
INSTITUTO TECNOLÓGICO DE CELAYA

Departamento de Ingeniería Química

*“Análisis de Modelos de Programación Estocástica Mixta-
Entera Lineal de Dos Etapas con Recurso”*

POR

Sergio Frausto Hernández

TESIS PRESENTADA AL DEPARTAMENTO DE INGENIERÍA QUÍMICA
COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE

DOCTOR EN CIENCIAS

Celaya, Guanajuato

Agosto, 2006

Resumen

Se desarrolló un algoritmo de solución para resolver problemas de programación estocástica mixta-entera lineal con recurso. Los modelos considerados son enteros en la primera etapa de un programa estocástico de dos etapas. El algoritmo de solución que se propone implementa diferentes técnicas de muestreo, un procedimiento que identifica una secuencia de soluciones incumbentes, y un criterio de paro estadístico. Para mostrar los alcances del algoritmo se introducen dos aplicaciones en la ingeniería de procesos; éstas son: *i)* optimización de cadenas de suministro y *ii)* seguridad en redes municipales de agua. En el problema de la cadena de suministro se consideró que las disponibilidades de la materia prima y las demandas del producto son inciertas, mientras que en el problema de seguridad en redes municipales de agua, la densidad de población y la probabilidad del riesgo de ataque involucraron las incertidumbres. En ambas áreas de investigación los parámetros inciertos fueron representados por funciones de distribución de probabilidad continua. Los resultados de estos programas estocásticos son la conformación, diseño y planeación de una cadena de suministro en términos de sus capacidades de producción y expansión, mientras que en los problemas de seguridad de redes municipales de agua se obtiene una configuración de sensores óptima que minimiza la cantidad de población afectada por ataque de un contaminante. Los ejemplos difieren entre sí por el número de variables de proceso y el número de parámetros inciertos, mostrando de esta manera que el algoritmo propuesto es aplicable a problemas de gran escala. Finalmente, algunos resultados obtenidos por el algoritmo propuesto son comparados con los obtenidos por otros métodos de solución estocástica reflejando que el algoritmo es estable y robusto.

Dirigida por:

Ph. D. Vicente Rico Ramírez

Dedicatorias

A mis padres

Martina y Teodoro

A mis hermanos

Carlos, Sara, Miguel, Teodoro, Javier, Raúl, Ma. de Jesús, Jaime y Mario

(cada uno con sus respectivas familias)

A mi novia

Rosy

A mis amigos

Pedro, Felipe, Abundio, Arnulfo, Morquecho, Chava, Lupe, César, Fello, Betillo, Lobito,

Oscar, Cisneros, Silicio, Quezada, Pérez, José, Liliana, Alex, Víc, Xóchitl, Rosa, el Don,

Lupita, Nelly, Claudia, Gabriel, Ulises, Florianne, Don Armando, Chema, Adrián,

Enedino, Carlos

Agradecimientos

A Dios

Por los dones y talentos otorgados, entre ellos la vida

A CONACYT

Por la Beca Nacional y la Beca Mixta otorgadas para los estudios de maestría y doctorado

A mi Asesor

Ph. D. Vicente Rico Ramírez

Por todo el apoyo y confianza para la realización del presente trabajo

A mis sinodales

Ph. D. Arturo Jiménez Gutiérrez, Ph. D. Ramiro Rico Martínez y

D. C. Salvador Hernández Castro

Por sus comentarios, disposición y tiempo para la revisión del presente trabajo

Al Chemical Engineering Department de la Carnegie Mellon University, en especial al

Ph. D. Ignacio E. Grossmann

Por hacer de la estancia de investigación una experiencia de vida

Al Instituto Tecnológico de Celaya

Por que lo que soy, en gran parte se lo debo a el

Contenido

RESUMEN	ii
LISTA DE FIGURAS	ix
LISTA DE TABLAS	xi
CAPÍTULO 1: INTRODUCCIÓN	1
1.1 Programación lineal e incertidumbres	2
1.2 Objetivo	4
1.3 Organización de la tesis	5
CAPÍTULO 2: PROGRAMACIÓN ESTOCÁSTICA DE DOS ETAPAS CON RECURSO FIJO	7
2.1 Introducción	8
2.2 Programas estocásticos de dos etapas con recurso	9
2.2.1 Teorema de la dualidad	10
2.3. Procedimiento general de solución de problemas estocásticos	11
2.3.1 Especificación de las incertidumbres usando distribuciones de probabilidad	12
2.3.2 Técnicas de muestreo en modelación estocástica	13
2.3.3 Propagación de los efectos inciertos a través del modelo y análisis de resultados	14

2.4 Solución aproximada de los métodos estocásticos	16
2.4.1 Límite inferior de Jensen	16
2.5 Diferencias entre los métodos L-shaped y descomposición estocástica	20
CAPÍTULO 3: DESCOMPOSICIÓN ESTOCÁSTICA	22
3.1 Introducción	23
3.2. Método de descomposición estocástica	24
3.2.1 Aproximación a la función de recurso $Q(x,u)$	25
3.2.2 Cortes de optimalidad	27
3.3 Un algoritmo de descomposición estocástica básico	29
3.4 Estabilización del algoritmo de descomposición estocástica	30
3.4.1 Soluciones incumbentes	31
3.4.2 Algoritmo SD con soluciones incumbentes	35
3.4.3 Aplicación ilustrativa	38
3.4.4 Análisis del error	43
3.5. Implementación computacional	46
 CAPÍTULO 4: CADENAS DE SUMINISTRO: UN PROBLEMA DE OPTIMIZACIÓN CON INCERTIDUMBRES EN LAS DISPONIBILIDADES DE LA MATERIA PRIMA Y EN LAS DEMANDAS DEL MERCADO	 49
4.1 Introducción	50
4.2 Cadenas de suministro	51
4.3 Modelo de planeación para la cadena de suministro estratégica	53

4.3.1	<i>Planteamiento del problema</i>	55
4.3.2	<i>Formulación SMILPwR</i>	56
4.4	Ejemplos y resultados	61
4.4.1	<i>Ejemplo 1</i>	61
4.4.2	<i>Ejemplo 2</i>	65
4.5	Métodos discretos vs. continuos	70
CAPÍTULO 5:	REDES DE AGUA MUNICIPALES: UN PROBLEMA ESTOCÁSTICO DE DOS ETAPAS PARA LA COLOCACIÓN DE SENSORES BAJO INCERTIDUMBRE	73
5.1	Introducción	74
5.2	Redes municipales de agua: planteamiento del problema	76
5.2.1	<i>Formulación como un problema estocástico entero de dos etapas con recurso fijo</i>	79
5.3	Modelos estocásticos de dos etapas y equivalentes deterministas	82
5.4	Ejemplos y resultados	84
5.4.1	<i>Ejemplo 1</i>	85
5.4.2	<i>Ejemplo 2</i>	90
5.4.3	<i>Análisis de resultados</i>	93
CAPÍTULO 6:	CONCLUSIONES Y TRABAJO FUTURO	98
6.1	Conclusiones	99
6.2	Trabajo futuro	100
BIBLIOGRAFÍA	102

APÉNDICE	107
<i>Apéndice A</i>	107
<i>Apéndice B</i>	119
<i>Apéndice C</i>	122
<i>Apéndice D</i>	125

Lista de Figuras

Figura 2.1	Etapas de decisión	9
Figura 2.2	Ejemplos de funciones de distribución de probabilidad para modelación estocástica	13
Figura 2.3	Dos funciones de límite inferior posibles	17
Figura 2.4	El límite inferior de Jensen en un problema de minimización	19
Figura 3.1	Desarrollo de la optimización exacta en el algoritmo SD	26
Figura 3.2	Primera representación para encontrar si la solución incumbente pudiese ser cambiada	34
Figura 3.3	Segunda representación para encontrar si la solución incumbente pudiese ser cambiada	35
Figura 3.4	Soluciones para la variable x en el algoritmo SD básico e incumbente	42
Figura 3.5	Comportamiento asintótico de la función objetivo	42
Figura 3.6	Resultados obtenidos para la Aplicación Ilustrativa con criterio de paro y para diferentes técnicas de muestreo	45
Figura 3.7	Diagrama de flujo del algoritmo SD incumbente	47
Figura 3.8	Implementación computacional del algoritmo SD incumbente	48
Figura 4.1	Ejemplo de una cadena de suministro	52
Figura 4.2	Red de producción	55
Figura 4.3	Estructura para el Ejemplo 1	62
Figura 4.4	Convergencia del Ejemplo 1 para diferentes técnicas de muestreo	64
Figura 4.5	Red para el Ejemplo 2	66

Figura 4.6	Convergencia del Ejemplo 2 para diferentes técnicas de muestreo	68
Figura 5.1	Ejemplo de una red de agua municipal	77
Figura 5.2	Modelo de la propagación de un contaminante inyectado. a) Cuando la propagación ocurre de un nodo k a un nodo j. b) Cuando la propagación no ocurre de un nodo k a un nodo j	79
Figura 5.3	Red municipal de agua para el Ejemplo 1	86
Figura 5.4	Patrón de la dirección del flujo para el periodo I del Ejemplo 1	88
Figura 5.5	Resultados de la convergencia del Ejemplo 1	89
Figura 5.6	Configuración óptima del Ejemplo 1 cuando la zona industrial se considera en riesgo y el costo del sensor es bajo	90
Figura 5.7	Patrón de la dirección del flujo para el periodo I del Ejemplo 2	92
Figura 5.8	Resultados de la convergencia del Ejemplo 2	92
Figura 5.9	Configuración óptima para el Ejemplo 2 cuando la zona residencial se considera en riesgo y el costo del sensor es bajo	94
Figura B.1	Patrón de la dirección del flujo para el periodo II del Ejemplo 1	119
Figura B.2	Patrón de la dirección del flujo para el periodo III del Ejemplo 1	120
Figura B.3	Patrón de la dirección del flujo para el periodo IV del Ejemplo 1	121
Figura C.1	Patrón de la dirección del flujo para el periodo II del Ejemplo 2	122
Figura C.2	Patrón de la dirección del flujo para el periodo III del Ejemplo 2	123
Figura C.3	Patrón de la dirección del flujo para el periodo IV del Ejemplo 2	124

Lista de Tablas

Tabla 3.1	Costos de generación (g_{ij}) y otros datos de la Aplicación Ilustrativa	41
Tabla 4.1	Disponibilidad y demanda para el Ejemplo 1	62
Tabla 4.2	Datos de costos para el Ejemplo 1	63
Tabla 4.3	Datos para los coeficientes de reacción y capacidades existentes del Ejemplo 1	63
Tabla 4.4	Datos para la inversión, para las capacidades de expansión y para el VNP del Ejemplo 1	63
Tabla 4.5	Resultados del Ejemplo 1	65
Tabla 4.6	Disponibilidad y demanda para el Ejemplo 2	66
Tabla 4.7	Datos de costos para el Ejemplo 2	66
Tabla 4.8	Datos para los coeficientes de reacción y capacidades existentes del Ejemplo 2	67
Tabla 4.9	Datos para la inversión, para las capacidades de expansión, y para el VNP del Ejemplo 2	67
Tabla 4.10	Resultados del Ejemplo 2	69
Tabla 4.11	Comportamiento del VNP cuando se incrementa el número de escenarios	70
Tabla 4.12	Comportamiento del VNP cuando se disminuye el error del límite.....	70
Tabla 4.13	Características y esfuerzo computacional del Ejemplo 1	71
Tabla 4.14	Características y esfuerzo computacional del Ejemplo 2	71
Tabla 5.1	Distribución de nodos para el Ejemplo 1	86
Tabla 5.2	Probabilidad de ataque (%) usada para cada zona y escenario	89

Tabla 5.3	Resultados del Ejemplo 1 cuando la zona industrial se considera en riesgo	89
Tabla 5.4	Resultados del Ejemplo 1 cuando la zona residencial se considera en riesgo	89
Tabla 5.5	Distribución de nodos para el Ejemplo 2	91
Tabla 5.6	Resultados del Ejemplo 2 cuando la zona industrial se considera en riesgo	93
Tabla 5.7	Resultados del Ejemplo 2 cuando la zona residencial se considera en riesgo	93
Tabla 5.8	Ejemplo 1 con zona industrial en riesgo y densidad de población diferente en cada nodo	95
Tabla 5.9	Ejemplo 2 con zona residencial en riesgo y densidad de población diferente en cada nodo	95
Tabla 5.10	Características y esfuerzo computacional del Ejemplo 1	97
Tabla 5.11	Características y esfuerzo computacional del Ejemplo 2	97
Tabla D.1	Densidad de población para cada uno de los nodos del Ejemplo 1	125
Tabla D.2	Densidad de población para cada uno de los nodos del Ejemplo 2	126

Capítulo 1

Introducción

En este capítulo se presentan los antecedentes de los métodos de programación estocástica y se revisan algunos trabajos de esta área. Posteriormente, se plantean los objetivos de este trabajo y se describe la organización de la tesis.

1.1 Programación lineal e incertidumbres

Por varias décadas la Programación Lineal (LP) se ha establecido como una de las herramientas fundamentales para la planeación. Sus aplicaciones se han hecho rutinarias en varias disciplinas: ingeniería, administración, economía, ecología, etc. Se puede atribuir esta amplia extensión de su aceptación a: *i)* Un entendimiento del potencial y alcance de los problemas LP entre los practicantes, *ii)* buenos algoritmos, y *iii)* una amplia disponibilidad y fiabilidad de los programas de cómputo. Además, investigaciones sobre problemas especiales (por ejemplo, de asignación, de transporte, de redes, etc.) han hecho a la metodología LP indispensable para numerosas industrias, incluyendo las de distribución, energía, manufactura y telecomunicaciones, por nombrar algunas. A pesar de su éxito, se debe resaltar que los modelos LP tradicionales son modelos deterministas. Es decir, todos los coeficientes de la función objetivo y sus restricciones se consideran conocidos con precisión. La suposición de que todos los parámetros del modelo son conocidos con certeza limita la utilidad de la aproximación cuando se planifica bajo incertidumbre.

En el ámbito de la ingeniería de procesos, los modelos se consideran con frecuencia de una naturaleza determinista, pero existen situaciones en las que las incertidumbres no pueden despreciarse. Estas incertidumbres pueden deberse a variaciones de parámetros externos del proceso, como la calidad de las alimentaciones, la disponibilidad de la materia prima y las demandas de los productos, o a la incapacidad de establecer con exactitud los valores que pueden tomar los parámetros internos del proceso, tales como coeficientes de transferencia, constantes de reacción y algunas propiedades físicas. Esta clase de problemas que involucran incertidumbres en los datos o parámetros de entrada constituyen una rama de la optimización que es conocida como Programación Estocástica (PE) u Optimización Bajo Incertidumbres (OBI).

Una clase particularmente importante de modelos, que incorporan incertidumbres dentro de un modelo de optimización, son los programas estocásticos lineales de dos etapas con recurso. Para motivar esta clase de modelos, recordaremos que, en el análisis determinista, la planeación consiste de elegir niveles de actividad que satisfacen las restricciones originales mientras se maximiza el beneficio total (o minimiza el costo total). Note que toda la información necesaria para tomar decisiones se considera que está disponible al tiempo de la planeación. Bajo incertidumbre, no toda la información está disponible y parámetros tales como los recursos son a menudo modelados por variables aleatorias. En tales circunstancias, se aconseja planear solamente esas actividades que no pueden ser pospuestas hasta una fecha futura. Las actividades restantes podrían ser pospuestas hasta que una mejor información llegue a estar disponible. Debido a que la información se revela secuencialmente en el tiempo, las decisiones que se toman bajo incertidumbre naturalmente involucran un proceso multi-etapa. Los modelos más utilizados para planeación bajo incertidumbre se pueden acreditar a Dantzig (1955) y Beale (1955), y son a menudo referidos como programas estocásticos de dos etapas con recurso.

Recientemente, se han reportado algunos trabajos para resolver problemas de programación estocástica lineal (ver, por ejemplo, Birge y Louveaux, 1997; Haneveld y Vander Vlerk, 1999; Ponce 2003). Los resultados, a pesar del gran esfuerzo computacional, justifican la utilidad de estos métodos y reflejan una apreciable diferencia entre lo que se obtiene con respecto a un modelo determinista.

Por otro lado, los modelos de Programación Mixta Entera Lineal (MILP) son ampliamente usados para analizar una gran variedad de problemas de planeación de la producción, control de inventario, logística, y calendarización. A pesar de la dificultad

teórica de estos modelos, el desarrollo de nuevos algoritmos de solución y la creciente evolución de las computadoras ha permitido resolver problemas de tamaño real.

Para resolver los modelos MILP usualmente las incertidumbres en los datos o en los parámetros se desprecian con base en alguna sugerencia razonable o en la realización de alguna estimación estadística cuidadosa que, en combinación con un análisis de sensibilidad de los coeficientes inexactos, puede llegar a ser satisfactoria en muchos casos. Sin embargo, en algunos casos las soluciones óptimas pueden depender en gran medida de los valores de los datos inexactos, por lo que se requiere tomar en cuenta las incertidumbres de los coeficientes de una forma fundamental.

Al respecto, la teoría de la probabilidad proporciona técnicas para el análisis cuantitativo de las incertidumbres; por ejemplo se ha extendido el uso de técnicas de muestreo como las técnicas *Monte Carlo*, *Latin Hypercube*, *Latin Hypercube Media* y *Secuencia de Hammersley*. En estas técnicas de muestreo los valores de los datos o parámetros inciertos se interpretan como valores procedentes del muestreo de variables aleatorias a las cuales se les ha asignado una distribución de probabilidad.

Hasta la fecha, el desarrollo matemático para la programación estocástica entera es todavía insuficiente. Sólo se han desarrollado técnicas para resolver problemas específicos, o bien para usar una propiedad particular que considera formulaciones de programas equivalentes deterministas para los programas estocásticos enteros (Birge y Louveaux, 1997).

1.2 Objetivo

El objetivo de este trabajo consiste en desarrollar una estrategia de solución para problemas estocásticos mixto-entero lineales de dos etapas con recurso. Las propiedades

matemáticas de este tipo de problemas se abordarán con restricciones enteras en la primera etapa y con incertidumbres representadas por funciones de distribución de probabilidad continua.

Para resolver estos modelos se extenderá al caso mixto-entero el esquema de solución propuesto por Ponce y col. (2004). En tal esquema, además de la técnica de muestreo implementada dentro del algoritmo de Descomposición Estocástica (SD) básico (Higle y Sen, 1991; Higle y Sen, 1996), se incorporará un procedimiento para identificar una secuencia de soluciones incumbentes que proporcionen un criterio de paro adicional para el algoritmo.

Por otra parte, aplicaciones interesantes de problemas MILP deterministas serán reformulados como problemas estocásticos y se utilizarían como casos de estudio para validar el algoritmo propuesto.

1.3 Organización de la tesis

A continuación se presenta una descripción de cómo se ha distribuido este trabajo para lograr los objetivos propuestos.

En el Capítulo 2 se revisa la programación estocástica de dos etapas con recurso fijo, el tipo de decisiones que se ven involucradas en cada una de las etapas y la representación matemática de éstas. Enseguida se presenta el principio fundamental que da lugar a los diferentes métodos de solución estocástica, el límite inferior de Jensen. Por último, se discuten las diferencias, ventajas y desventajas de las técnicas de solución basadas en dicho límite.

En el Capítulo 3 se presentan los fundamentos teóricos del método de descomposición estocástica (SD) básico y el de soluciones incumbentes. Se describen

además los algoritmos de solución (básico y de soluciones incumbentes) que se proponen para resolver programas estocásticos mixto-enteros en la primera etapa. Se diseña además un criterio de paro que se basa en el comportamiento asintótico del promedio de los valores de la función objetivo. Finalmente, se resuelve un ejemplo ilustrativo para mostrar el desarrollo de los algoritmos que se han propuesto.

En el Capítulo 4 se trata una aplicación de interés actual en la ingeniería de procesos: la optimización de cadenas de suministro. Se plantea el problema a resolver y su formulación matemática como un problema estocástico de dos etapas que involucra incertidumbres en la disponibilidad de la materia prima y en las demandas del mercado. Se resuelven dos ejemplos reportados en la literatura, y de sus resultados se realiza un análisis comparativo con los obtenidos por otras técnicas de solución estocástica.

En el Capítulo 5 se propone un nuevo modelo para representar la seguridad en redes municipales de agua. El problema se formula como un problema estocástico de dos etapas que involucra incertidumbres en la densidad de población y en las probabilidades del riesgo de ataque. Por la naturaleza del modelo, se presentan dos ejemplos reportados en la literatura que involucran un número grande de variables de proceso y de parámetros inciertos. Al final, se realiza un análisis de resultados que resalta las ventajas del método SD comparado respecto de otros métodos de solución estocástica.

Finalmente, en el Capítulo 6 se presentan las conclusiones de este trabajo y las recomendaciones para trabajo futuro.

Capítulo 2

Programación Estocástica de Dos Etapas con Recurso Fijo

En este capítulo se revisan los aspectos fundamentales acerca de la programación estocástica de dos etapas con recurso fijo, incluyendo el tipo de decisiones que se ven involucradas en cada una de las etapas y su representación matemática. Se presenta también el principio básico que da lugar a los diferentes métodos de solución estocástica, el límite inferior de Jensen. Por último, se discuten las diferencias, ventajas y desventajas de dichos métodos de solución.

2.1 Introducción

Los programas estocásticos lineales son programas lineales en los que algunos datos del problema se consideran inciertos. Los programas con recurso son aquellos en los cuales algunas decisiones o acciones se pueden tomar después de que la incertidumbre es revelada o conocida. Para ser más precisos, *datos inciertos* significa que algunos datos del problema se pueden representar por variables aleatorias. Para estas variables aleatorias se considera que una descripción probabilística exacta se encuentra disponible, ya sea bajo la forma de distribuciones de probabilidad, densidades o, más generalmente, de medidas probabilísticas. Como es usual, el valor particular que tomará la variable aleatoria se conoce después del experimento aleatorio.

El arreglo de decisiones es dividido en dos grupos (ver Figura 2.1):

- i) Las decisiones se toman antes de que ocurra el experimento (problemas “aquí y ahora”). Todas estas decisiones se llaman decisiones de primera etapa y al periodo cuando se toman estas decisiones se llama primera etapa.
- ii) Las decisiones se toman después de que ocurre el experimento (problemas “esperar y ver”). Estas se llaman decisiones de segunda etapa. El periodo correspondiente se llama segunda etapa.

Las decisiones de primera etapa son representadas por el vector x , mientras que las decisiones de segunda etapa son representadas por el vector y o $y(u)$ o igualmente $y(u,x)$ si se quiere enfatizar que las decisiones de la segunda etapa dependen como funciones de los resultados del experimento aleatorio y de las decisiones de la primera etapa.

Observe aquí que las definiciones de primera y segunda etapa se relacionan solamente antes y después del experimento aleatorio y pueden en realidad contener secuencias de decisiones y eventos, tal y como se muestra en la Figura 2.1.

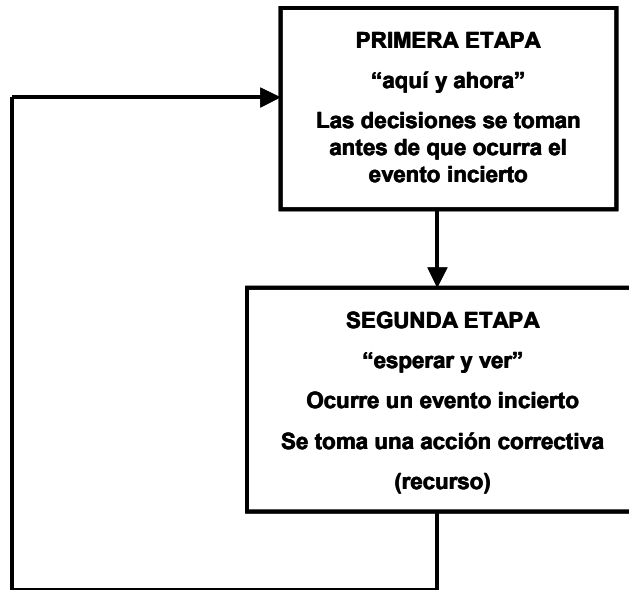


Figura 2.1. Etapas de decisión.

Enseguida se discutirá la representación matemática de esta secuencia lógica de decisiones.

2.2 Programas estocásticos de dos etapas con recurso

La formulación general de un programa estocástico lineal de dos etapas con recurso fue originada por Dantzig (1955) y Beale (1955). El modelo de la primera etapa se formula como:

$$\begin{aligned}
 \min_{x \in X} \quad & z = c^T x + Q(x) \\
 \text{s. a} \quad & Ax = b \\
 & x \geq 0
 \end{aligned} \tag{2.1}$$

donde A es una matriz de coeficientes y b y c son vectores de coeficientes, $Q(x)$ es la función de recurso definida por

$$Q(x) = E_u [Q(x, u)] \quad (2.2)$$

donde u representa al vector de variables estocásticas, E_u representa el operador esperanza matemática y $Q(x, u)$ se obtiene de la segunda etapa del problema:

$$\begin{aligned} Q(x, u) = \text{Min} \quad & q^T(u)y \\ \text{s. a} \quad & W(u)y = h(u) - T(u)x \\ & y \geq 0 \end{aligned} \quad (2.3)$$

Los vectores q y h así como las matrices W y T pueden depender de las variables u .

Por otra parte, la matriz W define el tipo de recurso: fijo o completo. En el caso de recurso fijo, W es independiente de u (es decir, es sólo una matriz de coeficientes). En el caso de recurso completo, cualquier valor de x que se seleccione en la primera etapa daría como resultado un problema factible en la segunda etapa.

La formulación general de un programa estocástico entero lineal de dos etapas con recurso es muy semejante al caso general de los programas estocásticos lineales, sólo que el primero requiere de la existencia de variables enteras o binarias en cualquiera de las etapas (Birge y Louveaux, 1997). De esta manera, para que las Ecuaciones 2.1 y 2.3 correspondan a un modelo estocástico entero lineal, algunas de las variables x y/o y deben ser variables binarias o enteras.

2.2.1 Teorema de la dualidad

Para cada problema de Programación Lineal (LP), también llamado como problema primo, existe otro problema denominado problema dual. Este problema es de gran utilidad cuando se realizan análisis de sensibilidad en la solución óptima del problema. Así, el problema definido en la segunda etapa no se resolverá directamente, sino que se resolverá

su problema dual para derivar restricciones lineales que serán utilizadas por las técnicas estocásticas que se discutirán en la Sección 2.3.3.

Dado el problema primo de la Ecuación 2.3, su dual asociado está dado por:

$$\begin{aligned} Q(x, u) = \text{Max} \quad & \pi^T [h(u) - T(u)x] \\ \text{s. a} \quad & \pi W(u) \leq q(u) \end{aligned} \quad (2.4)$$

donde π representan los multiplicadores de Lagrange de las restricciones del problema primo. El *Teorema de la Dualidad* expresa lo siguiente:

1. Si el problema primo tiene una solución óptima (y^*), entonces el problema dual tiene una solución óptima (π^*) que satisface:

$$q(u)(y^*)^T = (\pi^*)^T [h(u) - T(u)x] \quad (2.5)$$

2. Si uno de los problemas tiene un valor objetivo no acotado, el otro problema debe ser no factible.
3. Ambos problemas podrían ser no factibles.

En la literatura se puede encontrar una variedad de trabajos referentes a programación estocástica lineal, en donde se puede observar el éxito con que se han empleado sus algoritmos y esquemas de solución. En las secciones siguientes se revisará en una forma general el procedimiento de solución de los métodos estocásticos, así como la concepción de ellos en el límite de Jensen.

2.3. Procedimiento general de solución de problemas estocásticos

En general, los métodos de solución de problemas estocásticos consisten de un procedimiento iterativo que involucra los siguientes tres pasos:

1. Especificación de las incertidumbres en términos de distribuciones de probabilidad, las cuales reflejarán la cantidad de información que está disponible.
2. Muestreo del espacio de dominio de los parámetros inciertos.
3. Propagación de los efectos inciertos a través del modelo y aplicación de técnicas estadísticas para analizar los resultados.

A continuación se describen cada uno de los pasos por separado.

2.3.1 Especificación de las incertidumbres usando distribuciones de probabilidad

Para representar la diversa naturaleza de las incertidumbres se pueden utilizar diferentes distribuciones de probabilidad. Algunas distribuciones representativas se muestran en la Figura 2.2. El tipo de distribución elegida para una variable incierta refleja la cantidad de información que está disponible. Por ejemplo, las distribuciones uniforme y log-uniforme representan una probabilidad igual de encontrar un valor en cualquier parte dentro de un rango específico, sobre una escala lineal o logarítmica, respectivamente.

La distribución normal (Gausiana) refleja una desviación simétrica de la probabilidad del valor de un parámetro por encima o por debajo de su valor medio. En contraste, las distribuciones lognormal y algunas triangulares son esquematizadas de una forma tal que hay una probabilidad mas alta de los valores colocados sobre un lado de la media que del otro.

La distribución beta proporciona un amplio rango de formas y es una manera muy flexible de representar variabilidad sobre un rango fijo. Finalmente, en algunos casos especiales, las distribuciones definidas por el usuario se usan para representar alguna caracterización arbitraria de incertidumbres, incluyendo una distribución al azar (por ejemplo, probabilidades fijas de valores discretos).

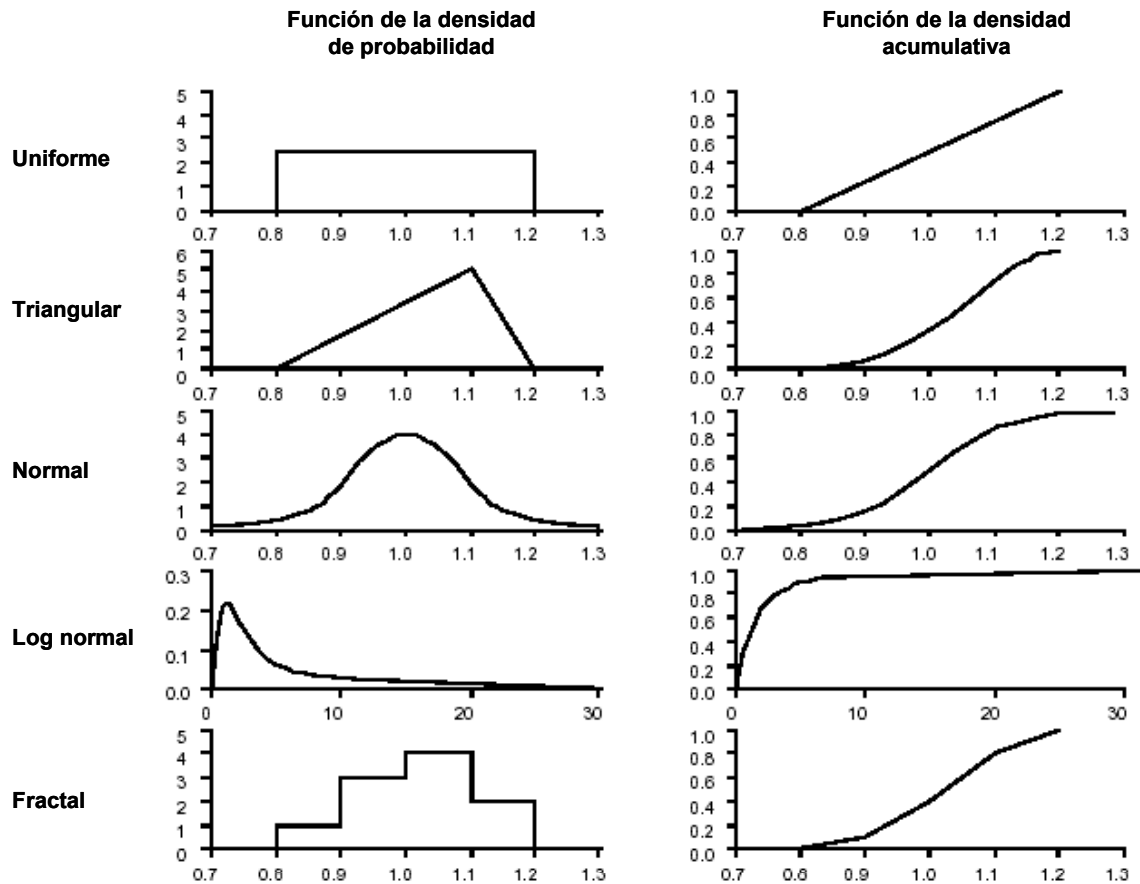


Figura 2.2. Ejemplos de funciones de distribución de probabilidad para modelación estocástica.

2.3.2 Técnicas de muestreo en modelación estocástica

Una vez que las distribuciones de probabilidad se asignan a los parámetros inciertos, el siguiente paso es el de desarrollar una operación de muestreo del dominio del parámetro de la variable incierta. Entre estas técnicas de muestreo se pueden mencionar a las técnicas *Monte Carlo*, *Latin Hypercube*, *Latin Hypercube Media* y *Secuencia de Hammersley*.

La técnica *Monte Carlo* (MC) es una de las más empleadas para muestrear las distribuciones de probabilidad. Esta técnica se basa en un generador pseudo-aleatorio para aproximar a una distribución uniforme (por ejemplo, con igual probabilidad en el rango de 0 a 1). Los valores específicos para cada variable de entrada se seleccionan por transformación inversa de la distribución acumulativa. Las técnicas Monte Carlo tienen la propiedad de que los puntos sucesivos en el muestreo son independientes.

Por otro lado, la técnica *Latin Hypercube* (LHS) es una forma de muestreo estratificado que puede producir estimados más precisos de la función de distribución de probabilidad. En esta técnica, el rango de cada parámetro incierto se subdivide en intervalos no traslapados de igual probabilidad. Mientras que en la técnica *Latin Hypercube Media* (MLHS), la subdivisión del intervalo se realiza con base a su punto medio. Sin embargo, el principal inconveniente de este esquema de estratificación es que sólo es uniforme en una dimensión y no proporciona propiedades de uniformidad en k-dimensiones.

Por último, la técnica de la *Secuencia de Hammersley* (HSS) usa un esquema de diseño óptimo para colocar n puntos sobre un hipercubo de k-dimensiones. Este esquema asegura que el muestreo es más representativo de la población y muestra grandes propiedades de uniformidad en k-dimensiones.

2.3.3 Propagación de los efectos inciertos a través del modelo y análisis de resultados

La propagación de los efectos inciertos en el modelo se puede realizar a través de los métodos analíticos, tales como, el método *L-Shaped* (Birge y Louveaux, 1997; Birge y Louveaux, 1998) y el método de *Descomposición Estocástica* (SD) (Higle y Sen, 1991; Higle y Sen, 1996). En estos métodos las aproximaciones por muestreo de los parámetros inciertos pueden presentar el inconveniente de que cuando se resuelve la n -th

aproximación, algunos cálculos podrían ser desechados por la optimización cuando la aproximación no es muy exacta (Birge y Louveaux, 1997). Para estructuras específicas donde se aplica el método *L-shaped*, dos aproximaciones evitan estos problemas por incrustar muestreo dentro de algún algoritmo sin optimización completa. Estas dos aproximaciones son el método de *Dantzig y Glynn* (1990), que utilizan un muestreo eficiente para reducir la varianza en cada corte basado en un número de muestras grande, y el método SD propuesto por Hagle y Sen (1991). Este último método utiliza una colección de muestras para derivar muchos cortes conforme se incrementa el número de iteraciones. Estos métodos requieren condiciones de convexidad, y estructuras de bloque-dual angular.

Painton y Diwekar (1995) establecieron que la exactitud en el cálculo del valor esperado de la función objetivo es dependiente del número de muestras. Es decir, para un número dado de muestras ($N_{muestras}$), el error en el cálculo del valor esperado decrece cuando $N_{muestras}$ incrementa. Este error puede ser calculado como:

$$\epsilon_{\mu} \propto (N_{muestras})^{-0.5}$$

Además, el número de muestras requerido para una exactitud dada en un problema de optimización estocástica también puede depender de otros factores, como lo son el tipo de incertidumbre y los valores de los puntos de las variables de decisión (Painton y Diwekar, 1995).

En la siguiente sección se discute cómo se conciben los métodos estocásticos para encontrar una solución aproximada desde el punto de vista del límite de Jensen.

2.4 Solución aproximada de los métodos estocásticos

En esta sección se discute una forma para limitar los problemas estocásticos debido a que, cuando se tiene un gran número de parámetros inciertos, no es posible calcular el valor esperado exacto de un programa lineal; sin embargo, es posible identificar un intervalo que contiene el valor esperado. En términos prácticos, se requiere encontrar el valor de la solución “esperar y ver” (ver Figura 2.1).

Los métodos para limitar son útiles si se desean usar las técnicas de descomposición determinista en problemas que contienen un gran número de variables aleatorias, tales como el método de descomposición *L-shaped*. Una alternativa más donde se aplican estos límites es en el desarrollo de aproximaciones que usan métodos estocásticos, entre los cuales se puede mencionar a los métodos SD.

Como se ha mencionado, los límites se pueden usar ya sea para aproximar el valor esperado de algunos programas lineales o para limitar el problema de la segunda etapa de un problema de dos etapas. El problema consiste en encontrar el valor esperado de un programa lineal, en términos de una función $\phi(u)$, la cual en el caso de dos etapas corresponden a la función de recurso $Q(x,u)$ para una x fija.

2.4.1 Límite inferior de Jensen

Se considera que el vector q de la función objetivo de la segunda etapa no depende de variables aleatorias, por lo que las incertidumbres sólo afectarán el lado derecho de la Ecuación 2.3. El propósito de esta sección es el de encontrar un límite inferior sobre la función $Q(x,u)$. Fijando el valor de x , $x = \bar{x}$, se usará la función $\phi(u)$ para representar a $Q(\bar{x},u)$; es decir, $\phi(u) \equiv Q(\bar{x},u)$.

Suponiendo que $\phi(u)$ es una función convexa, se puede obtener su límite inferior a través de una función lineal

$$L(u) = cu + b. \quad (2.6)$$

Como el objetivo siempre será el de encontrar un límite inferior que sea el más grande posible, se requerirá que este límite sea tangente a $\phi(u)$ en algún punto \tilde{u} . La Figura 2.3 muestra dos ejemplos de estas funciones de límite inferior.

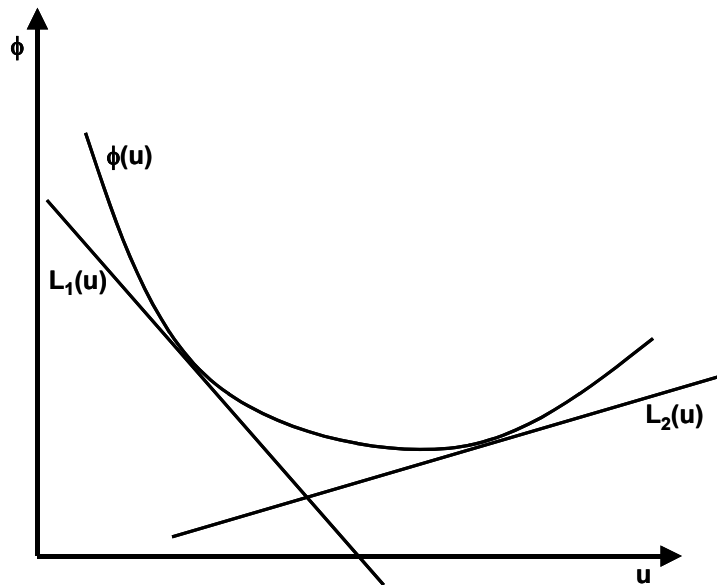


Figura 2.3. Dos funciones de límite inferior posibles.

Si se hace que la función límite inferior $L(u)$ sea tangente a $\phi(u)$ en \tilde{u} , la pendiente de dicha función es $\phi'(\tilde{u})$, y se tiene que

$$\phi(\tilde{u}) = \phi'(\tilde{u})\tilde{u} + b \quad (2.7)$$

Puesto que $\phi(\tilde{u}) = L(\tilde{u})$, de la Ecuación 2.7 se tiene que

$$c = \phi'(\tilde{u}) \quad (2.8)$$

y

$$b = \phi(u) - \phi'(\tilde{u})\tilde{u} . \quad (2.9)$$

Sustituyendo las Ecuaciones 2.8 y 2.9 en la Ecuación 2.6, la función límite inferior estará dada por

$$L(u) = \phi(\tilde{u}) + \phi'(\tilde{u})(u - \tilde{u}) . \quad (2.10)$$

Como la función límite inferior definida por la Ecuación 2.10 es una función lineal, fácilmente se puede calcular su valor esperado:

$$E[L(\tilde{u})] = \phi(\tilde{u}) + \phi'(\tilde{u})(E[\tilde{u}] - \tilde{u}) = L(E[\tilde{u}]) . \quad (2.11)$$

En otras palabras, se ha encontrado el límite inferior esperado a través de la evaluación de la función límite inferior en $E[\tilde{u}]$. De aquí se puede demostrar que se obtiene el mejor límite inferior (el más grande) al hacer $\tilde{u} = E[\tilde{u}]$. Esto no sólo se ve del hecho que la función no lineal que soporta $\phi(u)$ puede tener un valor más grande que $\phi(E[\tilde{u}])$ en $E[\tilde{u}]$, sino también de la siguiente diferenciación:

$$\frac{d}{d\tilde{u}} L(E[\tilde{u}]) = \phi'(\tilde{u}) - \phi'(\tilde{u}) + \phi''(\tilde{u})(E[\tilde{u}] - \tilde{u}) \quad (2.12)$$

Si se hace esta ecuación igual a cero se encuentra que $\tilde{u} = E[\tilde{u}]$. Con lo anterior se obtiene lo que se conoce como el límite inferior de Jensen o desigualdad de Jensen y que expresa lo siguiente:

Si $\phi(u)$ es convexa sobre el soporte de \tilde{u} entonces

$$E[\phi(\tilde{u})] \geq \phi(E[\tilde{u}]) \quad (2.13)$$

Este es el mejor límite inferior y se ilustra en la Figura 2.4. Se puede observar que el límite inferior de Jensen puede ser visto de dos maneras. Primero, se puede ver como un

límite donde una distribución se reemplaza por su media, y el problema en sí permanece sin cambiar. Esto es cuando se calcula $\phi(E[\tilde{u}])$. Segundo, se puede ver como un límite donde la distribución permanece sin cambiar y la función se reemplaza por una función lineal afín, como por ejemplo una línea recta. Esto es cuando se integra $L(u)$ sobre el soporte de \tilde{u} . Dependiendo de la situación dada, ambos puntos de vista pueden ser útiles.

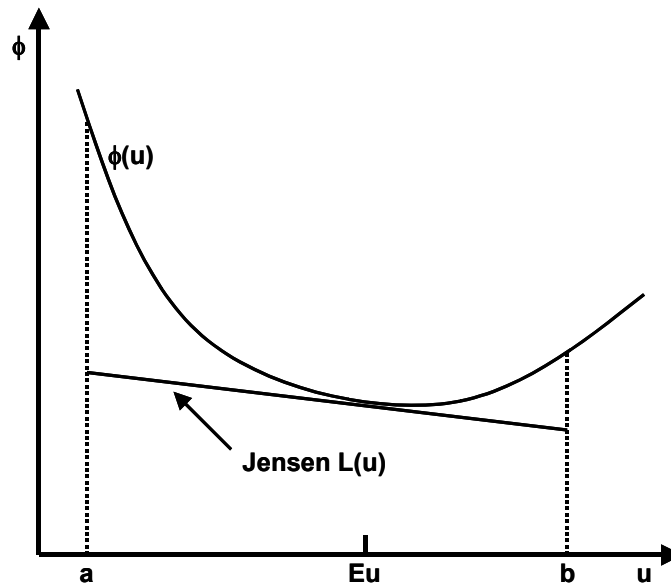


Figura 2.4. El límite inferior de Jensen en un problema de minimización.

Hay además una tercera interpretación, en donde se considera primero resolver el dual de $\phi(E[\tilde{u}])$ para obtener un conjunto V de bases óptimas π (ver Sección 2.2.1). En esta base, como u no entra en las restricciones del dual de ϕ , el dual es factible para todos los valores posibles de u . Se considera entonces que se resuelve la versión dual de $\phi(u)$ para toda u , pero con la restricción de que en la optimización sólo se usará la base dada V .

En una situación semejante, se podría afirmar que se usa la función correcta, la

distribución correcta, pero se optimiza solamente en una forma aproximada. El límite inferior de Jensen puede ser interpretado en este marco como una representación de una optimización aproximada usando el problema correcto y una distribución correcta, pero solamente una base dual factible (en SD no se usa una, sino un número finito de bases).

Desde esta concepción, a continuación se establecen algunas diferencias entre el algoritmo SD y el método L-shaped.

2.5 Diferencias entre los métodos L-shaped y descomposición estocástica

Existen dos grandes diferencias entre el método SD y el método *L-shaped* exacto. La primera, en el método SD se opera sobre un muestreo más que en todos los resultados, y segunda, lo que se calcula es un estimado del límite inferior sobre $Q(x_k)$ mas que $Q(x_k)$ mismo. Sin embargo, la razón por la que el límite inferior es algo diferente radica en que el método *L-shaped* está basado en esperanzas condicionales, mientras que el SD se basa en una optimización inexacta. Por otro lado, en la sección anterior se discutió que el límite inferior de Jensen tiene tres diferentes interpretaciones, una de las cuales permite usar esperanzas condicionales (como en el procedimiento de limitación del *L-shaped*) y otra corresponde a la optimización inexacta (como en SD).

Para que las tres interpretaciones del límite de Jensen sean equivalentes, el arreglo limitado de bases tiene que obtenerse a partir de la solución del problema de recurso en los puntos de las esperanzas condicionales. Eso no es el caso en SD. Ahí los puntos son aleatorios (acordes con el muestreo u_j). Aún así, usando un número limitado de bases se produciría un límite inferior, pero no el límite inferior de Jensen.

Por lo tanto SD y la versión límite del *L-shaped* son en realidad muy diferentes. La razón es porque la concepción del límite inferior es diferente, y el valor objetivo en SD es

solamente un límite inferior en términos de esperanzas (producto del muestreo). Uno de los métodos mantiene el número limitado de puntos en una forma muy cuidadosa, y el otro aleatoriamente. Un método tiene un criterio de paro exacto (error de límite), el otro se basa en un análisis estadístico. Así, más que cualquier cosa, ambos son alternativas de aproximación. Si uno no puede resolver el problema exacto, el otro recurre a un método que se basa en un límite o en el muestreo.

Capítulo 3

Descomposición Estocástica

En este capítulo se presentan la teoría del método SD básico y se presenta su algoritmo de solución. Posteriormente se discuten algunas extensiones a dicho algoritmo para lograr una estabilización y convergencia a través de la identificación de una secuencia de soluciones incumbentes. Enseguida, se propone un algoritmo SD incumbente para resolver programas estocásticos mixto-enteros en la primera etapa. Finalmente, se resuelve un ejemplo ilustrativo para mostrar el desarrollo de los algoritmos que se han propuesto.

3.1 Introducción

El método de descomposición *L-shaped* es un método determinista. Ello porque se entiende que si el algoritmo es repetido con los mismos datos de entrada, proporcionará los mismos resultados cada vez. En contraste con esto, se tienen los llamados métodos estocásticos. Estos son métodos que idealmente no darán los mismos resultados en dos corridas, aún con los mismos datos de entrada. Se dice *idealmente* porque es imposible en el mundo real crear verdaderamente números aleatorios, y de ahí que, en la práctica, sea posible repetir una corrida. Por otro lado, estos métodos tienen criterios de paro que son de una naturaleza estadística o estocástica, y donde normalmente convergen con probabilidad de uno.

La razón para llamar a estas técnicas de solución como métodos estocásticos es porque están guiados por un efecto aleatorio, como puede ser un muestreo. En este trabajo, se ha centrado la atención en el método SD, y aquí mismo se describirán los conceptos principales sobre los que se basa la teoría del algoritmo SD para los programas mixto-entero lineales en la primera y/o segunda etapa.

Para exponer la teoría del algoritmo SD, el presente capítulo se ha organizado de la siguiente manera: en la Sección 3.2 se discuten los conceptos fundamentales de la teoría SD, posteriormente ésta teoría se resume paso por paso en la Sección 3.3, donde se esquematiza el algoritmo de solución SD básico; en la Sección 3.4 se describen los conceptos de estabilización del algoritmo SD básico, entre los que se incluyen la identificación de una secuencia de soluciones incumbentes y un criterio de paro; por último, en la Sección 3.5 se presenta la implementación computacional para generar el algoritmo SD que considera estos conceptos de estabilización

3.2. Método de descomposición estocástica

Considérese el problema de la Ecuación 3.1:

$$\begin{aligned} \min_{x \in X} \quad & z = c^T x + Q(x) \\ \text{s. a} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{3.1}$$

donde $Q(x)$ se obtiene del primer momento de inercia

$$Q(x) = \int Q(x, u) f(u) du \tag{3.2}$$

con f como una función de densidad para \tilde{u} y

$$Q(x, u) = \min \{ q(u)^T y / Wy = h(u) - T(u)x, y \geq 0 \}. \tag{3.3}$$

Usando teoría de la dualidad (ver Sección 2.2.1), se tiene la siguiente formulación alternativa de $Q(x, u)$:

$$Q(x, u) = \max \{ \pi^T [h(u) - T(u)x] \mid \pi^T W \leq q(u)^T \}. \tag{3.4}$$

En la Ecuación 3.4, se observa que u y x no están presentes en las restricciones de la formulación dual, por lo que, si para una u y x dada se obtiene una solución, el problema es dual factible para toda u y x . De esta manera, si π^0 es una solución dual factible,

$$Q(x, u) \geq (\pi^0)^T [h(u) - (u)x] \tag{3.5}$$

para toda u y x , debido a que π^0 es factible pero no necesariamente óptimo en un problema de maximización. Esta observación es una interpretación del límite inferior de Jensen y una parte central del algoritmo SD, véase la Sección 2.4.1. En esa discusión se han dado tres diferentes interpretaciones, una de las cuales fue optimización aproximada usando un número finito de bases duales factibles, más que todas las bases posibles. En cada una de las optimizaciones del algoritmo SD se utiliza este subarreglo de bases duales factibles y por tal razón, el algoritmo sólo producirá un límite inferior a la solución.

3.2.1 Aproximación a la función de recurso $Q(x,u)$

El algoritmo SD es una técnica que se basa en el muestreo, por lo que u^v y x^v se definen como la muestra más reciente y el valor encontrado en la solución óptima de la iteración v , respectivamente. Para obtener una aproximación de la función de recurso, en la iteración actual se evalúa $Q(x^v, u^k)$ para todas las muestras encontradas hasta el momento. Para la muestra más reciente u^v , primero se resuelve

$$Q(x^v, u^v) = \max \{ \pi^T [h(u^v) - T(u^v)x^v] \mid \pi^T \leq q(u)^T \}, \quad (3.6)$$

para obtener una solución dual óptima π_v^v . Note que esta optimización es exacta por ser la primera en involucrar la muestra u^v . Después, se define un vector V como el conjunto de soluciones duales factibles que se han obtenido hasta la iteración v , por lo que se deberá de incorporar a éste vector la solución dual óptima π^v .

Después, en lugar de evaluar $Q(x^v, u^k)$ para cada una de las muestras anteriores, $k = 1, \dots, v-1$, simplemente se resuelve

$$\max_{\pi} \{ \pi^T (h(u^k) - x^v) \mid \pi \in V \} \quad (3.7)$$

para obtener π_k^v . Nótese que para todas las muestras, con excepción de la más reciente u^v , se ha desarrollado una optimización aproximada que toma en cuenta sólo el número finito de soluciones duales factibles contenidas en el vector V . Esta situación se ilustra en la Figura 3.1. En el esquema se observa una situación para el tercer punto de muestreo. Primero, se optimiza exactamente para el nuevo punto de muestreo, u^3 , con una solución dual óptima verdadera π^3 (punto G). Esta solución se representa en la Figura 3.1 por el hiperplano que está soportado por u^3 , $Q(x^3, u^3)$ (línea \overline{AB}). Después, se optimiza inexactamente para los dos puntos de muestreo anteriores. Hay tres bases disponibles para

esta optimización inexacta. Estas bases son representadas por las tres líneas delgadas (\overline{AB} , \overline{CD} y \overline{EF}). Como se puede ver, ninguno de los dos puntos de muestreo anteriores encuentran su base óptima verdadera, pues se obtendrían como óptimos los puntos H e I que no yacen sobre la curva $Q(x^3, u)$.

Por ejemplo, si se define el conjunto $\Xi(\bar{u}) = \{u_1, u_2, u_3\}$, con una probabilidad de $1/3$ cada una, se puede calcular un límite inferior sobre $Q(x^3)$ mediante

$$L(x^3) = \frac{1}{3} \sum_{k=1}^3 (\pi_k^3)^T [h(u^k) - T(u^k)x^3]. \tag{3.8}$$

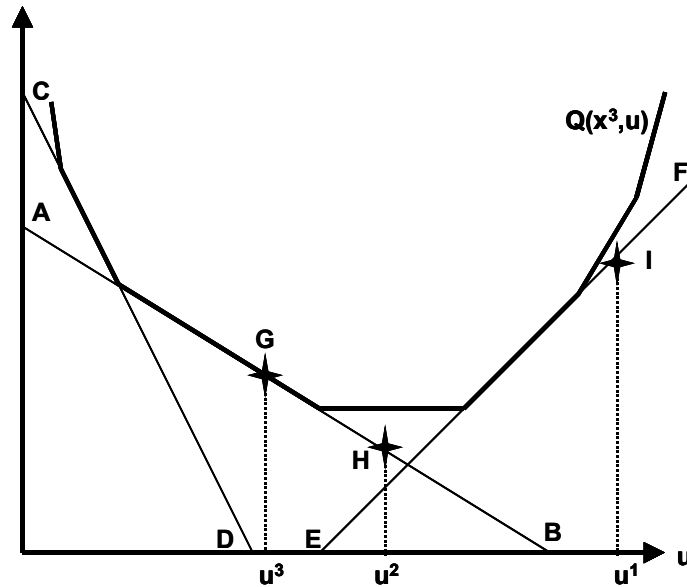


Figura 3.1. Desarrollo de la optimización exacta en el algoritmo SD.

Este es un límite inferior que ha sido obtenido a partir de la optimización inexacta de los puntos de muestreo anteriores. Es decir, los tres puntos de muestreo pudieran no representar correctamente la distribución de probabilidad, por lo que en términos de

esperanza matemática lo que se obtendría sería sólo un límite inferior. No obstante, como eventualmente este término convergerá a $Q(x)$, en general se puede escribir que

$$Q(x^v) = \frac{1}{v} \sum_{k=1}^v (\pi_k^v)^T [h(u_k) - T(u_k)x^v]. \quad (3.9)$$

Recuerde, sin embargo, que este no es el valor verdadero de $Q(x^v)$, sino justamente un estimado. De esta manera, en la siguiente sección se muestra como se interpretan estos límites inferiores como estimados de los cortes de optimalidad dentro del método SD.

3.2.2 Cortes de optimalidad

El algoritmo SD se basa entonces en la adición de restricciones lineales (conocidas como cortes) a la primera etapa del problema. Estos cortes se denominan cortes de optimalidad y son adicionados sucesivamente durante el procedimiento de solución. Un corte de optimalidad es una aproximación lineal de la función de recurso $Q(x)$ que se determina por la solución del problema dual de la segunda etapa.

A continuación se obtienen los cortes de optimalidad correspondientes a x^v (aunque estos cortes pudieran no estar vinculados entre sí e igualmente no representar un límite inferior, al menos son estimados de un límite inferior). El término $Q(x)$ de la función objetivo y de sus restricciones se reemplaza por θ . Así, el corte generado en la iteración v es dado por

$$\theta \geq \frac{1}{v} \sum_{k=1}^v (\pi_k^v)^T [h(u_k) - T(u_k)x] = \alpha_k^v + (\beta_k^v)^T x. \quad (3.10)$$

El doble arreglo de índices sobre α y β indican que el corte fue generado en la iteración k y que ha sido actualizado en la iteración v . Note entonces que para el corte generado en la Ecuación 3.10, k es igual a v .

En el método SD se toman en cuenta los cortes generados en las iteraciones anteriores. La razón es que, aunque se espera que estos cortes se pierdan (debido a que se ha utilizado una optimización inexacta), en realidad no están demasiado lejos de un ajuste (ya que para su obtención se han basado en un muestreo). El hecho de ser cortes pasados implica que están basados en un muestreo más pequeño que el presente, y por tanto con un error mayor por lo que se pretende irlos reemplazando, pero no desecharlos. Considérese entonces que existe un límite inferior L sobre $Q(x,u)$, tal que $Q(x,u) \geq L$ para toda x y u .

Los cortes anteriores

$$\theta \geq \alpha_k^{v-1} + (\beta_k^{v-1})^T x \quad \text{para } k = 1, \dots, v-1 \quad (3.11)$$

serán reemplazados por

$$\theta \geq \frac{v-1}{v} [\alpha_k^{v-1} + (\beta_k^{v-1})^T x] + \frac{1}{v} L = \alpha_k^v + (\beta_k^v)^T x \quad \text{para } k = 1, \dots, v-1 \quad (3.12)$$

Por razones de implementación se prefiere $L = 0$, pero en general este límite tendría que ser calculado. Esta desigualdad es más inexacta que la previa, debido a que $L \leq Q(x,u)$. El problema maestro ahora se describe como

$$\begin{aligned} \min \quad & c^T x + \theta \\ \text{s. a} \quad & Ax = b \\ & -(\beta_k^v)^T x + \theta \geq \alpha_k^v \quad \text{para } k = 1, \dots, v \\ & x \geq 0 \end{aligned} \quad (13)$$

y que produce el valor de la próxima iteración x^{v+1} . Higle y Sen (1991) suponen recurso completo (no se requieren cortes de factibilidad) y proponen una simplificación al método considerando sólo un conjunto restringido de posibles soluciones al problema dual de la segunda etapa, lo cual reduce significativamente el tiempo de cómputo requerido.

3.3 Un algoritmo de descomposición estocástica básico

Los pasos del algoritmo SD básico propuesto por Higle y Sen (1991), son los siguientes:

- Paso 0. Defina $\nu=0$, $V_0=\{\emptyset\}$, $\theta^\nu=-\infty$. x^I se supone conocido.
- Paso 1. Haga $\nu = \nu + I$ y tome una muestra de las variables inciertas, u^ν .
- Paso 2. Determine la aproximación lineal a la función de recurso $Q(x)$ (corte de optimalidad):

a) Resuelva el problema dual del problema de la segunda etapa:

$$\begin{aligned} \max \quad & \pi^T (h_\nu - T_\nu x^\nu) \\ \text{s. a} \quad & \pi^T W \leq q \end{aligned}$$

para encontrar los valores del vector de variables del problema dual π , π_ν^ν , y

haga $V_\nu = V_{\nu-1} \cup \pi_\nu^\nu$

b) Obtenga los coeficientes del corte de optimalidad:

$$\alpha_\nu^\nu = \frac{1}{\nu} \sum_{k=1}^{\nu} (\pi_k^\nu)^T h_k$$

$$\beta_\nu^\nu = \frac{1}{\nu} \sum_{k=1}^{\nu} (\pi_k^\nu)^T T_k$$

donde π_k^ν es la solución al problema (para toda $k|k \neq \nu$):

$$\begin{aligned} \max \quad & \pi^T (h_k - T_k x^\nu) \\ \text{s. a} \quad & \pi \in V_\nu \end{aligned}$$

Observe que el vector de solución a esta serie de problemas sólo puede ser alguno de los vectores que se incluyen ya en el conjunto de soluciones, V_ν .

c) Actualice los coeficientes de los cortes anteriores:

$$\alpha_k^v = \frac{v-1}{v} \alpha_k^{v-1}, \quad k = 1, \dots, v-1$$

$$\beta_k^v = \frac{v-1}{v} \beta_k^{v-1}, \quad k = 1, \dots, v-1$$

- Paso 3. Resuelva el problema de la primera etapa después de la adición del corte de optimalidad:

$$\begin{aligned} \min \quad & c^T x + \theta^v \\ \text{s. a} \quad & Ax = b \\ & \theta^v + \beta_k^v x \geq \alpha_k^v, \quad k = 1, \dots, v \end{aligned}$$

para obtener x^{v+1} . Vaya al paso 1.

- Paso 4. El algoritmo termina si:
 - a) Los cambios en la función objetivo son pequeños.
 - b) No se requiere la adición de nuevos vectores de solución al conjunto de soluciones, V .

3.4 Estabilización del algoritmo de descomposición estocástica

En esta sección se proporcionan algunas extensiones al método SD básico que ayudan por un lado a superar los problemas de implementación computacional, y por el otro, a estabilizar las aproximaciones e iteraciones generadas por el método. Las extensiones que aquí se discuten giran en torno a la identificación de una secuencia de soluciones incumbentes que se extraen de la secuencia de soluciones del programa principal (Higle y Sen, 1991). Los fundamentos teóricos de éste procedimiento se presentan a continuación.

3.4.1. Soluciones incumbentes

Una solución incumbente es un vector de variables de decisión x óptimo que garantiza el estimado más bajo de la función objetivo de todos los demás vectores de solución encontrados en el proceso del algoritmo SD básico.

La solución incumbente se usa dentro del algoritmo SD de una forma tal que se resuelven dos problemas, uno usando la solución incumbente, y otro resolviendo el paso 3 del algoritmo SD básico para encontrar un nuevo vector de soluciones x (solución candidata). Si el estimado del objetivo que se obtiene usando la solución candidata es menor que el que se obtiene usando la solución incumbente, ésta se toma como una nueva solución incumbente, si no, se sigue manteniendo la solución incumbente anterior hasta que haya una nueva solución candidata que satisfaga la condición.

Las literales que no tienen una barra encima corresponden a las variables del algoritmo SD básico, y las literales que tienen una barra encima son las variables asociadas con la solución incumbente. Por ejemplo, en la iteración ν^{th} , x^ν es la solución del programa maestro en la iteración $\nu - 1$, mientras que $\bar{x}^{\nu-1}$ es el valor actual de la variable x en la solución incumbente. En esta misma iteración, el siguiente valor de la solución incumbente será seleccionado entre los valores de la solución del programa maestro y del actual de la solución incumbente, es decir $\bar{x}^\nu \in \{\bar{x}^{\nu-1}, x^\nu\}$. Ahora bien, en la construcción de los planos cortantes, $\bar{\pi}_k^\nu$ es la solución dual que se obtiene cuando la operación se desarrolla por el uso de la solución incumbente de la primera etapa $\bar{x}^{\nu-1}$, más que por el uso de la solución del programa maestro x^ν . En resumen, el conjunto de soluciones $\{\bar{x}^\nu\}$ es una secuencia de soluciones incumbentes, mientras que el conjunto de soluciones $\{x^\nu\}$ es una secuencia de soluciones candidatas. Por último, para dar seguimiento a la solución incumbente se

introduce un nuevo índice i_ν que indicará la iteración en que fue encontrada la nueva solución incumbente.

El método se inicia con $\nu = 0$ y se hace que $u_0 = E[\tilde{u}]$. De esta forma se resuelve

$$\begin{aligned} \min \quad & c^T x + q^T y \\ \text{s. a} \quad & Ax = b \\ & Wy = u_0 - T(u_0)x \\ & x, y \geq 0 \end{aligned} \tag{3.14}$$

para obtener una x^l inicial. Se define la primer solución incumbente $\bar{x} = x^l$ y se denota que fue encontrada en la iteración 1 ($i_0 = 1$).

A continuación se presenta lo que se hace en una iteración general del algoritmo. Antes que nada se incrementa el contador de iteraciones, $\nu = \nu + 1$, y se toma una muestra u_k . Enseguida se determina un nuevo corte ν como se presentó en la Sección 3.2 y se actualiza el corte correspondiente al incumbente. Después se resuelven

$$\max \{ \pi^T [h(u_k) - T(u_k)x^\nu] \mid \pi^T W \leq q^T \} \tag{3.15}$$

y

$$\max \{ \pi^T [h(u_k) - T(u_k)\bar{x}^{\nu-1}] \mid \pi^T W \leq q^T \} \tag{3.16}$$

para obtener π^ν y $\bar{\pi}^\nu$, respectivamente. Posteriormente se actualiza el arreglo de bases dual factibles haciendo que $V := V \cup \{ \pi^\nu, \bar{\pi}^\nu \}$.

Ahora, se necesita calcular el nuevo corte y actualizar los cortes anteriores. Para la obtención de este nuevo corte se requieren los valores de π_k para $k = 1, \dots, \nu - 1$ y estos se obtienen de resolver

$$\max \{ \pi^T [h(u_k) - T(u_k)x^\nu] \mid \pi \in V \} \tag{3.17}$$

De esta manera, el corte se calcula de

$$\theta \geq \frac{1}{\nu} \sum_{k=1}^{\nu} (\pi_k)^T [h(u_k) - T(u_k)x] = \alpha_{\nu}^{\nu} + (\beta_{\nu}^{\nu})^T x. \quad (3.18)$$

Para actualizar el corte incumbente i_{ν} , se hace de la misma forma que cuando se encontró el corte para la iteración ν , es decir, se resuelve

$$\max \{ \pi^T [h(u_k) - T(u_k)\bar{x}^{\nu-1}] \mid \pi \in V \} \quad (3.19)$$

para obtener $\bar{\pi}_k$, y se reemplaza el corte el corte anterior i_{ν} por

$$\theta \geq \frac{1}{\nu} \sum_{k=1}^{\nu} (\bar{\pi}_k)^T [h(u_k) - T(u_k)x] = \alpha_{i_{\nu-1}}^{\nu} + (\beta_{i_{\nu-1}}^{\nu})^T x. \quad (3.20)$$

Los cortes restantes se actualizan como antes

$$\theta \geq \frac{\nu-1}{\nu} [\alpha_k^{\nu-1} + (\beta_k^{\nu-1})^T x] + \frac{1}{\nu} L = \alpha_k^{\nu} + (\beta_k^{\nu})^T x \quad (3.21)$$

para $k = 1, \dots, \nu-1, k \neq i_{\nu-1}$

Ahora, es tiempo de verificar si la solución incumbente actual pudiese ser cambiada o no. Se usarán las Figuras 3.2 y 3.3 para ilustración, y se usará la función $\phi_{\nu}(x)$ para definir el estimado actual de $\phi(x) = c^T x + Q(x)$. En las figuras se tiene que $\nu = 3$. Cuando comienza la iteración ν , la aproximación de $\phi(x)$ está dada por $\phi_{\nu-1}(x) = \phi_2(x)$ (líneas \overline{AB} y \overline{BC}). La solución incumbente es $\bar{x}^{\nu-1} = \bar{x}^2$ y la solución en la iteración es $x^{\nu} = x^3$. En la figura la posición de \bar{x}^2 es algo arbitraria, puesto que no se puede saber como se observarán las cosas en una iteración previa. Se sabe que $\phi_{\nu-1}(x^{\nu}) - \phi_{\nu-1}(\bar{x}^{\nu-1}) = \phi_2(x^3) - \phi_2(\bar{x}^2)$ (segmento \overline{AD}) representa una aproximación de cuanto se podría ganar por hacer a $x^{\nu} = x^3$ el nuevo valor del incumbente. Sin embargo, $x^{\nu} = x^3$ podría estar en la región donde $\phi_{\nu-1}(x) = \phi_2(x)$ corresponde a la mala aproximación de $\phi(x)$ porque su obtención se ha basado en dos cortes (\overline{AB} y \overline{BC}). La función $\phi_{\nu}(x) = \phi_3(x)$, por otro lado,

fue desarrollada alrededor de $x^v = x^3$ (Figura 3.3), y pudiera ser una buena aproximación dentro de esta región, al igual que $\phi_{v-1}(x) = \phi_2(x)$ lo es alrededor de $\bar{x}^{v-1} = \bar{x}^2$, porque su obtención se ha basado en tres y dos cortes, respectivamente (líneas \overline{EF} , \overline{FG} y \overline{GH} de la Figura 3.3, y \overline{AB} y \overline{BC} de la Figura 3.2, respectivamente). De ahí, $\phi_v(x^v) - \phi(\bar{x}^{v-1}) = \phi_3(x^3) - \phi_3(\bar{x}^2)$ representa la medida de lo que realmente se ha ganado al cambiar la solución incumbente (segmento \overline{EI} de la Figura 3.3). Si ocurre que

$$\begin{aligned} \phi_v(x^v) - \phi_v(\bar{x}^{v-1}) &< r[\phi_{v-1}(x^v) - \phi_{v-1}(\bar{x}^{v-1})] \\ &= \phi_3(x^3) - \phi_3(\bar{x}^2) < r[\phi_2(x^3) - \phi_2(\bar{x}^2)] \\ &= \overline{EI} < r\overline{AI} \end{aligned} \tag{3.22}$$

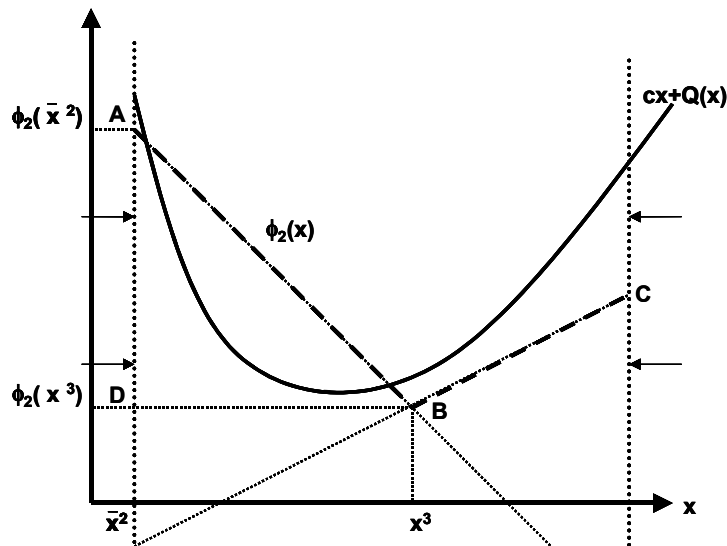


Figura 3.2. Primera representación para encontrar si la solución incumbente pudiese ser cambiada.

donde $r \in (0,1)$ e indica que al menos se está ganando una fracción r de lo que se espera, y se acepta a x^v como la nueva solución incumbente, es decir se hace $\bar{x}^v = x^v$ e $i_v = v$. Si no,

no se estuviese de acuerdo con el cambio, y se hace $\bar{x}^v = \bar{x}^{v-1}$ e $i_v = i_{v-1}$. Cuando se ha actualizado la solución incumbente, se resuelve un nuevo problema maestro para obtener x^{v+1} y se repite el proceso.

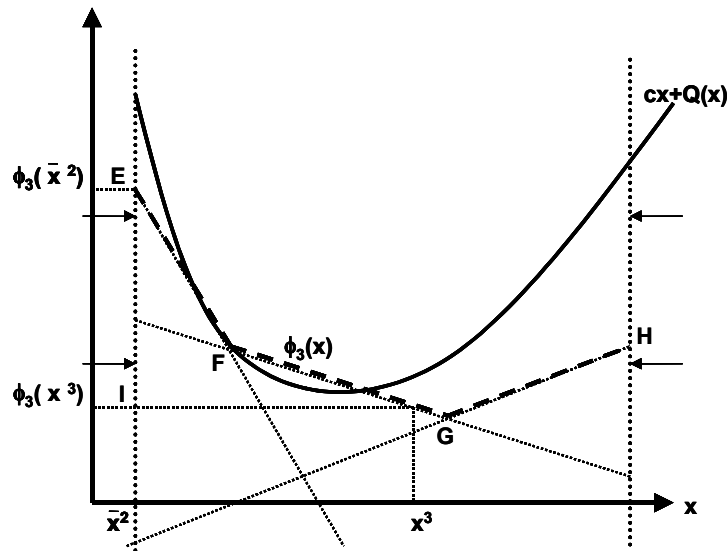


Figura 3.3. Segunda representación para encontrar si la solución incumbente pudiese ser cambiada.

3.4.2 Algoritmo SD con soluciones incumbentes

La estructura de un algoritmo SD que utiliza soluciones incumbentes para facilitar la convergencia se resume en los siguientes pasos (Higle y Sen, 1996):

- Paso 0. Defina $v=0$, $V_0=\{\emptyset\}$, $u^0 = E[u]$, x^l se obtiene de la Ecuación 3.14. $\bar{x}^0 = x^1$, $i_v = 0, f = 0, \bar{f} = 0, \rho^0 = 0$. Δ_l se fija un valor suficientemente grande y $r \in (0,1)$.
- Paso 1. Haga $v = v + 1$ y tome una muestra de las variables inciertas, u^v .
- Paso 2. Determine la aproximación lineal a la función de recurso $Q(x)$ (corte de optimalidad):

a) Resuelva el problema dual del problema de la segunda etapa:

$$\begin{aligned} \max \quad & \pi^T (h_v - T_v x^v) \\ \text{s. a} \quad & \pi^T W \leq q \end{aligned}$$

para encontrar los valores del vector de variables del problema dual π , π_v^v , y

haga $V_v = V_{v-1} \cup \pi_v^v$

b) Obtenga los coeficientes del corte de optimalidad:

$$\alpha_v^v = \frac{1}{v} \sum_{k=1}^v (\pi_k^v)^T h_k$$

$$\beta_v^v = \frac{1}{v} \sum_{k=1}^v (\pi_k^v)^T T_k$$

donde π_k^v es la solución al problema (para toda $k|k \neq v$):

$$\begin{aligned} \max \quad & \pi^T (h_k - T_k x^v) \\ \text{s. a} \quad & \pi \in V_v \end{aligned}$$

Observe que el vector de solución a esta serie de problemas sólo puede ser alguno de los vectores que se incluyen ya en el conjunto de soluciones, V_v .

Posteriormente se determina el corte que será agregado, así como la aproximación lineal del problema f .

$$\theta \geq \alpha_v^v + \beta_v^v x$$

$$f = \alpha_v^v + (c + \beta_v^v) x^v$$

c) Actualice los coeficientes de los cortes anteriores:

$$\alpha_k^v = \frac{v-1}{v} \alpha_k^{v-1}, \quad k = 1, \dots, v-1$$

$$\beta_k^v = \frac{v-1}{v} \beta_k^{v-1}, \quad k = 1, \dots, v-1$$

d) Si $v \equiv i_v + 1$ entonces reevaluar el corte i_v^{th} como sigue:

Resuelva el problema dual del problema de la segunda etapa

$$\begin{aligned} \max \quad & \pi^T (h_v - T_v x^v) \\ \text{s. a} \quad & \pi^T W \leq q \end{aligned}$$

para encontrar los valores del vector de variables incumbentes del problema

dual $\bar{\pi}$, $\bar{\pi}_v^v$, y haga $V_v = V_{v-1} \cup \bar{\pi}_v^v$.

Calcular los cortes del coeficiente de optimalidad de la solución incumbente

$$\alpha_{i_v}^v = \frac{1}{v} \sum_{k=1}^v (\bar{\pi}_k^v)^T h_k$$

$$\beta_{i_v}^v = \frac{1}{v} \sum_{k=1}^v (\bar{\pi}_k^v)^T T_k$$

Por último determine el corte que será agregado, así como la aproximación

lineal del problema \bar{f} .

$$\theta \geq \alpha_{i_v}^v + \beta_{i_v}^v x$$

$$\bar{f} = \alpha_{i_v}^v + (c + \beta_{i_v}^v) x^v$$

e) Examine el incumbente.

Si $f - \bar{f} < r\eta^v$ entonces $\bar{x}^v = x^v$, $i_v = v$, $\Delta_v = 2\Delta_{v-1}$.

En caso contrario, $\bar{x}^v = \bar{x}^{v-1}$, $i_v = i_{v-1}$, $\Delta_v = \Delta_{v-1}/2$.

- Paso 3. Resuelva el problema maestro de la primera etapa con la restricción adicional $-\Delta_v \leq x - \bar{x}^v \leq \Delta_v$

$$\begin{aligned} \min \quad & c^T x + \theta^v \\ \text{s. a} \quad & Ax = b \\ & \theta^v + \beta_k^v x \geq \alpha_k^v, \quad k = 1, \dots, v \\ & -\Delta_v \leq x - \bar{x}^v \leq \Delta_v \end{aligned}$$

para obtener x^{v+1} .

$$\eta = cx^{v+1} + \text{Max}_{1 \leq k \leq v} \{ \alpha_k^v + \beta_k^v x^{v+1} \},$$

$$\bar{\eta} = c\bar{x}^{v+1} + \text{Max}_{1 \leq k \leq v} \{ \alpha_k^v + \beta_k^v \bar{x}^v \},$$

$$\rho = \eta - \bar{\eta}.$$

- Paso 4. El algoritmo termina si:
 - a) No se requiere la adición de nuevos vectores de solución al conjunto de soluciones, V .
 - b) El estimado del valor de la función objetivo asociado con el incumbente es menor o igual a un error.

En la siguiente sección se resolverá un problema de planeación para la generación de energía, en el cual se requiere encontrar la capacidad óptima en la forma de un dispositivo de reserva. Como el dispositivo de reserva se trata con una sola variable, nos permite ilustrar gráficamente los conceptos aquí discutidos.

3.4.3 Aplicación ilustrativa

El siguiente ejemplo es un programa lineal y el único objetivo de presentarlo aquí es el de ilustrar la secuencia de soluciones incumbentes, $\{\bar{x}^v\}$, obtenida del algoritmo de la Sección 3.4.2 y compararla con la secuencia de soluciones $\{x^v\}$ generada por el algoritmo básico de la Sección 3.3.

El ejemplo consiste en un problema de planeación para una capacidad de reserva dentro del contexto de una planeación para la generación de energía (pgp). El problema pgp, especialmente la versión determinista, ha sido estudiado por varios autores (ver por ejemplo Luenberger, 1984; Sherali y col., 1984). El problema involucra la determinación de un incremento de la capacidad en la forma de un dispositivo de reserva, que será instalado de forma tal que se minimice la suma del costo de instalación y los costos de operación esperados. Para determinar los costos de operación se requiere información respecto a cada uno de los diferentes segmentos de carga l . Cada segmento de carga es caracterizado por dos cantidades: a) el número de horas de operación y b) la energía requerida, que en este problema es una variable aleatoria. De aquí, f_i denota el costo por kW-hr de energía producida por el generador i , y h_j denota el número fijo de horas de operación requerido por el segmento de carga j ; así $g_{ij} = f_i h_j$ proporciona el costo por kW del generador asignado i para proporcionar al segmento de carga j . Después se considera el elemento aleatorio del problema: la energía asociada con cada segmento de carga. Una vez que la energía requerida por cada segmento de carga es conocida, a través de un modelo de transporte se obtiene un costo mínimo que se asigna para cada capacidad de generación por cada segmento de carga; además, las restricciones de la demanda aseguran que la energía requerida por todos los segmentos se satisface, mientras que las restricciones de suministro corresponden a las limitaciones de capacidad por cada generador.

Para describir el problema notacionalmente, se definen

$l =$ el número de segmentos de carga

$N =$ el número de generadores existentes (el dispositivo de reserva se numera como $N + 1$).

$e_i =$ la capacidad del generador (en kW) $i, i = 1, \dots, N$

g_{ij} = el costo de operación por cada kW suministrado por el generador i al segmento de carga j , $i = 1, \dots, N+1$, $j = 1, \dots, l$

c = el costo por unidad (kW) de capacidad de reserva instalada

U = la capacidad de reserva máxima que puede ser instalada

Las variables aleatorias de interés son

\tilde{u}_j = la demanda (en kW) asociada con el segmento de carga j , $j = 1, \dots, l$

y las variables de decisión son

x = la capacidad del nuevo dispositivo

y_{ij} = el número de kW's suministrados por el dispositivo i al segmento de carga j .

El problema se establece como sigue:

$$\begin{aligned} \min \quad & cx + E[h(x, \tilde{u})] \\ \text{s. a} \quad & 0 \leq x \leq U \end{aligned}$$

donde

$$\begin{aligned} h(x, u) = \min \quad & \sum_{i=1}^{N+1} \sum_{j=1}^l g_{ij} y_{ij} \\ \text{s. a} \quad & \sum_{i=1}^{N+1} y_{ij} \geq u_j \quad j = 1, \dots, l \\ & \sum_{j=1}^l y_{ij} \leq e_i \quad i = 1, \dots, N \\ & \sum_{j=1}^l y_{N+1,j} \leq x \\ & y_{ij} \geq 0 \quad i = 1, \dots, N+1, j = 1, \dots, l \end{aligned}$$

Note que si uno de los N generadores originales corresponde a una opción de subcontrato con un apropiado límite de capacidad alto, el subproblema de recurso es factible para todos los valores de x y u ; de esta manera se satisface la consideración de recurso completo. Los datos que se usan para este problema se presentan en la Tabla 3.1. El

cuarto generador se ha designado como la opción de subcontrato. Cada una de las tres variables aleatorias corresponden a la demanda de varios segmentos de carga que fueron tratadas como variables aleatorias uniformemente distribuidas.

Tabla 3.1. Costos de generación (g_{ij}) y otros datos de la Aplicación Ilustrativa.

SEGMENTO DE CARGA (i)	GENERADORES EXISTENTES (j)				
	1	2	3	4	Reserva
1	10	20	30	1000	35
2	20/3	40/3	20	2000/3	70/3
3	10/3	20/3	10	1000/3	35/3
Capacidad existente	15	10	10	1000	0
$l = 3, N = 4, c = 500/\text{kW}, k = 150 \text{ kW}$ $\tilde{u}_1 \in [10, 50], \tilde{u}_2 \in [15, 35], \tilde{u}_3 \in [15, 75]$					

En la Figura 3.4 se ilustran dos trayectorias: una representa la trayectoria de soluciones incumbentes (MC_IN), y la otra representa la trayectoria generada por el algoritmo básico (MC_SD). Ambos métodos se inicializaron con las mismas soluciones de inicio y ambos fueron sujetos a las mismas secuencias de observaciones de \tilde{u} , los cuales fueron obtenidos para 100 muestras generados por Monte Carlo. Como se esperaba, se observa que ambas secuencias de solución regularmente visitan la región óptima, como indica la teoría presentada para estos algoritmos. Es importante mencionar que la línea asociada con las soluciones incumbentes proporciona una trayectoria mucho más estable que la línea asociada con el SD básico, la cual persiste cuando en el proceso del algoritmo se cae en puntos fuera de la región óptima (por ejemplo, $x^v = 0$). Esto sugiere que de los dos algoritmos, el de soluciones incumbentes proporciona una representación más confiable de una solución óptima.

En la Figura 3.5 se presenta el comportamiento de la función objetivo obtenido por el algoritmo básico (MC_SD) y por el valor objetivo promedio del algoritmo incumbente

(MC_IN). Del comportamiento asintótico de la función objetivo aquí ilustrado, se sugiere un criterio de paro sencillo para el algoritmo incumbente, el cual para mayor claridad será tratado en la siguiente sección.

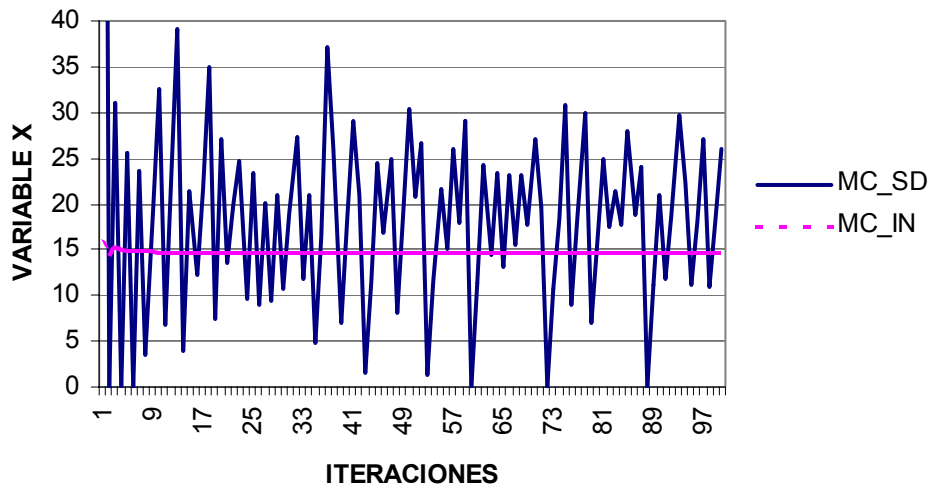


Figura 3.4. Soluciones para la variable x en el algoritmo SD básico e incumbente.

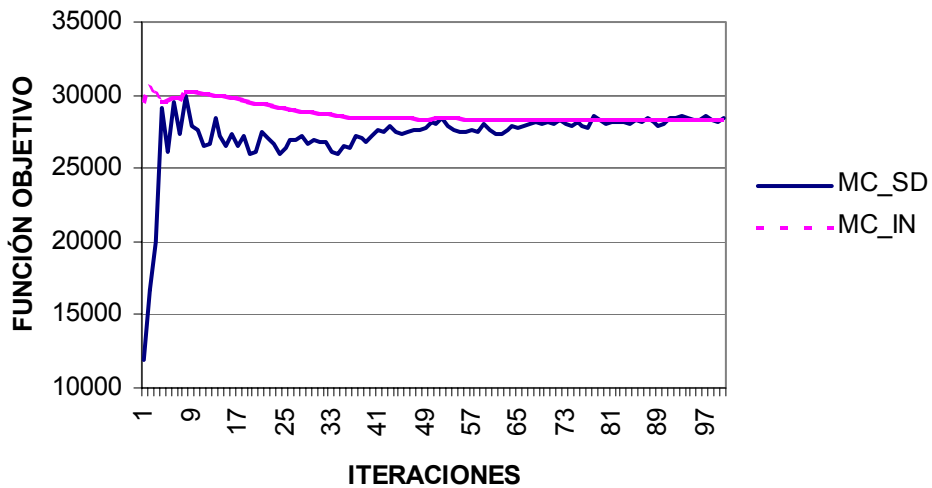


Figura 3.5. Comportamiento asintótico de la función objetivo.

3.4.4 Análisis del error

Higle y Sen (1991) propusieron dos clases de criterio para detener el curso de un algoritmo SD. La primera clase está basada sobre las propiedades asintóticas de la solución incumbente, y la segunda se basa en pruebas estadísticas de las condiciones de optimalidad.

En la primera clase de estos criterios, el valor objetivo óptimo se estima cuando un número suficientemente grande de iteraciones han pasado y el algoritmo termina cuando haya una evidencia clara de que el valor objetivo asociado con el incumbente es suficientemente cercano al valor objetivo óptimo.

Este proyecto se centra en realizar un análisis del error de los resultados asintóticos del valor estimado del objetivo como criterio de paro, basándose para ello en la metodología propuesta por Painton y Diwekar (1995). En dicho trabajo se incluye explícitamente en la función objetivo un término de error que está en función del número de muestras. El planteamiento es el siguiente:

Para m observaciones y_i de la variable aleatoria u , el estimado para la media o valor esperado y varianza son:

$$\bar{y} = \frac{1}{m} \sum_{i=1}^m y_i, \quad (3.22)$$

$$\hat{\sigma}_y^2 = \frac{1}{m-1} \sum_{i=1}^m (y_i - \bar{y})^2. \quad (3.23)$$

La distribución de muestreo de \bar{y} depende esencialmente de la distribución de u . Por ejemplo, si u tiene una distribución normal, \bar{y} se describe por una distribución t . La distribución t converge a una distribución normal cuando m es suficientemente grande. Se sabe además que para una m grande (no importa qué tipo de distribución sea u), la distribución de \bar{y} converge a una distribución normal con desviación estándar de la media

de muestreo igual a $\hat{\sigma}_y / \sqrt{m}$. Este es un resultado del teorema del límite central que permite calcular los límites de error probabilísticos del valor esperado, dado por $(\bar{y} - c\hat{\sigma}_y / \sqrt{m}, \bar{y} + c\hat{\sigma}_y / \sqrt{m})$, donde c se obtiene de una distribución normal de media cero y desviación unitaria de una variable aleatoria φ . Por ejemplo $\Pr(-c < \varphi < c) = \delta$. Para un intervalo de confianza alto, como $\delta = 0.95$, se encuentra que del inverso de la distribución normal unitaria de forma que c es aproximadamente igual a dos.

Mientras que por un lado el teorema del límite central proporciona los estimados de los valores esperados para un número de muestras m dadas por la técnica de muestreo, por el otro este estimado proporciona el límite superior de la banda de error para un muestreo más uniforme y menos independiente. En este trabajo, a diferencia del de Painton y Diwekar (1995), no se incorporará en la función objetivo el término que penaliza el error $(2\hat{\sigma} / \sqrt{m})$, ya que le daría un carácter no lineal al modelo. Así, se usará únicamente un análisis de error para alcanzar este límite superior a través de manipular el número de muestras en el algoritmo SD.

Considérese que γ^v representa la suma estadística de los valores objetivos de las soluciones incumbentes obtenidas durante las primeras v iteraciones. Se considera además, que γ^v satisface la condición de que $|\bar{f}_v(\bar{x}^{v-1}) - \gamma^v| \rightarrow 0$ si la secuencia $\{\bar{f}_v(\bar{x}^{v-1})\}$ converge. γ^v es un promedio estadístico,

$$\gamma^v = \frac{1}{v} \sum_{k=1}^v \bar{f}_k(\bar{x}^k) + \frac{2\hat{\sigma}_{\bar{f}_v}}{\sqrt{v}}. \quad (3.24)$$

El algoritmo terminaría si la siguiente condición llegara a satisfacerse:

$$\left| \frac{\gamma^v - \gamma^{v-1}}{\gamma^v} \right| \leq \varepsilon \quad (3.25)$$

donde $\varepsilon > 0$ es un nivel de tolerancia aceptable. En caso contrario, se debe aumentar el número de muestras v .

Para el ejemplo mostrado en la Sección 3.4.2, ahora se resuelve con el criterio de paro incluido en el algoritmo SD incumbente. Los resultados obtenidos se presentan en la Figura 3.6.

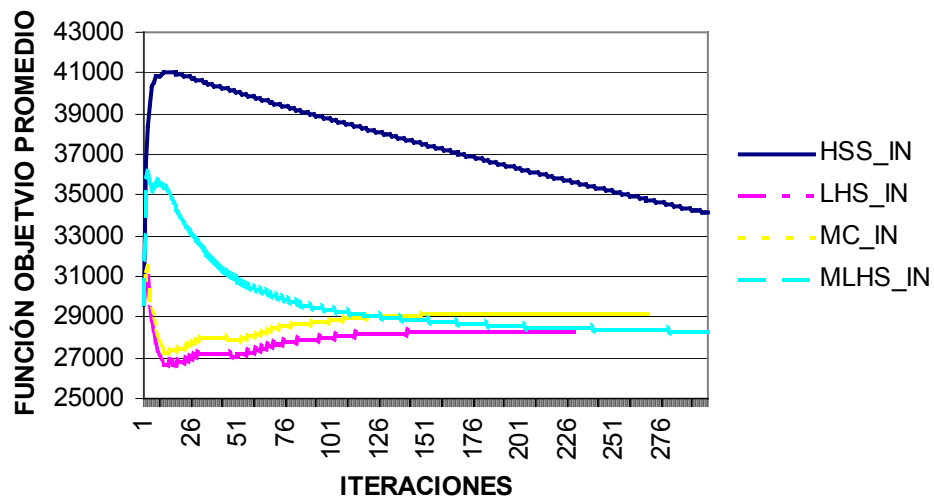


Figura 3.6. Resultados obtenidos para la Aplicación Ilustrativa con criterio de paro y para diferentes técnicas de muestreo.

Los resultados presentados en la Figura 3.6 corresponden a un error del límite superior menor o igual a 1×10^{-6} . La gráfica muestra el comportamiento de la función objetivo promedio para 300 muestras obtenidas por diferentes técnicas de muestreo: la secuencia de Hammersley (HSS_IN), Latin Hypercube (LHS_IN), Monte Carlo (MC_IN) y para la Median Latin Hypercube (MLHS_IN). En esta misma gráfica se puede observar que cuando se utiliza un muestreo Latin Hypercube o una Monte Carlo, el algoritmo converge

en 230 y 270 iteraciones, respectivamente, mientras que con las otras técnicas restantes el algoritmo no logra converger en el rango de muestras dado. Se puede inferir con ello que para obtener una rápida convergencia en un menor número de muestras, las técnicas de muestreo Latin Hypercube y Monte Carlo son las más adecuadas para utilizar en un algoritmo incumbente. Es importante señalar que este resultado no era el esperado, puesto que trabajos previos reportan a la secuencia Hammersley como la técnica más apropiada. En el capítulo siguiente y en los ejemplos a resolver se seguirán utilizando las cuatro técnicas de muestreo para poder establecer una conclusión más convincente.

En la siguiente sección se presenta la implementación computacional del algoritmo que utiliza soluciones incumbentes.

3.5. Implementación computacional

La implementación computacional del algoritmo involucra una interfase que integra la técnica de muestreo de la secuencia de Hammersley (Kalagnanam y Diwekar, 1997) codificada en Fortran, el ambiente de modelación GAMS (Brooke y col., 1998), y un programa principal en C++. El programa principal genera los problemas MILP y LP apropiados para cada una de las iteraciones del SD incumbente, transfiere el control de ejecución y verifica la convergencia del algoritmo. En la Figura 3.7 se ilustra el diagrama de flujo con los pasos de solución, en la Figura 3.8 se presenta su implementación computacional, y en el Apéndice A se muestra un listado de codificación del algoritmo SD incumbente.

En el diagrama de flujo de la Figura 3.7 se requiere primero de la generación de un muestreo de los parámetros inciertos, de inicializar las variables de primera etapa y de la solución incumbente para resolver el primer problema de la segunda etapa. Para encontrar

estos valores iniciales se recomienda resolver el problema de la Ecuación 3.14 con los valores medios de los parámetros estocásticos.

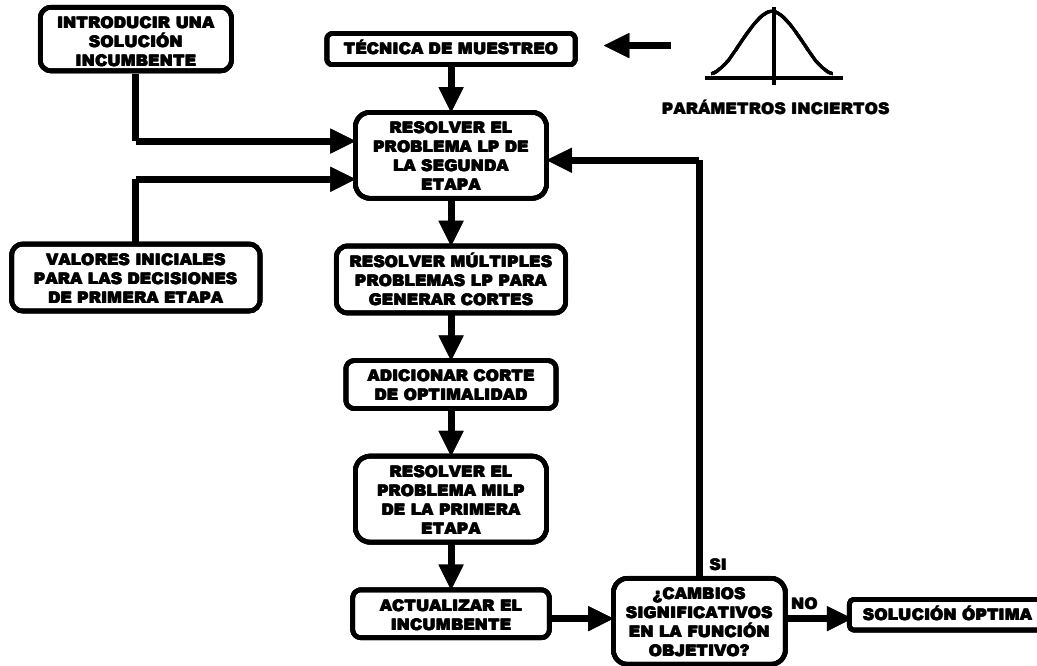


Figura 3.7. Diagrama de flujo del algoritmo SD incumbente.

Una vez resuelto el problema de la segunda etapa se genera el corte para las soluciones regular e incumbente. Ambos cortes se adicionan al problema de la primera etapa. La solución de este problema se obtiene a través de un resolvidor de GAMS. Note que en esta parte se obtienen los nuevos valores de las variables de decisión de primera etapa y el estimado del valor de la función objetivo, y que se puede calcular un estimado del valor de la función objetivo de la solución incumbente. Posteriormente se verifica si la solución incumbente será actualizada o no, así como también si existen diferencias significativas entre los estimados de la función objetivo. En caso de existir diferencias

significativas se continúa con el algoritmo, en caso contrario, se ha obtenido una solución óptima.

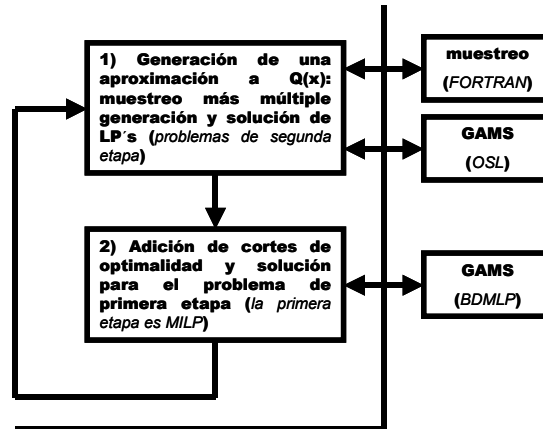


Figura 3.8. Implementación computacional del algoritmo SD incumbente.

En los capítulos siguientes se presentará la formulación de dos problemas interesantes en ingeniería de procesos: la optimización de cadenas de suministro y la seguridad en redes de agua municipales. En ambos casos los problemas se formulan como modelos estocásticos mixto-entero lineales de dos etapas con recurso, lo que permite el empleo del algoritmo que se ha propuesto en este capítulo.

Capítulo 4

Cadenas de Suministro: Un Problema de Optimización con Incertidumbres en las Disponibilidades de la Materia Prima y en las Demandas del Mercado

En este capítulo se aborda el problema de optimización de cadenas de suministro. Se define lo que es una cadena de suministro y la clase de problemas que se han propuesto para describirla. Se describe el planteamiento del problema y se presenta su formulación matemática como un problema estocástico de dos etapas que involucra incertidumbres en la disponibilidad de la materia prima y en las demandas del mercado. Para mostrar los alcances del modelo propuesto se resuelven dos ejemplos, donde sus resultados son comparados con los obtenidos por otras técnicas de solución estocástica.

4.1 Introducción

En fechas recientes, la optimización de las cadenas de suministro de las empresas ha llamado la atención de la industria y de la academia (ver Bok y col, 2000; Gupta y Maranas, 2003; Ryu y Pistikopoulos, 2005). A pesar de que no hay una forma sistemática de definir el alcance del problema, Chopra y Meindl (2001), dependiendo del marco del tiempo sobre el cual la decisión se está aplicando, proponen categorizarlo en problemas estratégicos o de diseño, problemas tácticos o de planeación, y en problemas operacionales o de calendarización.

En este trabajo se abordarán los problemas estratégicos o de diseño, en los cuales las empresas deciden como se estructurará o configurará la cadena de suministro y los procesos que se desarrollarán en cada etapa. En estas decisiones se incluyen capacidades de producción y ubicación de almacenes, productos que serán procesados o almacenados, el tipo de transporte y el sistema de información que se intercambiará a lo largo de la cadena.

En general, una decisión estratégica se realiza a largo plazo y tiene un costo alto si se pretendiera cambiar una vez que se ha establecido. Por lo tanto, para tomar estas decisiones, deben considerarse las incertidumbres de los próximos años (demanda del producto, disponibilidad de materia prima y sus costos) además de las incertidumbres debidas al proceso (constantes de reacción, calidad de la materia prima y tiempos de procesamiento). Aunque por un lado el hecho de involucrar estas incertidumbres en una forma substancial y primordial convertiría al problema en uno mas realista (Sahinidis y col., 1989; Iyer y Grossmann, 1998), por otro lado lo convertiría también en uno más complejo y difícil de resolver (Applequist y col., 2000).

Como una contribución en este trabajo, se propone un modelo de diseño de la cadena de suministro donde se incluyen explícitamente las incertidumbres en las

disponibilidades de la materia prima y en las demandas de producto del mercado. Aunque existen varias propuestas en este sentido, estos problemas se formularán aquí como modelos estocásticos mixto-entero lineales de dos etapas con recurso fijo y se considerará el caso entero en la primera etapa. Además, se aplicará el algoritmo SD para resolver los problemas de optimización resultantes.

Para exponer una idea clara de este trabajo de investigación, este capítulo se ha organizado de la siguiente manera: en la Sección 4.2, se define lo que es una cadena de suministro; enseguida, en la Sección 4.3 se plantea y formula el problema como un modelo estocástico mixto-entero lineal en la primera etapa de un problema de dos etapas con recurso; posteriormente, en la Sección 4.4 se resuelven dos ejemplos numéricos y se comprara con resultados reportados previamente en la literatura; y por último, en la Sección 4.5 se presenta una discusión entre las ventajas y desventajas de resolver un problema por un método de solución discreto o uno continuo.

4.2 Cadenas de suministro

Una cadena de suministro se puede definir como una red de socios comerciales que interactúan entre sí para transformar una materia prima en productos terminados para los clientes (ver Figura 4.1). Cada uno de estos socios comerciales o partes que constituyen la cadena de suministro desarrollan las actividades necesarias para llevar a cabo esta transformación. Algunas de las partes que pueden constituir a una cadena de suministro se presentan en la Figura 4.1, pero el análisis no se limita a las ahí mostradas. Estos elementos pueden ser un mercado que proporciona la materia prima, las plantas de fabricación o de procesamiento del producto, los centros de almacenamiento del producto terminado, los

diferentes tipos o medios de distribución, los centros de venta del producto (ya sea mayoristas o minoristas), y hasta las necesidades particulares de cada uno de los clientes.

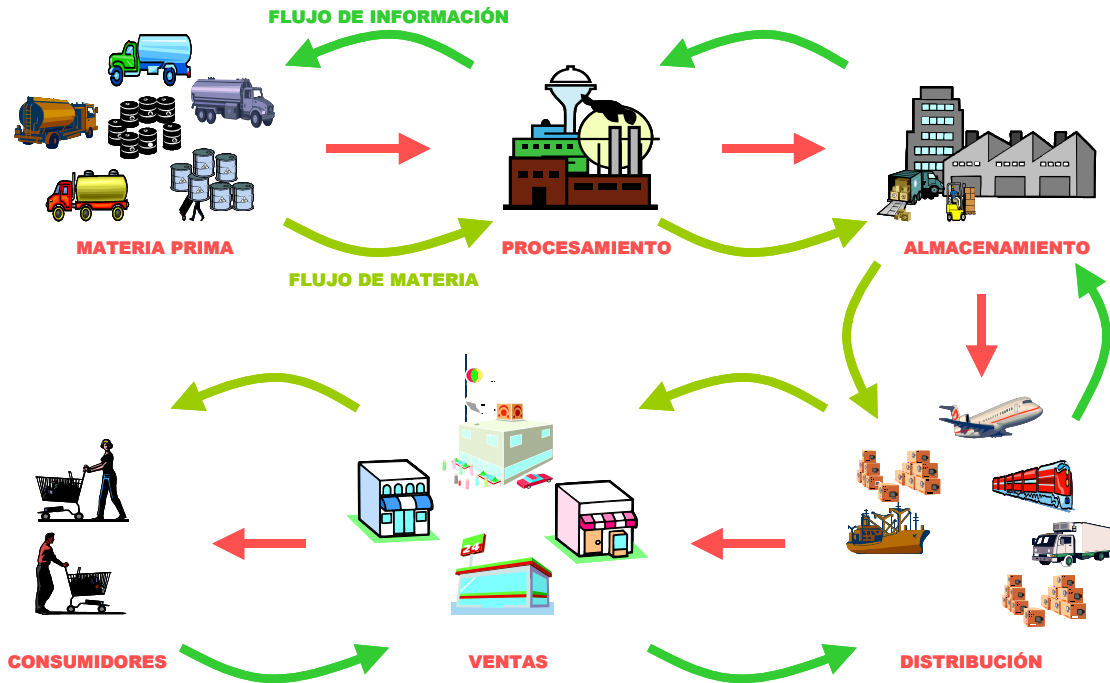


Figura 4.1. Ejemplo de una cadena de suministro.

La interacción entre cada una de estas partes de la cadena de suministro es a través de flujos de información, los cuales pueden tomar la forma ya sea de flujo de materia, o bien, de flujo de capital. Por lo general siempre existe un flujo de información oportuno cuando la dirección de éste es hacia delante de la cadena, mientras que en sentido inverso siempre se tiene una información con retraso debida, entre otros factores, a las incertidumbres de los mercados. Por ello, en un mundo globalizado y competitivo una cadena de suministro eficiente requiere que ambos flujos de información sean oportunos.

Desde el punto de vista de la cadena de suministro, las plantas de fabricación o de procesamiento ya no competirían como unidades independientes, sino más bien como una liga de socios comerciales, en donde el éxito primordial de la empresa dependerá de la habilidad de su directivo para integrar la red de socios comerciales (Shapiro, 2001).

La planeación de la cadena de suministro es entonces la gestión de los flujos de información entre cada una de las etapas que constituyen la cadena para maximizar su rentabilidad. El objetivo de una cadena de suministro es la de minimizar el costo total de la cadena, mientras se satisface la demanda final del cliente, siendo los costos por inventario, los costos por faltantes y los costos por distribución algunas de las áreas mayormente afectadas por las decisiones tomadas en las cadenas de suministro (Simchi-Levi y col., 1998).

Nótese entonces que la planeación de la cadena de suministro se basa en la premisa de que la dependencia creada entre los participantes del suministro requieren que las actividades de coordinación tengan un adecuado flujo de información, ya sea de capital o de materia, entre cada uno de sus miembros. Esta definición de coordinación requiere de información exacta y oportuna que pueda ser compartida por todos los miembros de la cadena. Esto también significa que, si no hay interdependencia, no hay nada por coordinar.

4.3 Modelo de planeación para la cadena de suministro estratégica

La planeación de la producción es un tópico de la industria de procesos químicos y es la parte central de todas las operaciones de la planta (Clay y Grossmann, 1997). Dentro de las aplicaciones de la planeación se manejan características como selección y disposición de la materia prima, así como de la optimización del balance de materia total y su conversión. Por su naturaleza, la planeación involucra el tomar decisiones óptimas

acerca de futuros eventos basados en la información que se encuentra disponible y de proyecciones a futuro.

El problema general se puede establecer como sigue: dado el modelo de procesos (por ejemplo, un arreglo de restricciones), de información al momento y de eventos futuros, y de una función objetivo que refleja las preferencias costo/riesgo, encontrar la solución que minimice la función objetivo sin violar las restricciones. Mientras que la información al momento puede ser conocida, los eventos futuros son inevitablemente inciertos.

En la Figura 4.2 se muestra una red simple de producción con múltiples alimentaciones, productos intermedios, productos finales, todos o parte de los cuales dependen de las condiciones inciertas del mercado que gobiernan el suministro, demanda y precio. Los puntos de distribución de flujos se representan por nodos y las unidades de procesamiento por rectángulos. Las materias primas de la entrada a la red e intermedias se compran a los mercados sujetas a los precios establecidos, contratos y disponibilidades. Los productos se venden a los mercados basados en las demandas, precios establecidos, contratos y capacidades de producción.

Las incertidumbres se pueden involucrar en la función objetivo y las restricciones. La función objetivo incluye la incertidumbre en los costos futuros, precios y acciones. Las restricciones del modelo del proceso pueden incluir dos tipos de parámetros estocásticos: endógenos (por ejemplo, los coeficientes de producción) y exógenos (por ejemplo, demandas futuras del producto).

Las próximas secciones describen el planteamiento del problema de la cadena de suministro y su reformulación como un problema de programación estocástica de dos etapas mixto-entero lineal con recurso (SMILPwR).

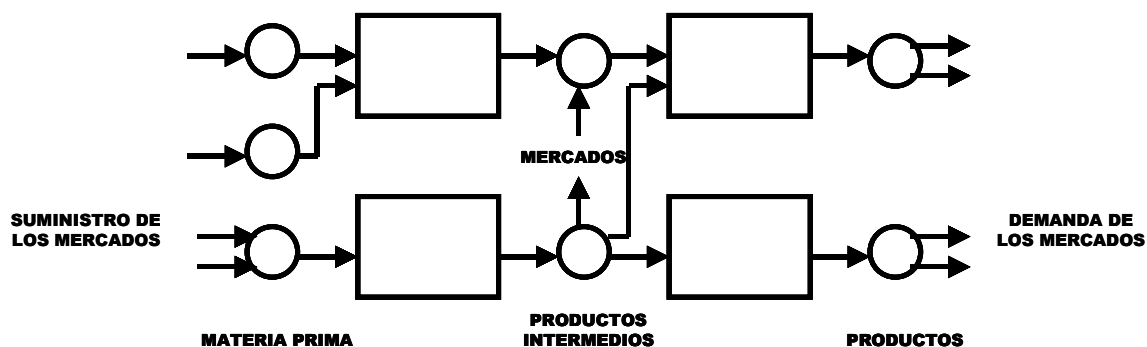


Figura 4.2. Red de producción.

4.3.1 Planteamiento del problema

Considérese el problema propuesto por Iyer y Grossmann (1998) que consiste de una superestructura de NP procesos y NC químicos (entre los que se incluyen materias primas, productos intermedios y productos finales). Hay NK corrientes en la red, y los balances de materia de cada proceso se expresan linealmente con respecto a la velocidad de producción del producto principal para ese proceso. La capacidad de los procesos se determina por la velocidad de flujo del producto principal. Las demandas y los precios de los químicos pueden variar por cada periodo de tiempo sobre el horizonte de planeación.

La función objetivo a maximizar es el valor presente neto (VNP) del proyecto sobre el horizonte de planeación que consiste de NT periodos de tiempo. El VNP incluye los costos de inversión, costos de operación y los ingresos obtenidos por ventas en los NM mercados.

4.3.2 Formulación SMILPwR

Aunque los fundamentos de nuestro modelo son los mismos que han propuesto Iyer y Grossmann (1998), la contribución principal a este respecto es la reformulación del problema en uno que es mixto entero en la primera etapa de un problema estocástico mixto-entero lineal de dos etapas. Esto es significativo por dos razones: *i*) primera, las incertidumbres son formalmente incorporadas dentro del modelo, y *ii*) el algoritmo incumbente SD (Higle y Sen, 1991) se puede usar para proporcionar una aproximación a la solución óptima del problema.

En la formulación del modelo estocástico se ha respetado la notación original y se ha agregado la parte que le da la dimensión estocástica al problema. Esta se define a continuación:

Índices

- $i =$ procesos ($i = 1, \dots, NP$)
- $j =$ químicos ($j = 1, \dots, NC$)
- $t =$ periodos de tiempo ($t = 1, \dots, NT$)
- $k =$ corrientes en la red ($k = 1, \dots, NK$)
- $l =$ mercados ($1, \dots, NM$)

Parámetros

- $I(j) =$ arreglo de corrientes del químico j que se produce en el complejo
- $O(j) =$ arreglo de corrientes del químico j consumido en el complejo
- $L_i =$ arreglo de corrientes que corresponde a las entradas/salidas del proceso i
- $m_i =$ corriente que corresponde al producto principal del proceso $i \in (m_i \in L_i)$
- $NEXP(i) =$ número máximo de expansiones permitidas del proceso i

$CI(t)$ = límite de inversión de capital en el periodo t

α_{it} = coeficiente de costo de inversión del proceso i ajustado en el periodo t
(\$/capacidad)

$\overline{\alpha}_i$ = coeficiente de costo de inversión del proceso i no ajustado en el periodo t
(\$/capacidad)

β_{it} = coeficiente de costo fijo de inversión del proceso i ajustado en el periodo t
(\$)

$\overline{\beta}_i$ = coeficiente de costo fijo de inversión del proceso i no ajustado en el periodo t
(\$)

δ_{it} = costo de operación del proceso i en el periodo t (\$/cantidad de producción)

γ_{jlt} = precio de venta del químico j en el mercado l en el periodo t (\$/venta)

Γ_{jlt} = precio del químico j en el mercado l en el periodo t (\$/compra)

θ_{jlt} = precio del producto j en el mercado l en el periodo t (\$/compra)

$u1_{jlt}$ = disponibilidad del químico j en el mercado l en el periodo t (\$/venta)

$u2_{jlt}$ = demanda del químico j en el mercado l en el periodo t (\$/compra)

Variables

y_{it} = variable de decisión para la expansión del proceso i en el periodo t , es igual a
1 si hay una expansión, 0 en caso contrario

Q_{it} = capacidad del proceso i en el periodo t

QE_{it} = capacidad de expansión del proceso i en el periodo t

F_{jlt} = faltante del producto j en el periodo t en el mercado l

P_{jlt} = cantidad de producto j comprado en el periodo t en el mercado l

S_{jlt} = cantidad del producto j vendida en el periodo t en el mercado l

W_{kt} = cantidad de flujo de la corriente k en el periodo t

Superíndices

$()^L$ = límite inferior

$()^U$ = límite superior

En el modelo se incluyen explícitamente las incertidumbres en la disponibilidad y demanda de los productos. Se considera una función de distribución de probabilidad continua para representar a cada uno de estos parámetros inciertos, consideración que convierte al problema en un programa estocástico. De esta forma, se propone la separación de las restricciones del modelo en dos etapas, siguiendo el concepto de recurso.

En la primera etapa se toman las decisiones de cuáles procesos y a qué capacidades van a producir tales procesos a través de las variables binarias y_{it} . La capacidad total del proceso Q_{it} y la capacidad a expandir QE_{it} también son variables continuas de primera etapa. La función objetivo en la Ecuación 4.1 es el VNP, el cual incluye los costos de inversión de los procesos y sus expansiones (α_{it} y β_{it}) más los costos esperados de operación, compra de materia prima y venta de producto, $E_{it}R(Q_{it}, u_{jlt})$. La Ecuación 4.2 limita la expansión de la capacidad en cada periodo de tiempo t . La Ecuación 4.3 define la capacidad total en cada periodo de tiempo. La Ecuación 4.4 limita el número de expansiones $NEXP(i)$ para el proceso i sobre el horizonte de planeación, y la Ecuación 4.5 limita el capital disponible a invertir, CI_t , en cada periodo de tiempo. Los parámetros $\bar{\alpha}_{it}$ y $\bar{\beta}_{it}$ son coeficientes de costos no ajustados al VNP.

Primera etapa:

$$\text{Min} \quad - \sum_{t=1}^{NT} \sum_{i=1}^{NP} (\alpha_{it} QE_{it} + \beta_{it} y_{it}) + E_u R(Q_{it}, u_{jlt}) \quad (4.1)$$

s. a

$$y_{it}QE_{it}^L \leq QE_{it} \leq y_{it}QE_{it}^U \quad i = 1, \dots, NP; t = 1, \dots, NT \quad (4.2)$$

$$Q_{it} = Q_{i,t-1} + QE_{it} \quad i = 1, \dots, NP; t = 1, \dots, NT \quad (4.3)$$

$$\sum_{t=1}^{NT} y_{it} \leq NEXP(i) \quad i \in I' \subseteq \{1, 2, \dots, NP\} \quad (4.4)$$

$$\sum_{i=1}^{NP} (\bar{\alpha}_{it}QE_{it} + \bar{\beta}_{it}y_{it}) \leq CI_t \quad t = 1, \dots, NT \quad (4.5)$$

$$\begin{aligned} QE_{it}, Q_{it} &\geq 0 \\ y_{it} &\in \{0, 1\} \end{aligned} \quad (4.6)$$

En la segunda etapa, la función de recurso $R(Q_{it}, u_{jlt})$ toma en cuenta los costos de ventas, costos de operación, costos de compras y costos de faltantes de demanda no satisfecha (Ecuación 4.7). P_{jlt} , y S_{jlt} son las cantidades de compra y venta de producto j en el mercado l , y W_{kt} es la cantidad de flujo de la corriente k . Todas ellas se consideran variables de decisión de segunda etapa.

La Ecuación 4.8 establece límites de producción basados en la capacidad existente y la Ecuación 4.9 es un balance de materia para cada químico en el complejo. La Ecuación 4.10 es el balance de masa de los químicos en cada proceso. Las Ecuaciones 4.11 y 4.12 incorporan las incertidumbres en las disponibilidades ($u1_{jlt}$) y demandas de los productos ($u2_{jlt}$), respectivamente. En este modelo se considera que las disponibilidades y demandas del químico j son continuas, a las cuales puede asignárseles una función de distribución de probabilidad. Observe en la Ecuación 4.12 que la venta total disponible del químico j puede no satisfacer toda la demanda del mercado l , y que, para que el problema de la segunda etapa siempre produzca una solución factible (condición de recurso completo), se considera un faltante de producción F_{jlt} con respecto a la demanda. Este último término será

penalizado en la función objetivo por un costo θ_{jlt} por falta de una pérdida potencial de ventas o incumplimiento de contrato.

Segunda etapa:

$$R(Q_{it}, u_{jlt}) = \text{Min} - \left(\sum_{t=1}^{NT} \sum_{j=1}^{NC} \sum_{l=1}^{NM} \gamma_{jlt} S_{jlt} - \sum_{t=1}^{NT} \sum_{i=1}^{NP} \delta_{it} W_{m,t} - \sum_{t=1}^{NT} \sum_{j=1}^{NC} \sum_{i=1}^{NM} \Gamma_{jlt} P_{jlt} - \sum_{t=1}^{NT} \sum_{j=1}^{NC} \sum_{i=1}^{NM} \theta_{jlt} F_{jlt} \right) \quad (4.7)$$

s. a

$$W_{m,t} \leq Q_{it} \quad i = 1, \dots, NP; t = 1, \dots, NT \quad (4.8)$$

$$\sum_{j=1}^{NM} P_{jlt} + \sum_{k \in I(j)} W_{kt} = \sum_{j=1}^{NM} S_{jlt} + \sum_{k \in O(j)} W_{kt} \quad j = 1, \dots, NC; t = 1, \dots, NT \quad (4.9)$$

$$W_{kt} = \mu_{ik} W_{m,t} \quad k \in L_i / m_i; i = 1, \dots, NP; t = 1, \dots, NT \quad (4.10)$$

$$P_{jlt} \leq u1_{jlt} \quad l = 1, \dots, NM; t = 1, \dots, NT \quad (4.11)$$

$$S_{jlt} + F_{jlt} = u2_{jlt} \quad l = 1, \dots, NM; t = 1, \dots, NT \quad (4.12)$$

$$W_{kt}, P_{jlt}, S_{jlt}, F_{jlt} \geq 0 \quad (4.13)$$

Una alternativa a esta formulación es crear el modelo de la cadena de suministro como un problema de programación estocástica mixto-entero lineal multi-etapa (MSMILPwR), aunque en este trabajo se ha restringido a la solución de un modelo de dos etapas. Para algunos modelos de programación multi-etapa no es necesario que los periodos de tiempo correspondan a las etapas como se ha hecho aquí (ver Dantzig, 1988; Bienstock y Shapiro, 1985). En ciertos casos, todos los periodos de tiempo futuro pueden ser agrupados dentro de la segunda etapa.

4.4 Ejemplos y resultados

Los ejemplos tienen el mismo modelo y solamente varían en estructura y tamaño. El objetivo principal consiste en determinar la mejor estructura para una red de procesos. La calendarización de las expansiones de la capacidad consideran un límite en la inversión y un incremento en la demanda, teniendo en cuenta las incertidumbres en la disponibilidad de la materia prima y en la demanda de los productos.

El Ejemplo 1 es un problema pequeño con tres químicos y tres procesos. En el Ejemplo 2 hay ocho procesos, los cuales se traducen en un mayor número de variables binarias. Para comparar los resultados obtenidos por el algoritmo SD incumbente, se han implementado diferentes algoritmos de solución: Descomposición de Benders, Equivalente Determinista, y el Método L-shaped. Además, para el caso del algoritmo SD incumbente, se han utilizado diferentes técnicas de muestreo para analizar cuál de ellas tiene una más rápida convergencia y es más conveniente utilizar para ejemplos posteriores.

Las características específicas de cada ejemplo son descritas en su respectiva sección. Para la simulación numérica se ha utilizado una hp workstation xw6000 y los modelos generados LP y MILP han sido resueltos con CONOPT y BDMLP respectivamente, dentro de la interfase de modelación GAMS (Brooke y col., 1998),

4.4.1 Ejemplo 1

Considere la superestructura mostrada en la Figura 4.3 consistente de tres procesos que pueden ser usados para producir el químico 3 a partir de las materias primas del químico 1 y el químico 2.

Se considera sólo un periodo de operación dentro de un horizonte de 20 años. El objetivo es seleccionar cuál proceso y qué capacidad usar en este periodo de tiempo, con

demandas inciertas para el químico 3, así como disponibilidades inciertas para los químicos 1 y 2 (como en Sahinidis y col., 1989).

Comenzando con una capacidad inicial para el proceso 2 de 10,000 ton/yr, y con un límite de inversión de \$200,000, se debe encontrar la mejor configuración para las expansiones de capacidad dentro del horizonte de tiempo dado. Los valores de los parámetros inciertos se presentan en la Tabla 4.1, los datos para calcular los costos se reportan en la Tabla 4.2, los datos para los coeficientes de reacción y capacidades existentes se presentan en la Tabla 4.3, y finalmente, en la Tabla 4.4 se incluyen datos de inversión, capacidades de expansión y los valores para calcular el VNP.

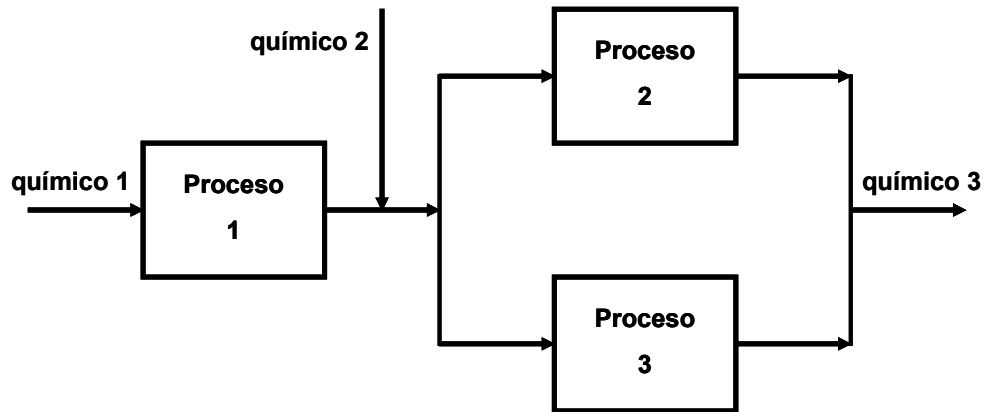


Figura 4.3. Estructura para el Ejemplo 1.

Tabla 4.1. Disponibilidad y demanda para el Ejemplo 1

Demanda	Ch1	Ch2	Ch3
Baja	6000	18000	27000
Alta	5400	20000	30000

Los resultados de la convergencia para este ejemplo con el algoritmo incumbente y diferentes técnicas de muestreo se presentan en Figura 4.4. Se requirieron 300 muestras

para alcanzar un error del límite superior menor o igual a 1×10^{-6} . En esta gráfica se observa que con las técnicas de muestreo Latin Hypercube y Monte Carlo se obtiene una rápida convergencia, necesiéndose para ello de 112 y 128 iteraciones, respectivamente. Mientras que las técnicas Latin Hypercube media y la secuencia de Hammersley no logran una convergencia para este número de muestras. Nótese de nueva cuenta que con la técnica de muestreo de la secuencia de Hammersley no se presenta un comportamiento asintótico, solo un decaimiento, generándose la impresión de que no lograría converger ni para un número de muestras grande.

Tabla 4.2. Datos de costos para el Ejemplo 1

Proceso	α en 10^2 \$/ton	β en 10^5 \$	δ en 10^2 \$/ton
1	1.38	85	0.40
2	2.72	73	0.60
3	1.76	110	0.50
Químico		Γ en 10^2 \$/ton	
1		4.0	
2		9.6	
Químico		γ en 10^2 \$/ton	
3		26.20	

Tabla 4.3. Datos para los coeficientes de reacción y capacidades existentes del Ejemplo 1.

PROCESO	μ	$Q_{i,0}$
1	1.0	0.0
2	1.0	10 000
3	1.0	0.0

Tabla 4.4. Datos para la inversión, para las capacidades de expansión y para el VNP del Ejemplo 1.

Máxima capacidad a invertir	\$200 000
Capacidad de expansión mínima	5000 ton
Capacidad de expansión máxima	50 000 ton
Tiempo de vida del proyecto	20 años
Tasa de interés	0.10
Tasa de impuestos	0.45
Factor del capital de trabajo	0.15
Factor del valor de salvamento	0.10

Los resultados que se tienen para las variables a analizar entre las técnicas de muestreo Latin Hypercube y Monte Carlo son iguales, existiendo sólo una pequeña diferencia del 0.65% en el estimado del valor objetivo promedio, -1.01764×10^8 y -1.01099×10^8 , respectivamente. Esto debido a la variabilidad, producto del muestreo. Nótese que el signo negativo es porque se está minimizando el negativo del VNP (lo que equivale a maximizar el VNP).

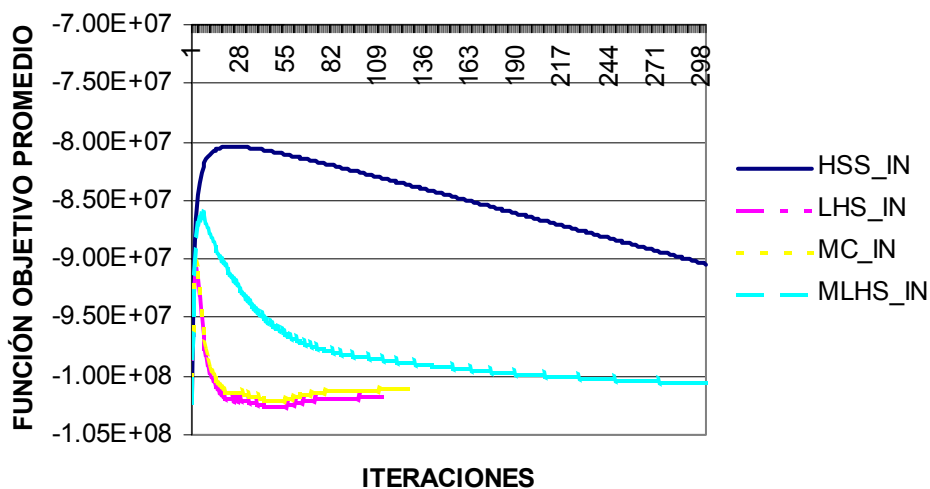


Figura 4.4. Convergencia del Ejemplo 1 para diferentes técnicas de muestreo.

En la Tabla 4.5 se presentan los resultados obtenidos por diferentes métodos de solución. Se puede observar que todos los métodos predicen la misma configuración a utilizar, así como las mismas capacidades y expansiones a considerar. Nótese que el valor estimado del VNP por el algoritmo SD tiene una diferencia de casi el 1.2 % con respecto a los valores obtenidos por los otros métodos de solución. Esto puede deberse a varias razones; una porque el número de escenarios considerados es mayor por el algoritmo SD

(128) contra sólo 8 de los métodos anteriores. Segundo, el algoritmo SD está basado en una técnica de muestreo, por lo que el criterio de paro se basa de igual manera en un estimado estadístico, contrario al de las técnicas anteriores, en donde su criterio de paro es exacto.

Tabla 4.5. Resultados del Ejemplo 1

VARIABLES	EQUIVALENTE DETERMINISTA (8 ESCENARIOS)	BENDERS (8 ESCENARIOS)	L-SHAPED (8 ESCENARIOS)	SD INCUMBENTE (128 MUESTRAS, MONTECARLO)
VNP	-1.02322E+8	-1.023E+8	-1.02321E+8	-1.01099E+8
Q	Q ₁ =5000 Q ₂ =22904.412	Q ₁ =5000 Q ₂ =22904.412	Q ₁ =5000 Q ₂ =22904.411	Q ₁ =5000.02 Q ₂ =22904.4
QE	QE ₁ =5000 QE ₂ =12904.412	QE ₁ =5000 QE ₂ =12904.412	QE ₁ =5000 QE ₂ =12904.411	QE ₁ =5000.02 QE ₂ =12904.4
y	y ₁ =1 y ₂ =1	y ₁ =1 y ₂ =1	y ₁ =1 y ₂ =1	y ₁ =1 y ₂ =1

4.4.2 Ejemplo 2

Este ejemplo es una modificación del problema resuelto por Acevedo y Pistikopoulos (1998) que tiene capacidades pre-existentes. La configuración de la red se muestra en la Figura 4.5, la cual tiene ocho procesos que pueden ser usados para producir los productos P1 y P2 de tres materias primas A, B y C. El objetivo es encontrar cuáles procesos son más recomendables para usar, dado que ambas disponibilidades de materias primas A y C son muy limitadas e inciertas, así como también la demanda para ambos productos. Se considera que la materia prima B está siempre disponible.

El horizonte de planeación considerado es de treinta años y el objetivo es encontrar si la inversión en esta red es rentable. Los demás datos para este ejemplo se pueden encontrar de las Tablas 4.6 a la 4.9.

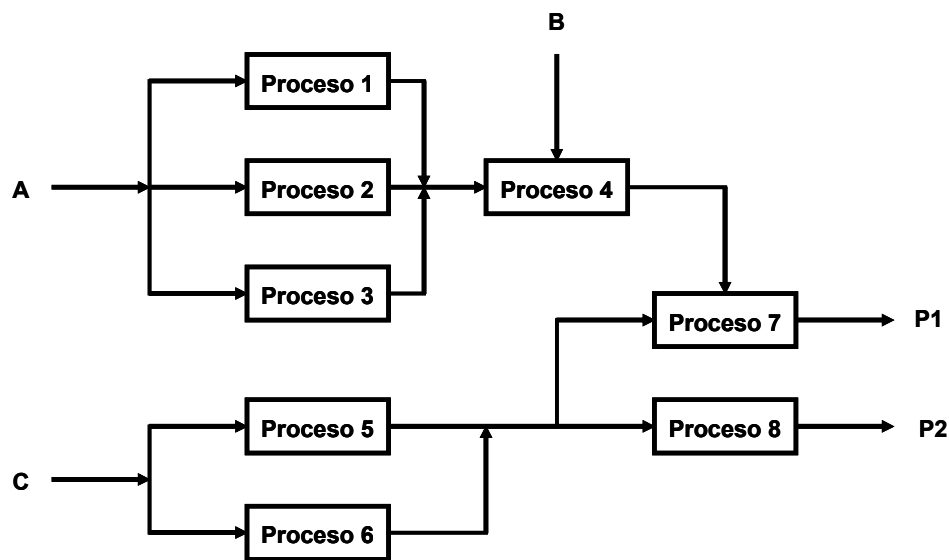


Figura 4.5. Red para el Ejemplo 2

Tabla 4.6. Disponibilidad y demanda para el Ejemplo 2

Demanda	P1	P2	A	C
Baja	19.8	16.8	21.25	27.7
Alta	24.2	20.57	25.7	33.9

Tabla 4.7. Datos de costos para el Ejemplo 2

Proceso	α en \$/ton	β en \$	δ en \$/ton
1	2.72	110	5
2	1.39	73	15
3	1.76	90	5
4	2.0	85	15
5	1.45	79	10
6	1.38	73	8
7	1.76	79	10
8	2.0	73	10
Químico		Γ en \$/ton	
A		200	
B		350	
C		300	
Químico		γ en \$/ton	
P1		900	
P2		850	

Tabla 4.8. Datos para los coeficientes de reacción y capacidades existentes del Ejemplo 2.

PROCESO	μ	$Q_{i,0}$
1	0.8	30
2	0.75	20
3	0.7	25
4	0.9	40
5	0.9	20
6	0.8	20
7	0.85	30
8	0.9	20

Tabla 4.9. Datos para la inversión, para las capacidades de expansión, y para el VNP del Ejemplo 2.

Máxima capacidad a invertir	\$1550
Capacidad de expansión mínima	10 ton
Capacidad de expansión máxima	80 ton
Tiempo de vida del proyecto	30 años
Tasa de interés	0.10
Tasa de impuestos	0.45
Factor del capital de trabajo	0.15
Factor del valor de salvamento	0.10

Primero, el modelo fue resuelto considerando el peor escenario posible, baja demanda para todos los productos y baja disponibilidad para toda la materia prima, para verificar si es rentable bajo estas circunstancias.

Resolviendo para el peor escenario posible, se tiene un VNP de \$ -91 559.864, lo cual indica que el problema es rentable aún en estas circunstancias extremas. Los resultados de la convergencia del algoritmo incumbente para diferentes técnicas de muestreo se presentan en la Figura 4.6. Estos resultados se obtuvieron para 400 muestras y con un error del límite superior menor o igual a 1×10^{-8} .

De la Figura 4.6 se observa que con la técnica de muestreo Montecarlo se obtiene una rápida convergencia, necesitándose de tan sólo 139 iteraciones. Mientras que las técnicas Latin Hypercube, Latin Hypercube Media y la secuencia de Hammersley no logran una convergencia para este número de muestras. Nótese otra vez que con la técnica de

muestreo de la secuencia de Hammersley no se presenta un comportamiento asintótico, sino ascendente, no logrando la convergencia ni para un número de muestras grande.

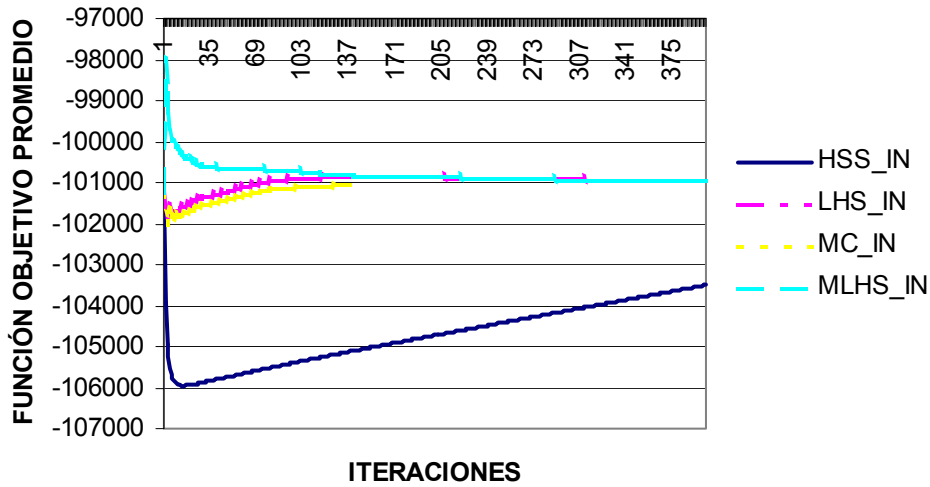


Figura 4.6. Convergencia del Ejemplo 2 para diferentes técnicas de muestreo.

En la Tabla 4.10 se presentan los resultados que se obtuvieron por diferentes métodos de solución. De esta tabla se puede observar que sólo los métodos Equivalente Determinista y SD predicen la misma configuración a utilizar, aunque el valor de una de sus expansiones difiere un poco. Note así que el valor del VNP es diferente en todos los métodos de solución, pero dentro del mismo orden de magnitud. Esto puede deberse a que conforme aumenta el tamaño y complejidad del problema, el espacio de búsqueda aumenta, así como el número de estructuras factibles que satisfacen las condiciones del modelo. Nótese que aunque pudiera realizarse una menor inversión en el costo de expansiones (caso del método L-shaped), los costos de operación y procesamiento esperados pueden aumentar de una forma tal que compensa el ahorro en la inversión de las expansiones, caso contrario a las estructuras obtenidas por los otros métodos de solución.

Tabla 4.10. Resultados del Ejemplo 2.

VARIABLES	EQUIVALENTE DETERMINISTA (16 ESCENARIOS)	BENDERS (16 ESCENARIOS)	L-SHAPED (16 ESCENARIOS)	SD INCUMBENTE (139 MUESTRAS, MONTECARLO)
VNP	-100 891.605	-97 507.33	-98 716.731	-101 061
Q	Q ₁ =30 Q ₂ =20 Q ₃ =25 Q ₄ =40 Q ₅ =33.88 Q ₆ =20 Q ₇ =30 Q ₈ =30	Q ₁ =40 Q ₂ =20 Q ₃ =25 Q ₄ =40 Q ₅ =33.88 Q ₆ =20 Q ₇ =30 Q ₈ =20	Q ₁ =30 Q ₂ =20 Q ₃ =25 Q ₄ =40 Q ₅ =33.905 Q ₆ =20 Q ₇ =30 Q ₈ =20	Q ₁ =30 Q ₂ =20 Q ₃ =25 Q ₄ =40 Q ₅ =30.8 Q ₆ =20 Q ₇ =30 Q ₈ =30
QE	QE ₅ =13.88 QE ₈ =10	QE ₁ =10 QE ₅ =13.88	QE ₅ =13.905	QE ₅ =10.8 QE ₈ =10
y	y ₅ =1 y ₈ =1	y ₁ =1 y ₅ =1	y ₁ =1	y ₅ =1 y ₈ =1

Además, se puede observar que el valor estimado del VNP por el algoritmo SD presenta una diferencia que va del 0.16% hasta el 3.5 % con respecto a los valores obtenidos por los otros métodos de solución, lo que puede deberse al número de escenarios considerados, por una técnica de muestreo y al criterio de paro estadístico.

En ese sentido, en las Tablas 4.11 y 4.12 se presenta un análisis de cómo se ve afectado el VNP de los Ejemplos 1 y 2 cuando se incrementa el número de escenarios y se disminuye el error del límite respectivamente. De los resultados presentados en estas tablas se puede inferir el comportamiento asintótico del VNP y la tendencia hacia una mejora de la solución cuando se aumenta el número de escenarios o se disminuye el error del límite superior. Para el caso del Ejemplo 1 se aprecia que la tendencia de la solución es hacia el valor obtenido por el equivalente determinista, no así para el caso del ejemplo 2. Sin embargo, recuerde que para este ejemplo, aunque los métodos de solución equivalente de determinista y SD predicen la misma configuración, no predicen la misma cantidad a expandir.

Tabla 4.11. Comportamiento del VNP cuando se incrementa el número de escenarios.

	Solución Equivalente Determinista	Solución SD (convergencia)	Solución SD (300 muestras)	Solución SD (500 muestras)
Ejemplo 1	-1.02322E+8	-1.01099E+8 (128 muestras)	-1.019277E+8	-1.022327E+8
Ejemplo 2	-100 891.605	-101 061.00 (139 muestras)	-101 235.162	-101 239.1396

Tabla 4.12. Comportamiento del VNP cuando se disminuye el error del límite.

	Solución Equivalente Determinista	Solución SD (error del límite 1)	Solución SD (error del límite 2)
Ejemplo 1	-1.02322 E+8	-1.01099E+8 (1×10^{-6})	-1.020487E+8 (1×10^{-7})
Ejemplo 2	-100 891.605	-101 061.00 (1×10^{-8})	-101 266.2008 (1×10^{-9})

Por último, se puede establecer que el algoritmo SD incumbente presenta una estabilidad cuando se aumenta el número de escenarios y variables a considerar en el problema. Además, la técnica de muestreo Monte Carlo fue la que presenta mejores propiedades de convergencia asintótica, pudiéndose conseguir con ello una reducción en el tiempo de cómputo cuando los problemas a resolver sean a gran escala.

En la siguiente sección se discuten las ventajas y desventajas de resolver los problemas ya sea por un método discreto o bien por un método que usa distribuciones de probabilidad continua.

4.5 Métodos discretos vs. continuos

Cuando se tiene un problema que involucra parámetros inciertos existen diversos métodos que se pueden usar para resolverlo, ya sea por un análisis de escenario o bien utilizando distribuciones de probabilidad continua. Aquí la pregunta sería ¿Cuál de las dos alternativas es la más recomendable? Las respuestas son diversas y las opiniones

encontradas. Es por eso que en esta sección se presenta un análisis del trabajo implicado para resolver los dos problemas de este capítulo.

En las Tabla 4.13 y 4.14 se presentan las características de los problemas que se resolvieron en el Ejemplo 1 y en el Ejemplo 2, en cuanto a tamaño y complejidad del modelo, así como del trabajo invertido para obtener su solución. Solamente se comparan entre sí el método L-shaped y el método SD incumbente.

Tabla 4.13. Características y esfuerzo computacional del Ejemplo 1.

ANÁLISIS	L-SHAPED	SD INCUMBENTE
Variables continuas	6	6
Variables binarias	3	3
Variables estocásticas	3	3
No. iteraciones	5	128
No. escenarios	8	300
Problemas resueltos	40	512

Tabla 4.14. Características y esfuerzo computacional del Ejemplo 2.

ANÁLISIS	L-SHAPED	SD INCUMBENTE
Variables continuas	16	16
Variables binarias	8	8
Variables estocásticas	4	4
No. iteraciones	9	139
No. escenarios	16	300
Problemas resueltos	144	556

A pesar de que no hay mucha diferencia entre el valor objetivo obtenido por ambos métodos de solución (ver Tablas 4.5 y 4.6); en cuanto a esfuerzo computacional se refiere, el realizado por el método SD incumbente es mayor en los dos ejemplos (ver Tablas 4.11 y 4.12). Sin embargo, observe que conforme se aumenta la complejidad del problema (variables de proceso y parámetros estocásticos), la cantidad de problemas a resolver por el método L-shaped aumenta significativamente, es decir, se resuelven 104 problemas más en

el Ejemplo 2 que en el Ejemplo 1, mientras que en el método SD éste incremento es de sólo 42 problemas a resolver.

En el siguiente capítulo se resolverán problemas en los que se aumenta el número de variables y escenarios a considerar (problemas de gran escala) con una aplicación en redes de agua municipales. Se mostrará de esta manera la diversidad de aplicaciones que el algoritmo SD incumbente puede tener así como su robustez.

Capítulo 5

Redes de Agua Municipales: Un Problema Estocástico de Dos Etapas para la Colocación de Sensores Bajo Incertidumbre

En este capítulo el problema de seguridad en redes municipales de agua se formula como un problema estocástico de dos etapas que involucra incertidumbres en la densidad de población y en las probabilidades del riesgo de ataque. Por la naturaleza del modelo, se presentan dos ejemplos reportados en la literatura que involucran un número grande de variables de proceso y de parámetros inciertos. Al final se realiza un análisis de resultados que trata de resaltar las ventajas del método SD con respecto a otros métodos de solución estocástica.

5.1 Introducción

Recientemente, Berry y col. (2003) presentaron una formulación de programación entera determinista para la colocación óptima de sensores en redes de agua municipales. La formulación minimiza la fracción de la población que está en riesgo de ataque por un contaminante. El ataque lo modelan como la inyección de un volumen grande de contaminante dañino en un nodo de la red, por lo cual es difícil conocer a priori donde ocurrirá el ataque.

La solución considera diversos escenarios expuestos a un posible ataque. Los patrones de flujo de agua los obtienen con el simulador de agua EPANET 2.0 (Rossman, 1999), para lo cual consideran que existe por lo menos una fuente de abastecimiento de agua y que su demanda se mantiene por un tiempo largo. Para obtener una solución robusta consideran factores de ruido en la fase de riesgo e incertidumbres en los escenarios de ataque. Además, en el modelo se suponen diseños de demanda fijos, pero no se supone por cuanto tiempo se mantiene fijo el diseño, que tan a menudo aparece, o el orden en que estos aparecen. En la dirección del flujo, cada nodo toma en cuenta el número de gente que potencialmente consumiría agua en ese punto y correlacionan el diseño de flujo con el periodo de tiempo por día que demanda esta gente (por ejemplo, se representa gente en el trabajo durante el día y gente en la casa durante la noche). Note, sin embargo, que el número de la población no necesariamente es proporcional a la demanda, es decir, una zona industrial podría consumir demasiada agua aunque poca gente estuviera sobre esa zona. La formulación supone probabilidades constantes en los riesgos de ataque y en las densidades de población, convirtiendo el problema en un programa entero determinista.

Los resultados muestran que la configuración óptima es sensible a las perturbaciones en la probabilidad del riesgo de ataque y en la densidad de población como

una función del número fijo de sensores usados en la formulación. Sin embargo, esta formulación no incluye el costo y tipo de sensores, lo cual puede ser un factor mayor que cambie la solución.

Por otro lado, la programación estocástica ha proporcionado algoritmos que tratan a los parámetros inciertos de una forma fundamental para obtener una solución aproximada a problemas de planeación y calendarización. Sin embargo, el gasto computacional y la complejidad del problema aún persisten cuando se aumenta el número de variables, el número de parámetros inciertos o el número de escenarios a considerar.

En este trabajo se extiende el modelo de Berry y col. (2003) para que incluya explícitamente las incertidumbres en el riesgo de ataque y en la densidad de población, así como la consideración del costo en los sensores, cambiando de esta forma el problema determinista por un problema bajo incertidumbre. En esta formulación se incluirá el número de sensores como variables de decisión en la primera etapa en un problema estocástico de dos etapas y en donde los costos de los sensores se incluyen en la función objetivo. Además, por las dimensiones del modelo, se espera que el algoritmo SD incumbente que se propone sea computacionalmente robusto para resolver problemas con un gran número de variables, parámetros inciertos o escenarios a considerar.

Para exponer una idea clara del trabajo de investigación, el presente capítulo se ha organizado de la siguiente manera. En la Sección 5.2 se plantea el problema de redes de agua municipales y su formulación como un modelo estocástico mixto-entero lineal en la primera etapa de un problema de dos etapas con recurso. Enseguida, en la Sección 5.3, se presenta una discusión de las ventajas de resolver un problema estocástico de su contraparte equivalente determinista. Por último, los resultados de dos problemas reportados en la literatura se presentan en la Sección 5.4.

5.2 Redes municipales de agua: planteamiento del problema

El modelo matemático que aquí se presenta es una extensión del modelo que considera Berry y col. (2003). El problema consiste en colocar el número óptimo de sensores sobre la tubería de la red, de una forma tal que se espera minimizar el porcentaje de población vulnerable en un escenario de ataque consistente en inyección de un contaminante. Una persona es vulnerable si hay una vía directa de contaminación de un punto de ataque a otro mientras no exista un sensor sobre la tubería que detecte la propagación y favorezca su eliminación.

En la modelación de la red de agua se requiere de un diagrama $G = (V, E)$. En donde E es un arreglo de contornos que representan tuberías y V es un arreglo de vértices o nodos que representan los puntos en donde las tuberías se encuentran. Los vértices pueden representar fuentes que introducen el agua, tales como depósitos o tanques; también pueden representar puntos de agotamiento (o de demanda) de agua que se consume. En general la red se representa en una escala, donde los nodos representan colonias o regiones de una ciudad. La Figura 5.1 muestra una configuración típica de una red municipal de agua. Los análisis se desarrollan bajo un número de patrones de flujo, donde se conoce la dirección del flujo en cada tubería. Los vértices v_i y v_j conectan cada tubería y se denota por (v_i, v_j) . La mayoría de las redes son topológicamente estructuradas y por cuestiones prácticas se suponen planas.

Se considerará bajo riesgo sólo un número fijo de nodos dentro del patrón de flujo, el cual requiere sólo de su dirección en cada tubería. De esta manera, si el flujo se especifica para la tubería que va del nodo v_i al nodo v_j , entonces el flujo que iría de v_j a v_i es cero. Así, el modelo que se genera requiere de los siguientes datos de entrada:

- La red $G = (V, E)$. $V = v_1, \dots, v_n$ y $E = e_1, \dots, e_m$.

- La probabilidad de un ataque u_{ip} al nodo v_i durante el diagrama de flujo p , de lo cual se tiene que $\sum_{v_i \in V, p \in 1, \dots, P} u_{ip} = 1$, donde P es el numero de patrones de flujo.

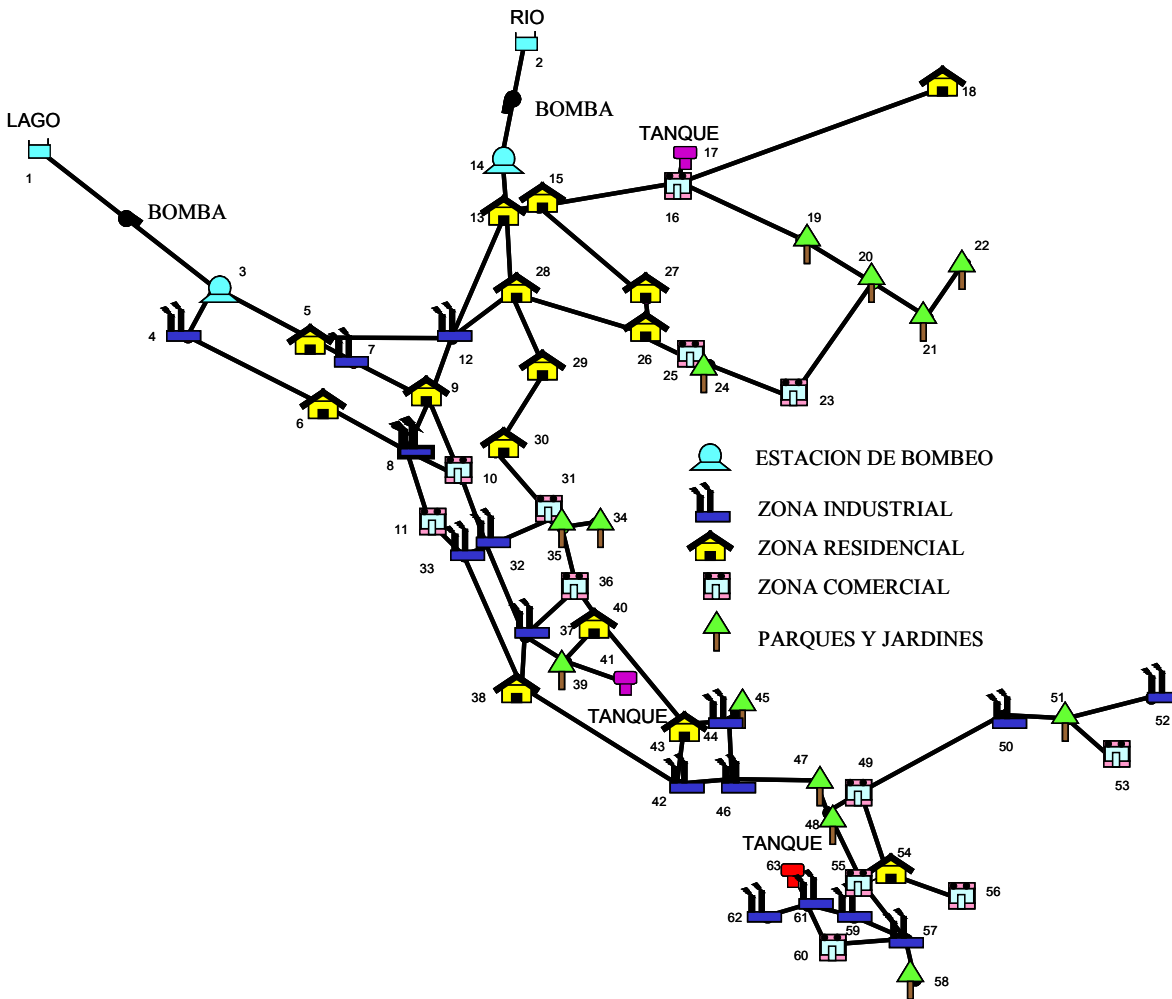


Figura 5.1 Ejemplo de una red de agua municipal.

- La densidad de población (cantidad de gente) δ_{ip} en el nodo v_i , mientras el flujo p es activo. Si el nodo v_i no demanda agua durante el diagrama de flujo p , $\delta_{ip} = 0$.

Se pueden replicar los patrones de flujos y asociar múltiples densidades de población con ellos, por ejemplo cuando el patrón de flujo se asocia con periodos de tiempo y estos aparecen varias veces en un día.

- Los parámetros que describen el patrón de flujo p se denotan por $f_{ijp} \equiv f_{cp} \in \{0,1\}$. Si el flujo es positivo en la dirección $e = (v_i, v_j)$, $f_{ijp} = 1$ durante el patrón de flujo p y 0 en caso contrario.

Además, $f_{ijp} f_{jip} = 0$ porque el agua no fluye en ambas direcciones de una tubería.

- El número máximo de sensores que se pueden colocar, X_{max} .
- Los costos de los sensores, C_{ij} , y el costo que implicaría si una persona es afectada por el contaminante, S .

Si ocurre un ataque sobre el nodo v_i durante el patrón de flujo p , un nodo $v_j \neq v_i$ se contamina si hay una vía de v_i a v_j que no tiene sensor y que tiene direcciones de flujo positivas en el patrón de flujo p . Más específicamente, v_j se contamina si existe una vía $v_i \equiv v_1, v_2, \dots, v_j \equiv v_l$ tal que $(v_k, v_{k+1}) \in E$ y $f_{k,(k+1),p} = 1$ para toda $k = 1, \dots, l-1$, y si además no existe un sensor sobre la tubería en la vía de contaminación. Si un nodo de demanda v_j se contamina durante el patrón de flujo p , entonces se dice que toda la gente del nodo v_j durante el tiempo p está expuesta; se desea minimizar la cantidad de gente expuesta a la contaminación. La Figura 5.2 ilustra la forma de propagación de la emisión de un contaminante.

En la siguiente sección se presenta la formulación matemática del problema aquí planteado.

5.2.1 Formulación como un problema estocástico entero de dos etapas con recurso fijo

Aunque los fundamentos del modelo propuesto son los mismos que los propuestos por Berry y col. (2003), la principal contribución es la reformulación del problema como uno entero en la primera etapa, de un problema estocástico mixto-entero lineal de dos etapas. Al igual que en el caso de las cadenas de suministro, esto es significativo por dos razones: *i*) primero, las incertidumbres son formalmente incorporadas dentro del modelo, y *ii*) el algoritmo SD incumbente se puede utilizar para proporcionar una solución aproximada al problema.

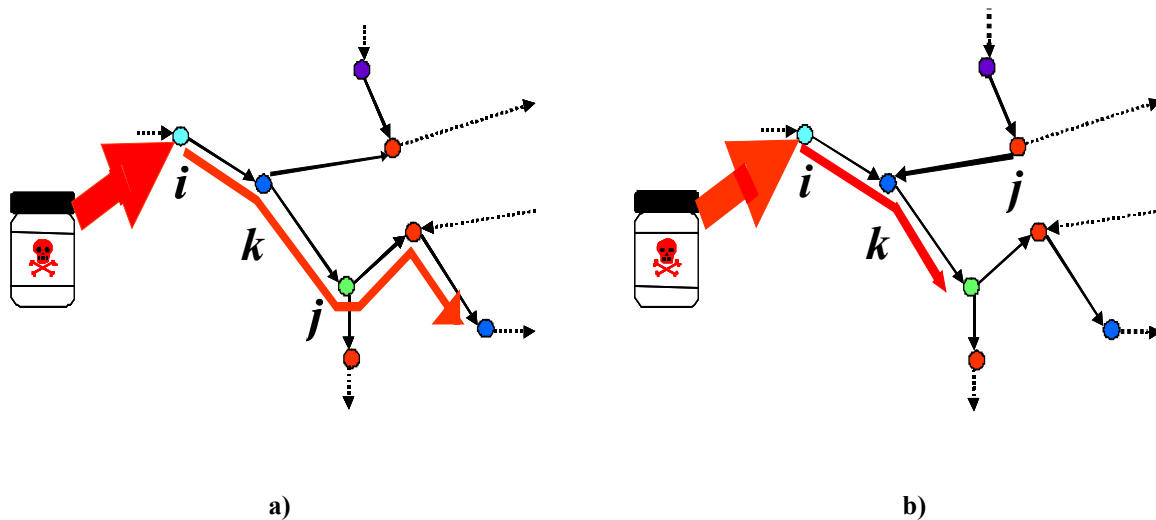


Figura 5.2. Modelo de la propagación de un contaminante inyectado. a) Cuando la propagación ocurre de un nodo *k* a un nodo *j*. b) Cuando la propagación no ocurre de un nodo *k* a un nodo *j*.

En este trabajo se incluyen las incertidumbres en el riesgo de ataque y en la densidad de población. Se considera que una función de distribución de probabilidad continua se asigna a cada uno de estos parámetros inciertos. Ésta consideración hace que el

problema se convierta en un programa estocástico. Por lo tanto, se propone la separación de las restricciones del modelo en dos etapas, siguiendo el concepto de recurso.

En la primera etapa se toman decisiones respecto de en donde se van a colocar los sensores a través de las variables binarias x_{ij} . Por lo tanto, los costos de los sensores (y la colocación de los sensores), C_{ij} , se incluyen en la función objetivo de la primera etapa. De aquí que, por incluir tales costos, la solución óptima podría no utilizar el número total de sensores, por lo que el número de sensores en la configuración es también una variable de optimización.

Las variables derivadas (las cuales representan la propagación de los contaminantes), y_{ipj} , se consideran como decisiones de segunda etapa. La función de recurso (el valor esperado de la función objetivo de la segunda etapa) es casi idéntica a la función objetivo del modelo MIP original. En ella se incluye el cálculo de la población en riesgo de sufrir un ataque. Sin embargo, un factor S se ha incluido en el cálculo. S es un costo asociado por cada miembro de la población que ha sido afectado por el ataque. Se podrá interpretar también como un factor de penalización en la función objetivo por afectación. Por lo tanto, la primera etapa intenta minimizar el costo de los sensores, así como también el valor esperado (E) del costo asociado con la población que ha sufrido el ataque (calculado a través del objetivo de la segunda etapa); note que ambos términos se calculan en una misma base monetaria. No obstante, como se mencionó C_{ij} y S se podrían interpretar como factores de peso (o de escala) desde un punto de vista numérico.

En resumen las aproximaciones de las dos etapas trabajan de la siguiente manera: *i*) la primera etapa decide el número y la localización de los sensores, *ii*) con las decisiones de la primera etapa, para algunos valores de muestreo de los parámetros inciertos, la segunda etapa evalúa la población en riesgo para ese escenario, *iii*) El cálculo de la función objetivo

de la segunda etapa se usa (a través de un corte de optimalidad) para modificar las decisiones de la primera etapa en la próxima iteración; tal es precisamente la idea del recurso.

La Ecuación 5.1 y la Ecuación 5.2 muestran las restricciones de las dos etapas. Las restricciones de la primera etapa limitan el número de sensores (menor o igual al máximo) que serán colocados en la red. Las restricciones de la segunda etapa propagan la contaminación, siempre y cuando no exista un sensor que lo prevenga, y evalúa la población en riesgo de sufrir el ataque. Note que la primera etapa es un problema de programación entera.

Primera etapa (entero):

$$\begin{aligned}
 & \text{Min} \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} + E_u [Q(x, u)] \\
 & \text{s. a} \\
 & x_{ij} - x_{ji} = 0 \quad \forall \quad i = 1, \dots, n-1 \quad i < j \\
 & \sum_{(i,j) \in E} x_{ij} - X_{\max} \leq 0 \\
 & x_{ij} \in \{0,1\} \quad \forall \quad i, j \in E
 \end{aligned} \tag{5.1}$$

Segunda etapa:

$$\begin{aligned}
 & \text{donde} \\
 & Q(x, u) = \min \sum_{i=1}^n \sum_{p=1}^P \sum_{j=1}^n S y_{ipj} \omega_{ip} \delta_{jp} \\
 & \text{s. a} \\
 & y_{ipi} = 1 \quad \forall \quad i = 1, \dots, n \\
 & \quad \quad \quad \forall \quad p = 1, \dots, P \\
 & y_{ipk} - y_{ipj} \leq x_{kj} \quad \forall \quad (k, j) \in E \quad \text{s.t.} \quad f_{kjp} = 1
 \end{aligned} \tag{5.2}$$

Observe que, por la naturaleza de este modelo, conforme se aumente el número de nodos y de tuberías en la representación de la red de agua, las dimensiones del problema aumentan significativamente con respecto al número de variables y parámetros inciertos en

el modelo. Para dar una idea más clara de esto que se está hablando, en la siguiente sección se presenta una discusión de la cantidad de problemas que se estarían resolviendo si se llegaran a utilizar algunos de los métodos estocásticos que se basan para su solución en un análisis de escenarios.

5.3 Modelos estocásticos de dos etapas y equivalentes deterministas

Dada la estructura del problema anterior, el modelo consiste de un problema estocástico mixto-entero lineal con recurso (SMILPwR). La solución a este tipo de problemas se trata aquí como una extensión a la solución del problema lineal descrita en el Capítulo 3, por las razones que se exponen a continuación. La clase principal de problemas SMILPwR involucra dos etapas y aquí se discutirá el caso cuando es entero en la primera de ellas.

Primera etapa:

$$\begin{aligned} \text{Min } & c^T x + Q(x) \\ \text{s. a } & Ax = b \\ & x \in \{0,1\} \end{aligned} \quad (5.3)$$

Segunda etapa:

$$\begin{aligned} Q(x,u) = \text{Min } & q^T(u)y \\ \text{s. a } & W(u)y = h(u) - T(u)x \\ & y \geq 0 \end{aligned} \quad (5.4)$$

En la primera etapa se elige la variable entera de decisión, x . En la segunda etapa, siguiendo la observación de los valores de los parámetros inciertos, u , y la evaluación de la función objetivo, se incorpora una acción correctiva, representada por las decisiones de la segunda etapa, y . A es una matriz de coeficientes y b y c son vectores de coeficientes, $Q(x)$

es la función de recurso definida por $Q(x) = E_u [Q(x, u)]$. E_u representa el operador esperanza matemática, $Q(x, u)$ es obtenido de la segunda etapa del problema, q es un vector de coeficientes y W , h y T son matrices de coeficientes que en principio pueden depender de las variables estocásticas u .

Se considera ahora que el vector de las variables estocásticas u tiene un soporte finito. Sea $k = 1, \dots, K$ los índices de sus posibles realizaciones y p_k sus probabilidades. Bajo estas suposiciones, se puede escribir ahora un programa equivalente determinista en la forma extensiva. Esta forma se crea por la unión de un arreglo de decisiones de segunda etapa, es decir, y_k , para cada realización de u , y obviamente, para cada realización de q_k , h_k , y T_k . Este es un problema mixto-entero lineal a gran escala que se define como la forma extensiva (EF):

$$\begin{aligned}
 (EF) \quad \min \quad & c^T x + \sum_{k=1}^K p_k q_k^T y_k \\
 \text{s. a} \quad & Ax = b \\
 & T_k x + W y_k = h_k \quad k = 1, \dots, K \\
 & x \in \{0,1\}, \quad y_k \geq 0, \quad k = 1, \dots, K
 \end{aligned} \tag{5.5}$$

Note que el problema (EF) es una representación exacta del problema dado por las Ecuaciones 5.3 y 5.4 para un tamaño de muestras finito. En general, su desventaja es que su tamaño crece exponencialmente, y por consiguiente puede llegar a ser demasiado grande para ser resuelto explícitamente. Como un ejemplo, considere un problema con 20 parámetros inciertos y con 6 valores discretos por cada uno; el número de eventos posibles o escenarios que podrían surgir sería igual a $6^{20} \cong 3.7 \times 10^{15}$, requiriéndose de igual cantidad de problemas a resolver (Clay y Grossmann, 1997).

De este modo, si se pretendiera resolver el problema dado por las Ecuaciones 5.1 y 5.2, el tamaño y complejidad combinatorial incrementaría con el número de escenarios, con

el número de variables estocásticas y con el número de diagramas por cada periodo de tiempo. Como resultado, las técnicas disponibles de espacio completo pueden fracasar al resolver la optimalidad de estos problemas al grado de llegar a ser intratables (Iyer y Grossmann, 1998). Motivados por el problema de manejar un gran número de variables y un gran número de parámetros estocásticos, así como por la disponibilidad que se tiene de usar una distribución de probabilidad continua en lugar de un análisis de escenario, se propone aplicar el algoritmo SD incumbente implementado en este trabajo para resolver también problemas SMILPwR que son enteros en la primera etapa.

A continuación, se resolverán dos ejemplos numéricos que muestran la robustez del algoritmo SD incumbente, no obstante la gran cantidad de variables y parámetros inciertos involucrados en el modelo.

5.4 Ejemplos y resultados

Los ejemplos tienen el mismo modelo y solamente varían en estructura y tamaño. El objetivo principal consiste en determinar la colocación óptima de un número limitado de sensores en una red de agua municipal, de forma tal que se minimice el riesgo de que la población pueda resultar afectada. Se consideran costos fijos para los sensores, teniendo en cuenta las incertidumbres en la probabilidad de ataque.

El Ejemplo 1 es un problema con 36 nodos y 40 tuberías, lo que se traduce en 40 parámetros inciertos y 40 variables binarias, sin contar las variables derivadas por la propagación del ataque. El Ejemplo 2 cuenta con 63 nodos y 94 tuberías, lo que equivale a tener 68 parámetros inciertos y 94 variables binarias, también sin contar las variables derivadas por la propagación del ataque. Como no es posible, bajo estas circunstancias, resolver por algún otro método estocástico, aquí sólo se comparará la solución obtenida con

la que se obtiene resolviendo un problema determinista, es decir, con un solo escenario (usando los valores medios de los parámetros inciertos).

Se utilizará el algoritmo SD incumbente con la técnica de muestreo Monte Carlo, ya que, como se ha establecido en el Capítulo 3, es la más conveniente a usar por este método de solución.

Las características específicas de cada ejemplo son descritas en su respectiva sección. Para la simulación numérica se ha utilizado una hp workstation xw6000 y los modelos generados LP y MILP han sido resueltos con CONOPT y BDMLP respectivamente, dentro de la interfase de modelación GAMS (Brooke y col., 1998).

5.4.1 Ejemplo 1

El caso de estudio a tratar corresponde al Ejemplo 1 que resolvieron Berry y col. (2003). La red municipal de agua se compone de 36 nodos, 40 tuberías, una estación de bombeo, una zona residencial, una zona comercial y una zona industrial. En la Figura 5.3 se muestra la red de agua municipal, y en la Tabla 5.1 se presenta la distribución de los nodos para cada una de las zonas.

Se considera que la zona industrial y residencial es susceptible de sufrir un ataque (dos escenarios) y que se cuenta con 7 sensores como máximo para prevenirlo. Se consideran dos escenarios con posibilidad de ser atacados. Los nodos que pertenecen a la zona considerada en riesgo tienen uniformemente una probabilidad alta de ser atacados, mientras que los nodos de las zonas restantes tienen una probabilidad baja. La Tabla 5.2 muestra los datos de probabilidad para cada uno de los dos escenarios y para los cuales se considera una distribución normal.

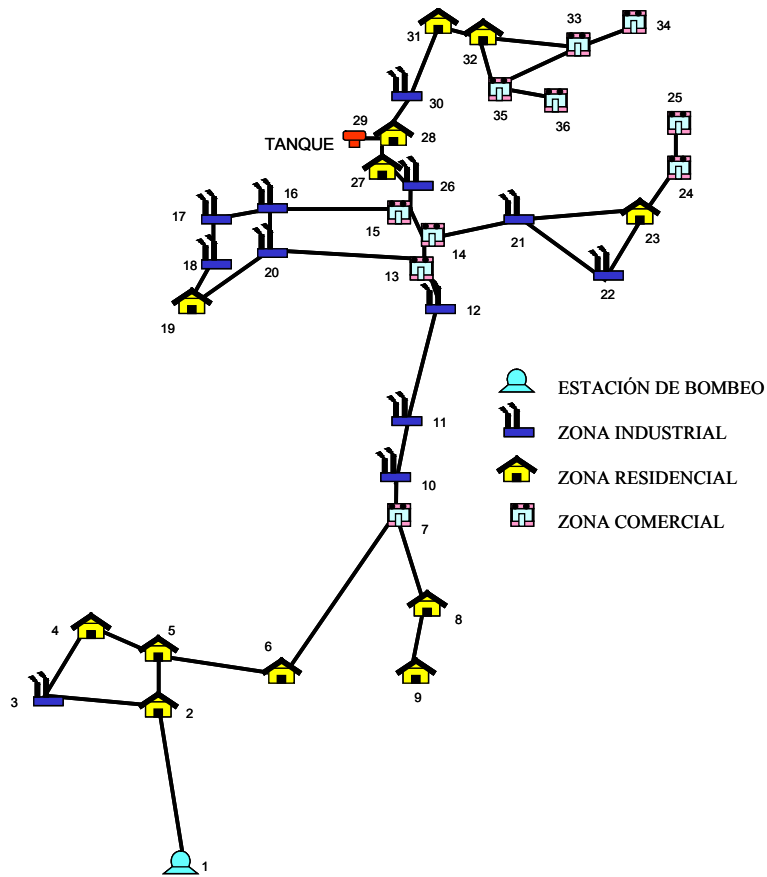


Figura 5.3. Red municipal de agua para el Ejemplo 1.

Tabla 5.1. Distribución de nodos para el Ejemplo 1.

ZONA	NODOS
ESTACIÓN DE BOMBEO	1
INDUSTRIAL	3, 10, 11, 12, 16, 17, 18, 20, 21, 22, 26, 30
RESIDENCIAL	2, 4, 5, 6, 8, 9, 19, 23, 27, 28, 31, 32
COMERCIAL	7, 13, 14, 15, 24, 25, 29, 33, 34, 35, 36

Tabla 5.2. Probabilidad de ataque (%) usada para cada zona y escenario

ESCENARIO	ESTACIÓN DE BOMBEO	ZONA INDUSTRIAL	ZONA RESIDENCIAL	ZONA COMERCIAL
1	1	76	12	11
2	1	12	76	11

La densidad de población en cada nodo es igual a 500 con un $\pm 25\%$ de población en tránsito y se representa con una distribución de probabilidad normal. Las densidades de población y las probabilidades de ataque posteriormente se normalizan para asegurar que el tamaño de la población total no cambie (18 000) y que la suma total de las probabilidades sea igual a uno. Por ejemplo, la información de la Tabla 5.2 implica que, para el escenario 2, la probabilidad de ataque de todos los nodos de la zona residencial es la misma para cada una de ellas, y su suma es aproximadamente 0.76 (0.76 también podría ser la media de la distribución de probabilidad normal para esta zona).

Se proponen dos tipos de sensores, uno de bajo costo (15 000 000) y otro de alto costo (45 000 000). El costo para cada sensor permanece fijo en la optimización y la decisión de colocarlo estará en función del costo que se tendría que pagar por indemnización individual (30 000) en caso de que la población resultara afectada.

Finalmente, para la red municipal de agua se han creado cuatro patrones de dirección del flujo, cada uno representando un periodo de 6 horas para una corrida de un día en el simulador EPANET 2.0 (Rossman, 1999). En la Figura 5.4 se presenta un patrón de flujo para el primer periodo de operación; los otros tres patrones de flujo se presentan en las Figuras B.1, B.2 y B.3 del Apéndice B.

El problema se resolvió utilizando 300 muestras obtenidas por Monte Carlo y se usaron para cada una de las zonas consideradas en riesgo y para cada uno de los costos de los sensores hasta lograr alcanzar un error del límite superior menor o igual 1×10^{-6} . Los resultados se resumen en la Figura 5.5. Se requirió de 276 iteraciones para la zona industrial con un costo de sensor alto (IND_ALTO) y de 118 iteraciones para la zona industrial con costo de sensor bajo (IND_BAJO). Similarmente, se requieren de 132

iteraciones para la zona residencial con un costo alto (RES_ALTO) y 111 iteraciones para la zona residencial con un costo de sensor bajo (RES_BAJO).

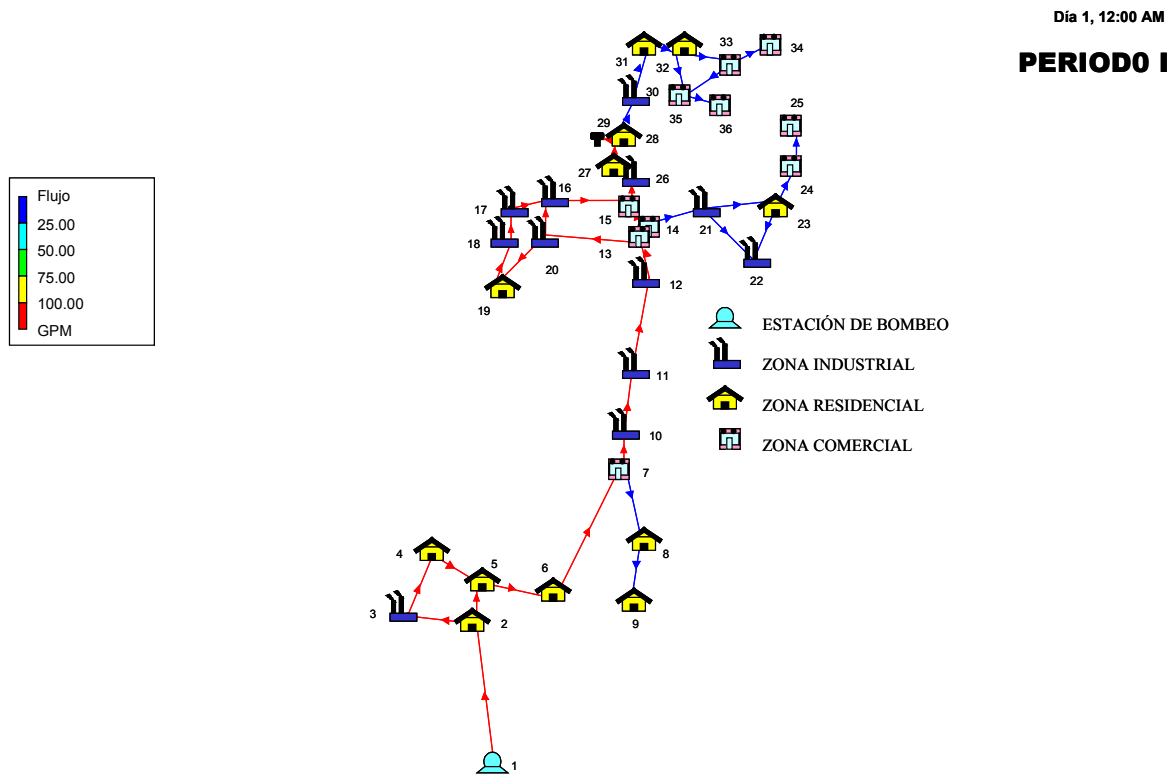


Figura 5.4. Patrón de la dirección del flujo para el periodo I del Ejemplo 1.

En las Tablas 5.3 y 5.4 se muestran los valores que se obtuvieron para la solución estocástica óptima en cada una de las zonas consideradas en riesgo, y en la Figura 5.6 se presenta la red con la configuración de sensores óptima cuando la zona considerada en riesgo es la industrial y el costo del sensor es bajo.

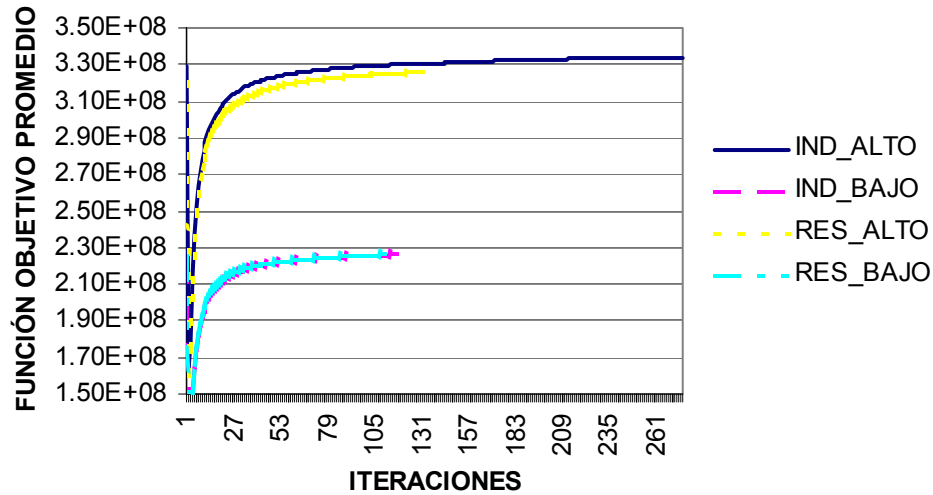


Figura 5.5. Resultados de la convergencia del Ejemplo 1.

Tabla 5. 3. Resultados del Ejemplo 1 cuando la zona industrial se considera en riesgo.

C. S.	Análisis	SOLUCIÓN ALGORITMO SD INCUMBENTE	SOLUCIÓN DETERMINISTA
Bajo	Sensores: % de población: Costo total:	$X_{5-6}, X_{12-13}, X_{21-23}, X_{26-27}, X_{30-31}$ 27.97 2.26088E+8	$X_{5-6}, X_{12-13}, X_{21-23}, X_{26-27}, X_{30-31}$ 27.75 2.2485E+8
Alto	Sensores: % de población: Costo total:	X_{12-13}, X_{26-27} 45.20 3.34112E+8	X_{12-13}, X_{26-27} 44.12 3.2825E+8

C.S. = Costo del sensor

Tabla 5.4. Resultados del Ejemplo 1 cuando la zona residencial se considera en riesgo.

C. S.	Análisis	SOLUCIÓN ALGORITMO SD INCUMBENTE	SOLUCIÓN DETERMINISTA
Bajo	Sensores: % de población: Costo total:	$X_{6-7}, X_{12-13}, X_{27-28}, X_{30-31}$ 30.74 2.26044E+8	$X_{6-7}, X_{12-13}, X_{27-28}, X_{30-31}$ 31.15 2.2585E+8
Alto	Sensores: % de población: Costo total:	X_{6-7}, X_{28-30} 43.67 3.25837E+8	X_{6-7}, X_{28-30} 42.61 3.2010E+8

C.S. = Costo del sensor

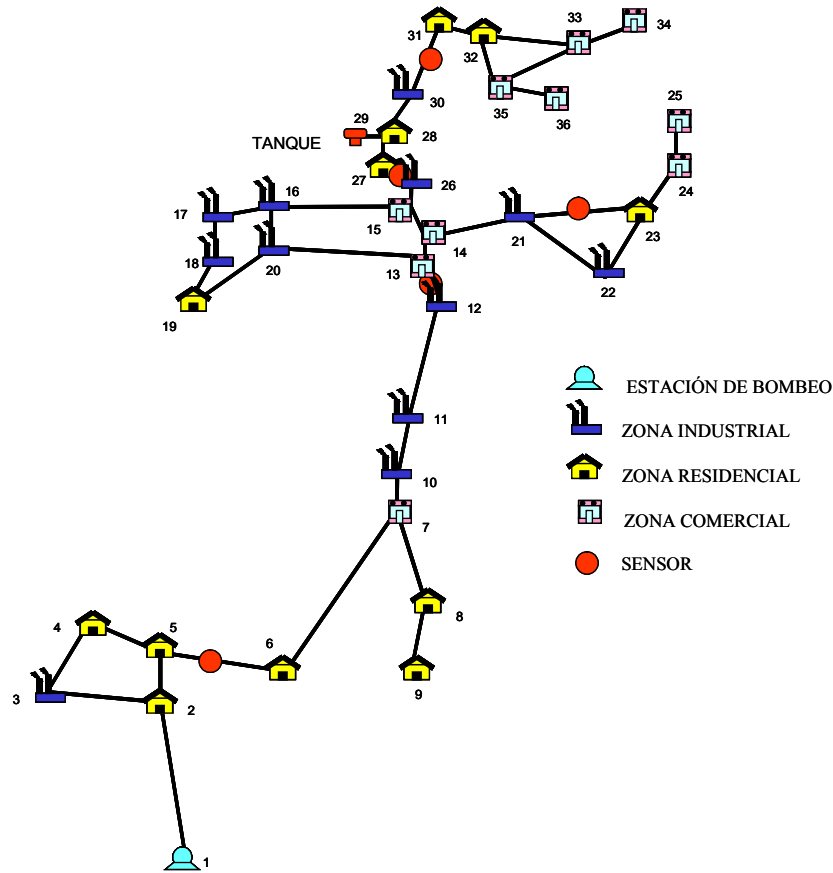


Figura 5.6. Configuración óptima del Ejemplo 1 cuando la zona industrial se considera en riesgo y el costo del sensor es bajo.

5.4.2 Ejemplo 2

El caso de estudio a tratar es una derivación del Ejemplo 2 que resolvieron Berry y col. (2003). La red municipal de agua modificada se compone de 63 nodos, 94 tuberías, una estación de bombeo, una zona industrial, una zona residencial, una zona comercial y una zona de parques y jardines. En la Figura 5.1 se ha mostrado esta red municipal de agua, y en la Tabla 5.5 se presenta la distribución de los nodos para cada una de las zonas.

La densidad de población para este problema es de 2000 habitantes por nodo con un $\pm 25\%$ de población en tránsito. La población total de la ciudad es de 126 000. Los costos

para los sensores son de 100 000 000 para el de bajo costo y de 350 000 000 para el costo alto. Los datos restantes corresponden a los del ejemplo anterior.

Tabla 5.5. Distribución de nodos para el Ejemplo 2.

ZONA	NODOS
ESTACIÓN DE BOMBEO	1,2,3,14
INDUSTRIAL	4,7,8,12,32,33,37,42,44,46,50,52,57,59,61,62,63
RESIDENCIAL	5,6,9,13,15,17,18,26,27,28,29,30,38,40,41,43,54
COMERCIAL	10,11,16,23,25,31,36,49,53,55,56,60
PARQUES Y JARDINES	19,20,21,22,24,34,35,39,45,47,48,51,58

El patrón de la dirección del flujo obtenidos por el simulador EPANET 2.0 (Rossman, 1999) para este problema, y para el primer periodo se presenta en la Figura 5.7; los otros tres patrones de flujo para los periodos II, III y IV se presentan en las Figuras C.1, C.2 y C.3 del Apéndice C.

El problema se resolvió utilizando 400 muestras para la zona industrial con un costo de sensor alto (IND_ALTO) y 300 muestras para las zonas restantes. Ambos arreglos de muestras fueron obtenidos por técnicas Monte Carlo y lograr tener un error del límite superior menor o igual a 1×10^{-5} . Los resultados se resumen en la Figura 5.8; se requirieron de 40 iteraciones para la zona industrial con costo de sensor bajo (IND_BAJO). Similarmente, se requieren de 66 iteraciones para la zona residencial con un costo alto (RES_ALTO) y 166 iteraciones para la zona residencial con un costo de sensor bajo (RES_BAJO).

En las Tablas 5.6 y 5.7 se muestran los valores que se obtuvieron para la solución estocástica óptima en cada una de las zonas consideradas en riesgo, y en la Figura 5.9 se presenta la red de agua con la configuración óptima de sensores cuando la zona considerada en riesgo es la residencial y el costo del sensor es bajo.

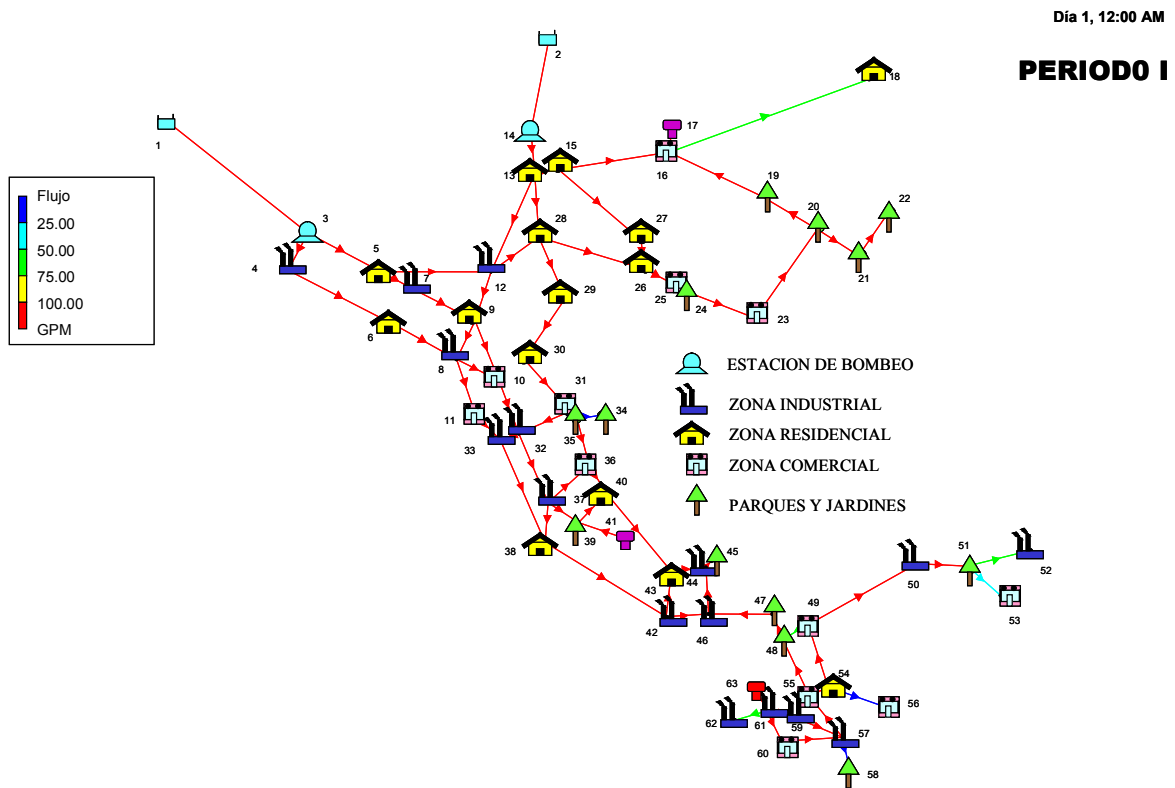


Figura 5.7. Patrón de la dirección del flujo para el periodo I del Ejemplo 2.

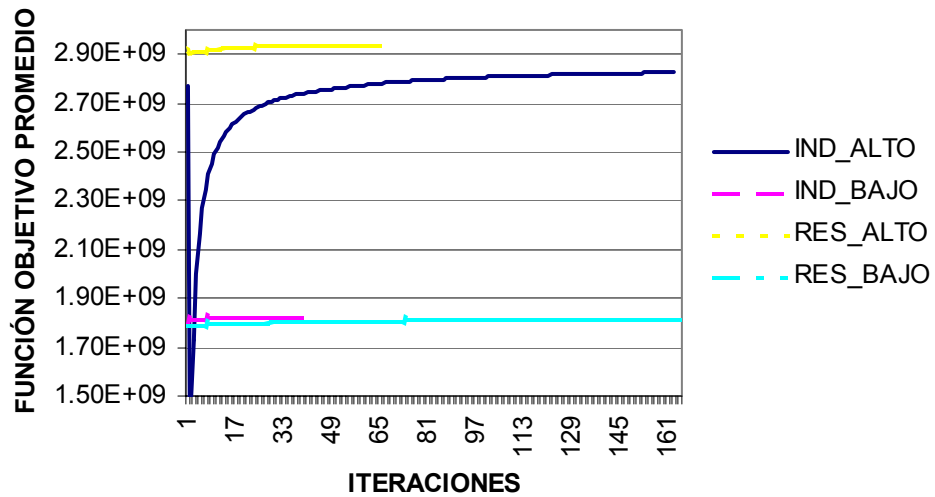


Figura 5.8. Resultados de la convergencia del Ejemplo 2.

Tabla 5.6. Resultados del Ejemplo 2 cuando la zona industrial se considera en riesgo.

C. S.	Análisis	SOLUCIÓN ESTOCÁSTICA	SOLUCIÓN DETERMINISTA
Bajo	Sensores:	$X_{14-13}, X_{26-28}, X_{30-31}, X_{32-37}, X_{33-38}, X_{47-48}$	$X_{14-13}, X_{26-28}, X_{30-31}, X_{32-37}, X_{33-38}, X_{47-48}$
	% de población: Costo total:	32.32 1.82184E+9	31.91 1.8063E+9
Alto	Sensores:	X_{32-37}, X_{38-42}	X_{32-37}, X_{38-42}
	% de población: Costo total:	56.24 2.82594E+9	54.69 2.7675E+9

C.S. = Costo del sensor

Tabla 5.7. Resultados del Ejemplo 2 cuando la zona residencial se considera en riesgo.

C. S.	Análisis	SOLUCIÓN ESTOCÁSTICA	SOLUCIÓN DETERMINISTA
Bajo	Sensores:	$X_{8-11}, X_{10-32}, X_{13-15}, X_{26-28}, X_{30-31}, X_{47-48}$	$X_{8-11}, X_{10-32}, X_{13-15}, X_{26-28}, X_{30-31}, X_{47-48}$
	% de población: Costo total:	32.06 1.81208E+9	31.41 1.7874E+9
Alto	Sensores:	$X_{10-32}, X_{26-28}, X_{30-31}, X_{38-42}$	$X_{10-32}, X_{26-28}, X_{30-31}, X_{38-42}$
	% de población: Costo total:	40.66 2.93731E+9	40.01 2.9124E+9

C.S. = Costo del sensor

5.4.3. Análisis de resultados

De los resultados de las Tablas 5.3, 5.4, 5.6 y 5.7 se observan significativas diferencias entre los casos de costos de sensores alto y bajo, aunque no exista gran diferencia entre los casos de la solución determinista y estocástica; por el contrario, el efecto de la distribución usada para la densidad de población se ve que no es importante.

El número de sensores usados y la configuración óptima cambian con respecto al costo de los sensores. En general, la fracción de población en riesgo decrece cuando se incrementa el número de sensores; por lo que cuando se usan sensores que tienen un costo bajo, el número de sensores tiende a ser el máximo. Por otro lado, la relación que existe entre el costo de los sensores y el costo asociado por cada persona afectada restringe a que en la configuración óptima se llegue a usar el número máximo de sensores. Esto es particularmente correcto para el caso de sensores con un costo alto.

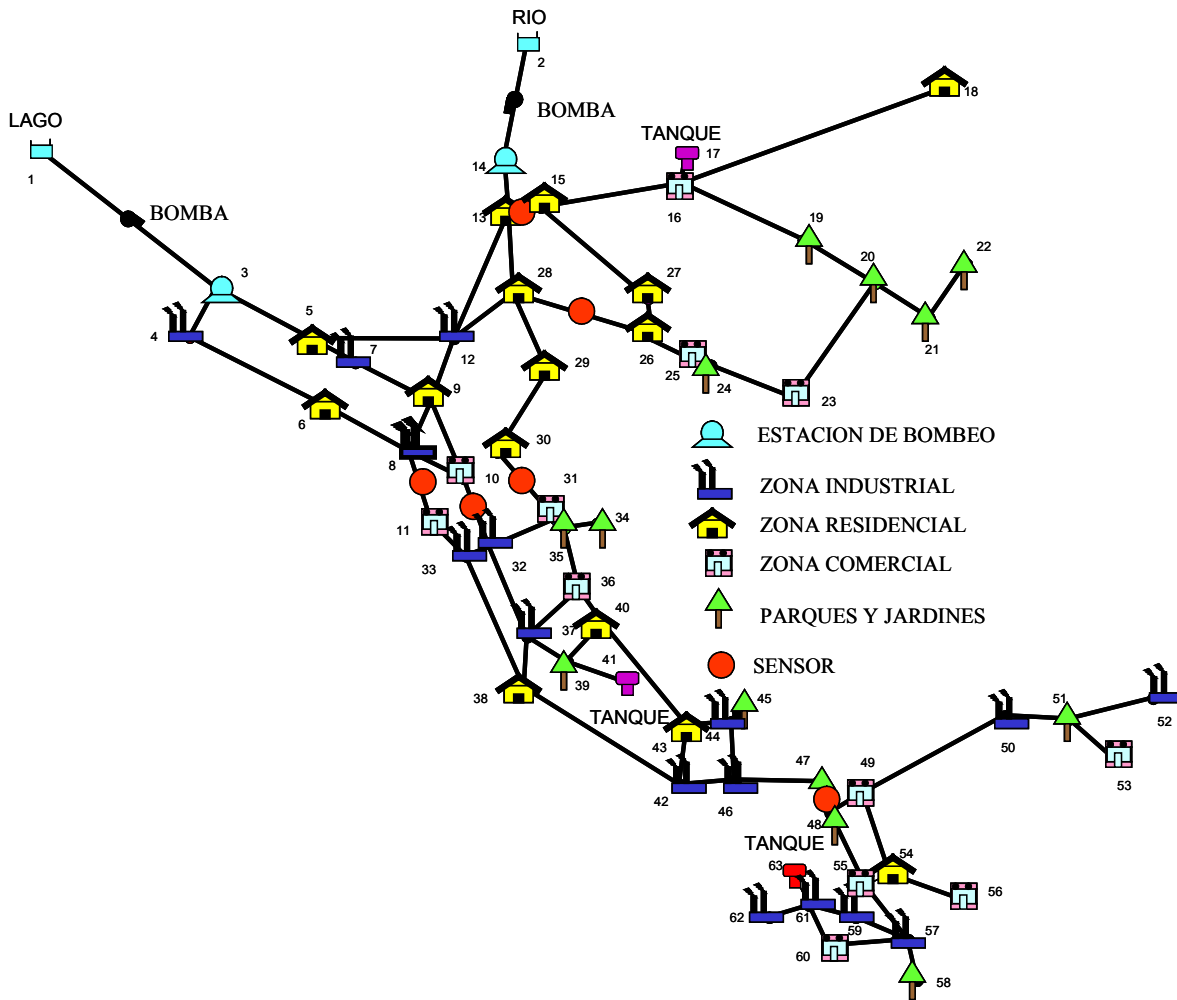


Figura 5.9. Configuración óptima para el Ejemplo 2 cuando la zona residencial se considera en riesgo y el costo del sensor es bajo.

Note sin embargo que para estos casos no existe una gran diferencia entre la solución determinista con la solución estocástica. Esto puede deberse a varios factores: i) las incertidumbres se encuentran sólo en la función objetivo, por lo que la única acción posible es la de planear en vista de los costos esperados de la población afectada; ii) patrones de flujo casi constantes, es decir, a pesar de que se han considerado 4 periodos de tiempo, la dirección del flujo no cambia drásticamente de un periodo a otro (ver patrones de flujo en los Apéndices C y D); iii) a pesar de que se considera incertidumbre de la

población en tránsito, para cada uno de los nodos se ha considerado una misma densidad de población (por ejemplo, una zona residencial puede tener más población que una zona industrial, o que una zona comercial); y iv) el costo de los sensores sólo se ha considerado como un factor de peso en relación directa con la población que pudiera resultar afectada, por lo que más conveniente sería considerar una función de costo para los sensores que varíe con respecto a la cantidad de personas afectadas.

En las Tablas 5.8 y 5.9 se presentan algunos resultados cuando se han considerado diferentes densidades de población en cada nodo. Los resultados reportados en estas tablas son para cuando la zona industrial se considera en riesgo en el Ejemplo 1 y para cuando la zona residencial se considera en riesgo para el Ejemplo 2. Las densidades de población usadas para cada nodo y en cada ejemplo se encuentran en el Apéndice D.

Tabla 5.8. Ejemplo 1 con zona industrial en riesgo y densidad de población diferente en cada nodo.

C. S.	Análisis	SOLUCIÓN ALGORITMO SD INCUMBENTE	SOLUCIÓN DETERMINISTA
Bajo	Sensores:	$X_{3-4}, X_{5-6}, X_{12-13}, X_{26-27}, X_{30-31}$	$X_{3-4}, X_{5-6}, X_{12-13}, X_{26-27}, X_{30-31}$
	% de población:	25.95	24.73
	Costo total:	2.15171E+8	2.0859E+8
Alto	Sensores:	$X_{5-6}, X_{12-13}, X_{26-27}$	$X_{5-6}, X_{12-13}, X_{26-27}$
	% de población:	33.80	32.53
	Costo total:	3.1755E+8	3.1069E+8

C.S. = Costo del sensor

Tabla 5.9. Ejemplo 2 con zona residencial en riesgo y densidad de población diferente en cada nodo.

C. S.	Análisis	SOLUCIÓN ESTOCÁSTICA	SOLUCIÓN DETERMINISTA
Bajo	Sensores:	$X_{8-11}, X_{10-32}, X_{13-15}, X_{26-28}, X_{30-31}, X_{38-42}$	$X_{8-11}, X_{10-32}, X_{13-15}, X_{26-28}, X_{30-31}, X_{38-42}$
	% de población:	35.23	34.29
	Costo total:	1.93191E+9	1.8965E+9
Alto	Sensores:	$X_{26-28}, X_{32-37}, X_{33-38}$	$X_{26-28}, X_{32-37}, X_{33-38}$
	% de población:	53.74	51.93
	Costo total:	3.08143E+9	3.0132E+9

C.S. = Costo del sensor

Los resultados las Tablas 5.8 y 5.9 presentan una diferencia que va de 2% hasta un 3% cuando se comparan los valores de la función objetivo para un costo de sensor bajo o uno de costo alto para una misma zona considera en riesgo. Siendo este aumento casi del 1% comparado con los casos en que se utilizó la misma densidad de población. Note nuevamente que para cada uno de los casos, la configuración óptima de la solución determinista y la de la solución estocástica son las mismas, y que la función objetivo es siempre mayor para la solución estocástica.

Del análisis anterior podría decirse que la razón del porqué la configuración óptima de la solución determinista y de la estocástica sean iguales, es porque las incertidumbres sólo se encuentran en la función objetivo de la segunda etapa y que una decisión que se tomara (configuración) en la primera etapa no sería afectada por ellas. Es decir, no existe ninguna restricción de segunda etapa que dependa de las incertidumbres y que sugiera realizar un cambio en la configuración obtenida de la primera etapa.

Por otro lado, en cuestiones computacionales y algorítmicas el método SD incumbente presenta gran estabilidad y robustez para resolver problemas a gran escala. En las Tablas 5.10 y 5.11 se muestran las características de los problemas aquí tratados. Para el análisis sólo se ha considerado el caso que utilizó mayor número de iteraciones para cada ejemplo, es decir, para el Ejemplo 1 se considera la zona industrial con un costo de sensor alto, y para el Ejemplo 2 la zona residencial con un costo de sensor bajo.

Observe también en las Tablas 5.10 y 5.11 que no se intentó siquiera resolver el problema con el método L-shaped por la cantidad inmensa de escenarios a considerar (para este cálculo se ha considerado que sólo se dispone de dos datos discretos y el valor de problemas a resolver se estima como en el caso del problema de la Ecuación 5.5). Note

entonces la robustez de la técnica SD incumbente para resolver problemas de gran magnitud.

Tabla 5.10. Características y esfuerzo computacional del Ejemplo 1.

ANÁLISIS	L-SHAPED	SD INCUMBENTE
Variables continuas	1815	1815
Variables binarias	43	43
Variables estocásticas	40	40
No. iteraciones	no se resolvió	276
No. escenarios	2^{40}	400
Problemas resueltos	2^{40} *no. de iteraciones	1104

Tabla 5.11. Características y esfuerzo computacional del Ejemplo 2.

ANÁLISIS	L-SHAPED	SD INCUMBENTE
Variables continuas	4503	4503
Variables binarias	96	96
Variables estocásticas	68	68
No. iteraciones	no se resolvió	166
No. escenarios	2^{68}	300
Problemas resueltos	2^{68} *no. de iteraciones	664

Capítulo 6

Conclusiones y Trabajo Futuro

En este capítulo se suman las contribuciones hechas en este trabajo, nuestras conclusiones y el trabajo futuro a seguir.

6.1 Conclusiones

1. Se implementó un algoritmo de solución para resolver problemas que son estocásticos enteros en la primera etapa de uno de dos etapas con recurso completo. Con este esquema se resolvieron diferentes problemas de aplicación, en la optimización de cadenas de suministro y en la seguridad de redes de agua municipales.
2. En los ejemplos de aplicación se consideraron explícitamente las incertidumbres en sus parámetros estocásticos, por ejemplo, a las disponibilidades de la materia prima, a las demandas de los mercados, a las densidades de población y a las probabilidades del riesgo de ataque se les asignaron una función de distribución de probabilidad continua.
3. Al algoritmo SD básico se le implementó un procedimiento que identifica una secuencia de soluciones incumbentes y un criterio de paro que permite manejar un error de su límite superior en un intervalo de confianza dado (algoritmo SD incumbente), consiguiéndose con ello una considerable disminución del número de problemas por resolver con respecto de un equivalente determinista.
4. Con el algoritmo SD incumbente no sólo se resolvieron problemas que se encuentran limitados por un cierto número de variables, parámetros inciertos y/o escenarios a considerar, sino que también se resolvieron problemas que son a gran escala (por ejemplo, se resolvió un problema de una red de agua municipal que tiene casi 5000 variables continuas, 90 variables binarias y 60 parámetros inciertos).
5. En los problemas que se han resuelto se observa que la técnica de muestreo de la secuencia de Hammersley no logra una convergencia cuando se utiliza el algoritmo SD incumbente, siendo mejor en estos casos la técnica de muestreo Monte Carlo.
6. El modelo propuesto para optimizar una cadena de suministro podría ser una herramienta útil que ayude a tomar decisiones sobre el como diseñar y planear una red

más robusta en términos de sus capacidades de producción, de sus capacidades de expansión y de sus incertidumbres de los mercados. Además, de que permite medir el riesgo de lo que se puede esperar si el peor escenario llega a ocurrir.

7. Los resultados que se obtuvieron en los problemas de seguridad en redes de agua municipales confirman que la fracción de población en riesgo o contaminada se ve influenciada en gran medida por la decisión acerca del nodo en que se colocará el sensor, la que a su vez es afectada por su costo de instalación. Lo anterior revela la importancia de considerar una función de costo para los sensores.
8. El porcentaje de población en riesgo disminuye conforme aumenta el número de sensores, lo que confirma la heurística que la mejor estrategia para colocar los sensores es en los lugares o puntos con mayor población.
9. El hecho de que la configuración óptima de la solución determinista y la solución estocástica sean iguales pudiera deberse a que las incertidumbres sólo se encuentran en la función objetivo de la segunda etapa, razón por la que la configuración determinada en la primera etapa no encuentre ninguna restricción dependiente de las incertidumbres en la segunda que sugiera ser modificada.

6.2 Trabajo futuro

1. Muchos problemas de operación y planeación involucran decisiones sobre el tiempo, por lo que sus formulaciones resultantes pudieran ser modelos estocásticos multi-etapa (ver Birge y Louveaux, 1991). Por consiguiente, el trabajo aquí presentado sería una buena referencia para atacar esta clase de problemas.
2. La dificultad para resolver el caso entero en la segunda etapa refleja la ausencia de un algoritmo de solución eficiente, por lo que este campo es un área prometedora y de gran

reto a futuro (Birge y Louveaux, 1991). El siguiente paso, sería el de extender el algoritmo aquí tratado al caso entero en la segunda etapa.

3. En este trabajo sólo se han abordado los problemas estratégicos o de diseño de la cadena de suministro, por lo que queda para trabajo futuro abordar los problemas tácticos o de planeación, y los estratégicos u operacionales (Gupta y Maranas, 2003; Bok, Grossmann y Park, 2000). Así como también el considerar incertidumbres en otro tipo de parámetros del modelo, tales como coeficientes de reacción, costos de materia prima, costos de producto, y tiempos de procesamiento (Clay Grossmann, 1997).
4. En los problemas de redes de agua municipales no se ha considerado una velocidad de reacción o de decaimiento del contaminante, por lo que algunos consumidores podrían no estar en riesgo (Berry y col., 2003). Por otro lado, esta clase de problemas también podría extenderse a una planeación y distribución del consumo desde el punto de vista de un recurso sustentable.

Bibliografía

- Acevedo, J. and E. N. Pistikopoulos (1998). Stochastic Optimization Based Algorithms for Process Synthesis Under Uncertainty. *Computers and Chemical Engineering*, 22 (4/5), 647.
- Applequist, G., J. F. Penky G. V. Reklaitis (2000). Risk and Uncertainty in Managing Chemical Manufacturing Supply Chains. *Computers and Chemical Engineering*, 24, 2211.
- Beale, E. M. L. (1955). On Minimizing a Convex Function Subject to Linear Inequalities. *J. R. Stat. Soc. B17*: 173-184.
- Berry J., L. Fletcher, W. Hart and C. Philips (2003). Optimal Sensor Placement in Municipal Water Networks. *In Proceedings of the World Water & Environmental Resources Congress*, Philadelphia, PA.
- Bienstock, D. and J. F. Shapiro (1985). Stochastic Programming Models for Strategic Planning: An Application to Electric Utilities. *MIT Technical Report OR 128-84*, revised.
- Birge, J. R. and F. Louveaux (1997). Introduction to Stochastic Programming. *Springer-Verlag*, New York.
- Birge, J. R. and F. Louveaux (1998), A Multicut Algorithm for Two Stage Stochastic Linear Programming. *European Journal of Operations Research*, 34, 384-392.
- Bok, J. K., I. E. Grossmann and S. Park (2000). Supply Chain Optimization in Continuous Flexible Process Networks. *Industrial and Chemical Engineering Chemistry Research*, 39(5), 1279.
- Brooke, A., D. Kendrick, A. Meeraus and R. Raman (1998). GAMS a User's Guide. *GAMS Development Corporation*.

-
- Clay, R. L. and I. E. Grossmann (1997). A Disgregation Algorithm for the Optimization of Stochastic Planning Models. *Computers and Chemical Engineering*, 21(7), 751.
- Chopra, S. and P. Meindl (2001). Supply Chain Management. Strategy, Planning and Operation. *Prentice Hall*, NJ.
- Dantzig, G. B. (1955). Linear Programming Under Uncertainty. *Management Sci.* 1: 197-206.
- Dantzig, G. B. (1988). Planning Under Certainty Using Parallel Computing. *Annals of Operations Research (Switzerland)*. 14, 1-4 (A01) 1-16.
- Dantzig, G. B. and P. Glynn (1990). Parallel Processors for Planning Under Uncertainty. *Annals of Operations Research*, 22, 1.
- Dempster, M. A. H.; J. K. Lenstra and A. H. G. Rinnooy Kan (1982). Deterministic and Stochastic Scheduling. *D. Reidel: Dordrecht*, Holland.
- Diwekar, U. M. (2003). Introduction to Applied Optimization. *Kluwer Academic Publishers*, Dordrecht, Netherlands and Norwell, MA.
- Gupta, A. and Maranas, C. D. (2003). Managing Demand Uncertainty in Supply Chain Planning. *Computers and Chemical Engineering*, 27, 1219.
- Haneveld, W. K. and M. H. Vander Vlerk (1999). Stochastic Integer Programming: General Models and Algorithms. *Annals of Operational Research*, 85, 39-57.
- Higle, J. L. and S. Sen (1991). Stochastic Decomposition: An Algorithm for Two-Stage Linear Programs with Recourse, *Mathematics of Operations Research*, 16, 650-669.
- Higle, J. L. and S. Sen (1996). Stochastic Decomposition. *Kluwer Academic Publisher*.
- Iyer, R. R. and I. E. Grossmann (1998). A Bilevel Decomposition Algorithm for Long-Range Planning of Process Networks. *Industrial and Engineering Chemistry Research*, 37(2), 474.
-

- Kalagnanam, J. and U. M. Diwekar (1997). An Efficient Sampling Technique for Off-line Quality Control. *Technometrics*, 39(3), 308.
- Kall, P. and S. W. Wallace (1994). Stochastic Programming. *John Wiley and Sons*, Chichester, England.
- Luenberger, D. G. (1984). Linear and Nonlinear Programming (Second Edition). *Addison Wesley Publishing Co.*, Reading MA.
- Painton, L. A. and U. M. Diwekar (1995). Stochastic Annealing under Uncertainty. *European Journal of Operations Research*, 83, 489.
- Ponce-Ortega, J. M., V. Rico-Ramírez, S. Hernández-Castro and U. M. Diwekar (2004). Improving Convergence of the Stochastic Decomposition Algorithm by Using an Efficient Sampling Technique, *Comp. Chem. Eng.*, 28, 767-773.
- Rico-Ramírez, V. and U. M. Diwekar (2002). Improving the Stochastic Decomposition Algorithm by Using an Efficient Sampling Technique. *INFORMS Annual Meeting*, San Jose, CA.
- Rossman, L. A. (1999). The EPANET Programmer's Toolkit for Analysis of Water Distribution Systems. *In Proceedings of the Annual Water Resources Planning and Management Conference*.
- Ryu, J. H. and E. N. Pistikopoulos (2005). Design and Operation of An Enterprise-Wide Process Network Using Operation Policies. 1. Design. *Industrial and Engineering Chemistry Research*. 44, 2174.
- Sahinidis, N. V., I. E. Grossmann, R. E. Fonari and M. Chathrati (1989). Optimization Model for Long Range Planning in the Chemical Industry. *Computers and Chemical Engineering*, 13(9), 1049.
- Shapiro, J. F. (2001). Modeling the Supply Chain. *Duxbury Press*, CA.

Sherali, H. D., F. H. Murphy, A. L. Soyster and S. Sen (1984). Allocation of Capital Costs in Electric Utility Capacity Expansion Planning Under Uncertainty. *Management Science*, 30, 1-19.

Simchi-Levi, D., P. Kaminsky and E. Simchi-Levi (1998). Designing and Managing the Supply Chain. *Irwin/McGraw-Hill*, New York, NY


```

cout<<"\nINTRODUZCA EL NÚMERO DE ITERACIONES=t";
cin>>nITER;

/*DEFINICIÓN DEL PROBLEMA PARA EL PASO 2
Max {PI^T(hv-Tv x^v)}
s.t. PI^T W <=q */

/* VARIABLES PARA EL PROBLEMA */

/* LOS COEFICIENTES DE PI[ITER][m],x[ITER][p],V[ITER][m],
alfa[ITER][p],beta[ITER][p],maxpi[ITER][p],argu[ITER][m]
ui[m], q[s],c[p], b[t], A[t][p]
SE DEBEN DE CAMBIAR DEACUERDO DEL PROBLEMA */

/* ASIGNACIÓN DINAMICA DE MEMORIA */
double (*PI)[m],(*PI2)[m],(*x1)[p],(*maxpi)[p],(*argu)[m];
double (*x2)[p],(*DELTA)[pp],*OBJF,*wsol,*eta,(*hv)[m-1],(*VV1)[m],(*VV2)[m];
double (*alfa)[p],(*beta)[p],(*alfa1)[p],(*beta1)[p],*desviacion,*limite;
int *ik;
PI=new double[nITER+1][m];
PI2=new double[nITER+1][m];
x1=new double[nITER+1][p];
VV1=new double[nITER+1][m];
VV2=new double[nITER+1][m];
alfa=new double[nITER+1][p];
beta=new double[nITER+1][p];
alfa1=new double[nITER+1][p];
beta1=new double[nITER+1][p];
maxpi=new double[nITER+1][p];
argu=new double[nITER+1][m];
hv=new double[nITER+1][m-1];
x2=new double[nITER+1][p];
ik=new int[nITER+1];
DELTA=new double[nITER+1][pp];
OBJF=new double[nITER+1];
wsol=new double[nITER+1];
eta=new double[nITER+1];
desviacion=new double[nITER+1];
limite=new double[nITER+1];

/* PASO 0 */
ITER=0;
L=-112652.676;

for(i=1;i<m;i++){
    VV1[ITER][i]=0;
    VV2[ITER][i]=0;
}

for(i=0;i<p;i++){
    x1[ITER+1][i]=0;
    x2[0][i]=0;
}

x1[1][0]=30;
x1[1][1]=20;
x1[1][2]=25;
x1[1][3]=40;
x1[1][4]=30.8;
x1[1][5]=20;
x1[1][6]=30;
x1[1][7]=30;
x1[1][12]=10.8;
x1[1][15]=10;
x1[1][20]=1.0;
x1[1][23]=1.0;

x2[0][0]=30;
x2[0][1]=20;
x2[0][2]=25;

```

```

x2[0][3]=40;
x2[0][4]=30.8;
x2[0][5]=20;
x2[0][6]=30;
x2[0][7]=30;
x2[0][12]=10.8;
x2[0][15]=10;
x2[0][20]=1.0;
x2[0][23]=1.0;

ik[0]=0;

OBJF[ITER+1]=-101229.044;
wsol[0]=-101229.044;

eta[0]=0;

DELTA[ITER][0]=2;
DELTA[ITER][1]=2;
DELTA[ITER][2]=2;
DELTA[ITER][3]=2;
DELTA[ITER][4]=2;
DELTA[ITER][5]=2;
DELTA[ITER][6]=2;
DELTA[ITER][7]=2;

/* SE ABRE UN ARCHIVO EN DONDE SE ALMACENAN LOS RESULTADOS IMPORTANTES */
ofstream resumen("RESUMEN.DAT");
if(!resumen){
    cout<<"No se puede abrir el archivo RESUMEN.DAT.\n";
    return 1;
}

resumen<<"\t\tRESUMEN DE LOS RESULTADOS\n";
resumen<<"\n\n\n";
resumen<<"IT.";
for(i=0;i<p;i++) resumen<<"\tX"<<i;
resumen<<"\tVAL.OBJ.";

/* PASO 1 */

/* SE LEEN LAS VARIABLES PROCEDENTES DEL MUESTREO Y
SON ALMACENADAS EN EL ARREGLO u[i] */
ifstream vestoca("LHSOUT.TMP"); //Variables del muestreo
if(!vestoca){
    cout<<"No se puede abrir el archivo LHSOUT.TMP.\n";
    return 1;
}

/* PASO 2 A */
do{
    ITER++;

    for(i=0;i<varest;i++) vestoca>>ui[i];

/* SE ASIGNAN LOS VALORES PARA hv(u) */
    for(i=0;i<m-1;i++) hv[ITER][i]=0;

    hv[ITER][8]=ui[0];
    hv[ITER][9]=ui[1];
    hv[ITER][10]=ui[2];
    hv[ITER][11]=ui[3];

/* SE GENERA EL PROBLEMA DE PROGRAMACIÓN LINEAL Y SE ESCRIBE EL
ARCHIVO EN CODIGO GAMS PARA DESPUES PODERLO COMPILAR
Max {PI^T(hv-Tv x^v)}
s.t. PI^T W <=q */
    ofstream proglin1("GAMS_1.GMS"); //Codigo GAMS del problema lineal

```

```

        if(!proglin1){
            cout<<"\nNo se puede abrir GAMS_1.GMS\n";
            return 1;
        }
        proglin1<<"$TITLE PASO_2_A"<<endl;
        proglin1<<"$OFFSYMXREF"<<endl;
        proglin1<<"$OFFSYMLIST"<<endl;
        proglin1<<"*n*DEFINICION DE VARIABLES, PARAMETROS Y ECUACIONES\n*n";

/* EL NÚMERO DE PI's DEPENDE DEL PROBLEMA ESPECIFICO */
        for(i=1;i<m;i++) proglin1<<"\nVARIABLES PI"<<i<<";";
        proglin1<<"\nVARIABLES OBJ;\n";
/* EL NÚMERO DE R's DEPENDE DEL PROBLEMA ESPECIFICO */
        proglin1<<"EQUATIONS PROB_LIN;\n";
        for(i=1;i<s;i++) proglin1<<"EQUATIONS R"<<i<<";\n";

        proglin1<<"PROB_LIN.. OBJ =E=";
        for(i=1;i<m;i++){
            suma=0.0;
            for(j=0;j<pp;j++) suma+=(T[i-1][j]*x1[ITER][j]);
            if((hv[ITER][i-1]-suma) != 0.0){
                if(i!=1) proglin1<<" + ";
                proglin1<<"("<<hv[ITER][i-1]-suma<<") * PI"<<i<<"\n";
            }
        }
        proglin1<<"";

        for(j=0;j<s-1;j++){
            proglin1<<"\nR"<<j+1<<"..";
            cont=0;
            for(i=0;i<m-1;i++){
                if(W[i][j]!=0.0){
                    cont=cont+1;
                    if(cont!=1) proglin1<<" + ";
                    proglin1<<"("<<W[i][j]<<") * PI"<<i+1;
                }
            }
            proglin1<<"=L= "<<q[j]<<";";
        }

        proglin1<<"\n\n*DEFINICION DE LAS ECUACIONES QUE FORMAN PARTE DEL MODELO";
        proglin1<<"\nMODEL PASO_2_A /ALL/;\n";
        proglin1<<"\nOPTION ITERLIM=1000000;";
        proglin1<<"\nOPTION RESLIM=1000000;";
        proglin1<<"\nOPTION LIMROW = 0;";
        proglin1<<"\nOPTION LIMCOL = 0;";
        proglin1<<"\nOPTION LP=CONOPT;\n";
        proglin1<<"\nPASO_2_A.WORKSPACE=10;";
        proglin1<<"\n*LLAMADA DE LA TECNICA DE SOLUCION";
        proglin1<<"\n\nSOLVE PASO_2_A USING LP MAXIMIZING OBJ;";
        proglin1<<"\n\nFILE RESULT_1 /C:\\GAMS_ESTO\\RESULT_1.DAT;";
        proglin1<<"\n\nPUT RESULT_1;";
        proglin1<<"\n\nRESULT_1.ND=6";

/* EL NÚMERO DE PI's DEPENDE DEL PROBLEMA ESPECIFICO */
        for(i=1;i<m;i++) proglin1<<"\nPUT PI"<<i<<".L;";
        proglin1<<"\nPUT OBJ.L;";
        proglin1<<"\n\nPUTCLOSE RESULT_1";

        proglin1.close();

/* SE MANDA COMPILAR EL PROGRAMA EN GAMS */
        system("gams GAMS_1");

/*****

/* La solución del problema anterior se adiciona al espacio
vectorial V al leer las variables del del archivo
RESULT_1.DAT */
        ifstream optimo_1("RESULT_1.DAT");           //Variables PI

```

```

        if(!optimo_1){
            cout<<"No se puede abrir el archivo RESULT_1.DAT.\n";
            return 1;
        }

/* PARA ESTE CICLO EL TAMAÑO DE m DEPENDE DEL NÚMERO DE PI'S
EL CUAL DEPENDE DEL TAMAÑO DE W Y hv O T */
        for(i=1;i<m;i++){
            optimo_1>>VV1[ITER][i];
            PI[ITER][i]=VV1[ITER][i];
        }
    optimo_1>>OBJ;
    optimo_1.close();

    ofstream proglin3("GAMS_3.GMS");
    if(!proglin3){
        cout<<"\nNo se puede abrir GAMS_3.GMS\n";
        return 1;
    }
    proglin3<<"$TITLE PASO_INCUM"<<endl;
    proglin3<<"$OFFSYMXREF"<<endl;
    proglin3<<"$OFFSYMLIST"<<endl;
    proglin3<<"*n*DEFINICION DE VARIABLES, PARAMETROS Y ECUACIONES\n*n";

    for(i=1;i<m;i++) proglin3<<"\nVARIABLES PI"<<i<<";";
    proglin3<<"\nVARIABLES OBJ;\n";

    proglin3<<"EQUATIONS PROB_LIN;\n";
    for(i=1;i<s;i++) proglin3<<"EQUATIONS R"<<i<<";\n";

    proglin3<<"PROB_LIN.. OBJ =E=";
    for(i=1;i<m;i++){
        suma=0.0;
        for(j=0;j<pp;j++) suma+=(T[i-1][j]*x2[ITER-1][j]);
        if((hv[ITER][i-1]-suma) != 0.0){
            if(i!=1) proglin3<<" + ";
            proglin3<<"("<<hv[ITER][i-1]-suma<<") * PI"<<i<<";\n";
        }
    }
    proglin3<<";";

    for(j=0;j<s-1;j++){
        proglin3<<"\nR"<<j+1<<";";
        cont=0;
        for(i=0;i<m-1;i++){
            if(W[i][j]!=0.0){
                cont=cont+1;
                if(cont!=1) proglin3<<" + ";
                proglin3<<"("<<W[i][j]<<") * PI"<<i+1;
            }
        }
        proglin3<<" =L= "<<q[j]<<";";
    }
    proglin3<<"\n\n*DEFINICION DE LAS ECUACIONES QUE FORMAN PARTE DEL MODELO";
    proglin3<<"\n\nMODEL PASO_INCUM /ALL/;\n";
    proglin3<<"\n\nOPTION ITERLIM=1000000;";
    proglin3<<"\n\nOPTION RESLIM=1000000;";
    proglin3<<"\n\nOPTION LIMROW = 0;";
    proglin3<<"\n\nOPTION LIMCOL = 0;";
    proglin3<<"\n\nOPTION LP=CONOPT;\n";
    proglin3<<"\n\nPASO_INCUM.WORKSPACE=10;";
    proglin3<<"\n\n*LLAMADA DE LA TECNICA DE SOLUCION";
    proglin3<<"\n\nSOLVE PASO_INCUM USING LP MAXIMIZING OBJ;";
    proglin3<<"\n\nFILE RESULT_3 /C:\\GAMS_ESTO\\RESULT_3.DAT/;";
    proglin3<<"\n\nPUT RESULT_3;";
    proglin3<<"\n\nRESULT_3.ND=6";

    for(i=1;i<m;i++) proglin3<<"\n\nPUT PI"<<i<<".L/;";
    proglin3<<"\n\nPUT OBJ.L;";
    proglin3<<"\n\nPUTCLOSE RESULT_3";

```

```

        proglin3.close();

        system("gams GAMS_3");

    ifstream optimo_3("RESULT_3.DAT");
    if(!optimo_3){
        cout<<"No se puede abrir el archivo RESULT_3.DAT.\n";
        return 1;
    }

    for(i=1;i<m;i++){
        optimo_3>>VV2[ITER][i];
        PI2[ITER][i]=VV2[ITER][i];
    }
    optimo_3>>OBJ;
    optimo_3.close();

/* PASO 2 B

PRIMERAMENTE OBTENER LOS COEFICIENTES DEL PI[ITER][k]
DONDE
    PI^V,k |
    MAX PI^T,k (h,k - T,k X^V)
    s.t: PI C V,v */

/* SE OPTIMIZAN LOS PI ANTERIORES */
    for(i=1;i<ITER;i++){
/* LOS ARREGLOS argu[i][k] EL NÚMERO DE k DEPENDE DEL PROBLEMA
ESPECIFICO */
        for(j=0;j<m-1;j++){
            suma=0.0;
            for(k=0;k<pp;k++) suma+=T[j][k]*x1[ITER][k];
            argu[i][j+1]=(hv[i][j]-suma);
        }
        for(r=1;r<ITER;r++){
            maxpi[r][0]=1E-80;
            for(k=1;k<=ITER;k++){
/* AQUI AUMENTARIAN LAS CADENAS DE V Y argu */
                suma=0.0;
                for(j=1;j<m;j++) suma+=VV1[k][j]*(argu[r][j]);
                pruebamax=suma;
                if(pruebamax>maxpi[r][0]){
                    maxpi[r][0]=pruebamax;
                    for(j=1;j<m;j++) PI[r][j]=VV1[k][j];
                }
                suma=0.0;
                for(j=1;j<m;j++) suma+=VV2[k][j]*(argu[r][j]);
                pruebamax=suma;
                if(pruebamax>maxpi[r][0]){
                    maxpi[r][0]=pruebamax;
                    for(j=1;j<m;j++) PI[r][j]=VV2[k][j];
                }
            }
        }

/* OBTENER LOS COEFICIENTES DEL CORTE DE OPTIMALIDAD: */
        alfa[ITER][0]=0.0;
        for(i=0;i<p;i++) beta[ITER][i]=0.0;
        for(k=1;k<=ITER;k++){
/* AQUI CAMBIARIAN LAS FUNCIONES DE ACUERDO CON EL PROBLEMA
Y PI's Y hv O T */
            for(j=1;j<m;j++) alfa[ITER][0]+=(PI[k][j]*hv[k][j-1]);
            for(i=0;i<pp;i++){
                for(j=1;j<m;j++) beta[ITER][i]+=(PI[k][j]*T[j-1][i]);
/* AQUI beta[ITER][p]CAMBIARIAN DEACUERDO DEL TAMAÑO DE X[p] */
            }
            alfa[ITER][0]/=double(ITER);
            for(i=0;i<pp;i++) beta[ITER][i]/=double(ITER);

```

```

suma=0.0;
for(i=0;i<p;i++) suma+=(c[i]+beta[ITER][i])*x1[ITER][i];
v1=alfa[ITER][0]+suma;

/*      if(ITER==ik[ITER-1]+1){*/
ikk=ik[ITER-1];
for(i=1;i<ITER;i++){
    for(j=0;j<m-1;j++){
        suma=0.0;
        for(k=0;k<pp;k++) suma+=T[j][k]*x2[ITER-1][k];
        argu[i][j+1]=(hv[i][j]-suma);
    }
}
for(r=1;r<ITER;r++){
    maxpi[r][0]=1E-80;
    for(k=1;k<=ITER;k++){
        suma=0.0;
        for(j=1;j<m;j++) suma+=VV2[k][j]*(argu[r][j]);
        pruebamax=suma;
        if(pruebamax>maxpi[r][0]){
            maxpi[r][0]=pruebamax;
            for(j=1;j<m;j++) PI2[r][j]=VV2[k][j];
        }
        suma=0.0;
        for(j=1;j<m;j++) suma+=VV1[k][j]*(argu[r][j]);
        pruebamax=suma;
        if(pruebamax>maxpi[r][0]){
            maxpi[r][0]=pruebamax;
            for(j=1;j<m;j++) PI2[r][j]=VV1[k][j];
        }
    }
}

alfa1[ITER][0]=0.0;
for(i=0;i<p;i++) beta1[ITER][i]=0.0;
for(k=1;k<=ITER;k++){
    for(j=1;j<m;j++) alfa1[ITER][0]+=(PI2[k][j]*hv[k][j-1]);
    for(i=0;i<pp;i++)
        for(j=1;j<m;j++) beta1[ITER][i]+=(PI2[k][j]*T[j-1][i]);
}
alfa1[ITER][0]/=double(ITER);
for(i=0;i<pp;i++) beta1[ITER][i]/=double(ITER);
alfa[ikk][0]=alfa1[ITER][0];
for(i=0;i<p;i++) beta[ikk][i]=beta1[ITER][i];

suma=0.0;
for(i=0;i<p;i++) suma+=(c[i]+beta[ikk][i])*x2[ITER][i];
v2=alfa[ikk][0]+suma;
/*      rho=0;
    */

/* PASO 2 C */

/* ACTUALICE LOS CORTES ANTERIORES SI LA ITERACIÓN ES MAYOR A 1 */
if(ITER!=1){
    for(k=1;k<ITER;k++){
        if(k!=ikk){
            alfa[k][0]=(double(ITER)-1)/double(ITER)*alfa[k][0]+L/double(ITER);
            for(i=0;i<p;i++) beta[k][i]=(double(ITER)-1)/double(ITER)*beta[k][i];
/* AQUI DEBERIAMOS DE CONSIDERAR LOS VALORES DE beta[k][p]
DEPENDIENDO DE X[p] */
        }
    }
}

if((v1-v2)<(rho*eta[ITER-1])){
    for(i=0;i<p;i++) x2[ITER][i]=x1[ITER][i];
    for(i=0;i<p;i++) DELTA[ITER][i]=2*DELTA[ITER-1][i];
    ik[ITER]=ITER;
}

```

```

    }
    else{
        for(i=0;i<p;i++) x2[ITER][i] = x2[ITER-1][i];
    for(i=0;i<p;i++) DELTA[ITER][i]=DELTA[ITER-1][i]/2;
        ik[ITER]=ik[ITER-1];
    }

/* PASO 3 */

/* RESOLVER EL PROBLEMA LINEAL
Min {c^T x + teta^v}
s.t.
    Ax=b
    teta^v + beta^v,k x >=alfa^v,k   k=1,...,v

    PARA OBTENER x^v+1 */

/* SE GENERA EL PROBLEMA DE PROGRAMACIÓN LINEAL Y SE ESCRIBE EL
ARCHIVO EN CODIGO GAMS PARA DESPUES PODERLO COMPILAR */
ofstream proglin2("GAMS_2.GMS");          //Codigo GAMS del problema lineal
if(!proglin2){
    cout<<"\nNo se puede abrir GAMS_2.GMS\n";
    return 1;
}

proglin2<<"$TITLE PASO_3"<<endl;
proglin2<<"$OFFSYMREF"<<endl;
proglin2<<"$OFFSYMLIST"<<endl;
proglin2<<"*\n*DEFINICION DE VARIABLES, PARAMETROS Y ECUACIONES\n*\n";
proglin2<<"\nVARIABLES OBJ,TETA"<<ITER<<";\n";

/* EL NÚMERO DE X's DEPENDE DEL PROBLEMA */
for(i=0;i<p-varint;i++)
    proglin2<<"\nPOSITIVE VARIABLES X"<<i<<";";
for(i=p-varint;i<p;i++)
    proglin2<<"\nBINARY VARIABLES X"<<i<<";";

/* AQUI EL NUMERO DE RR DEPENDEN DE LAS RESTRICCIONES DEL PROBLEMA,
ES DECIR DEL TAMAÑO DE b Y A */
proglin2<<"\nEQUATIONS PROB_LIN";
for(i=0;i<nea;i++)
    proglin2<<"\nEQUATIONS RR"<<i<<";";
for(i=1;i<=ITER;i++)
    proglin2<<"\nEQUATIONS R"<<i<<";";
for(i=0;i<pp;i++) proglin2<<"\nEQUATIONS RRR"<<i<<";";
for(i=0;i<pp;i++) proglin2<<"\nEQUATIONS RRRR"<<i<<";";

/* DEPENDIENDO DEL PROBLEMA SE CAMBIARIAN LOS ARGUMENTOS DE OBJ,
ASI COMO DE LAS ECUACIONES RR */
proglin2<<"\nPROB_LIN.. OBJ =E=";
for(i=0;i<p;i++){
    if(i!=0)
        proglin2<<" + ";
    proglin2<<("("<<<i<<") * X"<<i;
    if(i==p-1)
        proglin2<<" + TETA"<<ITER<<";";
}

/* AQUI EL NÚMERO DE RESTRICCIONES AUMENTARIAN DEPENDIENDO DEL
PROBLEMA */
for(i=0;i<nea;i++){
    proglin2<<"\nRR"<<i<<";";
    cont=0;
    for(j=0;j<p;j++)
        if(A[i][j]!=0.0){
cont=cont+1;
            if(cont!=1) proglin2<<" + ";
            proglin2<<("("<<<i<<A[i][j]<<") * X"<<j;
        }
}

```

```

        if(i<8) proglin2<<"=E="<<b[i]<<";
        if(i>=8) proglin2<<"=L="<<b[i]<<";
    }

/* AUMENTARIAN LAS ALFAS Y LAS BETAS DEPENDIENDO DEL TAMAÑO DE X */
for(i=1;i<=ITER;i++){
    proglin2<<"nR"<<i<<". TETA"<<ITER<<"=G= ("<<alfa[i][0]<<";
    for(j=0;j<pp;j++) proglin2 <<"- ("<<beta[i][j]<<") * X"<<j;
    proglin2 <<";
}
for(i=0;i<pp;i++){
    if(x2[ITER][i] != 0) proglin2<<"nRRR"<<i<<".X"<<i<<="L="<<DELTA[ITER][i]+x2[ITER][i]<<";
    if(x2[ITER][i] == 0) proglin2<<"nRRR"<<i<<".X"<<i<<="L="<<1.e-6<<";
}
for(i=0;i<pp;i++){
    if(x2[ITER][i] != 0) proglin2<<"nRRR"<<i<<".X"<<i<<="G="<<x2[ITER][i]-DELTA[ITER][i]<<";
    if(x2[ITER][i] == 0) proglin2<<"nRRR"<<i<<".X"<<i<<="G="<<0.<<";
}
proglin2<<"n\n*DEFINICION DE LAS ECUACIONES QUE FORMAN PARTE DEL MODELO";
proglin2<<"nMODEL PASO_3 /ALL;\n";
proglin2<<"nOPTION OPTCR=0.000001;";
proglin2<<"nOPTION OPTCA=0;";
proglin2<<"nOPTION ITERLIM=1000000;";
proglin2<<"nOPTION RESLIM=1000000;";
proglin2<<"nOPTION LIMROW = 0;";
proglin2<<"nOPTION LIMCOL = 0;";
/*
proglin2<<"nOPTION MIP=OSL;\n";*/
proglin2<<"nPASO_3.WORKSPACE=10;";
proglin2<<"n*LLAMADA DE LA TECNICA DE SOLUCION";
proglin2<<"n\nSOLVE PASO_3 USING MIP MINIMIZING OBJ;";
proglin2<<"n\nFILE RESULT_2 /C:\\GAMS_ESTO\\RESULT_2.DAT/;";
proglin2<<"nPUT RESULT_2;";
proglin2<<"nRESULT_2.ND=6";
for(i=0;i<p;i++) proglin2<<"nPUT X"<<i<<".L;";
proglin2<<"nPUT OBJ.L;";
proglin2<<"nPUTCLOSE RESULT_2";

proglin2.close();

system("gams GAMS_2");

/* La solución del problema anterior se lee del archivo
RESULT_2.DAT */
ifstream optimo_2("RESULT_2.DAT");           //Variables x
if(!optimo_2){
    cout<<"No se puede abrir el archivo RESULT_2.DAT.\n";
    return 1;
}

for(i=0;i<p;i++) optimo_2>>x1[ITER+1][i];
    optimo_2>>OBJ;
    cout<<"nOBJ=\t"<<OBJ<<"n";
optimo_2.close();
OBJF[ITER+1]=OBJ;
tau1=OBJF[ITER+1];

ofstream proglin4("GAMS_4.GMS");
if(!proglin4){
    cout<<"nNo se puede abrir GAMS_4.GMS\n";
    return 1;
}
proglin4<<"$TITLE PASO_4"<<endl;
proglin4<<"$OFFSYMXREF"<<endl;
proglin4<<"$OFFSYMLIST"<<endl;
proglin4<<"*n*DEFINICION DE VARIABLES, PARAMETROS Y ECUACIONES\n*n";
proglin4<<"nVARIABLES OBJ,TETA"<<ITER<<";n";
for(i=0;i<p-varint;i++)
    proglin4<<"nPOSITIVE VARIABLES X"<<i<<";
for(i=p-varint;i<p;i++)
    proglin4<<"nBINARY VARIABLES X"<<i<<";

```



```

proglin4<<"nEQUATIONS PROB_LIN;";
for(i=0;i<nea;i++)
    proglin4<<"nEQUATIONS RR"<<i<<";";
for(i=1;i<=ITER;i++)
    proglin4<<"nEQUATIONS R"<<i<<";";
proglin4<<"nPROB_LIN.. OBJ =E=";
for(i=0;i<p;i++){
    if(i!=0)
        proglin4<<" ";
    proglin4<<"("<<c[i]<<") * X"<<i;
    if(i==p-1)
        proglin4<<" + TETA"<<ITER<<";";
}
for(i=0;i<nea;i++){
    proglin4<<"nRR"<<i<<". ";
    cont=0;
    for(j=0;j<p;j++)
        if(A[i][j]!=0.0){
cont=cont+1;
            if(cont!=1) proglin4<<" ";
            proglin4<<"("<<A[i][j]<<") * X"<<j;
        }
    if(i<8) proglin4<<"L= "<<b[i]<<";";
    if(i>=8) proglin4<<"L= "<<b[i]<<";";
}
for(i=1;i<=ITER;i++){
    proglin4<<"nR"<<i<<". TETA"<<ITER<<" =G= ("<<alfa[i][0]<<");";
    for(j=0;j<pp;j++) proglin4 <<"- ("<<beta[i][j]<<") * X"<<j;
    proglin4 <<";";
}
proglin4<<"n\n*DEFINICION DE LAS ECUACIONES QUE FORMAN PARTE DEL MODELO";
proglin4<<"nMODEL PASO_4 /ALL/;\n";
proglin4<<"nOPTION ITERLIM=1000000;";
proglin4<<"nOPTION RESLIM=1000000;";
proglin4<<"nOPTION LIMROW = 0;";
proglin4<<"nOPTION LIMCOL = 0;";
/*
proglin4<<"nOPTION MIP=OSL;\n";*/
proglin4<<"nPASO_4.WORKSPACE=10;";
for(i=0;i<p;i++)
    proglin4<<"n X"<<i<<".FX = "<<x2[ITER][i]<<";";
proglin4<<"n*LLAMADA DE LA TECNICA DE SOLUCION";
proglin4<<"n\nSOLVE PASO_4 USING MIP MINIMIZING OBJ;";
proglin4<<"n\nFILE RESULT_4 /C:\\GAMS_ESTO\\RESULT_4.DAT/;";
proglin4<<"nPUT RESULT_4;";
proglin4<<"nRESULT_4.ND=6";
proglin4<<"nPUT OBJ.L;";
proglin4<<"nPUTCLOSE RESULT_4";
proglin4.close();
system("gams GAMS_4");
ifstream optimo_4("RESULT_4.DAT");
if(!optimo_4){
    cout<<"No se puede abrir el archivo RESULT_4.DAT.\n";
    return 1;
}
    optimo_4>>tau2;
optimo_4.close();

    eta[ITER]=tau1-tau2;

/*
    wsol[ITER]=(((double(ITER)-double(ik[ITER]))/(double(ITER)-double(ik[ITER])+1))*wsol[ITER-1])+
        ((1/(double(ITER)-double(ik[ITER])+1))*OBJF[ITER]);
*/
suma=0.0;
for(i=1;i<=ITER;i++) suma+=OBJF[i];
wsol[ITER]=suma/double(ITER);

    suma=0.0;
for(i=1;i<=ITER;i++) suma+=(OBJF[i]-wsol[ITER])*(OBJF[i]-wsol[ITER]);
    desviacion[ITER]=sqrt(suma/(double(ITER)-1));

```

```
limite[ITER]=wsol[ITER]+((2*desviacion[ITER])/sqrt(double(ITER)));

if(ITER>10) error=(fabs(limite[ITER]-limite[ITER-1])/fabs(limite[ITER]));
else error=1.0;

/* SE ALMACENAN LOS RESULTADOS*/

resumen<<"\n"<<ITER+1;
for(i=0;i<p;i++) resumen<<"\t"<<x1[ITER+1][i];
resumen<<"\t"<<OBJF[ITER+1];
resumen<<"\t"<<wsol[ITER];
resumen<<"\t"<<desviacion[ITER];
resumen<<"\t"<<error;

/* SE REGRESA AL PASO DOS */

} while((ITER<nITER) & (error>=1E-8));

resumen<<"\nIT.";
for(i=0;i<p;i++) resumen<<"\tX"<<i;
resumen<<"\tVAL.OBJ.";
resumen.close();
vestoca.close();
return 1;
}
```

Apéndice B

Patrones de la dirección del flujo de los periodos II, III y IV que se utilizan para resolver el Ejemplo 1 del Capítulo 5.

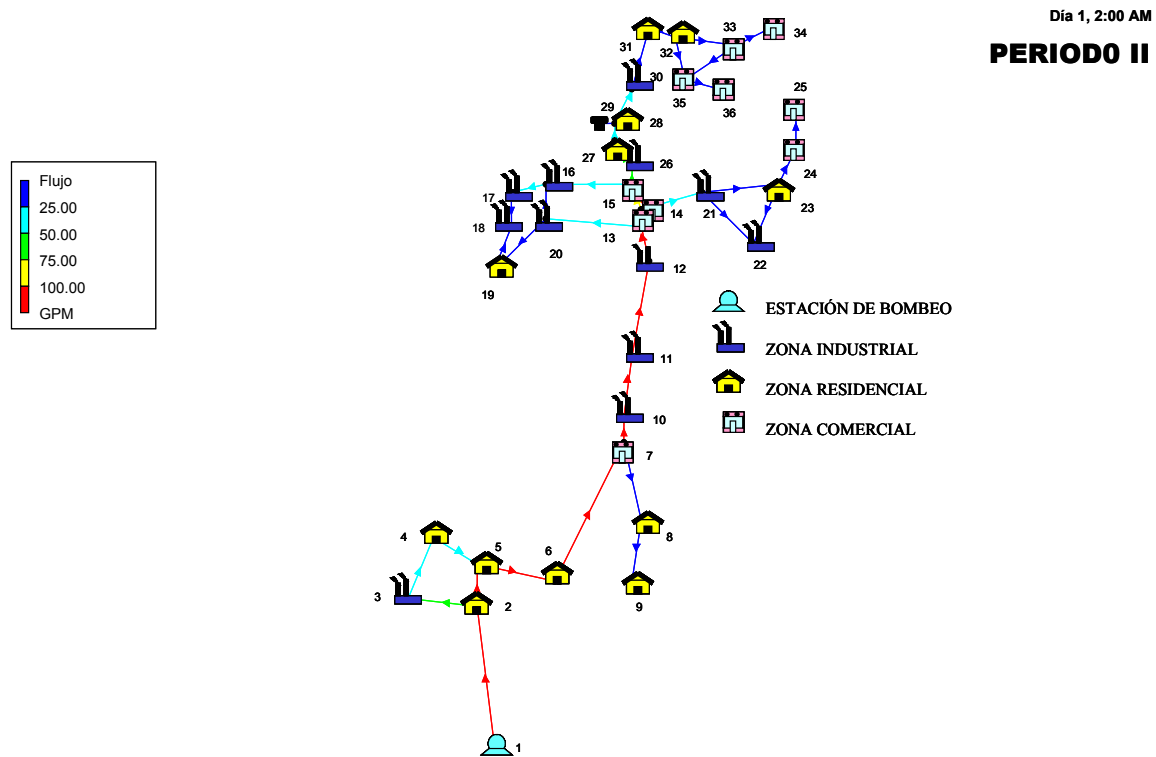


Figura B.1. Patrón de la dirección del flujo para el periodo II del Ejemplo 1.

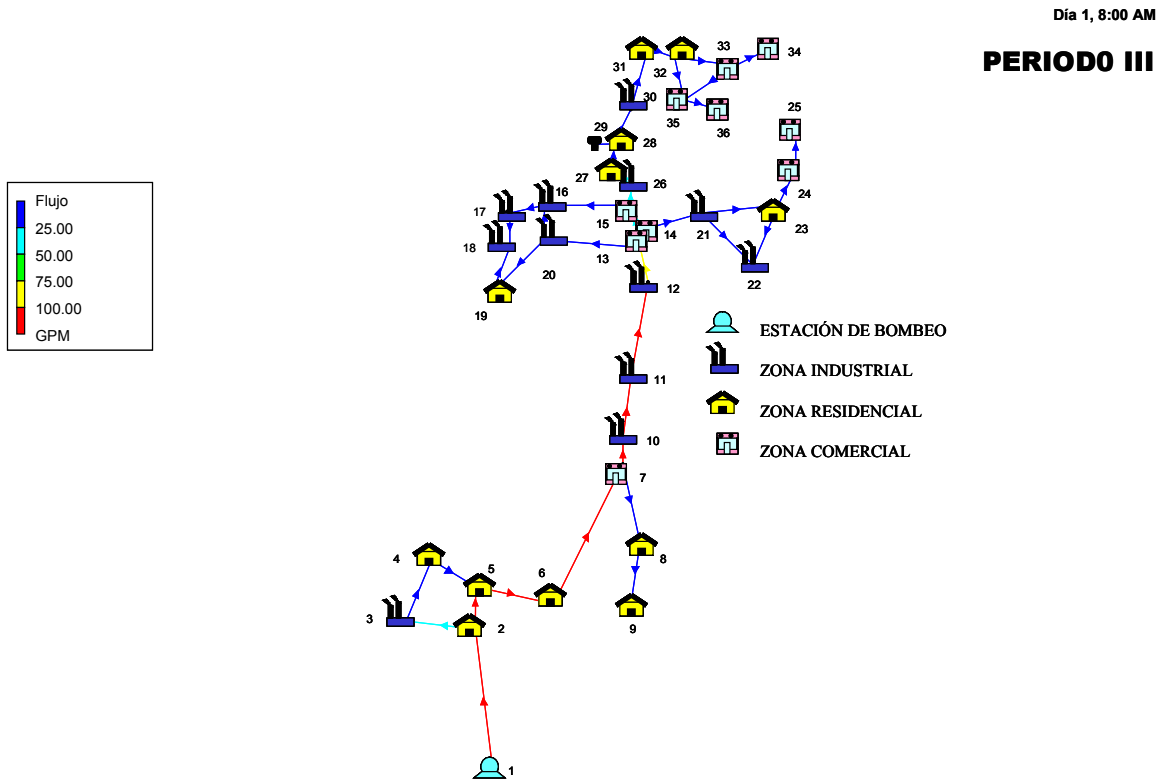


Figura B.2. Patrón de la dirección del flujo para el periodo III del Ejemplo 1.

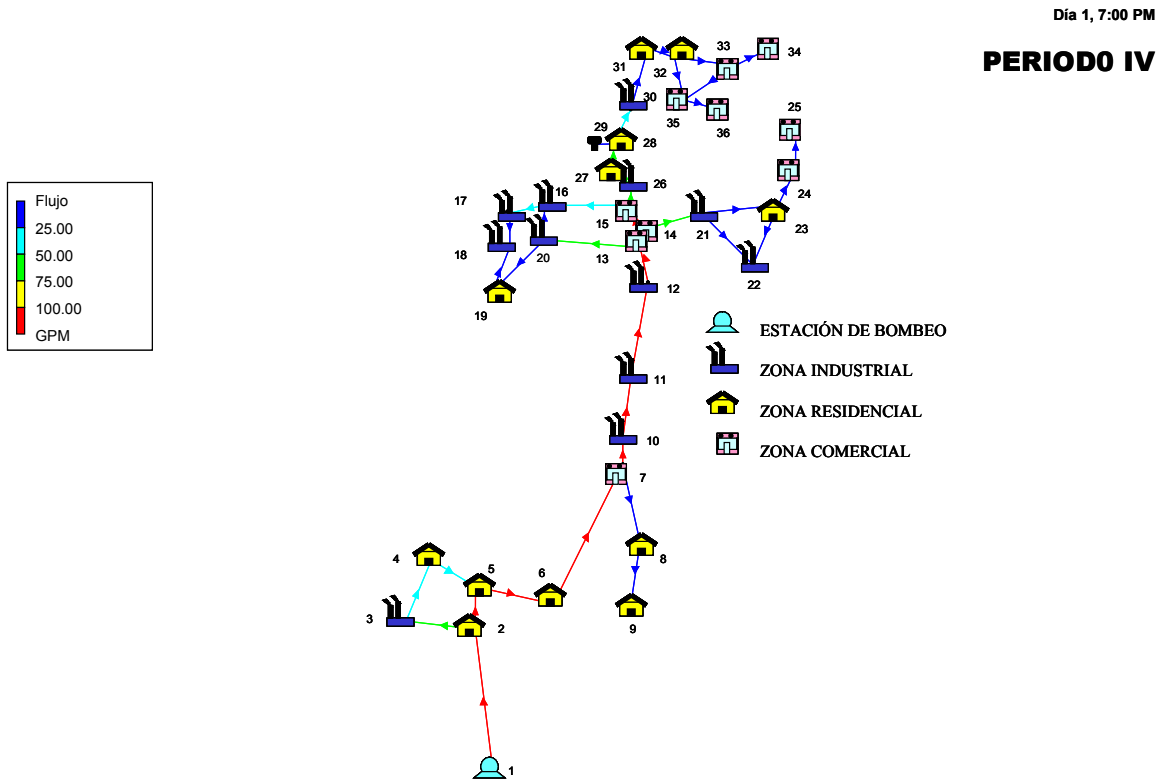


Figura B.3. Patrón de la dirección del flujo para el periodo IV del Ejemplo 1.

Apéndice C

Patrones de la dirección del flujo para los periodos II, III y IV que se utilizan para resolver el Ejemplo 2 del Capítulo 5.

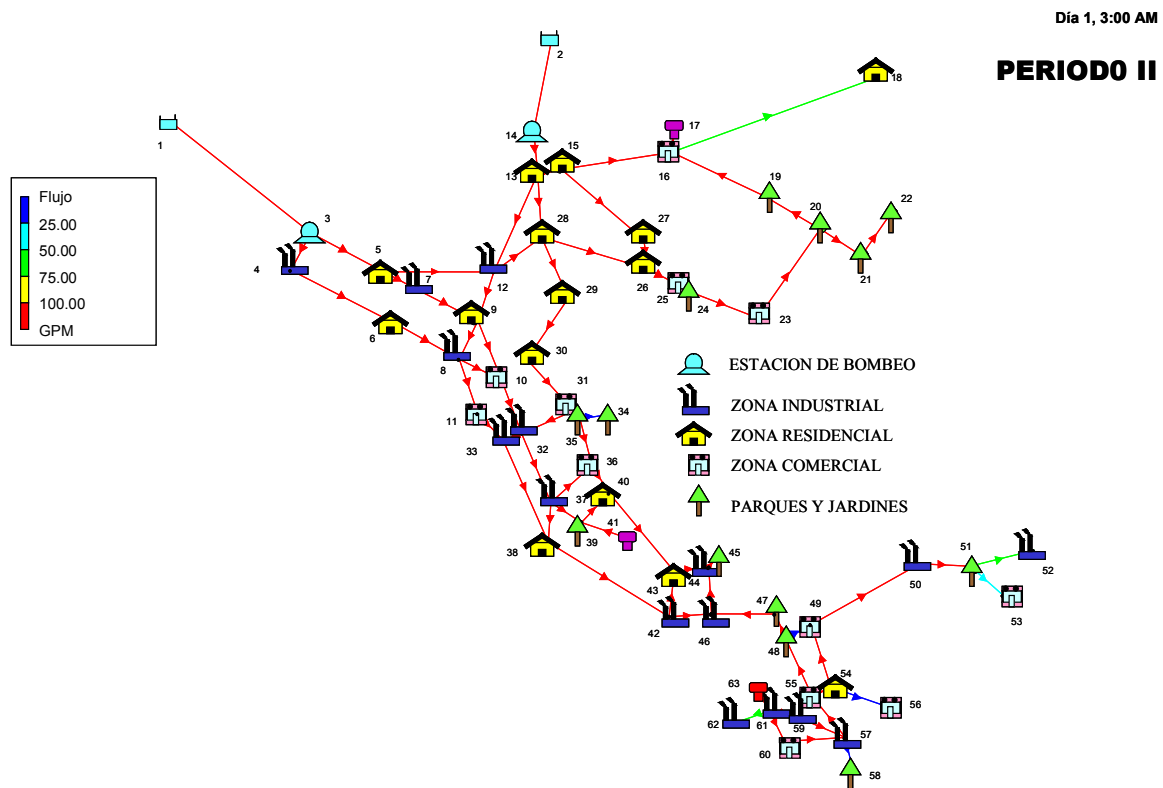


Figura C.1. Patrón de la dirección del flujo para el periodo II del Ejemplo 2.

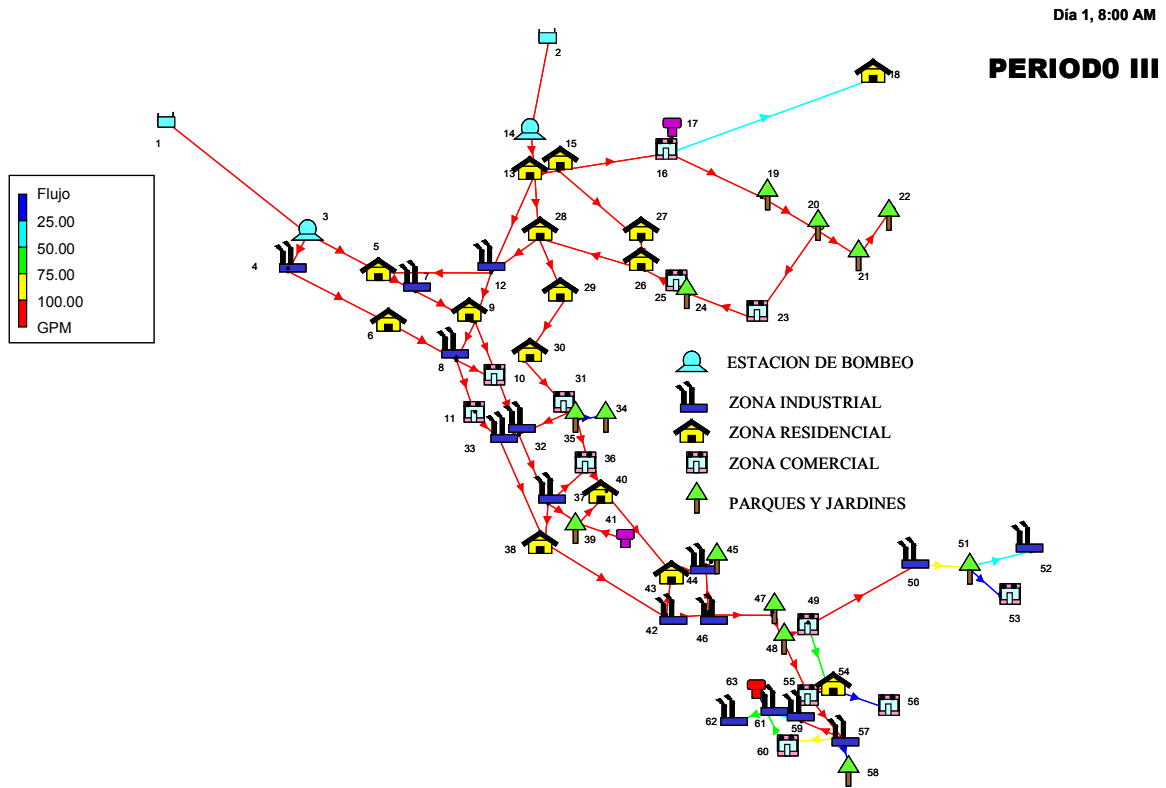


Figura C.2. Patrón de la dirección del flujo para el periodo III del Ejemplo 2.

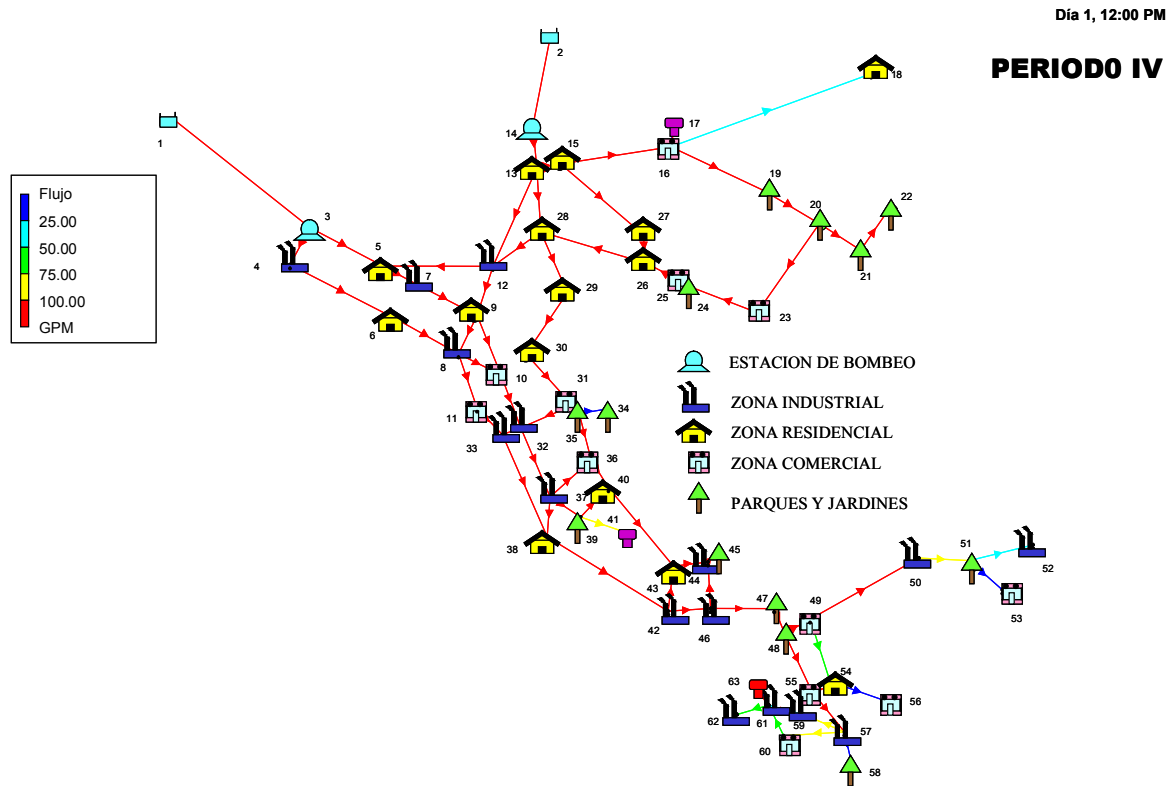


Figura C.3. Patrón de la dirección del flujo para el periodo IV del Ejemplo 2.

Apéndice D

Densidades de población para cada nodo que se utilizan para resolver el Ejemplo 1 y el Ejemplo del Capítulo 5.

Tabla D.1. Densidad de población para cada uno de los nodos del Ejemplo 1.

NODO	DENSIDAD DE POBLACIÓN	NODO	DENSIDAD DE POBLACIÓN
1	150	19	1050
2	750	20	350
3	400	21	450
4	750	22	200
5	1000	23	850
6	900	24	250
7	150	25	350
8	800	26	400
9	900	27	800
10	450	28	950
11	350	29	300
12	300	30	450
13	200	31	850
14	300	32	750
15	350	33	200
16	500	34	350
17	350	35	250
18	300	36	300

Tabla D.2. Densidad de población para cada uno de los nodos del Ejemplo 2.

NODO	DENSIDAD DE POBLACIÓN	NODO	DENSIDAD DE POBLACIÓN
1	1000	33	2500
2	2000	34	500
3	1000	35	500
4	1500	36	1500
5	3000	37	2000
6	3000	38	3500
7	2500	39	1000
8	2000	40	3000
9	3500	41	4000
10	2000	42	2000
11	1500	43	3500
12	2000	44	1500
13	3500	45	500
14	2000	46	1500
15	3000	47	1000
16	2500	48	500
17	3500	49	2000
18	3500	50	2500
19	1000	51	1000
20	500	52	1000
21	500	53	2000
22	1000	54	3500
23	1000	55	1000
24	1000	56	1500
25	1500	57	1500
26	4000	58	1000
27	3500	59	1000
28	4000	60	1500
29	4000	61	2000
30	4000	62	1500
31	2000	62	2000
32	1000		