades del lenguaje SIMAN que es parte integral de ARENA, teniendo control de muchas operaciones internas de ARENA. Tal y como se vio en la figura 4.2 esta línea permite visualizar un grupo de miembros entre los cuales se pueden citar los siguientes:

`ConstructString`: Con este miembro de la clase SIMAN se obtienen las cadenas de nombres de los constructores creados por ARENA, posee una gran cantidad de submiembros de los cuales los de mayor importancia son:

- `smSimanConstructResources`, *i*: el cual muestra el nombre de cada *i*-ésimo recurso.
- `smSimanConstructVariables`, *i*: igual que el anterior pero para las todas las variables implicadas en el sistema.

Para la interfaz se usan muchos otros códigos en Visual Basic con la ayuda de la librería dinámica de ARENA, lo anterior sirve como referencia.

4.2.1. Estructura de RSOptim

La interfaz RSOptim consta de cuatro partes generales que son llamadas a través de los botones en la parte superior de la ventana principal del programa, tal y como se muestra

en la figura 4.3, las partes de “Restricciones” es donde son ingresadas la o las restricciones que se implementarán en el sistema o modelo en estudio, la parte “Opciones” muestra la opción para cambiar el número de replicas del modelo. Las figuras 4.4 y 4.5 muestran las ventanas de las Restricciones y Opciones respectivamente.



Figura 4.3: Ventana Principal en RSOptim.



Figura 4.4: Ventana Restricciones en RSOptim.

Controles

En la interfaz en la parte de controles se le solicita a ARENA dos conjuntos de datos que son “Variables” y “Recursos”, los nombres de estos conjuntos son obtenidos a través de los comandos anteriormente descritos en la clase SIMAN, y para obtener sus respectivos valores se usan las líneas de código:

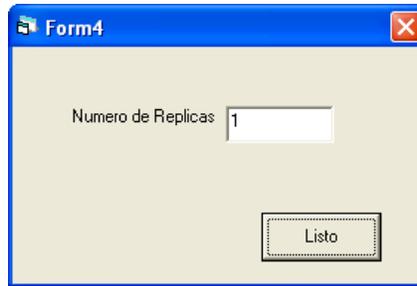


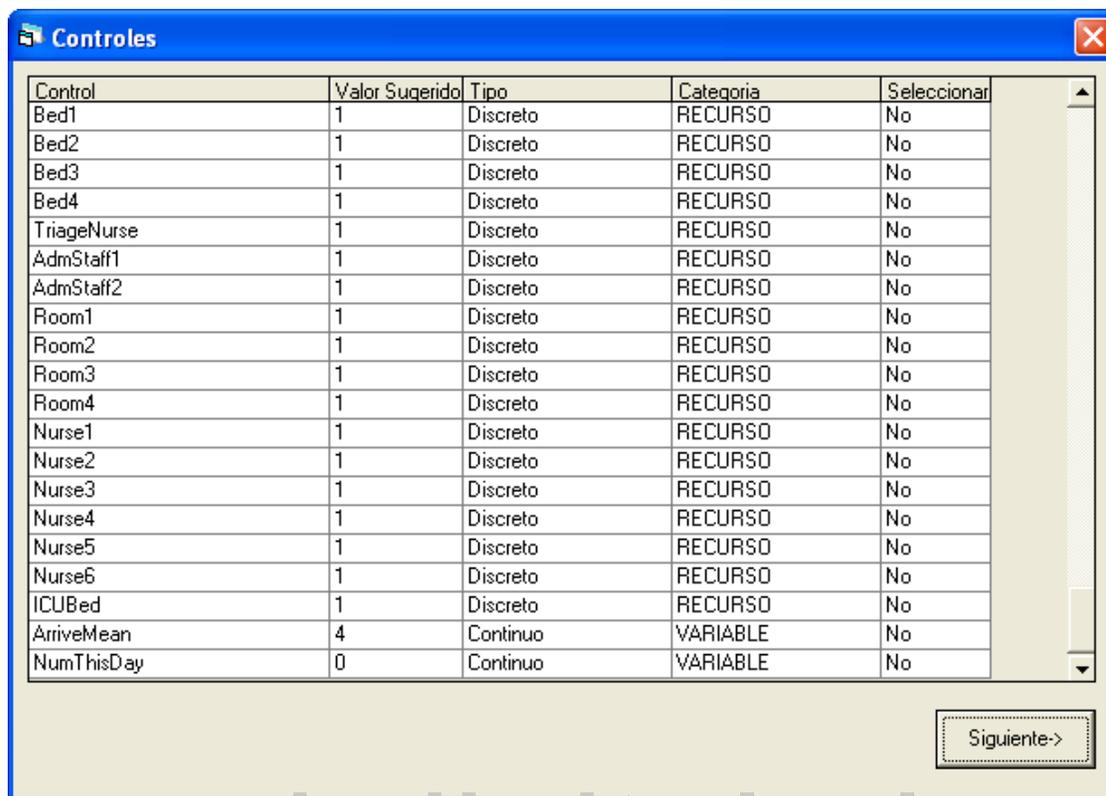
Figura 4.5: Ventana Opciones en RSOptim.

- `Application.ActiveModel.SIMAN.ResourceCapacity(i)`, para cada i -ésimo recurso.
- `Application.ActiveModel.SIMAN.VariableArrayValue(i)`, para cada i -ésima variable.

Estos datos son organizados junto a otras opciones en un objeto de tablas dinámicas, la cual esta compuesta por celdas y estas son altamente personalizables al poder modificar una gran cantidad de propiedades, permitiendo crear desde una simple rejilla de información, hasta una compleja aplicación donde se pueden modificar datos dentro de las celdas, obsérvese la figura 4.6.

Objetivo

En esta ventana se muestran todas las salidas, las creadas por el usuario en forma de variables y las generadas por ARENA, de estas se puede elegir sólo una para minimizar o maximizar dependiendo del caso que necesite el analista. Una vez más se usa el objeto de tablas dinámicas para mostrar los datos y las opciones de minimizar y maximizar tal y como lo muestra la figura 4.7



Control	Valor Sugerido	Tipo	Categoria	Seleccionar
Bed1	1	Discreto	RECURSO	No
Bed2	1	Discreto	RECURSO	No
Bed3	1	Discreto	RECURSO	No
Bed4	1	Discreto	RECURSO	No
TriageNurse	1	Discreto	RECURSO	No
AdmStaff1	1	Discreto	RECURSO	No
AdmStaff2	1	Discreto	RECURSO	No
Room1	1	Discreto	RECURSO	No
Room2	1	Discreto	RECURSO	No
Room3	1	Discreto	RECURSO	No
Room4	1	Discreto	RECURSO	No
Nurse1	1	Discreto	RECURSO	No
Nurse2	1	Discreto	RECURSO	No
Nurse3	1	Discreto	RECURSO	No
Nurse4	1	Discreto	RECURSO	No
Nurse5	1	Discreto	RECURSO	No
Nurse6	1	Discreto	RECURSO	No
ICUBed	1	Discreto	RECURSO	No
ArriveMean	4	Continuo	VARIABLE	No
NumThisDay	0	Continuo	VARIABLE	No

Figura 4.6: Ventana Controles en RSOptim.

4.3. Metodología Aplicada en RSOptim

En la construcción de esta herramienta de optimización se usa la metodología de superficies de respuestas, la cual es codificada a través de Visual C++ en una librería dinámica que se le da el nombre de “RSM.DLL”. En la construcción de la interfaz RSOptim en Visual Basic se tiene acceso a esta librería para realizar todos los cálculos matemáticos que se describen en el método de Superficies de Respuesta.

En Visual Basic se pueden usar las diferentes funciones que hayan sido construidas en una librería creada con lenguaje C++, mediante la línea de código:

```
Private Declare Function FUNC Lib "LIB.DLL" (X As Double, Y As Double)
```

donde FUNC es la función a utilizar de la librería LIB, X puede ser una variable de entrada

Selección	Respuesta	Valor
No	ICUBed.TimesUsed	FINAL
No	ICUBed.UsageCost	FINAL
No	ICUBed.Utilization	PROMEDIO
No	ICURoom.BusyCost	FINAL
No	ICURoom.IdleCost	FINAL
No	ICURoom.NumberBusy	PROMEDIO
No	ICURoom.NumberScheduled	PROMEDIO
No	ICURoom.ScheduledUtilization	FINAL
No	ICURoom.TimesUsed	FINAL
No	ICURoom.UsageCost	FINAL
No	ICURoom.Utilization	PROMEDIO
No	NumThisDay	FINAL
No	Nurse1.BusyCost	FINAL
No	Nurse1.IdleCost	FINAL
No	Nurse1.NumberBusy	PROMEDIO
No	Nurse1.NumberScheduled	PROMEDIO
No	Nurse1.ScheduledUtilization	FINAL
No	Nurse1.TimesUsed	FINAL
No	Nurse1.UsageCost	FINAL
No	Nurse1.Utilization	PROMEDIO
No	Nurse2.BusyCost	FINAL
No	Nurse2.IdleCost	FINAL
No	Nurse2.NumberBusy	PROMEDIO
No	Nurse2.NumberScheduled	PROMEDIO
No	Nurse2.ScheduledUtilization	FINAL
No	Nurse2.TimesUsed	FINAL

Figura 4.7: Ventana Objetivo en RSOptim.

para cálculos en la función y Y una variable de salida generada por la función.

Para la interfaz se crearon en la librería “RSM.DLL” las siguientes funciones que se explican a continuación:

- **GAUSS**: esta función recibe los valores creados desde Visual Basic ya codificados en forma de una matriz que contiene todas las interacciones de las variables de entrada, según el diseño factorial 2^k , escogidas para la optimización del modelo y un vector con los resultados obtenidos por cada interacción de las variables. Con esta matriz y vector se realiza el cálculo de regresión lineal según el método de Gauss.

- **DAPLI:** en esta función se realizan los cálculos requeridos para la obtención del paso para el método de máxima pendiente en ascenso, y con la función anterior ya se tiene la dirección para este método.
- **CRO:** aquí se lleva a cabo la recolección de las restricciones desde Visual Basic en forma de una matriz, que luego son usadas para calcular las diferentes trayectorias modificadas.

www.bdigital.ula.ve

Capítulo 5

Aplicación de RSOptim

En este capítulo se muestra la aplicación de la herramienta RSOptim a un modelo de simulación discreto, este estudio examina las alternativas para una atención automática en un pasillo del banco. El problema de optimización es minimizar el “costo total de los recursos” (cajeros y puntos de atención), mientras que se proporciona un servicio al cliente aceptable. Las opciones incluyen cuántos cajeros (Tellers) asignar a las transacciones de los clientes, cuántos puntos de atención (Kiosk) colocar en el banco, y qué proporción de clientes (Teller Percent) utilizarán los cajeros. En la siguiente figura 5.1 se puede apreciar el modelo construido en ARENA.

5.1. Aplicación en el Modelo

Una vez que se realiza la implementación en ARENA y probado el modelo que está en estudio, se inicializa la interfaz para obtener las entradas dentro de la herramienta creada. Se escogen tres variables: Kiosk, Tellers y Teller Percent y se fijan sus valores iniciales $(\text{Kiosk}, \text{Tellers}, \text{TellerPercent}) = (3, 5, 70)$ tal y como se muestra en la figura 5.2, se procede luego con un clic en el botón siguiente.

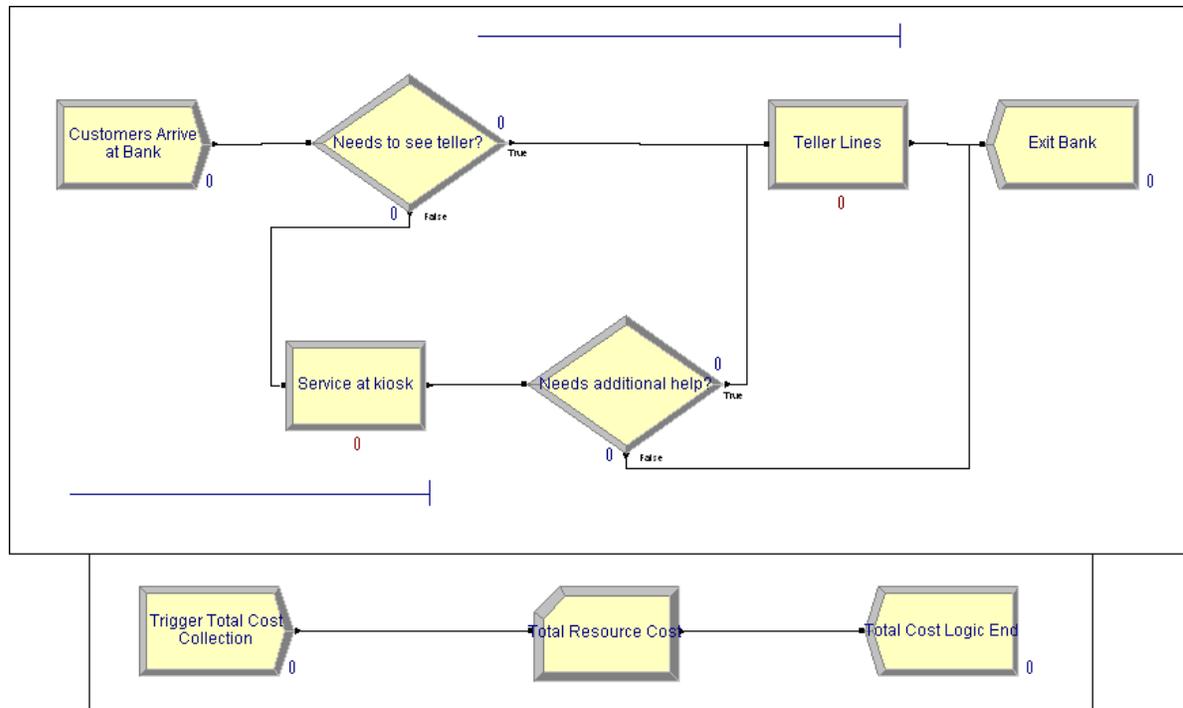


Figura 5.1: Modelo Estudiado en ARENA

Una vez escogidas las variables que se usan, se pasa a la ventana donde se introducen las restricciones del modelo en estudio. En la configuración de las restricciones que se analiza para este modelo en específico se debe tomar en cuenta el diseño y la implementación de la simulación, por ello se procede a explicar el uso de estas restricciones:

$$100 \cdot TellerPercent \geq 100$$

$$Kiosk + TellerPercent \geq 100$$

$$2 \leq Kiosk \leq 5$$

$$3 \leq Tellers \leq 8$$

$$4 \leq TellerPercent \leq 100$$

Se requiere para el modelo en estudio evitar un escenario donde el porcentaje de utilización de los cajeros por parte de los clientes sea 100 y donde siempre se use algún punto de atención. Además el proceso de optimización se debe mantener dentro de ciertos límites de requerimiento, los cuales están establecidos sobre el promedio de tiempo en espera para cajeros y puntos de atención. Las restricciones se establecen en la ventana de restricciones

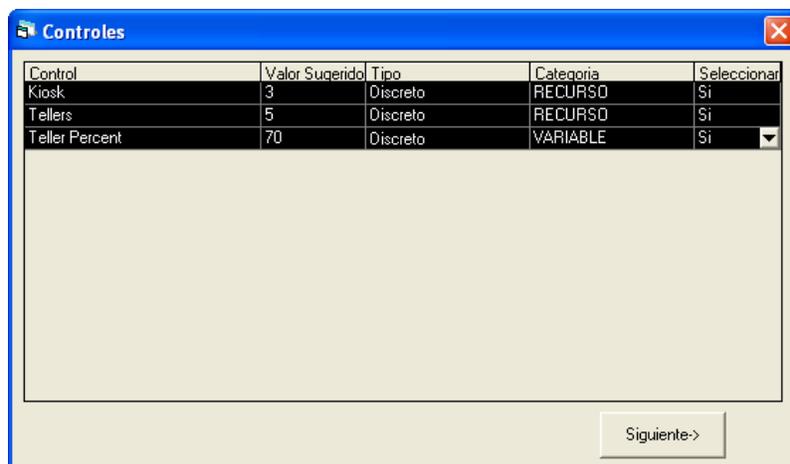


Figura 5.2: Selección de Entradas: Recursos y Variables

como se muestra en la figura 5.3, ahora se procede a la siguiente ventana.



Figura 5.3: Restricciones

Introducidas las restricciones se procede a escoger la salida a optimizar, tal y como se dijo anteriormente se requiere minimizar el costo total de los recursos (Total Resource Cost), se busca esta salida en la tabla de salidas que se muestra en la figura 5.4 para luego seleccionarla para una minimización.

Ya mostrado la elección de la función objetivo y continuando con el botón siguiente, la

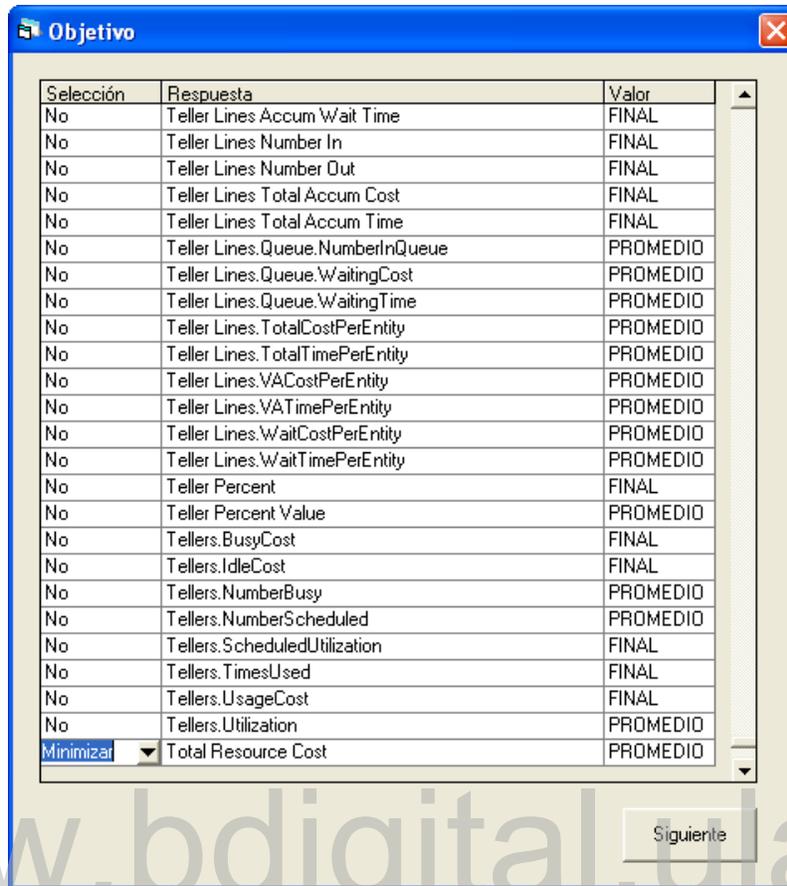
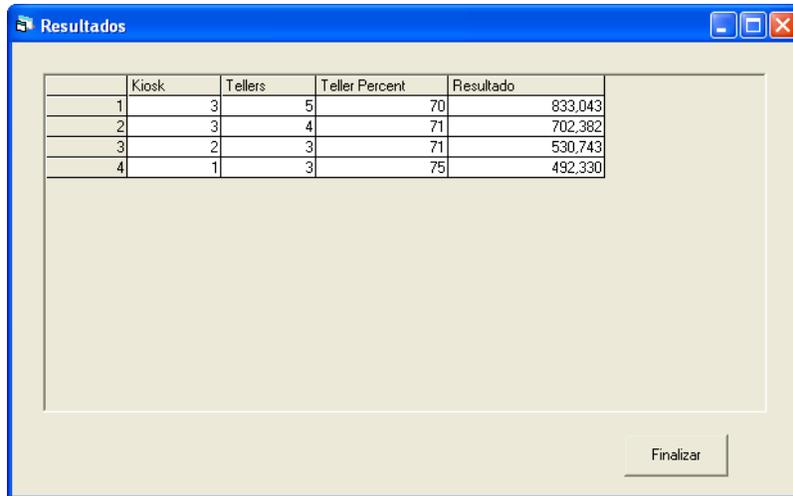


Figura 5.4: Función Objetivo

herramienta se sitúa en la ventana de opciones donde se pueden escoger el número de replicas, para el modelo en estudio se asignan 5 replicas, un clic en el botón “Listo” y comienzan los cálculos en la herramienta.

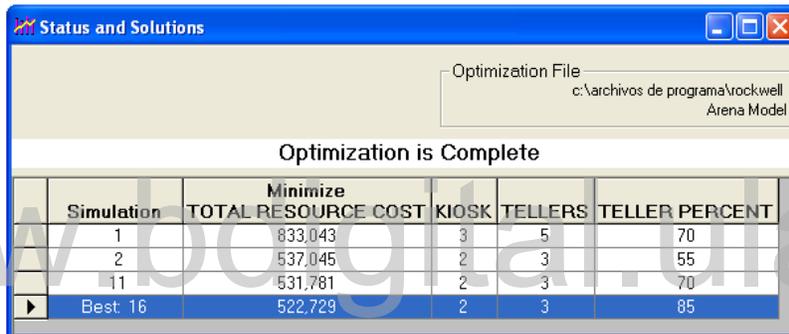
Realizados todos los cálculos a través de las diferentes iteraciones en la ejecución de esta herramienta, se muestran los resultados obtenidos de una manera satisfactoria, cumpliendo con todas las restricciones impuestas. Obsérvese la figura 5.5

Como referencia se usa el programa de optimización Optquest para analizar el mismo modelo con todos los parámetros iguales, dando como resultado lo indicado en la figura 5.6. **Debe señalarse que en este problema la herramienta RSOptim da mejores resultados que la herramienta OptQuest de ARENA.**



	Kiosk	Tellers	Teller Percent	Resultado
1	3	5	70	833,043
2	3	4	71	702,382
3	2	3	71	530,743
4	1	3	75	492,330

Figura 5.5: Ventana de Salida



Optimization File
c:\archivos de programa\rockwell
Arena Model

Optimization is Complete

Simulation	Minimize TOTAL RESOURCE COST	KIOSK	TELLERS	TELLER PERCENT
1	833,043	3	5	70
2	537,045	2	3	55
11	531,781	2	3	70
▶ Best 16	522,729	2	3	85

Figura 5.6: Ventana de Salida de OptQuest

Además se muestran las gráficas que permiten ver el comportamiento de las variables y de la función objetivo (salida del sistema).

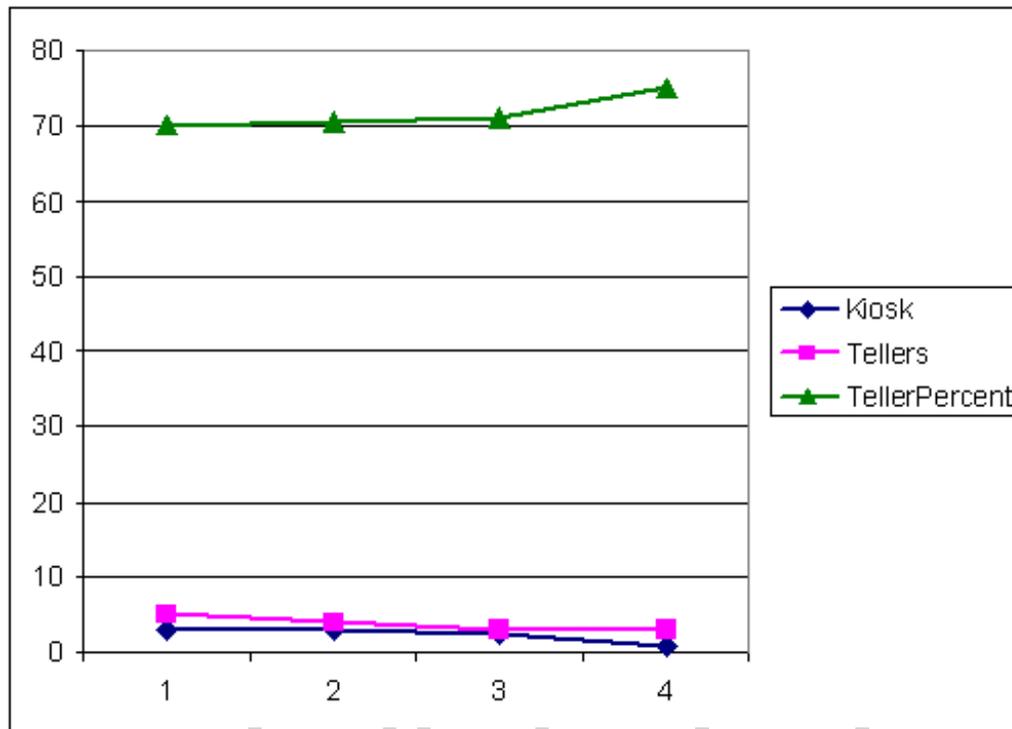


Figura 5.7: Salida de las Variables de RSOptim

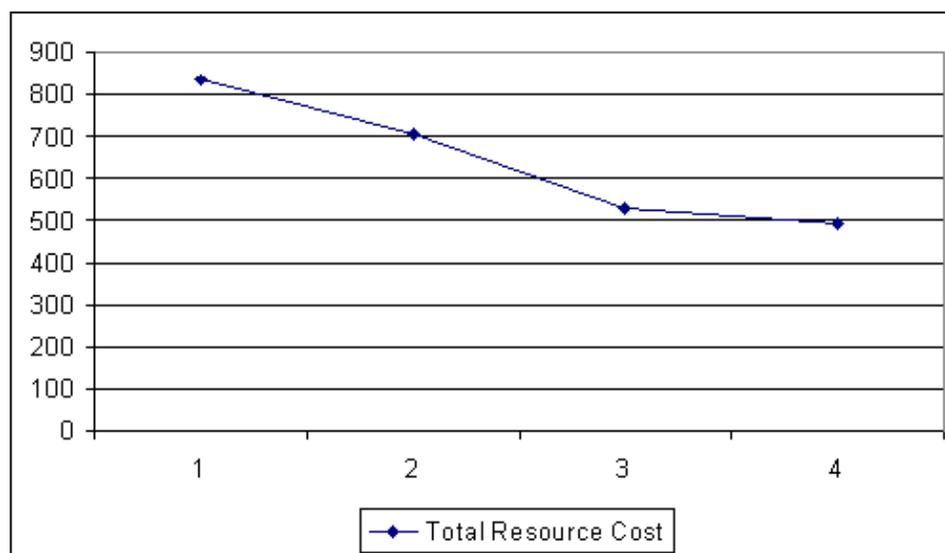


Figura 5.8: Salida Función Objetivo de RSOptim

Capítulo 6

Conclusiones y Recomendaciones

En este capítulo se presentan las conclusiones obtenidas con el desarrollo de este proyecto. Además se muestran varias recomendaciones respecto a este trabajo que permitirán profundizar el estudio que se ha realizado y una posible extensión del proyecto para abarcar otros métodos.

6.1. Conclusiones

Se estudiaron las herramientas computacionales de programación orientada a objetos en Visual Basic necesarias para implementar la interfaz entre los modelos de simulación en ARENA y el programa de optimización basado en la Metodología de Superficies de Respuesta, permitiendo alcanzar el objetivo planteado que era el de crear un software de optimización para modelos de simulación en ARENA.

Se implementó la Metodología de Superficies de Respuesta a la interfaz construida, y se demostraron las capacidades de la herramienta al implementar un método heurístico de optimización a un sistema complejo de tipo caja negra (en este caso representado por un modelo de simulación).

En este proyecto se propuso la optimización de modelos de simulación usando la Metodología de Superficies de Respuesta. Usualmente esta metodología es aplicada de forma no automatizada, y mucho del trabajo hecho en MSR es improvisado en partes separadas. Este proyecto es el primer intento de definir un algoritmo de MSR de forma clara, detallada y consistente.

Además de los conocimientos adquiridos al estudiar la metodología en uso, el desarrollo de esta herramienta de optimización aportó grandes descubrimientos de varias características del software de simulación ARENA, al momento de estudiar, analizar y utilizar la librería dinámica (DLL) que posee ARENA para su interacción con el software de programación Visual Basic.

Finalmente, se puede decir que la simulación es una de las herramientas de análisis más poderosas que disponen quienes se dedican al diseño y a la operación de procesos o sistemas complejos, entonces el agregar un método de optimización a ésta se logra una herramienta más eficaz capaz de dar resultados de simulación óptimos que ayuden en la toma de decisiones.

6.2. Recomendaciones

Basándonos en el análisis de los resultados obtenidos en este trabajo se pueden hacer las siguientes sugerencias:

El método que se aplicó para la creación de la interfaz es por primera vez adaptado para su uso en ARENA y como tal es susceptible a mejoras. Por lo tanto se recomienda investigar más a fondo las propiedades internas del software ARENA.

En este trabajo el ajuste de superficies de respuesta se hizo con modelos de primer orden, esto tiene dos consecuencias importantes, en primer lugar, el método es muy susceptible de “quedarse” en óptimos locales, en segundo lugar, la búsqueda del óptimo en las cercanías del mismo exige que se den pasos de longitud muy pequeña, lo cual implica un aumento del

número de réplicas de simulaciones necesario. Por esta razón se recomienda que en futuros trabajos se estudie la implementación de modelos de órdenes superiores. Es de esperar que esto logre reducir el costo computacional al disminuir el número de réplicas de simulación necesarias y adicionalmente se logren “mejores óptimos”.

De acuerdo a los resultados obtenidos, se recomienda el uso de la Metodología de Superficies de Respuesta para obtener buenas soluciones de optimización para modelos de simulación.

www.bdigital.ula.ve

Bibliografía

- [1] <http://msdn.microsoft.com/vbasic/using/>, 2004.
- [2] <http://msdn.microsoft.com/visualc/using/>, 2004.
- [3] Andradóttir, Sigrún. A review of simulation optimization techniques. *Winter Simulation Conference*, pages 151–158, 1998.
- [4] Azadivar, Farhad. Simulation optimization methodologies. *Winter Simulation Conference*, pages 93–100, 1999.
- [5] Bäck, Thomas. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, and Genetic Algorithms*. Oxford University Press, New York, 1996.
- [6] Carson, Yolanda y Anu, Maria. Simulation optimization: Methods and applications. *Winter Simulation Conference*, pages 118–126, 1997.
- [7] Fu, Michael C. *Optimization via simulation: a review*. *Annals of Operations Research*. 1994.
- [8] Glover, Fred. Tabu search - part i. *ORSA Journal Computing*, 1, 1989.
- [9] Glover, Fred y Laguna, Manuel. *Tabu Search*. Kluwer Academic Publishers, New York, 1997.
- [10] Kleijnen, Jack P.C. *Experimental design for sensitivity analysis, optimization, and validation of simulation models*. John Wiley and Sons, New York, 1998.
- [11] LANNER GROUP, INC., Houston. *WITNESS Version 9 User Manual*, 1998.
- [12] Law, Averill y Kelton, David. *Simulation Modeling and Analysis*. McGraw Hill, United State of America, 1982.
- [13] Marcano O., Freddy. Desarrollo de una herramienta de optimización basada en redes neuronales para modelos de simulación. Trabajo de grado, Departamento de Investigación de Operaciones, Universidad de Los Andes, Mérida, Enero 2005.

- [14] Medina P., Sebastián. Nuevos enfoques en la optimización de *Caja Negra*. Aplicaciones en optimización mediante simulación en ARENA. Trabajo de ascenso a la categoría de agregado, Departamento de Investigación de Operaciones, Universidad de Los Andes, Mérida. Por publicar.
- [15] Myers, Raymond H. y Montgomery, Douglas C. *Response Surface Methodology*. Wiley Series in Probability and Statistics. Jhon Wiley and Sons, Inc., 605 3ra Av., New York, NY, 1995.
- [16] Pflug, Georg Ch. *Optimization of stochastic models: The interface between simulation and optimization*. Kluwer Academic Publishers, Boston, 1996.
- [17] ProModel Corporation, Utah. *SimRunner2 User's Guide*, 1998.
- [18] Rhone, Eddie y Prasetio, Yanto. Tabu search. Proyecto de curso vía web, Industrial Engineering, University of Washington, 2002.
- [19] Ross, Sheldon M. *Simulación*. Prentice Hall, México, 1999.
- [20] Taha, Hamdy A. *Investigación de Operaciones, una introducción*. Prentice Hall, México, 1998.