

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE CIENCIAS

MODELO ESTOCÁSTICO Y ALGORITMOS EN-LÍNEA PARA EL PROBLEMA DE GESTIÓN DE SALDO EN CAJA EN EL BANCO CENTRAL DEL ECUADOR

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
MATEMÁTICO**

MARÍA CONSUELO SOTO LIMA

DIRECTOR: DR. LUIS MIGUEL TORRES

QUITO, SEPTIEMBRE 2007

DECLARACIÓN

Yo, MARÍA CONSUELO SOTO LIMA, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

María Consuelo Soto Lima

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por MARÍA CONSUELO SOTO LIMA bajo mi supervisión.

Dr. Luis Miguel Torres
DIRECTOR DE PROYECTO

AGRADECIMIENTOS

Mi sincero agradecimiento:

Al Dr. Luis Miguel Torres por su interés, paciencia y ayuda desinteresada en el desarrollo de esta tesis y sobre todo por transmitirme sus conocimientos.

Al Dr. Germán Rojas por sus enseñanzas, incentivos y su confianza en mí.

Al Mat. Diego Recalde por su ayuda en la realización de esta tesis.

A la Ing. Evelyn Quishpe por su sincera amistad y su ayuda incondicional a lo largo de mi carrera estudiantil.

DEDICATORIA

A Nuestro Padre Celestial, quien nos ha regalado la vida y los talentos

María Consuelo

TABLA DE CONTENIDO

RESUMEN	XII
Capítulo 1: CONCEPTOS DE PROGRAMACIÓN ESTOCÁSTICA	1
1.1. MODELOS DE OPTIMIZACIÓN ESTOCÁSTICA	3
1.2. EJEMPLOS DE PROGRAMACIÓN ESTOCÁSTICA	7
1.2.1. EL PROBLEMA DEL GRANJERO	7
1.2.2. EL PROBLEMA DEL VENDEDOR DE PERIÓDICO	16
1.2.3. PLANEACIÓN FINANCIERA Y CONTROL	18
1.3. PROGRAMA ESTOCÁSTICO BIETAPA	24
1.4. PROGRAMA ESTOCÁSTICO MULTIETAPA	27
1.5. EL VALOR DE INFORMACIÓN Y LA SOLUCIÓN ESTOCÁSTICA .	29
1.5.1. EL VALOR ESPERADO DE LA INFORMACIÓN PERFECTA EVPI	29
1.5.2. EL VALOR DE LA SOLUCIÓN ESTOCÁSTICA VSS	30
Capítulo 2: EL PROBLEMA DE GESTIÓN DE SALDO EN CAJA EN EL BANCO CENTRAL DEL ECUADOR (PGSC)	32
2.1. MODELO DE OPTIMIZACIÓN PARA EL PGSC	33
2.2. SOLUCIÓN BAJO INFORMACIÓN PERFECTA	35
2.3. MODELO ESTOCÁSTICO MULTIETAPA PARA EL PGSC	39
2.4. EJEMPLO	41
Capítulo 3: ALGORITMOS EN-LÍNEA PROPUESTOS PARA EL PGSC	51
3.1. ALGORITMOS DE BANDAS FIJAS	51
3.1.1. ALGORITMO 4-BANDAS	52
3.1.2. ALGORITMO 2-BANDAS	53
3.1.3. ALGORITMO LINEAL	54
3.1.4. ALGORITMO CUADRÁTICO	55
3.2. ALGORITMOS DE BANDAS MÓVILES	55
3.2.1. ALGORITMO 4-BANDAS MÓVIL	56
3.2.2. ALGORITMO 2-BANDAS MÓVIL	57
3.2.3. ALGORITMO LINEAL MÓVIL	58
3.2.4. ALGORITMO CUADRÁTICO MÓVIL	58
3.3. ALGORITMO ESTOCÁSTICO	60
3.4. EJEMPLO DE APLICACIÓN	61
3.5. CALIBRACIÓN DE PARÁMETROS	75

3.5.1. MODELO DE OPTIMIZACIÓN NO-LINEAL PARA LA CALIBRACIÓN DE PARÁMETROS	76
3.5.2. MÉTODO DE NELDER-MEAD	78
3.5.3. MÉTODO DEL DESCENSO PROFUNDO	81
Capítulo 4.: IMPLEMENTACIÓN COMPUTACIONAL	83
4.1. ESTRUCTURA GENERAL DEL PROGRAMA GSCaja	83
4.2. SIMULACIÓN DEL FLUJO DE CAJA	87
4.3. IMPLEMENTACIÓN DE LOS ALGORITMOS EN-LÍNEA	92
4.3.1. IMPLEMENTACIÓN DEL ALGORITMO ESTOCÁSTICO	93
4.4. CALIBRACIÓN DE PARÁMETROS	96
4.4.1. EVALUACIÓN DE LA UTILIDAD ESPERADA	96
4.4.2. IMPLEMENTACIÓN DE LOS MÉTODOS NUMÉRICOS DE OPTIMIZACIÓN	98
4.5. CÁLCULO DE LA BRECHA DE OPTIMALIDAD	101
Capítulo 5.: RESULTADOS NUMÉRICOS	102
5.1. CALIBRACIÓN DE PARÁMETROS	102
5.1.1. HORIZONTE DE 10 DÍAS	106
5.1.2. HORIZONTE DE 30 DÍAS	110
5.1.3. HORIZONTE DE 100 DÍAS	113
5.2. RESULTADOS PARA INSTANCIAS REALES	116
5.2.1. HORIZONTE DE 10 DÍAS	116
5.2.2. HORIZONTE DE 30 DÍAS	123
5.2.3. HORIZONTE DE 100 DÍAS	126
5.2.4. REGISTRO HISTÓRICO DE 500 DÍAS	130
Capítulo 6.: CONCLUSIONES	133
Apéndice A.: DISTRIBUCIONES DE PROBABILIDAD PARA EL FLUJO DE CAJA NETO	138
A.1. MODELOS UNIVARIANTES	138
A.2. MODELOS DE INTERVENCIÓN	145
A.3. SALDO MÍNIMO	150
Apéndice B.: SOLUCIÓN DEL PGSC	152
B.1. MODELO EXTENSO DEL PEGSC PARA EL EJEMPLO	152
B.2. FORMATO SMPS	156
B.3. FORMATO DIMACS	165

ÍNDICE DE FIGURAS

1.1. Árbol de escenarios para dos conjuntos de retornos y tres períodos de inversión.	20
2.1. Saldo en caja y retiros diarios desde enero del 2001 hasta junio del 2006. Fuente: Banco Central del Ecuador.	32
2.2. Saldo en caja y requerimientos neto diarios desde enero del 2001 hasta junio del 2006. Fuente: Banco Central del Ecuador.	33
2.3. Reducción del PGSC a un problema de flujo de costo mínimo.	38
2.4. Árbol de escenarios	43
3.1. Inversiones/Retiros realizados por los algoritmos de bandas fijas.	66
3.2. Saldos en caja obtenidos por los algoritmos de bandas fijas.	66
3.3. Inversiones/Retiros realizados por los algoritmos de bandas móviles.	69
3.4. Saldos en caja obtenidos por los algoritmos de bandas móviles.	70
3.5. Inversiones/Retiros realizados por el algoritmo Estocástico y por la solución óptima.	72
3.6. Saldos en caja obtenidos por el algoritmo Estocástico y por la solución óptima.	73
4.1. Estructura GSCaja.	84
4.2. Menú Principal del programa GSCaja.	86
4.3. Estructura fichero modelos.cpp	87
A.1. Flujo de caja neto enero-diciembre del 2005	139
A.2. Correlograma de la serie flujo de dinero	139
A.3. Desviación estándar vs la media	140
A.4. Modelo $SARIMA(1, 0, 0)(1, 0, 1)_5 - GARCH(0, 1)$	145
A.5. Flujo real versus el flujo estimado por el modelo	147
A.6. predicción diaria realizada con Sarima-Impulso.	149
A.7. predicción diaria realizada con Sarima-Impulso.	151

ÍNDICE DE TABLAS

1.1. Problema del granjero. Datos de entrada.	8
1.2. Soluciones óptimas determinísticas para cada escenario.	10
1.3. Solución óptima empleando la forma extensa del modelo estocástico	13
1.4. Solución del valor esperado.	14
1.5. Solución óptima del modelo estocástico para 3 períodos	22
2.1. Valores y probabilidades de las hojas del árbol de la Figura 2.4 . . .	44
2.2. Escenarios para el flujo de caja neto utilizados en el ejemplo	45
2.3. Solución del PEGSC para el Ejemplo.	46
2.4. Solución del ejemplo aplicada a un flujo real perteneciente al tercer escenario.	47
2.5. Solución óptima para el flujo de caja real del tercer escenario. . . .	47
2.6. Utilidad óptima para cada escenarios del ejemplo.	48
2.7. Solución del programa determinístico para el ejemplo	48
2.8. Decisiones tomadas respecto al Cuadro 2.7 para el flujo de caja del octavo escenario.	49
2.9. Decisiones tomadas respecto al Cuadro 2.3 para el flujo de caja del octavo escenario	49
2.10. Utilidades para cada escenario obtenidas para con la solución del EV.	50
3.1. Flujo y saldo mínimo para la instancia real de 30 días.	62
3.2. Parámetros utilizados para los algoritmos.	63
3.3. Desempeño de los algoritmos de bandas fijas para una instancia real de 30 días de duración.	65
3.4. Desempeño de los algoritmos de bandas móviles para una instancia real de 30 días de duración.	69
3.5. Desempeño del algoritmo estocástico y solución óptima para una instancia real de 30 días de duración.	71
3.6. Resumen de los algoritmos aplicados a la instancia del ejemplo. . .	74
5.1. Puntos iniciales.	105
5.2. Parámetros óptimos para un horizonte de tiempo de 10 días.	109
5.3. Parámetros óptimos para un horizonte de tiempo de 30 días.	113
5.4. Parámetros óptimos para un horizonte de tiempo de 100 días. . . .	116
5.5. Algoritmos con el mejor desempeño, para instancias reales con ho- rizontes de tiempo de 10 días.	121

5.6. Desempeño promedio de los algoritmos sobre instancias reales con horizontes de tiempo de 10 días.	123
5.7. Algoritmos con el mejor desempeño, para instancias reales con horizontes de tiempo de 30 días.	126
5.8. Desempeño promedio de los algoritmos sobre instancias reales con horizontes de tiempo de 30 días.	128
5.9. Algoritmos con el mejor desempeño, para instancias reales con horizontes de tiempo de 100 días.	128
5.10. Desempeño promedio de los algoritmos sobre instancias reales con horizontes de tiempo de 100 días.	130
5.11. Desempeño de los algoritmos sobre la instancia real con horizonte de tiempo de 500 días.	132
A.1. Información $SARIMA(1, 0, 0)(2, 0, 0)_5$	141
A.2. Información $SARIMA(1, 0, 0)(1, 0, 1)_5$	142
A.3. Criterios para la selección del modelo	143
A.4. Información $SARIMA(1, 0, 0)(1, 0, 1)_5 - GARCH(0, 1)$	146
A.5. Información ajuste del saldo mínimo y el flujo promedio cada 5 días.	151

ÍNDICE DE ALGORITMOS

3.1. 4-Bandas	52
3.2. 2-Bandas	53
3.3. Lineal	54
3.4. Cuadrático	55
3.5. 4-Bandas Móvil	56
3.6. 2-Bandas Móvil	57
3.7. Lineal Móvil	58
3.8. Cuadrático Móvil	59
3.9. Estocástico	60
3.10. Nelder-Mead	80
3.11. Descenso Profundo	82

RESUMEN

En este proyecto se aborda un problema de gestión de saldo en caja, que es un problema de programación estocástica multietapa, el cual tiene un horizonte de planificación amplio. El programa estocástico, es bastante extenso y crece exponencialmente conforme el número de períodos y escenarios para los cuales se toman las decisiones crece, implicando la utilización de más tiempo y recursos computacionales. Por esta razón, se estudia el desempeño de diversos algoritmos en-línea propuestos para el problema a partir de simulaciones computacionales, basadas en datos reales. El desempeño de estos algoritmos en-línea dependen de ciertos parámetros, los cuales son calibrados mediante la utilización de métodos numéricos de optimización. Se realiza una comparación de los diferentes algoritmos en-línea, con la política óptima para el modelo estocástico, que resulta ser un problema de flujo de costo mínimo.

Palabras claves: Programación estocástica, programación estocástica multietapa, algoritmos en-línea, problemas de flujo de costo mínimo, método de Nelder-Mead y método del Descenso Profundo.

Capítulo 1

CONCEPTOS DE PROGRAMACIÓN ESTOCÁSTICA

En un problema real de toma de decisión se puede estar ante una situación de certidumbre, riesgo e incertidumbre como lo describe Muñoz en [22]:

- Certidumbre: Si cada acción da un resultado predecible con exactitud.
- Riesgo: Si se conoce la probabilidad de todos los posibles resultados de cada una de las acciones que pueden tomarse. Una situación de certidumbre es un caso particular de una situación de riesgo con probabilidades cero y uno.
- Incertidumbre: Si cada acción implica un conjunto de posibles resultados, pero se desconocen las probabilidades exactas de éstos.

Muchos problemas reales nos confrontan con situaciones de riesgo e incertidumbre debido a que dependen de parámetros de entrada que no pueden ser conocidos con exactitud, bien sea debido a errores técnicos en la medición de los datos, o porque estos representan cantidades cuya naturaleza intrínseca alberga un cierto grado de incertidumbre. Ejemplos al respecto son frecuentes en la práctica: proyecciones de oferta y demanda de un cierto producto, comportamiento de los precios en el futuro, niveles previstos de producción, etc.

En algunos casos, es posible asumir simplemente que estos parámetros toman valores estimados, bien sea porque el nivel de imprecisión es bajo o porque los mismos tienen un impacto marginal sobre el sistema que se quiere analizar. Sin embargo, hay ocasiones en que los parámetros sujetos a incertidumbre juegan un papel importante para el análisis de un proceso de decisiones [23]. Surge entonces la necesidad de formular modelos de optimización adecuados que incorporen la incertidumbre de los datos de entrada como un aspecto fundamental. Cuantificar el efecto de una decisión tomada bajo la presencia de la incertidumbre acerca del futuro se ha convertido en uno de los problemas centrales de la investigación de operaciones y las ciencias administrativas [24].

Existen algunos paradigmas fundamentales para modelar problemas bajo situaciones de riesgo o incertidumbre. Su simplicidad depende de las circunstancias y de la información disponible acerca de los parámetros aleatorios. Se describen a continuación los cuatro más comúnmente utilizados: la estadística matemática, la programación estocástica, la programación robusta y la optimización en-línea.

Los *modelos de programación estocástica*, requieren de representaciones probabilísticas de los parámetros que no son conocidos con certeza; estos modelos pueden considerarse como una extensión de los modelos de programación lineal y no lineal [1].

El objetivo de los programas estocásticos es encontrar alguna política que sea factible en todos (o casi todos) los posibles escenarios y que maximice el valor esperado de la función objetivo, la cual depende tanto de las decisiones tomadas, como de los parámetros aleatorios [28].

En la *estadística matemática* se utiliza más el principio de “esperar y ver”: amplia información acerca de un problema es recogida y procesada previa a la toma de una decisión. Interesan conocer, por ejemplo, la distribución del óptimo del problema o algunos de sus momentos, como la esperanza y la varianza, condicionados a la toma de decisiones en el pasado. En contraposición, el enfoque empleado por la programación estocástica se conoce como “aquí y ahora”: una solución debe ser encontrada a priori en base a información incompleta [23].

La *optimización robusta* aborda problemas en los cuales se conoce que los parámetros tomarán valores dentro de cierto rango, pero se desconoce su distribución exacta de probabilidad [28]. Mientras que en la programación estocástica se pueden incorporar actitudes adversas al riesgo, por ejemplo, mediante la penalización de los peores escenarios, en la optimización robusta se hace uso de una generalización del concepto de protección frente al riesgo: Un modelo es robusto cuando es casi factible en todos los escenarios y una solución es robusta cuando es casi óptima para todos los escenarios [31]. Así, el objetivo de la optimización robusta es encontrar una solución que equilibre simultáneamente la optimalidad y la factibilidad.

La *optimización en-línea*, finalmente, aborda problemas de optimización que requieren una respuesta de manera inmediata. De hecho, los datos de entrada se van revelando paulatinamente, mientras el proceso de solución ya está en marcha, y las decisiones deben ser tomadas sobre la base de información incompleta [19], [8].

La optimización en-línea suele aplicarse en el modelamiento de sistemas de decisión continuos, para lo cual emplea técnicas y conceptos tomados de la Teoría de Juegos. El diseño de estrategias de solución en-línea y el establecimiento de criterios para evaluar la calidad de un algoritmo en-línea son algunos de los tópicos de interés dentro de esta área relativamente joven de la optimización.

Una manera adicional de abordar la incertidumbre, que ha sido empleada con éxito dentro de ciertos escenarios, consiste en asignar implícitamente o explícitamente valores para los parámetros estocásticos, empleando una estimación que se ajuste bien a datos históricos de los mismos [22]. Sin embargo, como ya se señaló antes hay ocasiones en las que este procedimiento resulta inaplicable, debido a que no se dispone de datos estadísticos suficientes acerca del fenómeno que se está modelando, o a que el nivel de variabilidad de los parámetros es tan elevado como para causar un impacto significativo sobre la función objetivo [23].

1.1. MODELOS DE OPTIMIZACIÓN ESTOCÁSTICA

En [27] se presenta una breve reseña de la historia de la programación estocástica, la cual comienza en 1955 cuando el primer documento teórico fue publicado por Dantzig [3]. A finales de la década de los 50's algunos modelos de programación matemática fueron introducidos por Dantzig [2], Beale [4], Tintner y Charnes & Cooper [25]. Algunos problemas de programación estocástica han sido formulados y resueltos, por ejemplo, en teoría de inventarios, micro-economía y sistemas de mantenimiento [23]. En la siguiente década se encontró una de las aplicaciones más populares de la programación estocástica: el problema de la selección del portafolio. La misma bastó para que su autor, H. Markowitz, consiguiera el premio Nobel [27].

Durante los últimos 50 años, la programación estocástica ha encontrado numerosas aplicaciones exitosas, la mayoría en finanzas, pero también en transportación, manejo energético, hidrología, planeación militar y muchas otras áreas. En la actualidad, este campo registra un alto nivel de actividad de investigación alrededor de todo el mundo [27].

Una definición sencilla de programación estocástica es aquella dada por Prékopa [26]: “la resolución de problemas de programación matemática en los que algunos o todos los parámetros son variables aleatorias”. Es decir, la programación estocástica considera problemas en los que algunos parámetros son desconocidos, pero se conoce una distribución de probabilidad asociada a los mismos. De manera más precisa, un problema de programación estocástica puede formularse

de la siguiente manera:

$$\min_x z(x, \xi) \tag{1.1}$$

$$\begin{aligned} \text{(PEG) s.r} \quad & g_i(x, \xi) \leq 0, \quad i = 1, 2, \dots, m \\ & x \in D, \end{aligned} \tag{1.2}$$

donde $D \subset \mathbb{R}^n$ y ξ es un vector aleatorio definido sobre un conjunto $\Omega \subset \mathbb{R}^q$, el cual es el espacio muestral de los parámetros aleatorios. Las funciones $z(x, \xi)$, $g_1(x, \xi)$, $g_2(x, \xi)$, ..., $g_m(x, \xi)$ dependen del vector de decisión $x \in \mathbb{R}^n$ y de los parámetros aleatorios, es decir, están definidas sobre el espacio $\mathbb{R}^n \times \Omega$. Se supone que la distribución de probabilidad P sobre las partes de Ω ,

$$P : \mathcal{P}(\Omega) \rightarrow [0, 1],$$

es conocida e independiente de las variables de decisión. Como algunos parámetros del problema son aleatorios, tanto la función objetivo (1.1), como los lados izquierda de las restricciones (1.2) son también variables aleatorias [22]. Notar que para cada realización de ξ el problema de programación estocástica es un problema determinístico de programación matemática. De esta forma, el conjunto de soluciones factibles del problema puede ser distinto para cada realización de ξ . De igual forma, se puede tener una realización ξ_1 para la que $z(x_1, \xi_1) < z(x_2, \xi_1)$ y otra realización ξ_2 para la cual $z(x_1, \xi_2) > z(x_2, \xi_2)$. Por lo tanto, en un problema de programación estocástica puede no existir un vector x que sea óptimo ni siquiera factible, para todas las realizaciones de los parámetros aleatorios [22].

En [22] se describen brevemente cuatro enfoques distintos para resolver PEG, los mismos que fueron planteados por Kall [29] y que mencionaremos a continuación.

En situaciones en las que no se conoce exactamente la distribución de probabilidad del vector de parámetros aleatorios y sólo puede suponerse que dicha función de distribución, P , pertenece a una cierta clase de funciones de distribución, \mathfrak{P} , se puede utilizar una estrategia propuesta en la *teoría de juegos*: elegir el punto de vista más pesimista, escogiendo aquella distribución $P \in \mathfrak{P}$, para la que el valor esperado del objetivo sea mayor [22]. Esto conduce a la formulación del siguiente problema estocástico:

$$\begin{aligned} \min_x \quad & \max_{P \in \mathfrak{P}} E[z(x, \xi)] \\ \text{s.r} \quad & g_i(x, \xi) \leq 0, \quad i = 1, 2, \dots, m \\ & x \in D. \end{aligned}$$

Otro enfoque para resolver los problemas estocásticos es *la eficiencia en distribución*, la cual se basa en la teoría clásica de la utilidad esperada de Von Newman y Morgenstern: existe una función de utilidad, u , tal que dadas dos variables aleatorias ξ y η , η es preferible o indiferente a ξ si y sólo si:

$$E[u(\eta)] \leq E[u(\xi)].$$

Esto, reduce el análisis de la eficiencia en distribución a fijar la función de utilidad y determinar la elección de las variables de decisión que maximiza el nivel de utilidad esperada:

$$\begin{array}{ll} \text{máx}_x & E[u(z(x, \xi))] \\ \text{s.r} & g_i(x, \xi) \leq 0, \quad i = 1, 2, \dots, m \\ \text{s.r} & x \in D \end{array}$$

La solución de este problema determinístico dependerá del tipo de preferencias establecidas y del tipo de función de utilidad del problema [22].

La resolución de problemas de programación estocástica mediante la *penalización de violación de restricciones* requiere de la transformación del problema estocástico en uno determinístico equivalente. Dicha transformación se realiza en base a las características del problema y a las preferencias del decisor. Para obtener el problema determinístico mediante el enfoque de penalización, se penaliza la posible violación del conjunto de restricciones del problema con la función $\mathcal{R}(x, \xi)$, que se define de la siguiente forma [22]:

$$\mathcal{R}(x, \xi) = \begin{cases} h(g_1(x, \xi), \dots, g_m(x, \xi)) & \text{si algún } g_i(x, \xi) > 0, i = 1, \dots, m. \\ 0 & \text{en otro caso,} \end{cases}$$

con $h : \mathbb{R}^m \rightarrow \mathbb{R}$.

La función que penaliza la violación de las restricciones, se interpreta como un coste extra o pérdida debida a la posible infactibilidad de la solución del problema. Así, el problema estocástico con penalización de las restricciones es:

$$\begin{array}{ll} \text{mín}_x & z(x, \xi) + \mathcal{R}(x, \xi) \\ \text{s.r} & x \in D \end{array}$$

El problema determinístico equivalente obtenido mediante el criterio de valor esperado será:

$$\begin{array}{ll} \text{mín}_x & E[z(x, \xi) + \mathcal{R}(x, \xi)] \\ \text{s.r} & x \in D \end{array}$$

El enfoque de la *programación con restricciones probabilísticas o de azar* transforma el problema estocástico en el problema determinístico equivalente. Para transformar las restricciones estocásticas en determinísticas se fija una probabilidad y se exige que se verifiquen las restricciones estocásticas con dicha probabilidad. Para la función objetivo se suele tomar su valor esperado [22].

En el *programa estocástico bietapa con recurso* se tiene un conjunto de decisiones que deben ser tomadas sin toda la información de algún evento aleatorio ω . Estas decisiones son llamadas las decisiones de la *primera etapa* y son usualmente representadas por un vector x . Después que toda la información es recibida acerca del vector aleatorio ξ se toman las decisiones de la *segunda etapa* y . Se suele usar la notación $\xi(\omega)$ para explicitar la dependencia del evento aleatorio ω . El programa estocástico con recurso puede formularse como:

$$\begin{aligned} \text{mín} \quad & c^T x + E_{\xi} Q(x, \xi) \\ \text{s.r} \quad & Ax = b, \\ & x \geq 0, \end{aligned} \tag{1.3}$$

donde

$$\begin{aligned} Q(x, \xi) = \text{mín} \quad & q(\omega)^T y \\ \text{s.r} \quad & Wy = h(\omega) - Z(\omega)x, \\ & y \geq 0, \end{aligned}$$

ξ es el vector aleatorio formado por las componentes de q^T , h^T y Z y E_{ξ} es la esperanza matemática respecto al vector aleatorio ξ . Además, W es llamada la *matriz de recurso*, que se asume constante y Z es una matriz aleatoria.

A la formulación (1.3) también se conoce como la *representación implícita del programa estocástico*. Una representación implícita más condensada es obtenida definiendo $Q(x) = E_{\xi} Q(x, \xi)$ como el *valor de la función* o *función de recurso*, así que (1.3) puede escribirse como:

$$\begin{aligned} \text{mín} \quad & c^T x + Q(x) \\ \text{s.r} \quad & Ax = b, \\ & x \geq 0. \end{aligned}$$

La elección de un método de resolución u otro depende de las características particulares del problema a resolver y, en gran medida, de las preferencias del decisor. Incluso, se puede considerar la posibilidad de combinar más de un criterio para poder recoger aspectos deseables de la solución del problema, pero, en general no es posible ordenar estos conceptos [22].

1.2. EJEMPLOS DE PROGRAMACIÓN ESTOCÁSTICA

A continuación se presentan ejemplos que describen de manera intuitiva las ideas fundamentales empleadas por la programación estocástica para modelar la incertidumbre. Estos ejemplos reflejan también algunos de los diferentes aspectos estructurales de los problemas de optimización estocástica.

1.2.1. EL PROBLEMA DEL GRANJERO

El ejemplo del granjero, introducido por Birge [1], es un problema de programación estocástica bietapa. Este ejemplo ilustra el fundamento básico de la programación estocástica y la ventaja de la solución estocástica sobre las aproximaciones determinísticas.

Un granjero cultiva maíz, trigo y remolacha azucarera en 500 hectáreas (has) de tierra. Durante el invierno, él quiere decidir cuánto de tierra asignar a cada cultivo en el próximo ciclo. El granjero necesita mínimo 200 toneladas (tons) de trigo y 240 tons de maíz para alimento del ganado. Estas cantidades pueden ser cultivadas en la granja o compradas a un mayorista. Cualquier exceso en la producción de alimento puede ser vendido, a precios de 170 y 150 dólares por tonelada para el trigo y el maíz respectivamente. Los precios de compra del trigo y el maíz son 238 y 210 dólares por tonelada respectivamente. Los costos de plantación son 150, 230 y 260 dólares por hectárea (ha) para el trigo, el maíz y la remolacha respectivamente. La remolacha azucarera es rentable, se vende a 36 dólares por tonelada, aunque cualquier producción mayor a 6000 tons puede ser vendida sólo a 10 dólares/ton. Se conoce que la producción media para cada cultivo es de 2,5, 3 y 20 toneladas por hectárea para el trigo, el maíz y la remolacha respectivamente. Sin embargo, el nivel de producción de cada cultivo depende de las condiciones del clima y puede variar alrededor del 20% bajo o sobre el valor medio, con iguales probabilidades de ocurrencia. En el Cuadro 1.1 se resumen los datos de entrada del problema.

Cultivo	trigo	maíz	remolacha
Producción media (ton/ha)	2,5	3	20
Costo plantación (\$/ha)	150	230	260
Precio venta (\$/ton)	170	150	36 bajo 6000 tons 10 sobre 6000 tons
Precio compra (\$/ton)	238	210	
Requerimiento (tons)	200	240	
Superficie disponible:	500 has		

Cuadro 1.1: Problema del granjero. Datos de entrada.

El problema de encontrar la asignación óptima de la superficie cultivable (PAS) puede formularse como el siguiente programa de optimización:

$$\begin{aligned} \text{mín} \quad & 150x_1 + 230x_2 + 260x_3 + 238y_1 - 170w_1 \\ & + 210y_2 - 150w_2 - 36w_3 - 10w_4 \end{aligned} \quad (1.4)$$

$$\text{s.r} \quad x_1 + x_2 + x_3 \leq 500 \quad (1.5)$$

$$z_1x_1 + y_1 - w_1 \geq 200 \quad (1.6)$$

$$\text{(PAS)} \quad z_2x_2 + y_2 - w_2 \geq 240 \quad (1.7)$$

$$w_3 + w_4 \leq z_3x_3 \quad (1.8)$$

$$w_3 \leq 6000 \quad (1.9)$$

$$x_1, x_2, x_3, y_1, y_2, w_1, w_2, w_3, w_4 \geq 0 \quad (1.10)$$

donde:

x_1 : hectáreas de tierra destinadas para el trigo

x_2 : hectáreas de tierra destinadas para el maíz

x_3 : hectáreas de tierra destinadas para la remolacha azucarera

w_1 : toneladas de trigo vendidas

y_1 : toneladas de trigo compradas

w_2 : toneladas de maíz vendidas

y_2 : toneladas de maíz compradas

w_3 : toneladas de azúcar de maíz vendidas a un precio favorable (36 dólares)

w_4 : toneladas de azúcar de maíz vendidas a un precio bajo (10 dólares)

z_1 : nivel de producción de trigo (ton/ha)

z_2 : nivel de producción maíz producidas (ton/ha)

z_3 : nivel de producción remolacha azucarera (ton/ha)

La ecuación (1.4) es la función de utilidad de los cultivos, cambiada de signo debido a que el problema se ha formulado como uno de minimización; (1.5) es la restricción de disponibilidad de superficie cultivable; (1.6) y (1.7) expresan que el saldo de la cantidad producida más la cantidad comprada y menos la cantidad vendida debe ser suficiente para satisfacer el requerimiento mínimo de trigo y maíz respectivamente; la restricción (1.8) indica que la cantidad total de remolacha azucarera vendida, ya sea a precio favorable o a precio bajo, no puede exceder a la cantidad producida; la restricción (1.9) indica que se puede vender hasta 6000 tons de remolacha azucarera a precio favorable y finalmente la restricción (1.10) reúne las restricciones de no negatividad.

Notar que este modelo no es lineal, debido a que las restricciones (1.6), (1.7) y (1.8) contienen términos de la forma $z_i x_i$.

Una manera muy simplificada de abordar la incertidumbre en la producción de los diferentes cultivos sería considerar por separado las posibilidades de que el próximo año sea bueno, promedio, o malo para la cosecha. De esta manera se obtienen tres escenarios:

- a. Escenario promedio: La producción media alcanza los valores medios dados en el Cuadro 1.1.
- b. Escenario pesimista: La producción está el 20% bajo la producción media, así $z_1 = 2$, $z_2 = 2,4$ y $z_3 = 16$.
- c. Escenario optimista: La producción está el 20% sobre la producción media, así $z_1 = 3$, $z_2 = 3,6$ y $z_3 = 24$.

Para cada uno de estos tres escenarios (PAS) puede resolverse como un programa lineal determinístico clásico. Las soluciones óptimas se muestran en el Cuadro 1.2.

En los tres escenarios el granjero debe asignar la suficiente superficie de tierra para alcanzar la cuota de remolacha de 6000 ton. Cuando la producción es promedio el debe asignar la superficie necesaria al maíz para obtener el requerimiento mínimo de alimento y el resto de superficie dedicarla al trigo que deja una mayor utilidad. Si la producción es alta, se requiere de asignaciones pequeñas de tierra para obtener los requerimientos mínimos de remolacha y maíz y el resto de superficie es asignado al trigo. Si la producción es baja se requiere grandes superficies

Escenario Promedio			
Cultivo	trigo	maíz	remolacha
Superficie	120	80	300
Producción	300	240	6000
Ventas	100	-	6000
Compras	-	-	-
Utilidad: 118600			
Escenario Pesimista			
Cultivo	trigo	maíz	remolacha
Superficie	100	25	375
Producción	200	60	6000
Ventas	-	-	6000
Compras	-	180	-
Utilidad: 59950			
Escenario Optimista			
Cultivo	trigo	maíz	remolacha
Superficie	183	67	250
Producción	550	240	6000
Ventas	350	-	6000
Compras	-	-	-
Utilidad: 167667			

Cuadro 1.2: Soluciones óptimas determinísticas para cada escenario.

de tierra para alcanzar los requerimientos mínimos y aún así el requerimiento de maíz no es satisfecho, por lo cual se deben comprar algunas toneladas de maíz. Analizando estos tres escenarios, se observa que la solución óptima es muy sensible a la variación de los niveles de producción por hectáreas. La superficie de trigo sembrada varía de 100 a 183 hectáreas, la superficie de maíz varía entre 25 y 87 hectáreas y la superficie de remolacha azucarera de 250 a 375 hectáreas. Desafortunadamente, no es posible predecir con exactitud el clima seis meses adelante y por lo tanto no se puede conocer si el año será bueno o malo para los diferentes cultivos.

Notar que la decisión más importante que debe realizarse es cuánto sembrar de remolacha. Si se planta demasiada superficie con remolacha, se deberá vender la remolacha que sobrepasa la cuota a un precio desfavorable. Si se planta una superficie muy pequeña se podría perder la oportunidad de vender un cultivo muy rentable.

El granjero desea tomar una decisión que sea la “mejor”, en promedio, para todas las circunstancias. Esta decisión no puede partir del supuesto de que ya se conoce cuál de los tres escenarios se presentará.

Una forma más adecuada para hacer frente a la incertidumbre en cuanto al clima del año próximo consiste en asignar a cada escenario una probabilidad, y buscar una solución que maximice la utilidad esperada sobre los tres escenarios.

Asumiendo que los tres escenarios tienen la misma probabilidad de ocurrir, el problema puede plantearse como se indica a continuación:

Representando con $s \in \{1, 2, 3\}$ a cada escenario (1=optimista, 2=medio, 3=pesimista) se redefinen las variables de decisión:

w_{is} : la cantidad de cultivo i vendida en el escenario s , donde 1 representa trigo, 2 representa maíz, 3 representa remolacha vendida a un precio favorable y 4 representa remolacha vendida a un precio desfavorable.

y_{js} : la cantidad de cultivo j comprado en el escenario s , para todo $j \in \{1, 2\}$.

Por ejemplo, w_{32} representa la cantidad de remolacha vendida a un precio favorable si la producción está en el promedio.

Se obtiene así el programa lineal.

$$\begin{aligned}
\text{mín} \quad & 150x_1 + 230x_2 + 260x_3 - \frac{1}{3}[170w_{11} - 238y_{11} \\
& + 150w_{21} - 210y_{21} + 36w_{31} + 10w_{41}] - \frac{1}{3}[170w_{12} - 238y_{12} \\
& + 150w_{22} - 210y_{22} + 36w_{32} + 10w_{42}] - \frac{1}{3}[170w_{13} - 238y_{13} \\
& + 150w_{23} - 210y_{23} + 36w_{33} + 10w_{43}]
\end{aligned} \tag{1.11}$$

$$\text{s.r} \quad x_1 + x_2 + x_3 \leq 500 \tag{1.12}$$

$$3x_1 + y_{11} - w_{11} \geq 200 \tag{1.13}$$

$$2,5x_1 + y_{12} - w_{12} \geq 200 \tag{1.14}$$

$$2x_1 + y_{13} - w_{13} \geq 200 \tag{1.15}$$

$$3,6x_2 + y_{21} - w_{21} \geq 240 \tag{1.16}$$

$$3x_{22} + y_{22} - w_{22} \geq 240 \tag{1.17}$$

$$2,4x_{23} + y_{23} - w_{23} \geq 240 \tag{1.18}$$

$$w_{31} + w_{41} \leq 24x_3 \tag{1.19}$$

$$w_{32} + w_{42} \leq 20x_{32} \tag{1.20}$$

$$w_{33} + w_{43} \leq 16x_{33} \tag{1.21}$$

$$w_{31}, w_{32}, w_{33} \leq 6000 \tag{1.22}$$

$$x_i, y_{ji}, w_{ki} \geq 0 \quad i = 1, 2, 3, \quad j = 1, 2, \quad k = 1, \dots, 4 \tag{1.23}$$

donde la ecuación (1.11) es la utilidad esperada de los cultivos cambiada de signo; (1.12) es la restricción de disponibilidad de superficie. Las restricciones (1.13)-(1.15) y (1.16)-(1.18) son las restricciones de requerimiento mínimo para el trigo y el maíz en cada uno de los tres posibles escenarios respectivamente. Las restricciones (1.19)-(1.21) indican que en cualquiera de los tres escenarios la cantidad de la remolacha azucarera vendida, ya sea a precio favorable o a precio bajo no puede exceder la cantidad producida. La restricción (1.22) indica que en cada escenario se pueden vender hasta 6000 ton. de remolacha azucarera a precio favorable. Finalmente, (1.23) contiene las restricciones de no negatividad.

El modelo anterior se conoce como *forma extensa* de un programa estocástico, porque el mismo contiene variables de decisión para todos los escenarios de producción de los cultivos. El Cuadro 1.3 indica la solución óptima de este problema.

La superficie óptima a destinar al cultivo de remolacha es aquella que siempre evite ventas a un precio desfavorable, aunque esto implique que no se llegue al tope de la cuota (para precios favorables) cuando la producción está bajo o en el

	Cultivo	trigo	maíz	remolacha
Escenario	Superficie	170	80	250
s=1 Optimista	Producción	510	288	6000
	Ventas	310	48	6000
	Compras	-	-	-
s=2 Promedio	Producción	425	240	5000
	Ventas	225	-	5000
	Compras	-	-	-
s=3 Pesimista	Producción	340	192	4000
	Ventas	140	-	4000
	Compras	-	48	-
Utilidad:	108390			

Cuadro 1.3: Solución óptima empleando la forma extensa del modelo estocástico

promedio. El área óptima para el maíz cumple exactamente con los requerimientos de alimento cuando la producción alcanza el nivel medio, lo que implica que si la producción está bajo o sobre el promedio deben realizarse compras o ventas, respectivamente. El resto de superficie se destina al trigo, esta área cubre los requerimientos mínimos y produce un excedente para la venta en todos los escenarios.

Las decisiones óptimas en un modelo estocástico corresponden a soluciones que están balanceadas o equilibradas frente a todos los escenarios. Las características de este equilibrio tienen un impacto importante sobre la utilidad esperada. Supongamos que la variación de la producción es cíclica, es decir que, un año bueno es seguido por un año promedio y luego un año malo. Si el granjero conociese de antemano el orden en el que se presentarían estos tres escenarios, entonces él podría tomar las decisiones de acuerdo al Cuadro 1.2. Así, obtendría una utilidad de \$167.667 para el primer año, de \$118.600 para el segundo año y \$59.950 para el tercer año. La utilidad media de los tres años sería de \$115.406. Este valor se conoce como solución óptima bajo la situación de *información perfecta*.

Desafortunadamente, el granjero no puede conocer a priori el clima de los siguientes años, de manera que la mejor decisión que puede tomar está dada por

la solución del modelo estocástico descrita en el Cuadro 1.3 y corresponde a una utilidad de \$108.390. La diferencia entre la utilidad obtenida bajo incertidumbre y la utilidad bajo la situación de información perfecta es $115.406 - 108.390 = \$7.016$, cantidad que se conoce como valor esperado de la información perfecta, (*EVPI, the expected value of perfect information*), y que representa la utilidad perdida debido a la presencia de incertidumbre.

Otra aproximación al problema del granjero podría consistir en asumir simplemente que la producción alcanzará su valor medio y resolver el programa lineal determinístico correspondiente. Una solución calculada de tal manera se conoce como *la solución del valor esperado*. En el Cuadro 1.4 se describen las decisiones de compra y venta de los cultivos para los diferentes escenarios, cuando las asignaciones de superficie para los cultivos han sido tomadas por el escenario promedio del Cuadro 1.2. Las decisiones tomadas con esta aproximación tienen

	Cultivo	trigo	maíz	remolacha
Escenario	Superficie	120	80	300
s=1	Producción	360	288	7200
Optimista	Ventas	160	48	6000
				(precio favorable)
				1200
				(precio no favorable)
	Compras	-	-	-
s=2	Producción	300	240	6000
Promedio	Ventas	100	-	6000
	Compras	-	-	-
s=3	Producción	240	192	4800
Pesimista	Ventas	40	-	4800
	Compras	-	48	-
Utilidad:	107240			

Cuadro 1.4: Solución del valor esperado.

consecuencias desfavorables, ya que consideran sólo un escenario. Por ejemplo, si la producción resulta ser alta el granjero deberá vender la remolacha sobrante de las 6000 ton a un precio desfavorable, lo que conlleva pérdidas en la utilidad. La

diferencia entre la utilidad obtenida con el valor esperado y la utilidad del modelo estocástico, $108.309-107.240=\$1.150$, es la pérdida en la utilidad por no considerar la incertidumbre en la producción de los cultivos. Este valor es llamado *valor de la solución estocástica (VSS, value of the stochastic solution)*, y representa la ganancia de emplear un enfoque estocástico durante la optimización.

El EVPI mide el valor de conocer el futuro con certeza, mientras que el VSS mide el valor de conocer y emplear distribuciones de probabilidad sobre parámetros sujetos a incertidumbre.

A continuación se ilustra la formulación general del *problema estocástico bietapa con recurso* para el problema de asignación (PAS). En la Sección 1.3 se analizarán con más detalle los problemas bietapa.

En el ejemplo del granjero se tiene una *decisión de la primera etapa* (la asignación de superficie de tierra a cultivos), la cual debe tomarse sin la información completa de ciertos eventos aleatorios futuros (el clima). Además, están las *decisiones de la segunda etapa o acciones correctivas*, que se toman después que la información de la producción de cada cultivo es recibida y son las decisiones respecto a cuánto vender y comprar de cada cultivo.

El problema de la segunda etapa para un escenario s en particular puede ser escrito como:

$$\begin{aligned}
 Q(x, s) = \text{mín} \quad & 238y_1 - 170w_1 + 210y_2 - 150w_2 - 36w_3 - 10w_4 \\
 \text{s.r} \quad & z_1(s)x_1 + y_1 - w_1 \geq 200 \\
 & z_2(s)x_2 + y_2 - w_2 \geq 240 \\
 & w_3 + w_4 \leq z_3(s)x_3 \\
 & w_3 \leq 6000 \\
 & y_j, w_k \geq 0, \quad j = 1, 2, k = 1, \dots, 4
 \end{aligned}
 \tag{1.24}$$

donde $z_i(s)$ representa la producción del cultivo i en el escenario s . El vector aleatorio, ξ , formado por los niveles de producción de los cultivos, sólo puede tomar tres diferentes valores, $\xi_1 = (z_1(1), z_2(1), z_3(1))$, $\xi_2 = (z_1(2), z_2(2), z_3(2))$ y $\xi_3 = (z_1(3), z_2(3), z_3(3))$ que representan a los tres escenarios. Definiendo $Q(x) = E_{\xi}Q(x, \xi)$ como la *función de recurso* se puede escribir el problema extenso como:

$$\begin{aligned}
\text{mín} \quad & 150x_1 + 230x_2 + 260x_3 + Q(x) \\
\text{s.r} \quad & x_1 + x_2 + x_3 \leq 500 \\
& x \geq 0,
\end{aligned}
\tag{1.25}$$

Esta forma se conoce como *modelo estocástico bietapa con recurso* y es una representación implícita del programa estocástico extenso.

1.2.2. EL PROBLEMA DEL VENDEDOR DE PERIÓDICO

Este es un ejemplo clásico de *programas estocásticos lineales bietapas con recurso*, que fue introducido por Birge [1].

Un vendedor de periódico compra cada mañana en la oficina de publicidad la cantidad de x periódicos a un precio c , usualmente él no puede comprar más de u unidades. Él vende los periódicos a un precio q recorriendo las calles de la ciudad. Los periódicos sobrantes son regresados a la oficina de publicidad a un precio r , inferior al precio de compra.

El vendedor cada mañana debe decidir cuantos periódicos comprar para maximizar su utilidad conociendo que la demanda varía cada día y está descrita por una variable aleatoria ξ .

Definiendo y como las ventas efectivas y w como el número de periódicos regresados a la oficina de publicidad al final del día, se formula el problema bietapa con recurso como:

$$\begin{aligned}
\text{mín} \quad & cx + Q(x) \\
\text{s.r} \quad & 0 \leq x \leq u,
\end{aligned}
\tag{1.26}$$

donde $Q(x) = E_{\xi}Q(x, \xi)$ y

$$Q(x, \xi) = \text{mín} \quad -qy(\xi) - rw(\xi) \tag{1.27}$$

$$\text{s.r} \quad y(\xi) \leq \xi, \tag{1.28}$$

$$y(\xi) + w(\xi) \leq x, \tag{1.29}$$

$$y(\xi), w(\xi) \geq 0, \tag{1.30}$$

donde E_{ξ} es la esperanza matemática con respecto a la demanda ξ ; la restricción (1.28) indica que la cantidad de periódicos vendida debe ser menor o igual a la cantidad de demanda y la restricción (1.29) indica que la cantidad de periódicos vendida más la regresada a la oficina debe ser igual a la cantidad de periódicos comprados. La función $-Q(x)$ es la ganancia esperada en ventas y retornos de periódicos, $-Q(x, \xi)$ es la ganancia si la demanda está en el nivel ξ . Se usa el signo negativo porque se ha formulado un problema de minimización.

Las decisiones de cuánto comprar deben ser tomadas antes de conocer la demanda, estas son las decisiones de la primera etapa. Cuando la demanda es conocida en la segunda etapa, se puede obtener la utilidad mediante la siguiente regla:

$$\begin{aligned} y^*(\xi) &= \min(\xi, x), \\ w^*(\xi) &= \max(x - \xi, 0). \end{aligned}$$

Es decir, la cantidad óptima de periódicos vendida es el mínimo entre la cantidad comprada y la demandada. La cantidad óptima de periódicos regresados es el sobrante de los periódicos vendidos. Esto es debido a que las ventas nunca pueden ser mayor al número de periódicos disponibles ni a la demanda y los retornos ocurren únicamente cuando la demanda es menor que el número de periódicos disponibles. Así, la función de esperanza para la segunda etapa es:

$$Q(x) = E_{\xi}[-q \min(\xi, x) - r \max(x - \xi, 0)]$$

$Q(x)$ puede ser calculada como:

$$\begin{aligned} Q(x) &= \int_{-\infty}^x (-q\xi - r(x - \xi))dF(\xi) + \int_x^{-\infty} -qx dF(\xi) \\ &= -(q - r) \int_{-\infty}^x \xi dF(\xi) - rx F(x) - qx(1 - F(x)) \end{aligned}$$

donde $F(\xi)$ representa la función de distribución acumulada de ξ . Integrando por partes se tiene:

$$\int_{-\infty}^x \xi dF(\xi) = xF(x) - \int_{-\infty}^x F(\xi)d\xi$$

de donde se concluye que,

$$Q(x) = -qx + (q - r) \int_{-\infty}^x F(\xi)d\xi$$

Para minimizar (1.26) se debe derivar e igual a cero: $(cx + Q(x))' = 0$. Si $Q(x)$ es continua y convexa entonces también es diferenciable donde ξ es continuo [1]. La derivada será:

$$Q'(x) = -q + (q - r)F(x).$$

Por lo tanto:

$$x - q + (q - r)F(x) = 0,$$

entonces $x = F^{-1}\left(\frac{q-c}{q-r}\right)$. En la frontera ($0 \leq x \leq u$), así $F(0) \leq \frac{q-c}{q-r} \leq F(u)$. La solución óptima será:

$$\begin{aligned} x^* &= 0, & \text{si } \frac{q-c}{q-r} < F(0), \\ x^* &= u, & \text{si } \frac{q-c}{q-r} > F(u), \\ x^* &= F^{-1}\left(\frac{q-c}{q-r}\right) & \text{caso contrario} \end{aligned}$$

donde $F^{-1}(\alpha)$ es el α -percentil de F . Si F es continua, $x = F^{-1}(\alpha)$ significa $\alpha = F(x)$. Ninguna representación razonable de la demanda debería implicar $F(0) = 0$, así que la solución nunca será $x^* = 0$. Notar que se requiere el conocimiento de la función de distribución de la demanda para poder calcular el valor numérico de x^* .

1.2.3. PLANEACIÓN FINANCIERA Y CONTROL

La esencia de la planeación financiera es la incorporación del riesgo dentro de las decisiones de inversión; debido a esto es una de las áreas donde más se aplica la programación estocástica.

En esta clase de problemas, las decisiones ocurren en diferentes puntos en el tiempo, así que estos problemas pueden ser vistos como tener múltiples etapas de observación y acción. Una característica importante es que las decisiones en cualquier etapa son altamente dependientes de los resultados pasados, y de las decisiones tomadas en etapas anteriores.

El siguiente problema, descrito por Birge [1], pertenece a los problemas de planeación financiera.

Se quiere ahorrar durante Y años para financiar la colegiatura de un niño, para lo cual se invertirá una cantidad b de dinero en un conjunto de fondos de inversión I . Después de los Y años se tendrá una fortuna que se esperaría que exceda al valor de la colegiatura G . Se supone que se puede rebalancear las inversiones en el conjunto de fondos cada v años, así que se tiene $H = Y/v$ períodos de inversión o decisión.

Si el monto acumulado excede la cantidad G después de los Y años se tiene un porcentaje de ingreso q sobre la cantidad excedida, si no se reúne el valor de la

colegiatura, se puede pedir la diferencia a un costo porcentual r . Para definir este exceso o déficit en el ahorro después de los Y años se utilizan las variables y y z respectivamente.

El problema de financiar la colegiatura (FIC) se puede plantear como el siguiente programa de optimización:

$$\text{máx } qy - rw \quad (1.31)$$

$$\text{s.r } \sum_i x(i, 1) = b, \quad (1.32)$$

$$\text{(FIC) } \sum_i -\xi(i, t-1)x(i, t-1) + \sum_i x(i, t) = 0, \quad \forall t = 2, \dots, H \quad (1.33)$$

$$\sum_i \xi(i, H)x(i, H) - y + w = G, \quad (1.34)$$

$$x(i, t) \geq 0, \quad \forall 1 \leq i \leq I, \quad \forall 1 \leq t \leq H, \quad (1.35)$$

donde $\xi(i, t)$ es el rendimiento del fondo i en el período t y $x(i, t)$ es la cantidad de dinero invertida en el fondo i en el período t .

La ecuación (1.31) representa el exceso o déficit en alcanzar el valor de la colegiatura una vez transcurridos los H períodos de inversión. Es decir, la cantidad ganada o pedida al final del horizonte de planificación. (1.32) indica que todo el dinero inicial debe ser invertido en los diferentes fondos. (1.33) indica que todo el dinero generado por los fondos en un período debe ser nuevamente invertido en el período siguiente. La restricción (1.34) indica que al final del horizonte de planificación se debe satisfacer el valor de la colegiatura.

Como el rendimiento de los fondos en todos los períodos de inversión es aleatorio, los distintos escenarios en los que deben tomarse las decisiones $x(i, t)$ también lo son. Así, el exceso o déficit en la colegiatura al final de los Y años es una variable aleatoria.

Las decisiones que se toman en cada período son conocidas en la programación estocástica como *no anticipadas*, ya que dependen exclusivamente de la información disponible en ese período, no pueden utilizar información futura [31].

A continuación se realiza el planteamiento del problema, para un caso sencillo con únicamente dos tipos de fondos de inversión: ventas $i = 1$ y bonos $i = 2$, para un horizonte de quince años, y revisión de de inversiones cada cinco años, es decir, se tienen tres períodos de inversión.

Se asume además, que sólo se tienen dos posibles conjuntos de retornos: el primer conjunto de retornos es 1,25 para ventas y 1,14 para bonos y el segundo conjunto de retornos es 1,06 para ventas y 1,12 para bonos. Existen en total ocho posibles *escenarios*, $s = 1, 2, \dots, 8$, cuyas probabilidades supondremos iguales, $p(s) = 0,125$.

Notando con el número 1 al primer conjunto de retornos y con 2 al segundo conjunto de retornos, se construye el *árbol de escenarios o de probabilidad* descrito en la Figura 1.1.

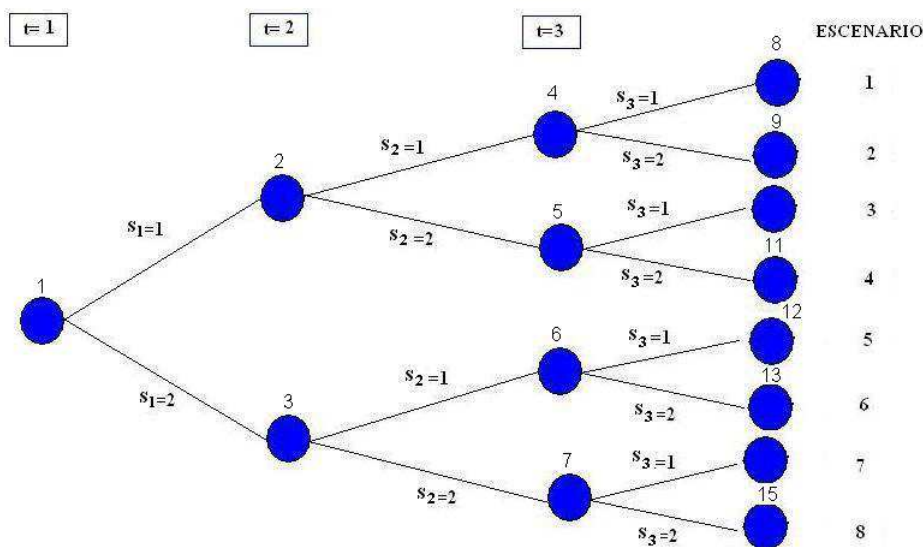


Figura 1.1: Árbol de escenarios para dos conjuntos de retornos y tres períodos de inversión.

Un escenario es cualquier trayectoria que va desde la raíz a las hojas del árbol. Los escenarios que comparten una misma historia hasta cierto período comparten una misma parte del árbol [31]. Por ejemplo, en el árbol de la Figura 1.1 la raíz es única, lo que implica que las decisiones del primer período son las mismas para cualquier escenario, en el segundo período se tienen dos nodos, que representan dos escenarios posibles para la toma de decisiones, una vez conocido el conjunto de retornos del primer período. En el tercer período se tienen cuatro nodos, lo que indica que se tienen múltiples escenarios.

El árbol de probabilidad es la forma explícita de representar la no anticipatividad de las decisiones [31].

Cada escenario s , encierra la historia de los retornos de los fondos en los períodos y puede ser descrito por un vector (s_1, \dots, s_t) , donde s_t indica el conjunto

de retornos ocurrido en el período t . Por ejemplo, el sexto escenario, $s = (2, 1, 2)$, que representa el camino desde la raíz hasta la hoja trece, significa que en el primer período se obtuvo el segundo conjunto de retornos (1,06 para ventas y 1,12 para bonos), en el segundo período de inversión se obtuvo el primer conjunto de retornos (1,25 para ventas y 1,14 para bonos) y en el último período se obtuvo nuevamente el segundo conjunto retornos.

Es necesario reformular el modelo FIC para incorporar en el mismo la estructura de árbol de escenarios descrita en los últimos párrafos. Las decisiones para el período, $t = 1$, son la cantidad de dinero a invertir en ventas, $x(1, 1)$, y bonos, $x(2, 1)$. Para $t = 2$, existen cuatro variables de decisión de la forma $x(i, 2, s_1)$ con $i \in \{1, 2\}$, y $s_1 \in \{1, 2\}$. Esto quiere decir que la cantidad invertida en el segundo período depende del retorno de las inversiones realizadas en el primer período. Igualmente, para $t = 3$, las variables de decisión $x(i, 3, s_1, s_2)$ dependen de las decisiones tomadas en los dos períodos anteriores. Finalmente, las variables de exceso o déficit también dependen del escenario o historia, y se tendrán en total ocho variables de la forma $y(s_1, s_2, s_3)$ y $z(s_1, s_2, s_3)$, con $s_i \in \{1, 2\}$ para $1 \leq i \leq 3$. Considerando una inversión inicial $b = 55$ un costo de colegiatura $G = 80$ (miles de dólares), una tasa de interés sobre el excedente $q = 1$, una tasa de interés sobre el faltante $r = 4$, se obtiene el siguiente modelo estocástico de planeación financiera EFIC:

$$\begin{aligned}
 & \text{máx} \quad \sum_{s_1=1}^2 \sum_{s_2=1}^2 \sum_{s_3=1}^2 0,125(y(s_1, s_2, s_3) - 4w(s_1, s_2, s_3)) \\
 & \text{s.r} \\
 & x(1, 1) + x(2, 1) = 55 \\
 & -1,25x(1, 1) - 1,14x(2, 1) + x(1, 2, 1) + x(2, 2, 1) = 0 \\
 & -1,06x(1, 1) - 1,12x(2, 1) + x(1, 2, 2) + x(2, 2, 2) = 0 \\
 & -1,25x(1, 2, 1) - 1,14x(2, 2, 1) + x(1, 3, 1, 1) + x(2, 3, 1, 1) = 0 \\
 & -1,06x(1, 2, 1) - 1,12x(2, 2, 1) + x(1, 3, 1, 2) + x(2, 3, 1, 2) = 0 \\
 & -1,25x(1, 2, 2) - 1,14x(2, 2, 2) + x(1, 3, 2, 1) + x(2, 3, 2, 1) = 0 \\
 & -1,06x(1, 2, 2) - 1,12x(2, 2, 2) + x(1, 3, 2, 2) + x(2, 3, 2, 2) = 0 \\
 & 1,25x(1, 3, 1, 1) + 1,14x(2, 3, 1, 1) - y(1, 1, 1) + w(1, 1, 1) = 80 \\
 & 1,06x(1, 3, 1, 1) + 1,12x(2, 3, 1, 1) - y(1, 1, 2) + w(1, 1, 2) = 80 \\
 & \text{(EFIC)} \quad 1,25x(1, 3, 1, 2) + 1,14x(2, 3, 1, 2) - y(1, 2, 1) + w(1, 2, 1) = 80 \\
 & \quad \quad \quad 1,06x(1, 3, 1, 2) + 1,12x(2, 3, 1, 2) - y(1, 2, 2) + w(1, 2, 2) = 80
 \end{aligned}$$

$$\begin{aligned}
1,25x(1, 3, 2, 1) + 1,14x(2, 3, 2, 1) - y(2, 1, 1) + w(2, 1, 1) &= 80 \\
1,06x(1, 3, 2, 1) + 1,14x(2, 3, 2, 1) - y(2, 1, 2) + w(2, 1, 2) &= 80 \\
1,25x(1, 3, 2, 2) + 1,14x(2, 3, 2, 2) - y(2, 2, 1) + w(2, 2, 1) &= 80 \\
1,06x(1, 3, 2, 2) + 1,12x(2, 3, 2, 2) - y(2, 2, 2) + w(2, 2, 2) &= 80 \\
x(i, t, s_1, \dots, s_{t-1}) \geq 0, \quad y(s_1, s_2, s_3) \geq 0, \\
w(s_1, s_2, s_3) \geq 0 \quad \forall t, s_1, s_2, s_3.
\end{aligned}$$

En el Cuadro 1.5 se presentan decisiones óptimas encontradas al resolver EFIC. La utilidad esperada de $-1,52$, suele llamarse el valor *RP* (*recourse pro-*

Período	Escenario	Ventas	Bonos
1	1-8	41,5	13,50
2	1-4	65,1	2,17
	5-8	36,7	22,4
3	1-2	83,0	0,00
	3-4	0,0	71,40
	5-6	0,0	71,40
	7-8	64,0	0,00
	Escenario	Superior a G	Inferior a G
	1	24,80	0,00
	2	8,87	0,00
	3	1,42	0,00
	4	0,00	0,00
	5	1,42	0,00
	6	0,00	0,00
	7	0,00	0,00
	8	0,00	12,2
Utilidad esperada $-1,52$ millones de dólares			

Cuadro 1.5: Solución óptima del modelo estocástico para 3 períodos

blem). La inversión inicial tiene más peso en ventas (\$41,5) que en bonos (\$13,5). En el segundo período la inversión en ventas es más destacada desde el primer al cuarto escenario, mientras que desde el quinto al octavo escenario la inversión

en ventas es más conservadora. En el último período las inversiones son completamente en ventas o en bonos.

Si se resuelve FIC suponiendo que los retornos toman sus valores esperados para cada período, el rendimiento para ventas y bonos será siempre \$1,15 y \$1,13, respectivamente. La solución óptima de este modelo determinístico sólo invierte en ventas en cada uno de los tres períodos y la utilidad obtenida es $-3,79$, la cual es llamada la solución *EV* (*expected value*). La diferencia entre los valores *RP* y *EV* es el valor de la solución estocástica: $VSS = RP - EV = -1,52 - (-3,79) = 2,27$. Este valor mide la ventaja de utilizar un modelo estocástico frente a un modelo determinístico.

En un caso general, definidas las variables de decisión como: $x(i, t, s_1, \dots, s_{t-1})$, $\forall t = 1, \dots, H$; $x(i, 1)$, $\forall i \in I$; $y(s_1, \dots, s_H)$ y $z(s_1, \dots, s_H)$ se puede formular el programa estocástico general que maximiza la utilidad esperada como se indica a continuación.

La función objetivo mide la esperanza, respecto a todos los posibles escenarios, del monto acumulado al final del período H :

$$\sum_{s_H} \dots \sum_{s_1} p(s_1, \dots, s_H) (qy(s_1, \dots, s_H) - rz(s_1, \dots, s_H))$$

La restricción en el primer período indica que la cantidad total invertida en los fondos debe ser igual a la cantidad de dinero inicial b :

$$\sum_i x(i, 1) = b$$

La condición de que todo el dinero obtenido al final de cada período debe invertirse en el período siguiente se expresa como una familia de restricciones para cada $t = 2, \dots, H$ y para cada historia (s_1, \dots, s_{t-1}) :

$$\sum_i -\xi(i, t-1, s_1, \dots, s_{t-1}) x(i, t-1, s_1, \dots, s_{t-2}) + \sum_i x(i, t-1, s_1, \dots, s_{t-1}) = 0,$$

Finalmente, las restricciones para el final del período H dependen de cada escenario (s_1, \dots, s_H) :

$$\sum_i \xi(i, H, s_1, \dots, s_H) x(i, H, s_1, \dots, s_{H-1}) - y(s_1, s_2, s_3) + z(s_1, s_2, s_3) = G.$$

Un gran inconveniente que limita la aplicabilidad de los modelos multietapa es la elevada cantidad de variables requeridas, aún para horizontes de tiempo cortos.

Por ejemplo, si consideramos en el caso anterior un problema con 20 períodos y tres alternativas para las tasa de retorno en cada período tendremos 3^{t-1} variables de decisión para los montos de inversión en el período $t \in \{1, \dots, 20\}$, 3^{20} variables para los posibles excedentes finales en cada escenario y 3^{20} variables para los posibles faltantes. Esto da como resultado un programa lineal con un total de

$$\sum_{t=1}^{20} 3^{t-1} + 2 \cdot 3^{20} \approx 8,72 \text{ mil millones de variables,}$$

que está muy lejos de lo que puede calcularse con los recursos computacionales actuales.

1.3. PROGRAMA ESTOCÁSTICO BIETAPA

Los dos primeros ejemplos de la sección anterior describen un mecanismo común utilizado en la programación estocástica para modelar ciertos procesos sujetos a incertidumbre: los *programas con recurso bietapa*. En los mismos, una decisión debe ser tomada cuya factibilidad o costo dependen del valor de ciertos parámetros asociados a un suceso aleatorio futuro. Se busca aquella solución que en promedio alcance un costo mínimo.

En los problemas bietapa con recurso el conjunto de decisiones está dividido en dos grupos [1].

- Decisiones de la primera etapa. Son todas las decisiones que se toman antes de que el suceso aleatorio tenga lugar.
- Decisiones de la segunda etapa. Son las decisiones que se toman después de conocido el resultado del suceso aleatorio, es decir, una vez que se conoce el valor tomado por el vector aleatorio. También son llamadas *acciones correctivas*.

Así, en el problema del granjero debe asignarse la tierra a los cultivos sin el conocimiento del clima para el próximo año. Por otra parte, para cada uno de los tres escenarios posibles, hay variables en el modelo que indican cuánto debe comprarse o venderse de cada producto, de manera que la utilidad *en ese escenario* sea máxima.

La utilidad (o el costo) de cada escenario depende tanto de las variables de decisión originales, como de las variables correctivas asociadas al mismo. El objetivo es optimizar la esperanza de la utilidad sobre todos los escenarios.

Un programa con recurso se denomina *completo* cuando para cualquier decisión de la primera etapa (independientemente de su factibilidad) existen decisiones factibles en la segunda etapa; si esto se cumple sólo para las decisiones factibles de la primera etapa se denomina *relativamente completo*; y se llama con recurso *parcial* cuando no siempre las decisiones factibles de la primera etapa tienen asociadas decisiones factibles en la segunda etapa [31].

Representamos en adelante la vector de variables de decisión de la primera etapa x , mientras que las decisiones de la segunda etapa estarán representadas por el vector y o $y(\omega, x)$ para clarificar que las decisiones de la segunda etapa son tomadas en función de los resultados del suceso aleatorio ω y de las decisiones tomadas en la primera etapa. La secuencia de eventos y decisiones se resume como:

$$x \rightarrow \xi(\omega) \rightarrow y(\omega, x)$$

Primero se toma una decisión x , conocido el suceso aleatorio se tiene los valores concretos para los parámetros $\xi(\omega)$, y se toman las decisiones de la segunda etapa, $y(\omega, x)$.

El objetivo del modelo general bietapa es encontrar el valor de x inicial que minimiza la suma de los costos de todos los escenarios, ponderada por sus probabilidades de ocurrencia.

El programa lineal bietapa fue formulado originalmente por Datzig [2] y Beale [4], como:

$$\begin{aligned} \text{mín} \quad & z = c^T x + E_{\xi}[\text{mín } q(\omega)^T y(\omega, x)] \\ \text{s.r} \quad & Ax = b, \\ & Z(\omega)x + Wy(\omega, x) = h(\omega), \quad \forall \omega \in \Omega \\ & x \geq 0, \quad y(\omega, x) \geq 0. \end{aligned} \tag{1.36}$$

donde la matriz A de dimensión $m_1 \times n_1$ y los vectores $c \in R^{n_1}$ y $b \in R^{m_1}$ pertenecen a la primera etapa y se suponen conocidos. En la segunda etapa, el vector de parámetros aleatorios ω puede tomar un valor específico dentro de un espacio muestral Ω . Para cada realización ω dada, los datos del problema $q(\omega) \in R^{n_2}$, $h(\omega) \in R^{m_2}$ y $Z(\omega)$ de dimensión $m_2 \times n_1$ son conocidos. La matriz W , perteneciente a la segunda etapa, de dimensión $m_2 \times n_2$, es llamada *matriz de*

recurso.

En los modelos bietapa están claramente diferenciadas la parte aleatoria y la parte determinística del problema. La parte aleatoria está compuesta por el vector $\xi^T(\omega) = (q(\omega)^T, h(\omega)^T, Z_1(\omega), \dots, Z_{m_2}(\omega))$ de dimensión $N = n_2 + m_2 + (m_2 \times n_1)$, cuyo valor depende, del resultado del suceso aleatorio.

La dependencia de y en ω es de diferente naturaleza, simplemente indica que las decisiones correctivas no son las mismas bajo diferentes realizaciones de ω . En el modelo habrá una variable distinta $y(\omega, x)$ para cada realización del fenómeno aleatorio [1].

La expresión de la segunda etapa es la más difícil porque para cada ω , el valor de $y(\omega, x)$ es la solución de un programa matemático determinístico. Así, para una realización ω dada se tiene el programa determinístico:

$$Q(x, \xi(\omega)) = \min_{y(\omega, x)} \{q(\omega)^T y(\omega, x) \mid W y(\omega, x) = h(\omega) - Z(\omega)x, y(\omega, x) \geq 0\} \quad (1.37)$$

que es el valor óptimo de la función para un vector x y un escenario ω particulares. El valor de la esperanza de la función objetivo para la segunda etapa se denota como:

$$Q(x) = E_{\xi}[Q(x, \xi(\omega))] \quad (1.38)$$

Así, el *programa determinístico equivalente* de un problema bietapa es:

$$\begin{aligned} \min \quad & z = c^T x + Q(x) \\ \text{s.r} \quad & Ax = b, \\ & x \geq 0. \end{aligned} \quad (1.39)$$

Notar que la mayor diferencia en la formulación de un problema determinístico y un problema estocástico bietapa radica en la expresión de la función de costo para la segunda etapa. Si la función $Q(x)$ fuera dada explícitamente, entonces este programa se convirtiera en un programa (no lineal) ordinario.

Esta representación ilustra además claramente la secuencia en la que ocurren los eventos en los problemas de recurso. Las decisiones de la primera etapa son tomadas en presencia de incertidumbre acerca de las futuras realizaciones de ξ . En la segunda etapa, el valor actual de ξ será conocido y algunas acciones correctivas o decisiones de recurso deberán ser tomadas. Sin embargo, las decisiones de la primera etapa se toman considerando sus efectos futuros. Estos efectos futuros

son medidos por el valor de la función o función de recurso, $Q(x)$, la cual calcula el valor esperado de tomar la decisión x [1]. Claramente, se observa que la dificultad del programa estocástico bietapa está en la evaluación de $Q(x)$.

1.4. PROGRAMA ESTOCÁSTICO MULTIETAPA

Muchos problemas de optimización y de planeación involucran sucesiones de decisiones en el tiempo. Estas decisiones pueden depender de los resultados de eventos aleatorios que no se conocen a priori, así como de decisiones tomadas en etapas anteriores. Se obtiene así un árbol de escenarios, donde cada escenario es un camino desde la raíz hasta una hoja, que corresponda a una selección específica de decisiones para todas las etapas del proceso, así como una realización específica de todos los eventos aleatorios.

En la Sección 1.2.3 fue presentado un ejemplo de un programa estocástico multietapa. En ese caso, la composición del fondo de inversiones era la decisión a tomar en cada etapa del proceso y las tasas de retorno de las distintas componentes del fondo en cada período correspondían a los eventos aleatorios.

Al igual que ocurre con los programas bietapa, los programas estocásticos multietapa con un número finito de posibles escenarios futuros tienen un programa lineal determinístico equivalente aunque el número de variables suele crecer explosivamente con el número de etapas (o períodos de decisión) del proceso [1].

Un problema de programación estocástica multietapa tiene la forma siguiente:

$$\begin{aligned}
 \text{mín} \quad & c_1^T x^1 + E_{\xi^1}[\text{mín } c_2(\omega^1)^T x^2(\omega^1) + \dots + E_{\xi^{H-1}}[\text{mín } c_H(\omega^{H-1})^T x^H(\omega^1, \dots, \omega^{H-1})] \dots] \\
 \text{s.r} \quad & W^1 x^1 = h^1, \\
 & Z^1(\omega^1) x^1 + W^2 x^2(\omega^1) = h^2(\omega^1), \quad \forall \omega^1 \in \Omega_1 \\
 & \vdots \\
 & Z^{H-1}(\omega^{H-1}) x^{H-1}(\omega^1, \dots, \omega^{H-2}) + W^H x^H(\omega^1, \dots, \omega^{H-1}) = h^H(\omega^{H-1}), \\
 & \quad \forall (\omega^1, \dots, \omega^{H-1}) \in \Omega_1 \times \dots \times \Omega_{H-1} \\
 & x^1 \geq 0; \quad x^t(\omega^1, \dots, \omega^{t-1}) \geq 0, t = 2, \dots, H,
 \end{aligned} \tag{1.40}$$

donde c_1 es un vector conocido en \mathbb{R}^{n_1} , h^1 es un vector conocido de \mathbb{R}^{m_1} , $\xi^t(\omega^t)^T =$

$(c_{t+1}(\omega^t)^T, h^{t+1}(\omega^t)^T, Z_1^t(\omega^t), \dots, Z_{m_t}^t(\omega^t))$ es un vector de dimensión N_t definido para todo $t = 2, \dots, H - 1$, en función del suceso aleatorio $\omega^t \in \Omega_t$; cada W^t es una matriz conocida de dimensión $m_t \times n_t$, y las variables de la t -ésima etapa están indexadas por el resultado de los eventos aleatorios ocurridos en etapas anteriores.

El problema determinístico equivalente puede formularse en términos de un programa dinámico para H períodos. Suponiendo que $x^t(\omega^1, \dots, \omega^{t-1})$ representa las decisiones a tomar al iniciar el período t :

$$Q^H(x^{H-1}(\omega^1, \dots, \omega^{H-2}), \xi^H(\omega^{H-1})) = \text{mín} \quad c_H(\omega^{H-1})^T x^H$$

s.r

$$W^H x^H = h^H(\omega^{H-1}) - Z^{H-1}(\omega^{H-1})x^{H-1},$$

$$x^H \geq 0.$$

Si para $t = 1, \dots, H - 1$ definimos

$$Q^{t+1}(x^t) = E_{\xi^{t+1}}[Q^{t+1}(x^t, \xi^{t+1})]$$

Se obtiene entonces la relación recursiva:

$$Q^t(x^{t-1}(\omega^1, \dots, \omega^{t-2}), \xi^t(\omega^t)) = \text{mín} \quad c_t(\omega)^T x^t + Q^{t+1}(x^t)$$

s.r

$$W^t x^t = h^t(\omega^{t-1}) - Z^{t-1}(\omega^{t-1})x^{t-1},$$

$$x^t \geq 0.$$

Es decir, x^t puede interpretarse como el estado del sistema al iniciar el período t . Hemos asumido que la distribución de ξ^t es independiente de los eventos pasados.

Para el programa principal, la formulación es similar al programa determinístico equivalente bietapa.

$$\text{mín} \quad c_1^T x^1 + Q(x^1)$$

s.r

$$W^1 x^1 = h^1,$$

$$x^1 \geq 0.$$

1.5. EL VALOR DE INFORMACIÓN Y LA SOLUCIÓN ESTOCÁSTICA

Debido a su complejidad computacional, los programas estocásticos no suelen utilizarse directamente para modelar problemas reales de mediana o gran escala. En su lugar, suelen resolverse versiones simplificadas de los mismos. Por ejemplo, se puede resolver el programa determinístico obtenido al reemplazar todas las variables aleatorias por sus valores esperados, o resolver varios programas determinísticos correspondientes a diferentes escenarios, y entonces combinar estas diferentes soluciones mediante alguna regla heurística [1].

La pregunta natural que surge es si las soluciones obtenidas de esta manera son cercanas al óptimo del programa estocástico o si pueden ser totalmente inadecuadas. Para responder esta pregunta se han desarrollado dos conceptos: el valor esperado bajo información perfecta y el valor de la solución estocástica.

1.5.1. EL VALOR ESPERADO DE LA INFORMACIÓN PERFECTA EVPI

El EVPI (*the expected value of perfect information*) cuantifica el valor promedio que el conocimiento de información completa y exacta sobre el futuro tiene para un problema estocástico específico. En otras palabras, indica en cuanto podría mejorar (en promedio) la solución del problema, si las decisiones se tomaran con conocimiento del futuro [1].

Considerar un programa estocástico bietapa de la forma (1.39), con un vector de parámetros aleatorios ξ cuyas realizaciones corresponden a los diferentes escenarios. Se define:

$$\begin{aligned} \min z(\xi) &:= \min c^T x + \min \{q^T y \mid Wy = h - Zx, y \geq 0\} \\ \text{s.r} \quad &Ax = b, \quad x \geq 0, \end{aligned} \tag{1.41}$$

como el valor óptimo del problema de optimización asociado con un escenario en particular ξ , donde $\xi(\omega)^T = (q(\omega)^T, h(\omega)^T, Z_1(\omega)^T, \dots, Z_m(\omega)^T)$.

Sea $\bar{x}(\xi)$ una solución óptima para este problema. El *problema de distribución* consiste en buscar la distribución de $\bar{x}(\xi)$ y de $z(\xi)$ en términos de ξ . Suponiendo que se puede encontrar la distribución de las decisiones óptimas $\bar{x}(\xi)$ y sus respectivas funciones objetivo $z(\xi)$, es posible calcular el valor esperado de la solución

óptima:

$$\begin{aligned} WS &= E_{\xi}[z(\xi)] \\ &= E_{\xi}z(\bar{x}(\xi)). \end{aligned} \tag{1.42}$$

Este valor es usualmente conocido como la solución de “esperar y ver”, debido a que el mismo puede ser calculado a posteriori, recopilando datos históricos sobre el problema y usando métodos estadísticos para estimar las distribuciones de $z(\xi)$ y $\bar{x}(\xi)$.

En contraposición, la solución llamada “aquí y ahora” corresponde a la solución del problema de recurso (1.39). El valor esperado de la información perfecta se define como la diferencia entre la solución de WS y la solución de RP,

$$EVPI = RP - WS$$

En el ejemplo del granjero, la solución de espera y ver fue \$115.406 y la solución del problema con recurso fue \$ 108.390. Así el valor esperado de la información perfecta es \$7.016 (se invierte al orden de los operandos en la resta porque se trata de un problema de minimización). Este valor indica cuánto beneficiaría anualmente al granjero, en promedio, el obtener información precisa sobre el clima de la siguiente temporada.

1.5.2. EL VALOR DE LA SOLUCIÓN ESTOCÁSTICA VSS

En algunas ocasiones, resolver el problema de recurso (1.39) es impracticable debido a su tamaño y a su alta complejidad computacional. La tentación natural es entonces resolver un problema más simple. Por ejemplo, reemplazando todas las variable aleatorias por sus valores esperados. Este problema se lo llama *el problema del valor esperado* o *el problema del valor promedio*, y puede formularse como:

$$EV = \min C^T x + Q(x, \bar{\xi}) \tag{1.43}$$

$$\text{s.r.} \tag{1.44}$$

$$Ax = b \tag{1.45}$$

$$x \geq 0, \tag{1.46}$$

donde Q está definida por (1.37) y $\bar{\xi} = E(\xi)$ denota el valor esperado de ξ .

La solución óptima de (1.43), $\bar{x}(\bar{\xi})$, es llamada la *solución del valor esperado*. No

hay ninguna razón para creer que $\bar{x}(\bar{\xi})$ es una buena aproximación a la solución del problema estocástico. El *resultado esperado de usar la solución EV* es

$$EEV = E_{\xi}[c^T \bar{x} + Q(\bar{\xi}, \xi)]. \quad (1.47)$$

El EEV mide cómo la solución del valor esperado actúa, permitiendo escoger las decisiones óptimas de la segunda etapa, en función de $\bar{x}(\bar{\xi})$ y ξ .

El valor de la solución estocástica se define como la diferencia del resultado esperado al usar la solución EV y la solución del problema de recurso:

$$VSS = EEV - RP$$

El valor VSS mide cuán buena o mala es una aproximación determinística a la solución al problema estocástico tomada únicamente sobre la base de valores promedio de los parámetros aleatorios.

Para el ejemplo del granjero, el valor $EEV = \$107.240$. Entonces el valor de la solución estocástica es $VSS = \$1.150$. Esta cantidad representa el costo de ignorar la incertidumbre dentro del proceso de una decisión.

Capítulo 2

EL PROBLEMA DE GESTIÓN DE SALDO EN CAJA EN EL BANCO CENTRAL DEL ECUADOR (PGSC)

Una de las funciones del Banco Central del Ecuador es el manejo de una cierta cantidad de dinero que la banca privada tiene depositada en esta entidad, con el fin de asegurar la suficiente liquidez monetaria para mantener el sistema financiero funcionando sin problemas. Si bien la función del banco no es hacer producir este dinero, a veces los saldos diarios en caja superan considerablemente a los retiros diarios, como se observa en la Figura 2.1. La diferencia es aún más grande si se toman en cuenta no sólo los retiros, sino también los depósitos diarios, es decir, el flujo neto diario. La Figura 2.2 indica valores históricos de *requerimiento neto diario*, comparados con los saldos en caja, para el período enero 2001 - junio 2006.

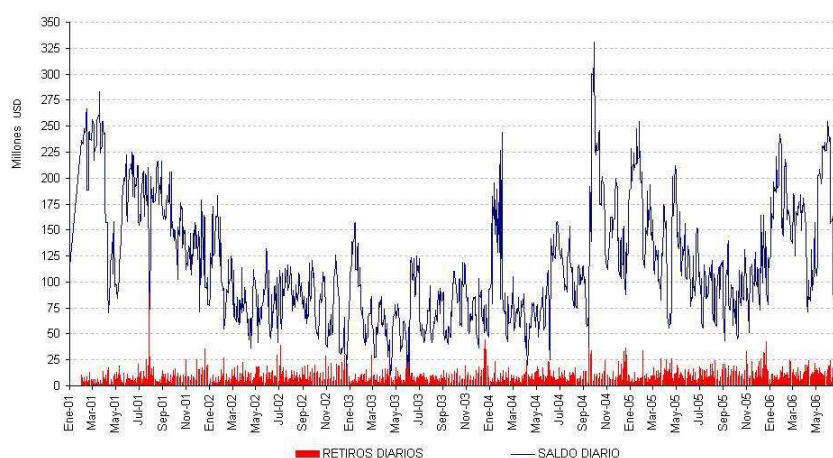


Figura 2.1: Saldo en caja y retiros diarios desde enero del 2001 hasta junio del 2006. Fuente: Banco Central del Ecuador.

Considerando estos datos se ha planteado la idea de invertir el excedente de dinero en el extranjero, de una manera segura, como una alternativa para generar recursos adicionales para el fisco. Para mantener la suficiente liquidez monetaria, el Banco Central ha establecido un valor de saldo en caja mínimo, el cual se ajusta semanalmente. El saldo en caja diario no debe ser nunca inferior a este valor.

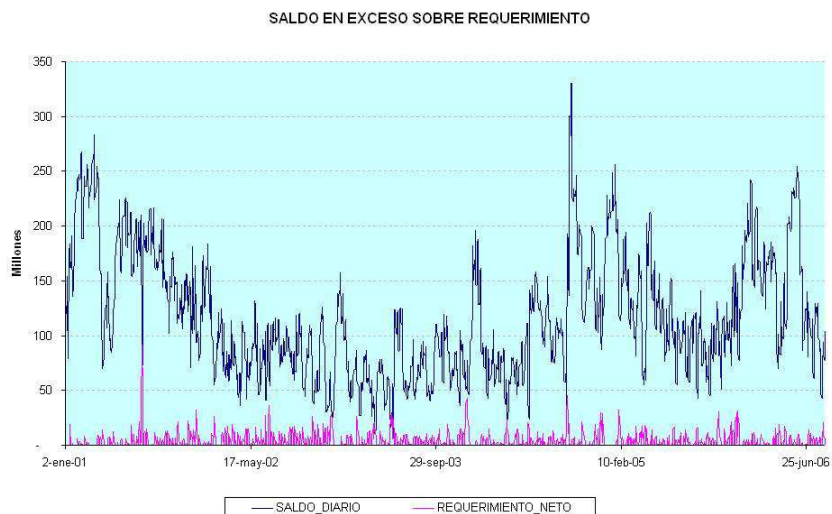


Figura 2.2: Saldo en caja y requerimientos neto diarios desde enero del 2001 hasta junio del 2006. Fuente: Banco Central del Ecuador.

A esta restricción del saldo se la llama *restricción de seguridad*. Las inversiones en el extranjero pueden realizarse a corto plazo. Cada retiro o envío de dinero está asociado a un costo de transacción y demora un determinado intervalo de tiempo en hacerse efectivo.

En la siguiente sección se formula un modelo de optimización para el PGSC, que considera los argumentos ya mencionados.

2.1. MODELO DE OPTIMIZACIÓN PARA EL PGSC

El problema de gestión de saldo en caja en el Banco Central del Ecuador consiste en dados un saldo en caja inicial, un flujo neto diario, decidir cuánto dinero enviar o retirar del extranjero cada día durante un horizonte de planificación T , de manera que se satisfaga la restricción de seguridad y a su vez se maximice la utilidad global percibida en la operación. Este problema puede formularse como el

siguiente programa de optimización:

$$\text{máx } U = \sum_{t=1}^T [r_t E_t - c_t^M M_t - c_t^I I_t] \quad (2.1)$$

s.r

$$E_t = E_{t-1} + I_{t-2} - M_t \quad t = 1, \dots, T \quad (2.2)$$

$$\text{(POGSC)} \quad C_t = C_{t-1} + Y_t - I_t + M_{t-2} \quad t = 1, \dots, T \quad (2.3)$$

$$C_t \geq L_t \quad \forall t = 1, \dots, T \quad (2.4)$$

$$I_t, M_t, E_t \geq 0 \quad \forall t = 1, \dots, T. \quad (2.5)$$

donde:

- Y_t es el flujo neto en caja (depósitos - retiros) en el día t , C_t es el saldo en caja al final del día t y E_t es el saldo exterior al final del día t . C_0 y E_0 son los saldos iniciales en caja y en el exterior respectivamente.
- Las variables de decisión están representadas por I_t , cantidad invertida en el extranjero en el día t y por M_t , cantidad retirada del saldo en el extranjero en el día t . Considerando que los envíos y retiros de dinero toman dos días para hacerse efectivos, el saldo diario en caja y el saldo diario en el extranjero satisfacen las ecuaciones 2.2 y 2.3. Por simplicidad, supondremos que $I_{-1} = M_{-1} = I_0 = M_0 = 0$.
- La función objetivo 2.1 refleja la utilidad generada durante el horizonte de planificación. La misma es igual al interés generado por el saldo diario en el extranjero menos los costos de envío y retiro de dinero. Denotamos por r_t a la tasa de interés, por c_t^M a la tasa para retiros y por c_t^I a la tasa para depósitos en el día t . Los intereses se capitalizan recién al finalizar el período.
- La ecuación 2.4 es la restricción de seguridad, donde L_t es el valor mínimo para el saldo en caja en el día t . Estos valores se ajustan semanalmente por el Banco Central.
- La restricción 2.5 indica que tanto las variables de decisión como el saldo en el exterior no pueden tomar valores negativos.
- Además, se ha asumido que las decisiones de envío/retiro de dinero se toman al final del día, una vez conocido el valor del flujo de dinero en dicho día, aunque antes de determinar los valores de saldo en caja y saldo en el exterior.

Notar que tanto el flujo neto en caja diario, como las tasas de interés y las tasas de cobro por transacción son variables aleatorias. Desde este punto de vista, POGSC es bastante similar al problema de planificación financiera descrito en el capítulo anterior. Sin embargo, debido a que en condiciones normales el comportamiento de las tasas de interés y los costos de transacción es predecible con un grado razonable de aproximación en el corto plazo, supondremos en esta tesis que dichos parámetros son conocidos de antemano. Así, la incertidumbre considerada en el modelo se refiere exclusivamente al flujo neto de dinero diario.

Formularemos en la Sección 2.3 un programa con recurso multietapa para el PGSC. Desafortunadamente, la aplicabilidad práctica del mismo es limitada, debido a que los horizontes de tiempo considerados en instancias reales ($T \geq 30$ días) conducirían a programas lineales con una cantidad astronómica de variables (en el orden de 10^{20}). Sin embargo, la programación estocástica puede aún ser empleada para el desarrollo de algoritmos en-línea que alcanzan buenos resultados como veremos en el Capítulo 3. Empezaremos a continuación el análisis de POGSC considerando la versión determinística del mismo.

2.2. SOLUCIÓN BAJO INFORMACIÓN PERFECTA

Si se suponen conocidos los valores del flujo neto diario para todos los días dentro del horizonte de planificación (por ejemplo para la estimación a posteriori de la solución bajo información perfecta), se puede reducir a un problema de flujo de costo mínimo.

El problema de flujo de costo mínimo (FCM) es uno de los problemas fundamentales de optimización en redes. Problemas de flujo de costo mínimo han surgido en muchos campos de la industria, como: la agricultura, las tele-comunicaciones, la educación, la energía, la salud, la fabricación, la medicina, la comercialización y el transporte [21].

El problema de flujo de costo mínimo puede formularse de la siguiente manera. Datos:

- una red dirigida, $R = (V, A, u, l)$, definida por un conjunto V de n nodos; un conjunto A de m arcos dirigidos, y dos vectores $u, l \in \mathbb{R}^A$ que especifican valores de flujo mínimo y capacidad máxima para cada arco.
- un vector, $b \in \mathbb{R}^V$, de demanda o suministro sobre los nodos, (asumiremos que

$b(i) > 0$, significa que el nodo i demanda flujo de la red, $b(i) < 0$ significa que i abastece con flujo a la red, y que $b(i) = 0$ no demanda ni abastece con flujo a la red, (en ese caso i es de transición)).

• un vector, $c \in \mathbb{R}^A$, de costos de transportación sobres los arcos, se pide encontrar un vector $x \in \mathbb{R}^A$ que satisfaga

$$f_x(i) = b(i), \quad \forall i \in V \quad (2.6)$$

$$l_{(i,j)} \leq x_{(i,j)} \leq \mu_{(i,j)}, \quad \forall (i,j) \in A \quad (2.7)$$

y para el cual el costo total de transportación del flujo a través de la red,

$$\sum_{(i,j) \in A} c_{(i,j)} x_{(i,j)},$$

sea mínimo. El valor

$$f_x(i) := \sum_{\{j:(j,i) \in A\}} x_{(j,i)} - \sum_{\{j:(i,j) \in A\}} x_{(i,j)}$$

se conoce como *flujo neto hacia el nodo i* , pues representa la cantidad total de flujo sobre los arcos que *entran* a i menos el flujo sobre los arcos que *salen* de i .

La ecuación (2.6) es la condición de *conservación de flujo*: el flujo neto en cada nodo debe ser igual a su demanda o suministro. La ecuación (2.7) establece la restricción de flujo mínimo/capacidad máxima sobre cada arco. Un vector $x \in \mathbb{R}^A$ que satisface las restricciones (2.6) y (2.7) se denomina un *flujo factible*.

El problema de flujo de costo mínimo es un problema polinomial, pues puede ser formulado como un programa lineal. Existen, además algoritmos combinatorios eficientes para su solución, el más conocido de estos es el *método del simplex para redes* [39].

Dada una instancia del POGSC en la cual los valores de r_t , c_t^M , c_t^I y y_t se suponen conocidos para todo el período de planificación, es posible formular a partir de la misma un problema de flujo de costo mínimo. Para ello se definen la red R , el vector de demandas b y el vector de costos c como se indica a continuación:

• Se requieren dos nodos por cada día $t \in \{1, \dots, T\}$ para explicar los movimientos de dinero. Un nodo v_t para indicar las transacciones que se realizan sobre el saldo en caja, al cual llamaremos *nodo interior*, y otro nodo w_t indica las transacciones que se realizan en el extranjero, llamado *nodo exterior*. Introducimos además dos

nodos auxiliares v_{T+1} y w_{T+1} .

Definimos cuatro conjuntos de arcos: un conjunto A_1 con arcos de la forma (v_t, v_{t+1}) , $1 \leq t \leq T$, que representan el dinero obtenido en caja entre un día y el siguiente; un conjunto $A_2 := \{(w_t, w_{t+1}) : 1 \leq t \leq T\}$ que representa el dinero mantenido en el extranjero; un conjunto $A_3 := \{(v_t, w_{t+2}) : 1 \leq t \leq T-1\} \cup \{(v_T, w_{T+1})\}$ para representar los envíos al extranjero; y un conjunto para los retiros del extranjero $A_4 := \{(w_t, v_{t+2}) : 1 \leq t \leq T-1\} \cup \{(w_T, v_{T+1})\}$. Adicionalmente, definimos un arco auxiliar de la forma (w_{T+1}, v_{T+1}) . La capacidad máxima de cada arco es $+\infty$. El flujo mínimo es 0 si el arco no pertenece a A_1 , y L_t para cada arco $(v_t, v_{t+1}) \in A_1$.

• La función, b , de demanda sobre los nodos está dada por:

$$b(v_t) = \begin{cases} -(C_0 + Y_1), & \text{si } t = 1 \\ -Y_t, & \text{si } 2 \leq t \leq T \\ -(C_0 + E_0 - \sum_{i=1}^T Y_i), & \text{si } t = T + 1 \end{cases}$$

$$b(w_t) = \begin{cases} -E_0, & \text{si } t = 1 \\ 0, & \text{en los demás casos.} \end{cases}$$

Así, las demandas simbolizan básicamente el flujo neto diario en caja, es decir, la cantidad que entra o sale del sistema.

Para que el problema de flujo admita al menos una solución factible, se requiere que la suma de demandas sobre los nodos sea igual a cero. Esto se garantiza definiendo la demanda de $b(v_{T+1})$ como se expresa arriba.

• La función, c , de costos sobre los arcos está dada por:

$$c_{(t,u)} = \begin{cases} 0 & \text{si } a \in A_1 \\ -r_t & \text{si } a = (w_t, w_{t+1}) \in A_2 \\ c_t^I & \text{si } a = (v_t, w_{t+2}) \in A_3 \\ c_t^M & \text{si } a = (w_t, v_{t+2}) \in A_4 \\ 0 & \text{si } a = (w_{T+1}, v_{T+1}) \end{cases}$$

La Figura 2.3 muestra un ejemplo para una instancia con $T = 3$.

Cualquier flujo factible sobre R puede usarse para definir una solución factible para POGSC: el valor de cada variable I_t se fija igual al valor del flujo sobre el arco $(v_t, w_{t+2}) \in A_3$, o igual al flujo sobre (v_T, w_{T+1}) .

Si $t = T$; el valor de cada variable M_t se fija de acuerdo al valor del flujo sobre el arco correspondiente A_4 . Puede verificarse entonces que las restricciones (2.2) y (2.3) son equivalentes a las restricciones de conservación de flujo sobre los nodos

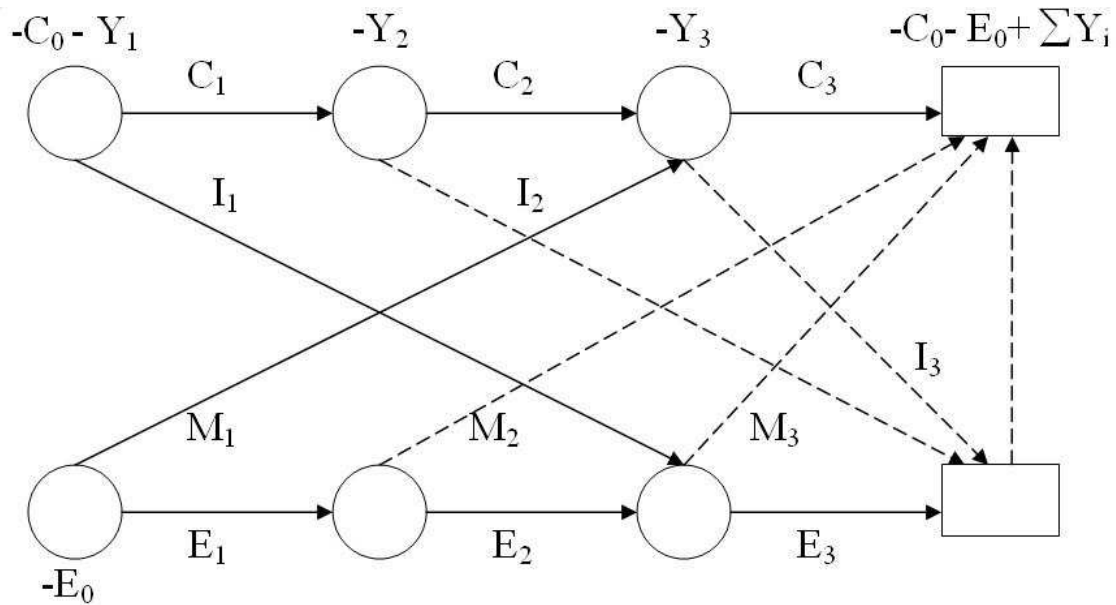


Figura 2.3: Reducción del PGSC a un problema de flujo de costo mínimo.

exteriores e interiores, respectivamente. Las restricciones (2.4) y (2.5) equivalen a las restricciones de flujo mínimo sobre los arcos. Como $u_a = +\infty, \forall a \in A$, las restricciones de capacidad máxima son redundantes. Finalmente al definir M_t e I_t de esta manera, los valores obtenidos para C_t y E_t corresponden al flujo sobre los arcos de A_1 y A_2 , respectivamente; y el costo de la solución dado por (2.1) es igual al costo del flujo. Por otra parte, es fácil constatar que toda solución al POGSC puede ser usada para definir un flujo factible en R con igual costo empleando la misma idea anterior.

El modelo determinístico para el PGSC supone el conocimiento a priori de todos los valores Y_t de flujo diario neto durante el horizonte de planificación. En la práctica estos valores no pueden predecirse con exactitud, ni siquiera para horizontes de tiempo pequeños. Por esto, el problema de gestión de saldo en caja del BCE es un problema de optimización bajo incertidumbre. Como se señaló en el capítulo anterior, problemas similares, por ejemplo el problema de planeación y control financiero, han sido formulados mediante modelos de programación estocástica.

Por otra parte, este problema requiere de la toma de decisiones secuenciales, donde las decisiones tomadas en un período influyen sobre los períodos subsiguientes, lo que sugiere el empleo de un modelo de programación estocástica multietapa.

2.3. MODELO ESTOCÁSTICO MULTIETAPA PARA EL PGSC

Para formular al problema de gestión de saldo en caja del BCE como un programa estocástico, consideramos al flujo de dinero en cada día t como una variable aleatoria Y_t . Cada escenario s se representa entonces por un vector (y_1, y_2, \dots, y_T) de observaciones para las variables aleatorias dentro del horizonte de planificación. Con el fin de tener un número finito de variables en el modelo, asumiremos en adelante que el flujo de dinero para un día t puede tomar valores únicamente de un conjunto finito \mathbb{Y}_t . Es decir,

$$y_t \in \mathbb{Y}_t, \quad \forall 1 \leq t \leq T$$

con $|\mathbb{Y}| =: k_t \in \mathbb{N}$.

Las variables de decisión del modelo representan las inversiones o pedidos a realizarse al final de cada día, dentro del horizonte de planificación y bajo cada escenario posible. Terminado el primer día se presentan k_1 escenarios posibles, de acuerdo al valor y_1 del flujo de dinero registrado en ese día. En cada uno de estos escenarios, se debe determinar el monto de la inversión para el primer día. De manera similar, al final del segundo día se tendrá $k_1 k_2$ escenarios posibles, de acuerdo a los valores registrados para $(y_1, y_2) \in \mathbb{Y}_1 \times \mathbb{Y}_2$. Representaremos por $I_1(y_1)$ y $M_1(y_1)$ a los montos de inversión y pedido respectivamente para el primer día y con $I_2(y_1, y_2)$ y $M_2(y_1, y_2)$ a los montos de inversión y pedido respectivamente para el segundo día. Prosiguiendo de esta forma, se obtiene un modelo con

$$\prod_{t=1}^T k_t$$

variables de decisión de la forma $I_t(y_1, \dots, y_t)$ y $M_t(y_1, \dots, y_t)$ donde $(y_1, \dots, y_t) \in \mathbb{Y}_1 \times \dots \times \mathbb{Y}_t \forall 1 \leq t \leq T$.

Asociado a los montos de inversión y pedido están valores de saldo diario en caja y en el extranjero definidos como $C_t(y_1, \dots, y_t)$ y $E_t(y_1, \dots, y_t) \forall 0 \leq t \leq T$. Denotaremos por $p(y_1, \dots, y_t)$ a la probabilidad de que el escenario (y_1, \dots, y_t) ocurra. Utilizando las ideas de la Sección 1.2.3, se puede formular como el siguiente pro-

grama multietapa con recurso:

$$\text{máx } \mathbf{E}[U] = \sum_{(y_1, \dots, y_T) \in \mathbb{Y}_1 \times \dots \times \mathbb{Y}_T} p(y_1, \dots, y_T) \left[\sum_{t=1}^T [r_t E_t(y_1, \dots, y_t) - c_t^I I_t(y_1, \dots, y_t) - c_t^M M_t(y_1, \dots, y_t)] \right] \quad (2.8)$$

s.r

$$\begin{aligned} E_t(y_1, \dots, y_t) &= E_{t-1}(y_1, \dots, y_{t-1}) + I_{t-2}(y_1, \dots, y_{t-2}) - \\ &\quad M_t(y_1, \dots, y_t), \\ \forall (y_1, \dots, y_t) &\in \mathbb{Y}_1 \times \dots \times \mathbb{Y}_t, \quad \forall t = 1, \dots, T \end{aligned} \quad (2.9)$$

$$\begin{aligned} \text{(PEGSC)} \quad C_t(y_1, \dots, y_t) &= C_{t-1}(y_1, \dots, y_{t-1}) + Y_t(y_1, \dots, y_t) - \\ &\quad I_t(y_1, \dots, y_t) + M_{t-2}(y_1, \dots, y_{t-2}), \\ \forall (y_1, \dots, y_t) &\in \mathbb{Y}_1 \times \dots \times \mathbb{Y}_t, \quad \forall t = 1, \dots, T \end{aligned} \quad (2.10)$$

$$\begin{aligned} C_t(y_1, \dots, y_t) &\geq L_t, \\ \forall (y_1, \dots, y_t) &\in \mathbb{Y}_1 \times \dots \times \mathbb{Y}_t, \quad \forall t = 1, \dots, T \end{aligned} \quad (2.11)$$

$$\begin{aligned} I_t(y_1, \dots, y_t), M_t(y_1, \dots, y_t), E_t(y_1, \dots, y_t) &\geq 0, \\ \forall (y_1, \dots, y_t) &\in \mathbb{Y}_1 \times \dots \times \mathbb{Y}_t, \quad \forall t = 1, \dots, T \end{aligned} \quad (2.12)$$

donde $\mathbf{E}[U]$ denota la esperanza de la utilidad y $M_{-1} = I_{-1} = M_0 = I_0 = 0$.

Notar que el tamaño del problema crece rápidamente conforme el número de escenarios aumenta. El número de escenarios crece de acuerdo al número de períodos y al tamaño de los conjuntos \mathbb{Y}_t . Por otra parte, esta formulación requiere que el flujo de dinero para cada día tenga una función de distribución discreta, y sobre un conjunto finito de valores.

PEGSC permite conservar la no anticipatividad de las decisiones, ya que marca la dependencia de las decisiones con el escenario en cada uno de los períodos. Es decir, los escenarios que comparten la misma historia hasta un período t , tomarán las mismas decisiones hasta ese período.

En programación estocástica cuando en un modelo se explicita las decisiones para todos los períodos, se lo llama modelo extenso. Así, PEGSC es el *modelo extenso de programación estocástica para el problema de gestión de saldo en caja en el BCE*.

Para visualizar de mejor forma la formulación del modelo estocástico. A continuación, se presenta una instancia del PGSC para un período corto de planificación.

2.4. EJEMPLO

Para ilustrar las dificultades que se pueden presentar en la formulación y solución del modelo extenso de programación estocástica para el PGSC, se presenta un ejemplo sencillo para un horizonte de planificación de seis días y con los siguientes supuestos:

- Las tasas de retorno son constantes e iguales para el horizonte de planificación, es decir, $r_t \% = r \% = 0,015 \%$, $t = 1, 2, \dots, T$.
- Las tasas de costos por invertir o retirar dinero del extranjero son constantes e iguales, es decir $c_t^M \% = c_t^I \% = c \% = 0,02 \%$, $t = 1, 2, \dots, T$
- El saldo en caja inicial $C_0 = 140.844$ millones de dólares
- El saldo en caja exterior inicial $E_0 = 20$ millones de dólares.

Para tomar en cuenta la incertidumbre en el flujo de caja neto en el PGSC se formuló PEGSG, el cual supone un conjunto de escenarios para el flujo de caja. Por lo tanto, para formular el PEGSC para esta instancia se requiere detallar un conjunto de posibles escenarios.

La forma normal de representar los escenarios es mediante un árbol de escenarios.

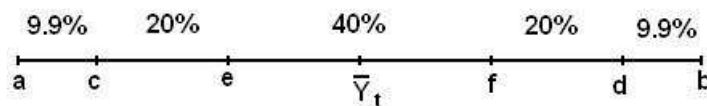
No conocemos a ciencia cierta la función de distribución del flujo de dinero, sin embargo se dispone de observaciones históricas de esta variable. El rango en el cual cae el flujo de dinero es aproximadamente $[-20; 20]$ millones de dólares. Se podría subdividir este intervalo y obtener los puntos medios para crear diferentes escenarios. Sin embargo, este procedimiento no toma en cuenta que el flujo neto de dinero tiene un comportamiento propio diferente cada día. De esta forma, el procedimiento anterior no crearía buenos escenarios.

Para el desarrollo de esta tesis se realizó un estudio de la serie de datos prevista. Del estudio univariante del flujo neto diario de dinero se obtuvo tres posibles modelos, los cuales están detallados en el Apéndice A de esta tesis. Estos modelos permiten realizar predicciones a un horizonte corto de tiempo. Además, ayudan en el conocimiento del comportamiento del flujo neto.

Para hacer predicciones acerca de los valores que puede tomar el flujo, estas distribuciones hacen uso de la información pasada del flujo y así, obtienen intervalos

de confianza para la predicción. Estos intervalos de confianza pueden ser obtenidos a diferentes niveles de confianza como se indica en el Apéndice A.

Para construir un árbol de escenarios para el flujo de caja neto diario; en cada período se utilizan intervalos de confianza a diferentes niveles para las predicciones. Por ejemplo, sea \bar{y}_t el valor medio estimado por el modelo, se puede obtener: un intervalo $[a, b]$ al 99,73% de confianza para la predicción de Y_t , luego obtener los intervalos $[c, d]$ y $[e, f]$ al 80% y 60% de confianza respectivamente. Así, se obtiene los subintervalos $[a, c]$, $[c, e]$, $[e, f]$, $[f, d]$ y $[d, b]$ con probabilidades de 0,099, 0,20, 0,40, 0,20 y 0,099 respectivamente como se indica en el gráfico:



Los puntos medios de cada intervalo son sus respectivos representantes y también son los nodos del árbol de escenarios, así las probabilidades de ocurrencia de cada nodo son las de sus respectivos intervalos. Para obtener más nodos del árbol a partir de un punto medio, se calcula nuevamente varios intervalos de confianza para la predicción del siguiente período, tomando a este punto medio como parte de la historia del flujo neto. Este proceso se puede repetir para cada uno de los puntos medios de los intervalos obtenidos en los diferentes períodos y de esta forma obtener cada vez un árbol de escenarios más espeso.

El conjunto de escenarios está formado por cada camino posible desde la raíz hasta cada uno de los nodos del árbol. De esta forma, se conserva la no anticipatividad, ya que cada valor tomado en un nodo (flujo de caja neto) depende del camino seguido (historia del flujo).

Para este ejemplo se construye un árbol pequeño y poco espeso. Obteniendo sólo dos intervalos de confianza para las predicciones del segundo hasta el quinto período y para el sexto período se toma la predicción del valor medio. El árbol de escenarios que se obtiene se indica en la Figura 2.4. Este árbol tiene 46 nodos que forman 16 escenarios.

En el Cuadro 2.1 se presenta el valor del flujo neto estimado en millones de dólares y la probabilidad de ocurrencia para cada una de los nodos del árbol. En el Cuadro 2.2 se especifica cada uno de los posibles escenarios, los cuales tienen igual probabilidad de ocurrencia de 0,0531.

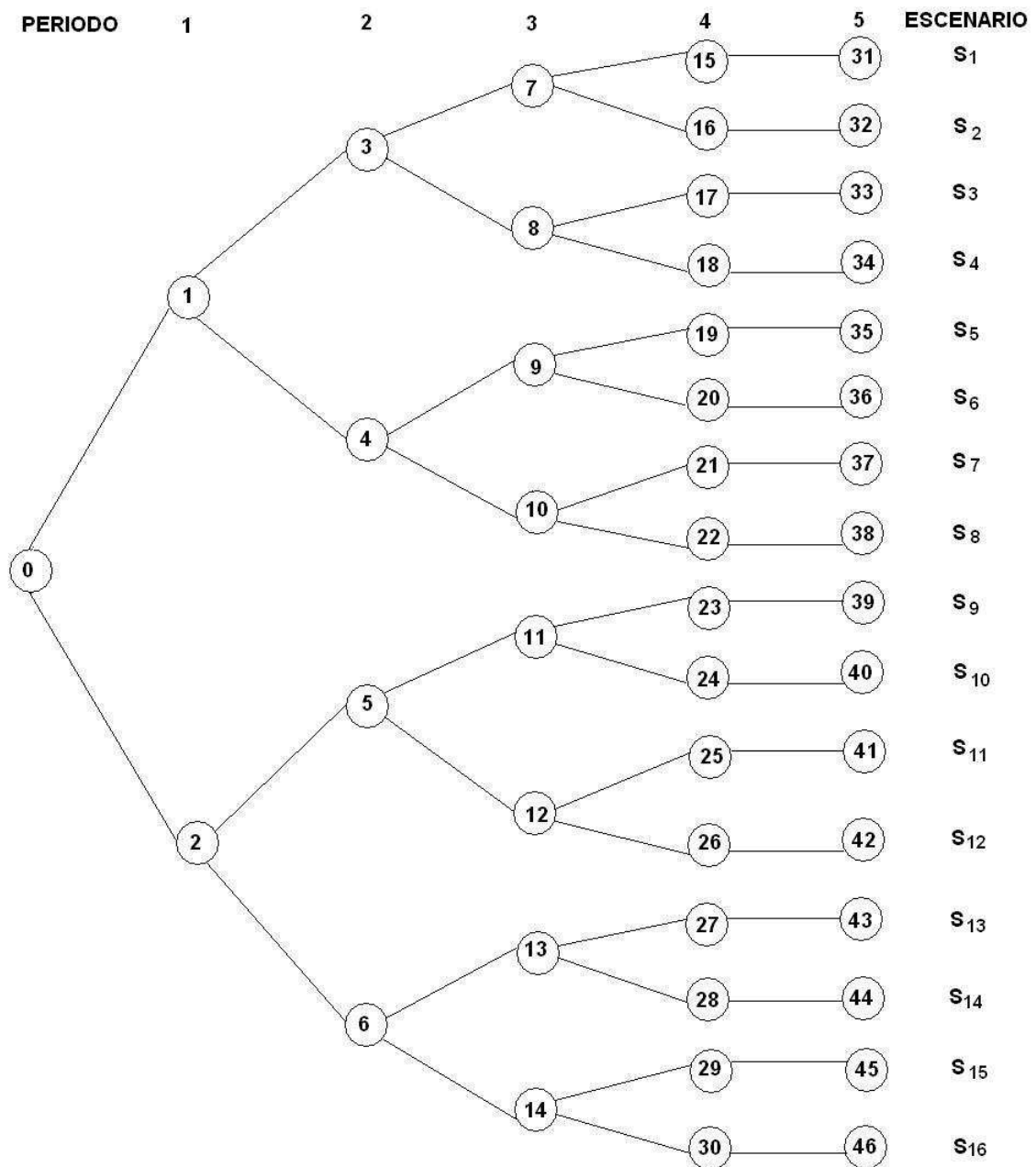


Figura 2.4: Árbol de escenarios

Nodo	Estimación	Probabilidad de ocurrencia	Nodo	Estimación	Probabilidad de ocurrencia
0	2,1200	1,0000	24	-3,5399	0,0531
1	17,6274	0,4800	25	11,2810	0,0531
2	-3,1337	0,4800	26	-10,6705	0,0531
3	17,1252	0,2304	27	16,1057	0,0531
4	-4,7049	0,2304	28	-5,8458	0,0531
5	10,3776	0,2304	29	8,9750	0,0531
6	-11,4524	0,2304	30	-12,9765	0,0531
7	22,8160	0,1106	31	22,2558	0,0531
8	0,8761	0,1106	32	15,1214	0,0531
9	15,7211	0,1106	33	19,9383	0,0531
10	-6,2188	0,1106	34	12,8039	0,0531
11	20,6230	0,1106	35	21,5064	0,0531
12	-1,3169	0,1106	36	14,3720	0,0531
13	13,5281	0,1106	37	19,1889	0,0531
14	-8,4119	0,1106	38	12,0544	0,0531
15	19,1244	0,0531	39	22,0242	0,0531
16	-2,8271	0,0531	40	14,8898	0,0531
17	11,9937	0,0531	41	19,7067	0,0531
18	-9,9578	0,0531	42	12,5722	0,0531
19	16,8184	0,0531	43	21,2747	0,0531
20	-5,1331	0,0531	44	14,1403	0,0531
21	9,6878	0,0531	45	18,9572	0,0531
22	-12,2637	0,0531	46	11,8228	0,0531
23	18,4116	0,0531			

Cuadro 2.1: Valores y probabilidades de las hojas del árbol de la Figura 2.4

Asumiendo que el valor mínimo que puede tomar el saldo en caja en los períodos 1 hasta 5 es 33,472 y en el período 6 es 40,739; y creados y definidos los escenarios se formular el problema estocástico extenso para esta instancia. Debido a que la formulación del modelo extenso para este ejemplo es bastante amplia se la detalla en el Apéndice B.

Escenario	Historia	Estimaciones
1	(1,3,7,15,31)	(17,627, 17,125, 22,816, 19,124, 22,256)
2	(1,3,7,16,32)	(17,627, 17,125, 22,816, -2,827, 15,121)
3	(1,3,8,17,33)	(17,627, 17,125, 0,876, 11,994, 19,938)
4	(1,3,8,18,34)	(17,627, 17,125, 0,876, -9,958, 12,804)
5	(1,4,9,19,35)	(17,627, -4,705, 15,721, 16,818, 21,506)
6	(1,4,9,20,36)	(17,627, -4,705, 15,721, -5,133, 14,372)
7	(1,4,10,21,37)	(17,627, -4,705, -6,219, 9,688, 19,189)
8	(1,4,10,22,38)	(17,627, -4,705, -6,219, -12,26, 12,054)
9	(2,5,11,23,39)	(-3,134, 10,378, 20,62, 18,412, 22,024)
10	(2,5,11,24,40)	(-3,134, 10,378, 20,62, -3,54, 14,89)
11	(2,5,12,25,41)	(-3,134, 10,378, -1,317, 11,281, 19,707)
12	(2,5,12,26,42)	(-3,134, 10,378, -1,317, -10,67, 12,572)
13	(2,6,13,27,43)	(-3,134, -11,45, 13,528, 16,106, 21,275)
14	(2,6,13,28,44)	(-3,134, -11,45, 13,528, -5,846, 14,140)
15	(2,6,14,29,45)	(-3,134, -11,45, -8,412, 8,975, 18,957)
16	(2,6,14,30,46)	(-3,134, -11,45, -8,412, -12,98, 11,823)

Cuadro 2.2: Escenarios para el flujo de caja neto utilizados en el ejemplo

La solución del problema estocástico para esta instancia se presenta en el Cuadro 2.3. Detalles acerca de la solución del PEGSC para este ejemplo se encuentran en el Apéndice B.

La solución del PEGSC indica que se debe invertir en el primer período 86,4941 millones de dólares en todos los escenarios. En los escenarios 1 a 8 se debe invertir 10,1709 millones de dólares y en los restantes escenarios no se debe invertir en el segundo día. En el tercer día, en los escenarios 1 a 4 se debe invertir 31,231

Día	Escenario	Pedidos	Inversión
1	1-16	0	86,4941
2	1-8	0	10,1709
	9-16	0	0
3	1-4	0	31,231
	5-8	0	0
	9-12	0	10,9875
	13-16	20,2435	0
4	1-16	0	0
5	1-16	0	0
6	1-16	0	0
utilidad óptima: 0,0508757			

Cuadro 2.3: Solución del PEGSC para el Ejemplo.

millones de dólares; en los escenarios 9 a 12 se debe invertir 10,9875 millones de dólares; en los escenarios 13 a 16 se debe realizar un pedido de 20,2435 millones de dólares; y en el resto de escenarios no se realizan inversiones ni pedidos. En todos los escenarios en los días 4, 5 y 6 no se realizan inversiones ni pedidos.

Aplicando la solución del ejemplo a un flujo de caja neto real, que pertenece al tercer escenario se obtiene una utilidad de 0,05826 millones de dólares. En el Cuadro 2.4 se detallan el flujo real y los saldo en caja, saldo en el exterior, costos y beneficio que resultaron de tomar las decisiones para este flujo según la solución del Cuadro 2.3 para el tercer escenario.

La solución óptima para este flujo de caja real, si se pudiera conocer la información a priori de este, se indica en el Cuadro 2.5.

La diferencia entre la utilidad obtenida por el PEGSC y la utilidad óptima para este flujo del tercer escenario es: $0,0731 - 0,05826 = 0,01484$, que es el costo por no conocer la información perfecta acerca del flujo.

Para obtener el valor bajo información perfecta, *EVPI*, para esta instancia es necesario conocer la solución óptima para cada uno de los 16 escenarios. El Cuadro 2.6 indica la solución óptima para cada escenario y la solución WS (“esperar

Día	Flujo	Caja	Exterior	Inversión	Costo	Beneficio	Utilidad
1	2,120	56,47	20,000	86,494	0,017	0,003	-0,014
2	37,256	83,555	20,000	10,170	0,002	0,003	0,0010
3	20,296	72,620	106,494	31,231	0,006	0,016	0,0097
4	7,197	79,817	116,665	0	0	0,018	0,0175
5	10,18	89,997	147,896	0	0	0,022	0,0222
6	4,224	94,221	147,896	0	0	0,022	0,0222
utilidad total: 0,05826							

Cuadro 2.4: Solución del ejemplo aplicada a un flujo real perteneciente al tercer escenario.

Día	Flujo	Saldo caja	Saldo Exterior	Inversión
1	2,120	33,472	20,000	109,49
2	37,256	33,472	20,000	37,256
3	20,296	33,472	129,492	20,296
4	7,197	40,669	166,748	0,000
5	10,180	50,849	187,044	0,000
6	4,224	55,073	187,044	0,000
utilidad : 0,0731				

Cuadro 2.5: Solución óptima para el flujo de caja real del tercer escenario.

y ver”), la cual es el promedio de estas soluciones óptimas.

Escenario	Utilidad óptima	Escenario	Utilidad óptima
1	0,067916	9	0,061581
2	0,067916	10	0,061581
3	0,067916	11	0,061449
4	0,066281	12	0,056992
5	0,065027	13	0,055962
6	0,065027	14	0,055962
7	0,063473	15	0,052598
8	0,056666	16	0,0445
WS = 0,060678			

Cuadro 2.6: Utilidad óptima para cada escenarios del ejemplo.

El valor del *EVPI* será $0,050876 - 0,060678 = 0,009802$, que es el costo por no conocer información perfecta sobre el flujo de caja.

Si por la dificultad de resolver el modelo estocástico para esta instancia se resolviera el POGSC reemplazando los valores del flujo de caja por sus valores esperados, con lo cual POGSC es un problema determinístico, se obtiene las decisiones del Cuadro 2.7.

Día	Flujo neto medio	Inversión
1	2,12	109,49
2	7,247	7,247
3	2,836	2,836
4	7,202	0
5	3,074	0
6	17,04	0
utilidad: 0,063892		

Cuadro 2.7: Solución del programa determinístico para el ejemplo

El valor esperado para esta instancia es $EV = 0,063892$. Recordar que la solución con valor esperado para el flujo neto de dinero no consideran todos los escenarios buenos y malos, tomando decisiones malas. Por ejemplo, si el flujo de caja neto pertenece al octavo escenario y las decisiones son tomadas conforme el Cuadro 2.7, los saldos en caja de los días 4, 5 y 6 no cumplen con la restricción de seguridad como se indica en el Cuadro 2.8. En cambio si se tomarán las decisiones para este mismo flujo de caja conforme la solución del problema estocástico, el saldo en caja cumple con la restricción de seguridad como indica el Cuadro 2.9.

Día	Flujo	Mínimo	Caja	Exterior	Inversiones
1	2,12	33,472	33,472	20	109,49
2	10,364	33,472	36,589	20	7,247
3	4,032	33,472	37,785	129,492	2,836
4	-10,438	33,472	27,347	136,739	0
5	-14,817	33,472	12,53	139,575	0
6	6,103	40,739	18,633	139,575	0
utilidad : 0,063892					

Cuadro 2.8: Decisiones tomadas respecto al Cuadro 2.7 para el flujo de caja del octavo escenario.

Día	Flujo	Mínimo	Caja	Exterior	Inversiones
1	2,12	33,472	56,47	20	86,494
2	10,364	33,472	56,663	20	10,171
3	4,032	33,472	60,695	106,494	0
4	-10,438	33,472	50,257	116,665	0
5	-14,817	33,472	35,44	116,665	0
1	6,103	40,739	41,543	116,665	0

Cuadro 2.9: Decisiones tomadas respecto al Cuadro 2.3 para el flujo de caja del octavo escenario

Para obtener el valor esperado de la solución esperada EV, se debe calcular cada una de las utilidades obtenidas en los escenarios al aplicar la solución del EV. En el Cuadro 2.10 se detallan estas utilidades. Con las decisiones del EV se tiene problemas de liquidez, en los escenarios 8-16.

Escenario	Utilidad	Escenario	Utilidad
1	0,05965	9	0,05431
2	0,05965	10	0,05431
3	0,05965	11	0,05419
4	0,05455	12	0,05337
5	0,05432	13	0,04197
6	0,05432	14	0,04197
7	0,05473	15	0,03474
8	0,04664	16	0,02008
EEV=0,049906			

Cuadro 2.10: Utilidades para cada escenario obtenidas para con la solución del EV.

El costo por ignorar la incertidumbre en el modelo, es el valor $VSS = 0,049906 - 0,050876 = -0,000970$, es decir, se perdería \$970 mil.

Por la discretización de la variable, se pierde en la solución del problema estocástico. Se puede obtener mejores resultados si se tuviera más subintervalos que discretizan el flujo neto. Sin embargo, el número de escenarios crece enormemente, con lo cual, el problema resulta más extenso y requiere más recursos computacionales y tiempo. Además en un horizonte de tiempo más extenso las estimaciones del flujo de dinero acumulan mayor error haciendo la solución del problema menos confiable.

Por esto se proponen algoritmos en-línea. Los cuales no requieren el conocimiento de las funciones de distribución para las variables sujetas a incertidumbre. Estos algoritmos se detallan en el siguiente capítulo.

Capítulo 3

ALGORITMOS EN-LÍNEA PROPUESTOS PARA EL PGSC

La elevada complejidad del modelo estocástico multietapa propuesto en el capítulo anterior para el problema de gestión de saldo en caja en el BCE hace imposible su implementación directa aún para horizontes de tiempo relativamente pequeños (por ejemplo 8 días).

Un enfoque de solución alternativo consiste en el empleo de algoritmos de optimización en-línea. Un algoritmo es llamado *en-línea* si toma una decisión (cálculo de una solución parcial) en cualquier tiempo en que un nuevo conjunto de datos requiere de una acción [8]. Estos algoritmos producen una secuencia de decisiones. Estas decisiones deben ser tomadas en base a eventos pasados sin información segura sobre el futuro. Para estudiar el desempeño de un algoritmo en-línea en cada secuencia de datos entrantes se compara su funcionamiento con el de un algoritmo óptimo, llamado *fuera-de-línea*. [10].

En la presente tesis se proponen y evalúan, mediante simulaciones computacionales, algunos algoritmos en-línea de tipo heurístico, para determinar cuándo y cuánto transferir desde/hacia el extranjero en base al saldo actual en caja.

En las siguientes secciones se describen los algoritmos en-línea que han sido estudiados: algoritmos de *bandas fijas*, uno de ellos sugerido por consultores del Banco Central; algoritmos de *bandas móviles*, que son modificaciones de los primeros algoritmos y finalmente un algoritmo que emplea técnicas de programación estocástica para la toma de decisiones.

En la mayoría de casos, estos algoritmos dependen de ciertos parámetros de entrada que deben ser calibrados para cada aplicación específica. En la sección 3.5 se detalla el proceso de calibración empleado para el PGSC en el Banco Central.

3.1. ALGORITMOS DE BANDAS FIJAS

La principal característica de estos algoritmos es que mantienen constantes los valores de las bandas durante el horizonte de planificación.

3.1.1. ALGORITMO 4-BANDAS

ENTRADA:

C : saldo actual en caja; E : saldo actual en el exterior y L : saldo mínimo.

SALIDA:

I : monto a invertir en el extranjero y M : monto a retirar del extranjero.

PARÁMETROS DE CALIBRACIÓN:

$b_{\text{máx}}$, $b_{\text{mín}}$: límites superior e inferior para el saldo en caja.

b_{low} , b_{high} : bandas intermedias baja y alta.

Tal que $L < b_{\text{mín}} < b_{\text{low}} < b_{\text{high}} < b_{\text{máx}}$

si $C > b_{\text{máx}}$ **entonces**

$I := C - b_{\text{high}}$; $M := 0$

caso contrario, si $C < b_{\text{mín}}$ **entonces**

$M := \text{mín}\{b_{\text{low}} - C, E\}$; $I := 0$

caso contrario

$I := 0$; $M := 0$

fin si

Algoritmo 3.1: 4-Bandas

4-Bandas transfiere dinero al exterior cada vez que el saldo en caja supera un valor límite $b_{\text{máx}}$ conocido como *banda máxima* y retira dinero del exterior cuando éste cae por debajo de una cierta *banda mínima* $b_{\text{mín}}$. Los montos de inversión y retiro no están pre-establecidos, se calculan de acuerdo al valor del saldo en caja. Para ello, 4-Bandas emplea dos bandas conocidas como banda intermedia baja b_{low} y banda intermedia alta b_{high} . Cuando se envía dinero, el monto de inversión es la diferencia entre el saldo en caja y b_{high} . Cuando se retira dinero, el monto a pedir se calcula a partir de la diferencia entre b_{low} y el saldo en caja actual. Así, entre más grande es el saldo en exceso sobre la banda máxima, más se invierte en el extranjero. De igual forma, cuando el saldo cae por debajo de la banda mínima, el monto del pedido a retirar será mayor mientras más pequeño sea el saldo en caja. Obviamente, los montos de dinero a invertir o retirar deben ser máximo las cantidades de dinero disponible en el saldo en el exterior o en el saldo en caja respectivamente. El funcionamiento de 4-Bandas se detalla en el cuadro Algoritmo 3.1. Cabe señalar que este algoritmo fue sugerido por expertos del Banco Central.

3.1.2. ALGORITMO 2-BANDAS

ENTRADA:

C : saldo actual en caja; E : saldo actual en el exterior y L : saldo mínimo.

SALIDA:

I : monto a invertir en el extranjero y M : monto a retirar del extranjero.

PARÁMETROS DE CALIBRACIÓN:

$b_{\text{máx}}$, $b_{\text{mín}}$: límites superior e inferior para el saldo en caja, tal que $L < b_{\text{mín}} < b_{\text{máx}}$.

a_1 , a_2 : montos de inversión y retiro.

si $C > b_{\text{máx}}$ **entonces**

$I := \text{mín}\{a_1, C - L\}; \quad M := 0$

caso contrario, si $C < b_{\text{mín}}$ **entonces**

$M := \text{mín}\{a_2, E\}; \quad I := 0$

caso contrario

$I := 0; \quad M := 0$

fin si

Algoritmo 3.2: 2-Bandas

Al igual que en el caso anterior, el algoritmo 2-Bandas invierte una cantidad fija a_1 en el extranjero cada vez que el saldo en caja supera el valor límite de la banda máxima $b_{\text{máx}}$ y retira del extranjero una cantidad fija a_2 cuando el saldo en caja cae por debajo de la banda mínima $b_{\text{mín}}$. Siempre y cuando la disponibilidad de los saldos en el exterior y en caja permitan realizar estos montos de inversión y retiro.

Notar que a diferencia de 4-Bandas en 2-Bandas los montos ha enviarse/retirarse del extranjero están ya pre-determinados. El cuadro Algoritmo 3.2 describe el funcionamiento de 2-Bandas.

El comportamiento del algoritmo depende del valor de los parámetros de las bandas $b_{\text{mín}}$, $b_{\text{máx}}$ y de los montos de transferencia a_1 y a_2 . Como ya se indicó anteriormente, estos valores deben ser calibrados para cada aplicación práctica.

3.1.3. ALGORITMO LINEAL

Este algoritmo trabaja de manera similar a 2-Bandas con la diferencia de que los montos de inversión y retiros, no son constantes sino que dependen linealmente de la diferencia entre el saldo en caja y las bandas. Así, cuando el saldo es superior a la banda máxima se envía la cantidad $f_1(x) = a_1x$, donde a_1 es un *factor de inversión constante* y x es la diferencia entre el saldo en caja y el valor de la banda máxima. Si el saldo en caja es inferior a la banda mínima se realiza un pedido por el monto $f_2(x) = a_2x$, donde a_2 es un *factor de retiro constante* y x es la diferencia entre la banda mínima y el saldo en caja. De esta manera, las inversiones realizadas son más grandes conforme el saldo en caja es más elevado y se retiran mayores montos de dinero cuando el saldo en caja es bastante inferior a la banda mínima.

Lineal se detalla en el cuadro Algoritmo 3.3. Notar que el algoritmo emplea como parámetros los valores de las bandas y los factores de inversión y retiro.

ENTRADA:

C : saldo actual en caja; E : saldo actual en el exterior y L : saldo mínimo.

SALIDA:

I : monto a invertir en el extranjero y M : monto a retirar del extranjero.

PARÁMETROS DE CALIBRACIÓN:

$b_{\text{máx}}$, $b_{\text{mín}}$: límites superior e inferior para el saldo en caja, tal que $L < b_{\text{mín}} < b_{\text{máx}}$.

a_1 , a_2 : factores de inversión y retiro.

si $C > b_{\text{máx}}$ **entonces**

$I := \min\{a_1(C - b_{\text{máx}}), C - L\}; \quad M := 0$

caso contrario, si $C < b_{\text{mín}}$ **entonces**

$M := \min\{a_2(b_{\text{mín}} - C), E\}; \quad I := 0$

caso contrario

$I := 0; \quad M := 0$

fin si

Algoritmo 3.3: Lineal

3.1.4. ALGORITMO CUADRÁTICO

Este algoritmo es similar al algoritmo Lineal, pero los montos a enviar o retirar se determinan mediante una función cuadrática. Cuando el saldo es superior a la banda máxima se envía al exterior la cantidad $f_1(x) = a_1x^2$, donde a_1 es un factor de inversión y x es la diferencia entre el saldo en caja y el valor de la banda máxima. De igual forma, si el saldo en caja es inferior a la banda mínima se realiza un pedido por el monto de $f_2(x) = a_2x^2$, donde a_2 es el factor de retiro y x es la diferencia entre la banda mínima y el saldo en caja. Los montos de dinero ha invertir o retirar deben ser máximo las cantidades de dinero disponible en el saldo en el exterior o en el saldo en caja respectivamente.

Cuadrático se detalla en el cuadro Algoritmo 3.4. Requiere como parámetros los valores de las bandas y los factores de inversión y retiro.

ENTRADA:

C : saldo actual en caja; E : saldo actual en el exterior y L : saldo mínimo.

SALIDA:

I : monto a invertir en el extranjero y M : monto a retirar del extranjero.

PARÁMETROS DE CALIBRACIÓN:

$b_{\text{máx}}$, $b_{\text{mín}}$: límites superior e inferior para el saldo en caja, tal que $L < b_{\text{mín}} < b_{\text{máx}}$.

a_1 , a_2 : factores de inversión y retiro.

si $C > b_{\text{máx}}$ **entonces**

$$I := \min\{a_1(C - b_{\text{máx}})^2, C - L\}; \quad M := 0$$

caso contrario, si $C < b_{\text{mín}}$ **entonces**

$$M := \min\{a_2(b_{\text{mín}} - C)^2, E\}; \quad I := 0$$

caso contrario

$$I := 0; \quad M := 0$$

fin si

Algoritmo 3.4: Cuadrático

3.2. ALGORITMOS DE BANDAS MÓVILES

Trabajar con bandas máxima y mínima constantes puede resultar muy poco flexible en la práctica. Por ello, los siguientes algoritmos determinan las bandas

en base al *saldo base* admisible en caja, el mismo que se considera un dato de entrada de cada instancia.

3.2.1. ALGORITMO 4-BANDAS MÓVIL

Es similar a 4-Bandas excepto que los valores de las cuatro bandas se fijan respecto a un saldo base variable. El algoritmo tiene cuatro parámetros de calibración A_1 , A_2 , B_1 y B_2 . En cada ejecución, recibe como datos de entrada el saldo en caja y un saldo base \tilde{L} . 4-Bandas móvil decide invertir una cantidad igual a la diferencia entre el saldo en caja y la banda intermedia alta $B_1 + \tilde{L}$ cuando el saldo en caja supera la banda máxima $A_1 + \tilde{L}$. Por el contrario, si el saldo en caja es inferior a la banda mínima $A_2 + \tilde{L}$, el algoritmo retira del extranjero un monto determinado por la diferencia entre la banda intermedia baja $B_2 + \tilde{L}$ y el saldo en caja.

El cuadro Algoritmo 3.5 describe en detalle el funcionamiento de 4-Bandas Móvil.

ENTRADA:

C : saldo actual en caja; E : saldo actual en el exterior; \tilde{L} : saldo base y L : saldo mínimo.

Donde $\tilde{L} \geq L$.

SALIDA:

I : monto a invertir en el extranjero y M : monto a retirar del extranjero.

PARÁMETROS DE CALIBRACIÓN:

A_1, A_2, B_1 y B_2 tal que $A_2 < B_2 < B_1 < A_1$.

si $C > \tilde{L} + A_1$ **entonces**

$I := C - (\tilde{L} + B_1); \quad M := 0$

caso contrario, si $C < \tilde{L} + A_2$ **entonces**

$M := \min\{(\tilde{L} + B_2) - C, E\}; \quad I := 0$

caso contrario

$I := 0; \quad M := 0$

fin si

Algoritmo 3.5: 4-Bandas Móvil

3.2.2. ALGORITMO 2-BANDAS MÓVIL

Este algoritmo trabaja de manera similar que 2-Bandas, con la única diferencia de que los valores de las bandas máxima y mínima no son parámetros fijos, sino que se calculan en función de un saldo base. El algoritmo 2-Bandas Móvil invierte una cantidad fija a_1 en el extranjero cuando el saldo en caja supere el valor de un saldo base \tilde{L} más un monto A_1 y retira una cantidad fija a_2 cuando el saldo en caja cae por debajo del saldo base más una cantidad A_2 . Los montos a retirar y invertir del exterior se realizan siempre y cuando exista la suficiente disponibilidad de dinero en los saldo en el exterior y en caja respectivamente.

El cuadro Algoritmo 3.6 muestra más detalles del funcionamiento de 2-Bandas móvil. Los parámetros A_1, A_2, a_1, a_2 son fijos y deben calibrarse previo el uso del algoritmo. Por otro lado, tanto el saldo en caja como el saldo base \tilde{L} son datos de entrada en cada corrida del algoritmo.

ENTRADA:

C : saldo actual en caja; E : saldo actual en el exterior; \tilde{L} : saldo base y L : saldo mínimo.

Donde $\tilde{L} \geq L$.

SALIDA:

I : monto a invertir en el extranjero y M : monto a retirar del extranjero.

PARÁMETROS DE CALIBRACIÓN:

A_1, A_2 : cantidades sumadas al saldo base para realizar inversiones y retiros, tal que $A_2 < A_1$.

a_1, a_2 : montos de inversión y retiro.

si $C > \tilde{L} + A_1$ **entonces**

$I := \min\{a_1, C - L\}; \quad M := 0$

caso contrario, si $C < \tilde{L} + A_2$ **entonces**

$M := \min\{a_2, E\}; \quad I := 0$

caso contrario

$I := 0; \quad M := 0$

fin si

Algoritmo 3.6: 2-Bandas Móvil

3.2.3. ALGORITMO LINEAL MÓVIL

Nuevamente, este algoritmo es similar a Lineal, con la diferencia que las bandas mínima y máxima están establecidas con respecto a un saldo base. Los montos de inversión y retiro son determinados por funciones lineales del exceso sobre la banda máxima o déficit respecto a la banda mínima. Igualmente, los montos a retirar y invertir del exterior se realizan siempre y cuando exista la suficiente disponibilidad de dinero en los saldo en el exterior y en caja respectivamente.

El cuadro Algoritmo 3.7 describe los detalles.

ENTRADA:

C : saldo actual en caja; E : saldo actual en el exterior; \tilde{L} : saldo base y L : saldo mínimo.

Donde $\tilde{L} \geq L$.

SALIDA:

I : monto a invertir en el extranjero y M : monto a retirar del extranjero.

PARÁMETROS DE CALIBRACIÓN:

A_1, A_2 : cantidades sumadas al saldo base para realizar inversiones y retiros, tal que $A_2 < A_1$.

a_1, a_2 : factores de inversión y retiro.

si $C > \tilde{L} + A_1$ **entonces**

$$I := \min\{a_1(C - (\tilde{L} + A_1)), C - L\}; \quad M := 0$$

caso contrario, si $C < \tilde{L} + A_2$ **entonces**

$$M := \min\{a_2(\tilde{L} + A_2 - C), E\}; \quad I := 0$$

caso contrario

$$I := 0; \quad M := 0$$

fin si

Algoritmo 3.7: Lineal Móvil

3.2.4. ALGORITMO CUADRÁTICO MÓVIL

Finalmente, consideramos una variante del algoritmo Cuadrático donde las bandas máxima y mínima se determinan sobre un saldo base variable. El cuadro Algoritmo 3.8 ilustra el funcionamiento de este algoritmo.

Tanto los algoritmos fijos como móviles tienen una característica en común. Basan sus decisiones sobre el monto de dinero a enviar o pedir únicamente en el saldo actual en caja y en el caso de los algoritmos móviles en un saldo base. Son métodos sencillos, fáciles de implementar y eficientes computacionalmente. Sin embargo, el uso limitado de información de entrada puede tener impacto negativo sobre la calidad de la solución. Por ejemplo, un algoritmo puede retirar dinero en el día t debido a que el saldo en caja es muy bajo. Como la transferencia demora dos días en hacerse efectivo, el flujo neto diario de dinero puede causar que durante este tiempo el saldo en caja aumente de tal forma que en el día $t+2$ sea necesario *re-enviar* al extranjero gran parte del dinero recibido. Más peligrosa es la situación inversa: un envío de dinero del algoritmo es seguido por fuertes débitos en el sistema interno, poniendo en riesgo la liquidez del mismo.

ENTRADA:

C : saldo actual en caja; E : saldo actual en el exterior; \tilde{L} : saldo base y L : saldo mínimo.

Donde $\tilde{L} \geq L$.

SALIDA:

I : monto a invertir en el extranjero y M : monto a retirar del extranjero

PARÁMETROS DE CALIBRACIÓN:

A_1, A_2 : cantidades sumadas al saldo base para realizar inversiones y retiros,

tal que $A_2 < A_1$.

a_1, a_2 : factores de inversión y retiro.

si $C > \tilde{L} + A_1$ **entonces**

$$I := \min\{a_1(C - (\tilde{L} + A_1))^2, C - L\}; \quad M := 0$$

caso contrario, si $C < \tilde{L} + A_2$ **entonces**

$$M := \min\{a_2(\tilde{L} + A_2 - C)^2, E\}; \quad I := 0;$$

caso contrario

$$I := 0; \quad M := 0$$

fin si

Algoritmo 3.8: Cuadrático Móvil

Si bien estos son problemas inherentes a la presencia de incertidumbre en el modelo, es natural preguntarse si el empleo de técnicas estocásticas puede ayu-

dar a obtener soluciones más robustas. A continuación, se describe un algoritmo basado en este enfoque.

3.3. ALGORITMO ESTOCÁSTICO

ENTRADA:

C : saldo en caja

E : saldo en el exterior

h : número de días

$L_t, t = 1, \dots, h$: saldos mínimo para los h días

$r_t, t = 1, \dots, h$: tasas de retorno para los h días

$c_t^M, t = 1, \dots, h$: tasa por retirar dinero del extranjero para los h días

$c_t^I, t = 1, \dots, h$: tasa por enviar dinero al extranjero para los h días

S : conjunto de escenarios

$p(s), \forall s \in S$: probabilidades de los escenarios

SALIDA:

I : monto a invertir en el extranjero

M : monto a retirar del extranjero

hacer

$C_0 := C, E_0 := E,$

Formular y Resolver el PEGSC para h días. (sección 2.3)

Obtener las decisiones para el primer día I_1 y M_1 .

Asignar

$M := M_1, I := I_1$

fin

Algoritmo 3.9: Estocástico

Estocástico recibe como entrada, toda la información necesaria para formular el modelo estocástico para el PGSC de la sección 2.3 para un horizonte de planificación de h días. Así, la información de entrada es un conjunto de escenarios

S , las tasas de retorno, las tasas por enviar/retirar dinero del extranjero, los saldos mínimos y saldos en el exterior y en caja, que servirán como saldo iniciales en la formulación del modelo estocástico. Estocástico al resolver el problema bajo incertidumbre, obtiene las decisiones de cuánto enviar/retirar del extranjero para cada escenario durante los h días. La decisión tomada en el primer día es siempre la misma para cada escenario, debido a que el PEGSC supone que el flujo neto en el primer día es conocido y la incertidumbre empieza a partir del segundo día. Por ejemplo, si los escenarios son generados mediante un árbol, el flujo neto del primer día es conocido y es la raíz del árbol. El algoritmo retorna el monto a enviar/retirar del extranjero del primer día encontrado al resolver PEGSC para el horizonte de h días. El cuadro Algoritmo 3.9 describe el funcionamiento de Estocástico. Cabe notar que este algoritmo no tiene parámetros de calibración.

Estocástico no toma decisiones basadas únicamente en los saldos exterior y en caja que se presentan en un cierto día. Sus decisiones tienen presente lo que podría suceder en los siguientes $h - 1$ días. Es decir, en base a todas las variables del problema como son las tasas de retorno, las tasas por enviar/retirar dinero, los saldos mínimos, los saldos en el extranjero y en caja disponibles, y principalmente por medio de la inclusión de los escenarios para el flujo neto incluye la incertidumbre al problema y a sus decisiones.

Como ya se mencionó en el Capítulo 2, plantear y resolver el modelo estocástico para un horizonte de planificación grande resulta muy complicado, debido al tamaño del problema y a lo difícil que resulta construir un buen árbol de escenarios (a causa del error en las estimaciones del flujo neto). Sin embargo, se presenta como alternativa a Estocástico, que va resolviendo día a día el PEGSC para un horizonte de tiempo manejable, permitiendo actualizar la información diariamente y así construir mejores escenarios y tomar decisiones más oportunas.

Con el fin de aclarar los algoritmos en-línea propuestos, en la siguiente sección se presenta como ejemplo la aplicación de los algoritmos para una instancia real de 30 días.

3.4. EJEMPLO DE APLICACIÓN

Para comprender y observar mejor el comportamiento de los algoritmos propuestos se presenta el desarrollo de estos en una instancia real. Esta instancia está formada por el flujo neto del Banco Central que se observó durante 30 días

desde el 24 de Enero del 2005 hasta el 4 de Marzo del mismo año. Se suponen que las tasas de retorno son constantes e iguales para el horizonte de planificación, $r_t = 0,00015$, $t = 1...T$, las tasas de costos por invertir o retirar dinero del extranjero son constantes e iguales, $c_t^M = c_t^I = 0,0002$, $t = 1...T$ y los saldos iniciales en caja y exterior son $E_0 = 20$ y $C_0 = 140,844$ millones de dólares.

En el Cuadro 3.1 se presentan los valores del flujo neto y su respectivo saldo mínimo para los treinta días.

Día	Fecha	Saldo mínimo	Flujo	Día	Fecha	Saldo mínimo	Flujo
1	24-01-05	43,073	2,186	16	14-02-05	54,672	-7,416
2	25-01-05	43,073	20,591	17	15-02-05	54,672	31,718
3	26-01-05	43,073	13,511	18	16-02-05	54,672	11,407
4	27-01-05	43,073	0,204	19	17-02-05	54,672	-0,892
5	28-01-05	43,073	0,28	20	18-02-05	54,672	5,518
6	31-01-05	56,351	-5,351	21	21-02-05	95,905	1,187
7	01-02-05	56,351	10,348	22	22-02-05	95,905	18,196
8	02-02-05	56,351	1,233	23	23-02-05	95,905	15,175
9	03-02-05	56,351	-15,997	24	24-02-05	95,905	1,961
10	04-02-05	56,351	-32,482	25	25-02-05	95,905	-2,673
11	03-02-03	59,215	-8,762	26	28-02-05	58,094	-8,548
12	04-02-03	59,215	3,764	27	01-03-05	58,094	13,813
13	09-02-05	59,215	-9,279	28	02-03-05	58,094	5,399
14	10-02-05	59,215	23,808	29	03-03-05	58,094	-10,382
15	11-02-05	59,215	14,288	30	04-03-05	58,094	-4,794

Cuadro 3.1: Flujo y saldo mínimo para la instancia real de 30 días.

El desempeño de los algoritmos depende en gran medida de los valores utilizados para los parámetros de calibración. En este ejemplo se trabaja con valores seleccionados arbitrariamente. En la siguiente sección se considera el problema de determinar valores más adecuados para los parámetros.

En el Cuadro 3.2 se presentan los valores de los parámetros utilizados para la aplicación de los algoritmos de bandas móviles y fijas.

Algoritmos de Bandas Fijas				
4-Bandas	$b_{\text{máx}} = 130$	$b_{\text{mín}} = 90$	$b_{\text{high}} = 120$	$b_{\text{low}} = 100$
2-Bandas	$b_{\text{máx}} = 130$	$b_{\text{mín}} = 90$	$a_1 = 40$	$a_2 = 40$
Lineal	$b_{\text{máx}} = 130$	$b_{\text{mín}} = 90$	$a_1 = 1$	$a_2 = 1$
Cuadrático	$b_{\text{máx}} = 130$	$b_{\text{mín}} = 90$	$a_1 = 0,054$	$a_2 = 0,05$
Algoritmos de Bandas Móviles				
4-Bandas Móvil	$A_1 = 80$	$A_2 = 40$	$B_1 = 65$	$B_2 = 50$
2-Bandas Móvil	$A_1 = 80$	$A_2 = 40$	$a_1 = 40$	$a_2 = 40$
Lineal Móvil	$A_1 = 80$	$A_2 = 40$	$a_1 = 1$	$a_2 = 1$
Cuadrático Móvil	$A_1 = 80$	$A_2 = 40$	$a_1 = 0,05$	$a_2 = 0,05$

Cuadro 3.2: Parámetros utilizados para los algoritmos.

Para los algoritmos de bandas fijas se establece las bandas mínima en 90 millones y máxima en 130 millones. Para los algoritmos de bandas móviles se establece un límite máximo de 80 millones sobre el saldo base y un límite mínimo de 40 millones sobre el saldo base. Los parámetros de los algoritmos difieren en los montos de los envíos y retiros del extranjero. Además, en los algoritmos de bandas móviles se asume que el saldo base es igual al saldo mínimo durante los días de aplicación de estos algoritmos.

Para observar el desempeño de los algoritmos, en el Cuadro 3.3 se presenta el saldo en caja junto con los montos de depósitos y retiros obtenidos al aplicar cada uno de los algoritmos de bandas fijas a la instancia del ejemplo. En la columna de In/Rt se presenta simultáneamente las inversiones y retiros del extranjero, los retiros se significan con signo negativo (-). Al final del cuadro se registran las utilidades alcanzadas por los algoritmos.

El flujo neto de dinero en el primer día es de 2,186 entonces se tiene un saldo en caja disponible al final del día igual al saldo en caja inicial más el flujo neto, $140,844 + 2,186 = 143,03$. Como este saldo en caja supera el valor de la banda máxima $b_{\text{máx}}$ ($143,03 > 130$) los algoritmos de bandas fijas deciden invertir dinero en el extranjero el siguiente día. 2-Bandas envía un monto de 40 millones de dólares. 4-Bandas envía un monto igual a la diferencia entre el saldo en caja y el valor de $b_{\text{high}} = 120$, es decir invierte $23,03 = 143,03 - 120$ millones de dólares. Lineal envía un monto igual a $13,03 = 1 * (143,03 - 130)$, la diferencia entre el saldo en caja y

Día	4-Bandas		2-Bandas		2-B Lineal		2-B Cuadrático	
	S. Caja	Iv/Rt	S. Caja	In/Rt	S. Caja	Iv/Rt	S. Caja	Iv/Rt
1	143,030	0,00	143,030	0	143,030	0,00	143,030	0,00
2	140,591	23,03	123,621	40	150,591	13,03	154,453	9,17
3	133,511	20,59	137,132	0	143,511	20,59	135,675	32,29
4	120,204	13,51	97,336	40	130,204	13,51	134,140	1,74
5	120,484	0,00	97,616	0	130,280	0,20	133,494	0,93
6	115,133	0,00	92,265	0	124,649	0,28	127,484	0,66
7	125,481	0,00	102,613	0	134,997	0,00	137,832	0,00
8	126,714	0,00	103,846	0	131,233	5,00	135,753	3,31
9	110,717	0,00	87,849	0	114,003	1,23	117,969	1,79
10	78,235	0,00	55,367	-40	81,521	0,00	85,487	0,00
11	69,473	-21,77	46,605	-40	72,759	-8,48	76,725	-1,02
12	73,237	-30,53	90,369	-20	76,523	-17,24	80,489	-8,81
13	85,723	-24,84	121,090	0	75,723	-13,48	72,228	-4,52
14	140,058	0,00	164,898	0	116,772	-14,28	104,848	-15,79
15	159,128	20,06	139,186	40	144,537	0,00	123,659	0,00
16	112,584	39,13	91,770	40	136,861	14,54	132,035	0,00
17	144,302	0,00	123,488	0	161,718	6,86	163,530	0,22
18	131,407	24,30	134,895	0	141,407	31,72	114,228	60,71
19	119,108	11,41	94,003	40	129,108	11,41	113,336	0,00
20	124,626	0,00	99,521	0	134,626	0,00	118,854	0,00
21	125,813	0,00	100,708	0	131,187	4,63	120,041	0,00
22	144,009	0,00	118,904	0	148,196	1,19	138,237	0,00
23	135,175	24,01	134,079	0	145,175	18,20	149,748	3,66
24	121,961	15,18	96,040	40	131,961	15,18	130,650	21,06
25	119,288	0,00	93,367	0	127,327	1,96	127,954	0,02

26	110,740	0,00	84,819	0	118,779	0,00	119,406	0,00
27	124,553	0,00	98,632	-40	132,592	0,00	133,219	0,00
28	129,952	0,00	104,031	0	135,399	2,59	138,058	0,56
29	119,570	0,00	133,649	0	125,017	0,00	127,676	0,00
30	114,776	0,00	128,855	0	120,223	0,00	122,882	0,00
utilidad:		0,2498	0,2510		0,2507		0,2868	

Cuadro 3.3: Desempeño de los algoritmos de bandas fijas para una instancia real de 30 días de duración.

el valor de la banda máxima multiplicado por el factor de inversión (1). Cuadrático invierte un monto igual a la diferencia al cuadrado entre el saldo en caja y $b_{\text{máx}}$ multiplicada por el factor de inversión (0,054), invierte $9,17 = 0,054 * (143,03 - 130)^2$.

Al final del noveno día, 2-Bandas decide realizar el primer retiro en el décimo día, debido a que el flujo neto del noveno día, $-15,997$, más el saldo en caja dejado en el octavo día, $103,846$, suman $87,85$ millones de dólares, siendo esta suma inferior a la banda mínima $b_{\text{mín}} = 90$. 2-Bandas retira en el décimo día el monto de 40 millones de dólares. Los demás algoritmos realizan su primer retiro en el día 11.

El saldo en caja al final del décimo día en el desarrollo de 4-Bandas es $78,235$, la suma entre el saldo en caja del noveno día, $110,717$ y el valor del flujo en el décimo día, $-32,482$. Como este saldo en caja es inferior al valor de $b_{\text{mín}}$, 4-Bandas decide realizar un retiro en el día 11. El monto retirado es $21,77$, la diferencia entre $b_{\text{low}} = 100$ y el saldo en caja.

Al final del décimo día, el saldo en caja de Lineal de $81,521$, es inferior a $b_{\text{mín}}$, el algoritmo realiza un retiro en el siguiente día por el monto de $8,48 = 1 * (90 - 81,521)$, la diferencia entre la banda mínima y el saldo en caja del décimo día, multiplicada por el factor de inversión.

Como el saldo en caja en el décimo día de Cuadrático es inferior a la banda mínima se retira en el décimo primer día un monto de $1,02$, la diferencia al cuadrado entre la banda mínima y el saldo en caja multiplicado por el factor de retiro ($0,054 * (90 - 85,487)^2$).

Para observar el comportamiento de los algoritmos en la Figura 3.1 se muestra los retiros e inversiones realizados por los algoritmos de bandas fijas y en la Figura 3.2 se muestra la evolución del saldo en caja de los algoritmos durante los 30 días.

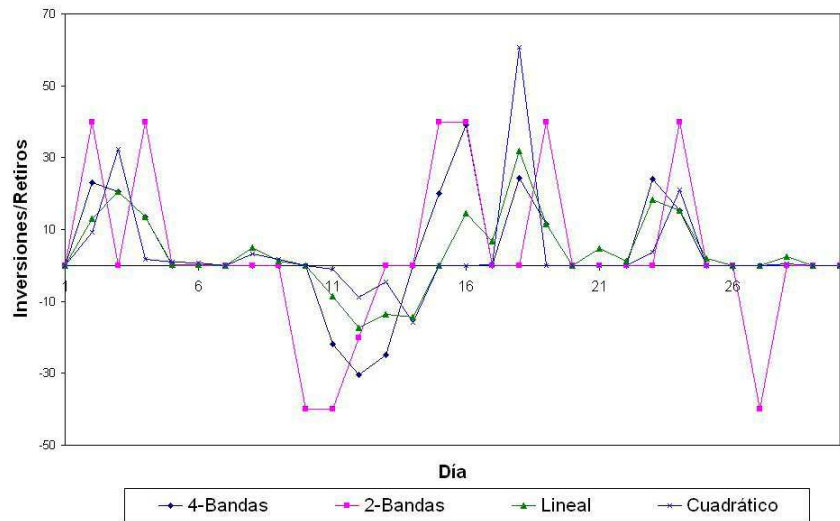


Figura 3.1: Inversiones/Retiros realizados por los algoritmos de bandas fijas.

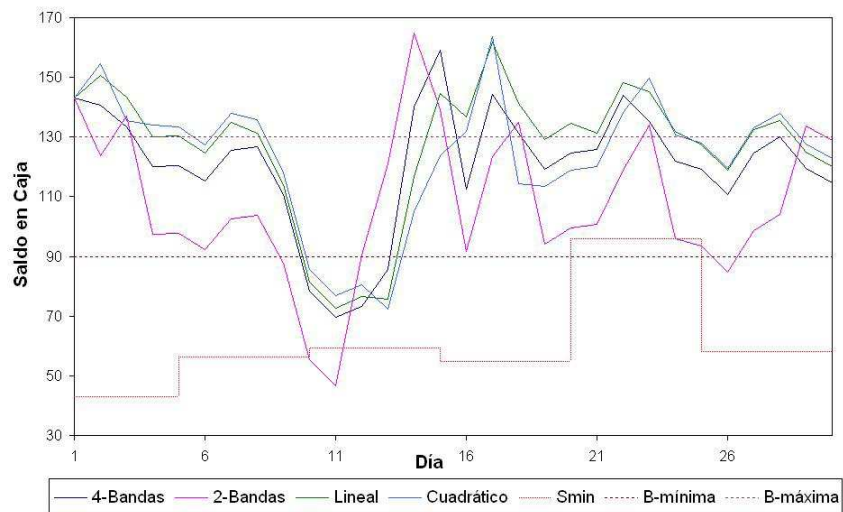


Figura 3.2: Saldos en caja obtenidos por los algoritmos de bandas fijas.

Las montos de dinero retirados e invertidos más elevados generalmente son realizados por 2-Bandas pero la inversión más elevada es hecha en el décimo octavo día por Cuadrático. Los algoritmos de bandas fijas realizan en los mismos tiempos sus inversiones y retiros. Se observa un comportamiento similar de los algoritmos en torno a las inversiones y retiros de dinero. También se observa un comportamiento similar en los saldos en caja de los algoritmos de bandas fijas. Los algoritmos de bandas fijas en los primeros cinco días realizan sus inversiones.

El flujo neto en caja en los días 9,10,11 y 13 toma valores negativos provocando que los saldo en caja caigan en los días 10, 11, 12 y 13 como indica en la Figura 3.2. Así, los algoritmos realizan retiros de dinero para evitar caer por debajo del saldo mínimo. Para 2-Bandas es inevitable fallar en la restricción de seguridad en los días 10 y 11, debido a los montos elevados de dinero enviado al extranjero. Como el dinero tarda dos días en hacerse efectivo, a partir del décimo cuarto día el saldo en caja comienza a crecer y los algoritmos nuevamente realizan inversiones de dinero (re-envían parte del dinero pedido días atrás). Lo anterior refleja el problema de los algoritmos: realizar elevadas inversiones en el extranjero para luego volver a retirar montos de dinero seguidos, provocando que el saldo en caja crezca y nuevamente re-enviar el dinero, esto ocasionando elevados costos por transferencia de dinero. Lo mejor es enviar/retirar cantidades moderadas de dinero para evitar realizar grandes saltos en el saldo en caja. Encontrando los valores adecuados para los parámetros de calibración de los algoritmos propuestos, estos pueden conseguir enviar/retirar montos más conservadores.

De los algoritmos de bandas fijas Cuadrático y 2-Bandas obtienen las mayores utilidades. 2-Bandas realiza menos inversiones que los otros dos algoritmos, pero las cantidades invertidas en los primeros días son más elevadas. A pesar de la utilidad obtenida 2-Bandas es el algoritmo que tuvo la mayor cantidad de días bajo el saldo mínimo. Las inversiones y retiros realizados por Cuadrático son más pequeñas debido a la función que utiliza para decidir cuánto dinero enviar y pedir del extranjero. De esta forma obtienen menores costos por transferencias y, además, no tiene problemas de liquidez.

En el Cuadro 3.4 se presenta el saldo en caja y las inversiones/retiros realizados por los algoritmos de bandas móviles, además, al final del cuadro se indica la utilidad obtenida por los algoritmos para esta instancia.

Se utilizó como saldo base el saldo mínimo dado. El saldo en caja al final del primer día 143,03 millones de dólares, que supera el saldo base de 43,073 más el monto $A_1 = 80$, entonces, los algoritmos bandas fijas deciden invertir en el segundo día. 2-Bandas móvil envía la cantidad de 40 millones de dólares. 4-Bandas envía la diferencia entre el saldo en caja y la banda intermedia alta $B_1 + \tilde{L} = 65 + 43,073$, es decir envía un monto de $34,96 = 143,03 - (65 + 43,073)$. Lineal Móvil invierte un monto igual a $34,96 = 1 * (143,03 - (43,073 + 80)) = 19,96$, la diferencia entre el saldo en caja y la adición del saldo base y la cantidad $B_1 = 65$, multiplicado por el factor de inversión. Cuadrático móvil invierte $19,91 = 0,05(143,03 - (65 + 43,073))^2$ el producto entre el factor de inversión y la diferencia entre el saldo en caja y la

Día	4-Bandas Móvil		2-Bandas Móvil		Lineal Móvil		Cuadrático M.	
	S. Caja	Iv/Rt	S. Caja	In/Rt	S. Caja	Iv/Rt	S. Caja	Iv/Rt
1	143,030	0,00	143,030	0	143,030	0,00	143,030	0,00
2	128,664	34,96	123,621	40	143,664	19,96	143,707	19,91
3	121,584	20,59	97,132	40	136,584	20,59	135,930	21,29
4	121,788	0,00	97,336	0	123,277	13,51	127,869	8,27
5	122,068	0,00	97,616	0	123,353	0,20	126,999	1,15
6	116,717	0,00	92,265	0	117,722	0,28	120,877	0,77
7	127,065	0,00	102,613	-40	128,070	0,00	131,225	0,00
8	128,298	0,00	103,846	0	129,303	0,00	132,458	0,00
9	112,301	0,00	127,849	0	113,306	0,00	116,461	0,00
10	79,819	0,00	95,367	0	80,824	0,00	83,979	0,00
11	71,057	-26,53	86,605	-40	72,062	-15,53	75,217	-7,65
12	74,821	-38,16	90,369	-20	75,826	-27,15	78,981	-28,79
13	92,074	-10,86	121,090	0	82,074	-23,39	77,355	-20,47
14	154,040	0,00	164,898	0	133,035	-8,47	129,958	-14,47
15	149,361	29,83	139,186	40	170,712	0,00	164,716	0,00
16	116,799	25,15	131,770	0	140,273	31,50	139,255	32,52
17	148,517	0,00	163,488	0	166,390	5,60	169,923	1,05
18	131,079	28,85	134,895	40	146,079	31,72	119,199	62,13
19	130,187	0,00	94,003	40	133,780	11,41	118,307	0,00
20	135,705	0,00	99,521	-40	139,298	0,00	123,825	0,00
21	120,859	16,03	100,708	0	135,859	4,63	125,012	0,00
22	139,055	-25,05	158,904	-40	154,055	-0,05	143,208	-5,93
23	154,230	0,00	174,079	0	169,230	0,00	158,383	0,00
24	181,237	0,00	216,040	0	171,237	0,00	166,277	0,00

25	158,232	20,33	173,367	40	168,564	0,00	163,604	0,00
26	149,684	0,00	164,819	0	160,016	0,00	155,056	0,00
27	136,907	26,59	138,632	40	151,907	21,92	154,484	14,39
28	142,306	0,00	104,031	40	143,493	13,81	146,452	13,43
29	131,924	0,00	93,649	0	133,111	0,00	136,070	0,00
30	127,130	0,00	88,855	0	128,317	0,00	131,276	0,00
utilidad:	0,1985	0,1190	0,1999	0,2100				

Cuadro 3.4: Desempeño de los algoritmos de bandas móviles para una instancia real de 30 días de duración.

adición del saldo base y el monto $B_1 = 65$.

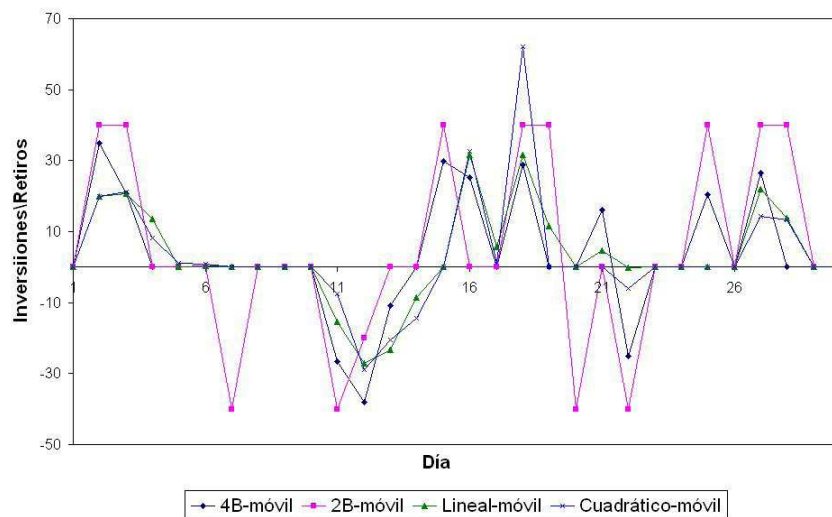


Figura 3.3: Inversiones/Retiros realizados por los algoritmos de bandas móviles.

El saldo en caja de 2-Bandas Móvil en el sexto día es de 92,265, que es inferior al monto de 96,351 la adición del saldo base en ese día 56,351 y la cantidad $A_2 = 40$, por esta razón decide realizar un retiro de 40 millones de dólares. El saldo en caja de los demás algoritmos decae sobre la banda mínima formada por el saldo base más el monto $A_2 = 40$ en el décimo día, así los algoritmos retiran dinero en el día 11. 4-Bandas Móvil realiza un retiro de $26,53 = (56,351 + 50) - 79,819$ la diferencia entre la banda media baja formada por el saldo base más el monto $B_2 = 50$ y el saldo en caja en el décimo día. Lineal Móvil realiza un retiro igual al producto entre el factor de retiro y la diferencia de la banda mínima y el saldo en

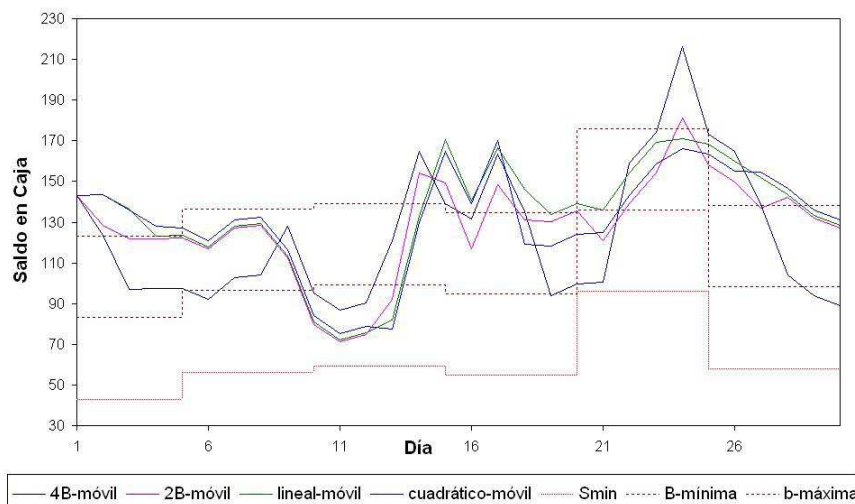


Figura 3.4: Saldos en caja obtenidos por los algoritmos de bandas móviles.

caja del décimo día $15,53 = 1 * ((56,351 + 40) - 80,824)$. Cuadrático Móvil realiza un retiro igual al producto entre el factor de retiro y la diferencia al cuadrado de la banda mínima y el saldo en caja del décimo día $7,65 = 0,05 * ((56,351 + 40) - 83,979)^2$.

En la Figura 3.3 se presentan los gráficos de las inversiones y los retiros realizados por los algoritmos de bandas móviles durante los 30 días. Los valores negativos significan los retiros de dinero del extranjero. En la Figura 3.4 se observan los diferentes saldos en caja generados por la aplicación de los algoritmos de bandas móviles.

La inversión más elevada es realizada por Cuadrático Móvil en el día 18. Sin embargo, las inversiones más fuertes de dinero son realizadas por 2-Bandas Móvil, este algoritmo realiza el primer retiro más temprano que los demás y sus retiros son de igual manera los más elevados. La mayoría de inversiones y retiros realizados por Lineal Móvil son más conservadores.

En los días 10-13 el saldo en caja decae, razón por la cual los días 14,15 y 17 se registran nuevamente valores positivos altos para el flujo, de manera que en estos días el saldo en caja crece. Esto ilustra una vez más la dificultad con anterioridad. Los pedidos se realizan en base al decaimiento del saldo en caja, pero mientras las transacciones se efectivizan puede haberse recuperado debido a depósitos locales. Como consecuencia, se tendrá un saldo en caja muy alto y deberá volver a enviarse el dinero de vuelta, causando únicamente costos de

transferencia inútiles.

Para este ejemplo todos los algoritmos de bandas móviles lograron cumplir con la restricción de seguridad durante el horizonte de planificación de 30 días. Cuadrático Móvil es el algoritmo con mayor utilidad, seguido por 2-Bandas lineal móvil.

Finalmente, en el Cuadro 3.5 se presenta el saldo en caja, junto con los montos de inversión y retiro para el algoritmo Estocástico y para la solución óptima o problema fuera-de-línea.

Día	Estocástico		Óptimo		Día	Estocástico		Óptimo	
	S. Caja	Iv/Rt	S. Caja	In/Rt		S. Caja	Iv/Rt	S. Caja	In/Rt
1	143,03	0,00	43,073	99,96	16	71,291	0,00	54,672	0,00
2	86,5408	77,08	47,707	15,96	17	103,009	0,00	79,872	6,52
3	81,1556	18,90	61,218	0,00	18	99,0891	15,33	91,279	0,00
4	80,1618	1,20	61,422	0,00	19	98,1971	-8,67	90,387	0,00
5	80,4418	-0,59	61,702	0,00	20	103,715	-26,97	95,905	0,00
6	69,2346	5,86	56,351	0,00	21	111,197	2,37	95,905	1,19
7	80,1687	0,00	66,699	-4,42	22	152,886	3,47	95,905	18,20
8	81,4017	0,00	67,932	-35,4	23	108,879	59,18	95,905	15,18
9	65,4047	-5,52	56,351	-8,76	24	88,2275	22,61	95,905	1,96
10	32,9227	-44,5	59,215	0,00	25	85,5545	0,00	82,022	11,21
11	29,6805	0,00	59,215	-5,52	26	73,452	3,55	73,474	0,00
12	77,9286	0,00	62,979	0,00	27	87,265	0,00	87,287	0,00
13	68,6496	-2,64	59,215	0,00	28	92,664	0,00	92,686	0,00
14	92,4576	-26,4	59,215	23,81	29	82,282	0,00	82,304	0,00
15	52,3346	57,05	62,088	11,42	30	77,488	0,00	77,51	0,00
utilidad de Estocástico: 0,3525 utilidad óptima:0,4709									

Cuadro 3.5: Desempeño del algoritmo estocástico y solución óptima para una instancia real de 30 días de duración.

Estocástico en el primer día resuelve el PEGSC para $h = 6$, con los datos dados de la instancia y con un árbol de escenarios formado por una distribución de probabilidad Sarima-Impulso (Apéndice A). Al resolver el PEGSC se obtiene deci-

siones sobre los montos a invertir y retirar del exterior desde el primer día hasta el sexto día. Los montos a invertir y retirar desde el segundo día dependen de cada uno de los posibles escenarios, así se tiene tantas soluciones como escenarios. Sin embargo las decisiones respecto a cuánto invertir/retirar en el primer día son generales para todos los escenarios, debido a que el problema está formulado suponiendo que todos los datos en el primer día son conocidos y la incertidumbre sobre el flujo empieza desde el segundo día. Así, en el segundo día se invierte un monto de 77,08 millones de dólares, este monto es la solución del PEGSC formulado el primer día. En el segundo día se formula un nuevo problema estocástico para los 6 días siguientes con el saldo exterior (20) y en caja (86,5408) del segundo día y el árbol de escenarios generado a partir del flujo neto del segundo día, 20,591. La solución encontrada a este problema para su primer período es 18,90, entonces está es la cantidad invertida en el tercer día. Notar que el problema estocástico se formula una vez conocido el valor del flujo neto para un cierto día y las decisiones encontradas son aplicadas en el siguiente día.

El problema fuera-de-línea se resuelve al final del horizonte de planificación una vez ya conocidos todos los valores del flujo neto. Como ya se dijo en el Capítulo 2 el PGSC se puede reducir a un problema de flujo de costo mínimo una vez conocidos los valores del flujo neto.

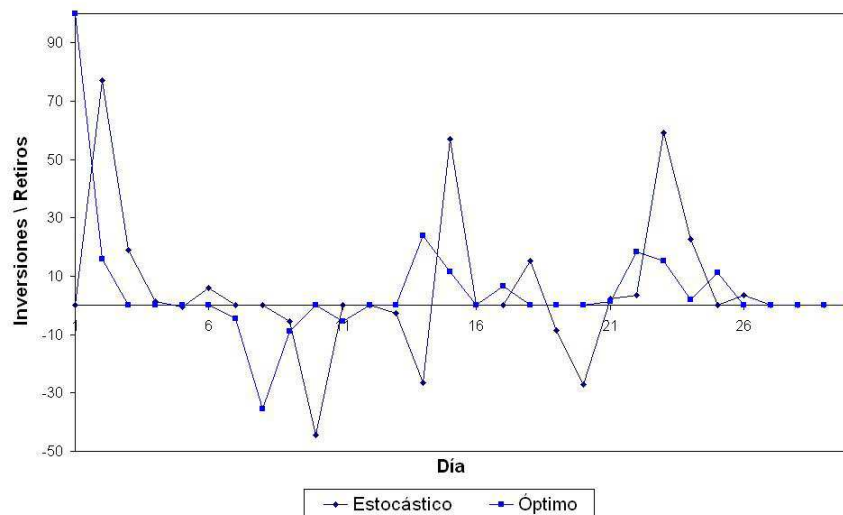


Figura 3.5: Inversiones/Retiros realizados por el algoritmo Estocástico y por la solución óptima.

En la Figura 3.5 se presenta el gráfico de las inversiones y retiros realizados

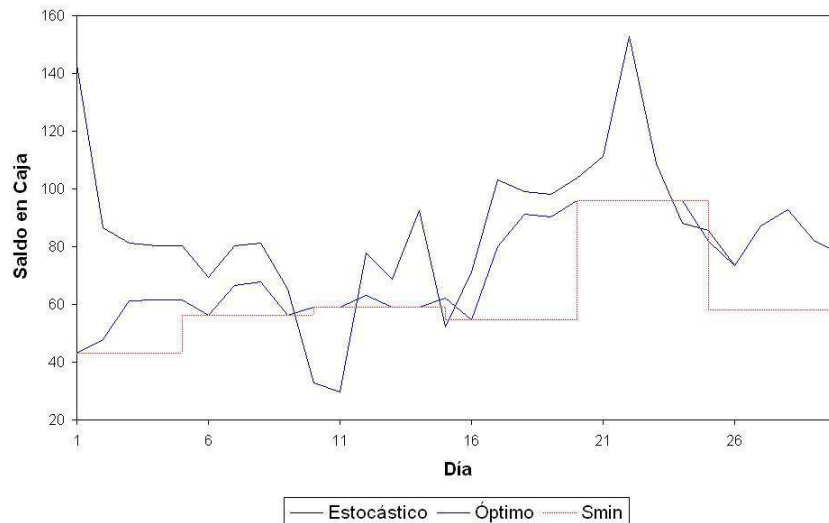


Figura 3.6: Saldos en caja obtenidos por el algoritmo Estocástico y por la solución óptima.

por el algoritmo Estocástico y por la solución del problema fuera-de-línea. En la Figura 3.6 se presenta el saldo en caja generado por el algoritmo estocástico y por el problema fuera-de-línea.

El óptimo realiza la inversión más elevada el primer día, en los siguientes días las inversiones son más moderadas y sólo tiene dos retiros, el primero es por un monto fuerte de dinero y el último es más bajo. A diferencia de los demás algoritmos que realizan un número mayor de retiros acumulando con esto costos de transacción.

El saldo en caja generado por la solución óptima a pesar de estar en algunos días en el valor límite del saldo mínimo nunca incumple con la restricción de seguridad. Debido al conocimiento del flujo neto para todos los días, la solución óptima puede realizar pedidos de dinero con la suficiente antelación para evitar incumplir con la restricción, de igual manera realiza las inversiones con los montos adecuados para obtener la mejor utilidad y evitar realizar transacciones de dinero innecesarias que sólo generan costos.

Estocástico se acerca a los límites del saldo mínimo razón por la cual en el día 9 y 10 cuando se presenta un flujo neto negativo viola la restricción de seguridad.

En general, dada la forma de la función objetivo, es más conveniente (aunque riesgoso) invertir un determinado excedente de una sola vez, que mediante varios

envíos sucesivos: el número de transacciones realizadas en el primer caso es más pequeño y, además, el dinero permanece más tiempo en el extranjero, generando más utilidades.

Dada la forma de los algoritmos propuestos estos pueden alcanzar en algunas instancias utilidades superiores a las del óptimo violando la restricción de seguridad. Cuando en un día no se cumple con la restricción de seguridad se lo llama *día malo*. Para penalizar a un día malo se supone que se puede realiza la “compra” de dinero a un costo alto, por un monto igual al faltante para cubrir el valor del saldo mínimo impuesto. Esta compra de dinero se hace sobre las decisiones tomadas por el algoritmo día a día, y así se corrige la utilidad.

En el Cuadro 3.6 se presenta un resumen de los algoritmos aplicados en este ejemplo: utilidad, utilidad corregida, número de días malos, número de inversiones y retiros, costos de transferencia y la brecha de optimalidad *gap*. El *gap* es la diferencia entre la utilidad óptima y la utilidad corregida del algoritmo dividida para la utilidad óptima.

Algoritmo	Utilidad	Utilidad Corregida	<i>gap</i>	# inv.	# ret.	Costos totales	Días malos
4-Bandas	0,2498	0,2498	0,47	9	3	0,054	
2-Bandas	0,2510	0,2220	0,53	6	4	0,076	2
Lineal	0,2507	0,2507	0,47	17	4	0,043	
Cuadrático	0,2868	0,2868	0,39	13	4	0,033	
4-Bandas Móvil	0,1985	0,1985	0,58	8	4	0,061	
2-Bandas Móvil	0,1190	0,1190	0,75	8	5	0,100	
Lineal Móvil	0,1999	0,1999	0,58	12	5	0,050	
Cuadrático Móvil	0,2100	0,2100	0,55	10	5	0,050	
Estocástico	0,3525	0,1591	0,66	11	7	0,076	5
Óptimo	0,4709	0		10	4	0,052	

Cuadro 3.6: Resumen de los algoritmos aplicados a la instancia del ejemplo.

Estocástico alcanzó la mejor utilidad de los algoritmos, por otro lado es también el algoritmo que tuvo más días malos, por esta razón su utilidad corregida es inferior. Cuadrático realizando 13 inversiones y 4 retiros con costes transac-

cionales de 0,033, alcanzó una de las mayores utilidades y sin ningún día malo. Los costos de Cuadrático son los más bajos de los algoritmos en esta instancia. Cuadrático alcanzó esta utilidad debido a que sus inversiones/retiros fueron en su mayoría conservadores, de esta forma no se debió realizar retiros elevados y así sus costos por transferir el dinero fueron pequeños. De igual forma el algoritmo Lineal alcanzó una buena utilidad realizando 17 inversiones y 4 retiros bajos, así obteniendo menores costos. Por el contrario, 2-Bandas Móviles realizó grandes inversiones y retiros de dinero, razón por la cual sus costos por transferencia son los más altos y su utilidad es la más baja.

El *gap* de los algoritmos generalmente es mayor que 0,3, lo que indica que están muy alejados de la solución óptima. Cuando el *gap* en un algoritmo se acerca a cero, se dice que su solución es más cercana a la del óptimo.

Los parámetros fijados para esta instancia no son los mejores para los algoritmos, esta es la razón por la cual en este punto no se puede etiquetar a cualquier algoritmo como bueno o malo. En la siguiente sección se aborda el tema de la calibración de parámetros.

3.5. CALIBRACIÓN DE PARÁMETROS

Un aspecto fundamental en la aplicación de los algoritmos en-línea descritos en las secciones anteriores es la selección de los valores adecuados para los parámetros de calibración. La búsqueda de estos valores debe realizarse durante una fase previa a la aplicación del algoritmo en un entorno determinado. Formularemos a continuación esta tarea como un subproblema de optimización no lineal y consideramos dos métodos numéricos de solución.

Cabe señalar, sin embargo, que un análisis exhaustivo de las particularidades numéricas del problema y de los algoritmos de solución (por ejemplo, en lo que respecta a estabilidad o eficiencia computacional) escapa de los propósitos de esta tesis.

3.5.1. MODELO DE OPTIMIZACIÓN NO-LINEAL PARA LA CALIBRACIÓN DE PARÁMETROS

Como ya se mencionó, los algoritmos en-línea propuestos requieren de ciertos parámetros para tomar sus decisiones. Dependiendo de los parámetros sus decisiones pueden ser generalmente buenas o malas. Siendo así, el objetivo encontrar valores de estos parámetros que permitan obtener la máxima utilidad cumpliendo la restricción de seguridad durante un horizonte de planificación T .

Para cada algoritmo se definen dos funciones $u : \mathbb{R}^N \rightarrow \mathbb{R}$, y $r : \mathbb{R}^N \rightarrow \mathbb{R}$, que dependen del vector x de parámetros del algoritmo, del vector aleatorio $y = (y_1, \dots, y_T)$ formado por las variables que representan el flujo neto de dinero y del vector q de parámetros del problema. El vector q está formado por los valores del saldo mínimo L_t , del saldo base \tilde{L}_t (en el caso de los algoritmos de bandas móviles), y de las tasas de retorno, envío y retiro de dinero r_t , c_t^I y c_t^M , para $t = 1, \dots, T$.

La función u está definida de la siguiente manera:

$$u(x, y, q) = \sum_{t=1}^T (r_t E_t - c_t^I I_t - c_t^M M_t)$$

donde

$$C_t = C_{t-1} - I_t + M_{t-2} + y_t, \quad t = 1, \dots, T$$

$$E_t = E_{t-1} + I_{t-2} - M_t, \quad t = 1, \dots, T$$

$$I_t = \begin{cases} g(x, y^t, q) & \text{si } g(x, y^t, q) > 0, \\ 0 & \text{caso contrario,} \end{cases}$$

$$M_t = \begin{cases} -g(x, y^t, q) & \text{si } g(x, y^t, q) < 0, \\ 0 & \text{caso contrario.} \end{cases}$$

para $t = 1, \dots, T$, con $I_{-1} := M_{-1} := I_0 := M_0 := 0$.

La función $g(x, y^t, q)$ retorna el monto enviado/retirado por el algoritmo en el día t conocidos los valores de flujo $y^t = (y_1, \dots, y_t)$ en los días anteriores. Asumimos

que $g(x, y^t, q) > 0$ indica que el algoritmo decide enviar dinero y que $g(x, y^t, q) < 0$ simboliza un retiro. De manera similar, la función $r(x, y, q)$ está dada por:

$$r(x, y, q) = \begin{cases} 1 & \text{si } C_t < L_t \text{ para algun } t, \quad t = 1, \dots, T, \\ 0 & \text{caso contrario.} \end{cases}$$

A partir de las funciones u y r definimos:

$$\begin{aligned} \bar{u}(x, q) &= E_y u(x, y, q) \\ \bar{r}(x, q) &= E_y r(x, y, q) \end{aligned}$$

Notar que $\bar{u}(x, q)$ refleja la utilidad promedio obtenida por el algoritmo para las instancias del problema cuyos parámetros determinísticos son iguales a q , suponiendo que los parámetros de calibración han sido fijados iguales a x . Por su parte \bar{r} refleja la fracción esperada de estas instancias en las cuales el algoritmo viola al menos una vez la restricción de seguridad.

Finalmente, usamos \bar{u} y \bar{r} para definir la función

$$f(x, q) = \begin{cases} \bar{u}(x, q) & \text{si } \bar{r}(x, q) < \epsilon, \\ -\infty & \text{en otro caso.} \end{cases}$$

Donde $\epsilon > 0$ es un parámetro de seguridad que debe fijarse de antemano. Esta función retorna la utilidad promedio obtenida por el algoritmo siempre y cuando la probabilidad de violar la restricción de seguridad sea menor a ϵ , y $-\infty$ en caso contrario.

El problema de calibración de parámetros consiste en, dado un vector q^* de valores para los parámetros determinísticos del modelo, encontrar un vector x de parámetros del algoritmo que maximice $f(x, q^*)$.

Para resolver esta clase de problemas existen algunos métodos numéricos. En las siguientes secciones se presentan dos de estos métodos.

3.5.2. MÉTODO DE NELDER-MEAD

El método de Nelder-Mead fue publicado en 1965 [33] y es uno de los métodos ampliamente utilizados para optimizar funciones no lineales sin restricciones [32]. Existe un libro [34] dedicado enteramente al método y sus variantes, es utilizado en MATLAB [36] en una herramienta para minimizar, y aparece en el libro *Numerical Recipes* [35] como el “amoeba algorithm”.

El objetivo del método de Nelder-Mead es minimizar (maximizar) una función escalar no lineal $f : \mathbb{R}^n \rightarrow \mathbb{R}$ sin asumir nada en cuanto a las propiedades o estructura de f . Lo único que el método requiere es el valor de $f(x)$ para cualquier $x \in \mathbb{R}^n$. De esta forma Nelder-Mead pertenece a *los métodos de búsqueda directa*.

La idea general del método consiste en construir una sucesión de símplexes en \mathbb{R}^n . Un símplex en \mathbb{R}^n es la envolvente convexa de $n + 1$ puntos afínmente independientes. Así un símplex en \mathbb{R}^2 es un triángulo y un símplex \mathbb{R}^3 es un tetraedro. En el cuadro Algoritmo 3.10 se describe el método de Nelder-Mead. Por facilidad se nota $f_i = f(x_i)$.

En cada iteración el algoritmo requiere un conjunto V_0 con $n+1$ puntos afínmente independientes ordenados y etiquetados en relación al valor de la función f . Así, el orden de los puntos será x_1, \dots, x_{n+1} , si $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$. De esta manera, x_1 es el mejor punto y x_{n+1} es el peor punto del conjunto V_0 .

Al finalizar, el algoritmo retorna un nuevo conjunto V de $n + 1$ puntos ordenados que son “mejores”.

En cada paso el algoritmo calcula el *centroide* de los n mejores puntos y calcula el punto de *reflexión*, x_r . A partir del valor f_r de la función evaluada en el punto de reflexión, se distinguen cuatro casos: en el primer caso, x_r resulta ser mejor que x_1 ; en el segundo caso, x_r está entre los n mejores puntos; en el tercer caso, x_r está entre el último de los n mejores puntos y el peor punto; en el cuarto caso, x_r resulta ser peor que x_{n+1} .

Los dos primeros caso aceptan un nuevo punto, es decir, se añade un punto más al conjunto V_0 . Los dos últimos casos pueden aceptar un nuevo punto o realizar un *encogimiento* de los puntos.

Dentro de los casos se realiza operaciones, ya sea para calcular nuevos puntos (como el punto de *expansión* y los puntos *interior* y *exterior* de *contracción*) o para realizar un encogimiento a los puntos del conjunto. Estas operaciones dependen de cuatro parámetros que son los coeficientes: ρ de reflexión, χ de expansión, γ

ENTRADA:

$f : \mathbb{R}^n \rightarrow \mathbb{R}$ función no lineal.

$V_0 = \{x_1, x_2, \dots, x_{n+1}\}$ conjunto inicial de $n + 1$ puntos ordenados respecto al valor de la función f .

SALIDA:

V conjunto de $n + 1$ puntos ordenados respecto al valor de f .

COEFICIENTES:

ρ de reflexión, χ de expansión, γ de contracción y σ de encogimiento.

· Calcular $\bar{x} = \sum_{i=1}^n x_i / n$ *centroide* de los n mejores puntos.

· Calcular $x_r = \bar{x} + \rho(\bar{x} - x_{n+1}) = (1 + \rho)\bar{x} - \rho x_{n+1}$ punto de *reflexión*.

· Evaluar $f_r = f(x_r)$.

si $f_r < f_1$ entonces

· Calcular $x_e = \bar{x} + \chi(x_r - \bar{x}) = \bar{x} + \rho\chi(\bar{x} - x_{n+1}) = (1 + \rho\chi)\bar{x} - \rho\chi x_{n+1}$ punto de *expansión*.

· Evaluar $f_e = f(x_e)$.

si $f_e < f_r$ entonces

· Aceptar x_e en V_0 , es decir hacer que $x_e \in V_0$

caso contrario

· Aceptar x_r en V_0 .

fin si

fin si

si $f_1 \leq f_r < f_n$ entonces

· Aceptar x_r en V_0 .

fin si

si $f_n \leq f_r < f_{n+1}$ entonces

· Calcular $x_c = \bar{x} + \gamma(x_r - \bar{x}) = \bar{x} + \rho\gamma(\bar{x} - x_{n+1}) = (1 + \rho\gamma)\bar{x} - \rho\gamma x_{n+1}$ punto exterior de contracción.

· Evaluar $f_c = f(x_c)$.

si $f_c \leq f_r$ entonces

· Aceptar x_c en V_0 .

caso contrario

· Calcular $x_i = x_1 + \sigma(x_i - x_1) = \sigma x_i + (1 - \sigma)x_1$, para $i = 2, \dots, n + 1$. Encogimiento.

· Hacer $V = V_0$.

· Terminar.

fin si

fin si

si $f_r \geq f_{n+1}$ entonces

· Calcular $x_{cc} = \bar{x} - \gamma(\bar{x} - x_{n+1}) = (1 - \gamma)\bar{x} + \gamma x_{n+1}$, punto interior de contracción.

· Evaluar $f_{cc} = f(x_{cc})$.

si $f_{cc} < f_{n+1}$ entonces

· Aceptar x_{cc} en V_0 .

caso contrario

· Calcular $x_i = x_1 + \sigma(x_i - x_1)$, para $i = 2, \dots, n + 1$. Encogimiento.

· Hacer $V = V_0$.

· Terminar.

fin si

fin si

· Ordenar los puntos de V_0 respecto al valor de f .

· V igual al conjunto de los $n + 1$ mejores puntos de V_0

Algoritmo 3.10: Nelder-Mead

de contracción y σ de encogimiento [32]. Estos parámetros deben satisfacer las condiciones:

$$0 < \rho < \chi, \quad \chi > 1, \quad 0 < \gamma < 1 \quad \text{y} \quad 0 < \sigma < 1.$$

Usualmente para el algoritmo de Nelder-Mead estándar se seleccionan los valores $\rho = 1$, $\chi = 2$, $\gamma = \frac{1}{2}$ y $\sigma = \frac{1}{2}$.

Cuando se ha aceptado un nuevo nodo, el algoritmo ordena nuevamente los puntos del conjunto inicial V_0 respecto al valor de la función f y toma los mejores $n + 1$ puntos para crear el nuevo conjunto V . Cuando se realiza un encogimiento a los puntos de V_0 , el conjunto V será igual al conjunto V_0 y se termina la iteración. De esta forma, en cada iteración se puede eliminar el peor punto del conjunto u obtener otros puntos que son el resultado de la combinación convexa entre el mejor punto y cada uno de los punto del conjunto ($x_i = \sigma x_i + (1 - \sigma)x_1$, para $x_i \in V_0$).

3.5.3. MÉTODO DEL DESCENSO PROFUNDO

Para una función f de n variables, la cual representa una superficie en \mathbb{R}^n ; el gradiente de la función en un punto x , $\nabla f(x)$, proporciona la dirección perpendicular al plano tangente a la superficie en dicho punto y representa la máxima variación de la función. Para aumentar el valor de la función basta desplazarse una distancia adecuada en la dirección del gradiente. Para disminuir el valor de la función se debe desplazar en dirección opuesta al gradiente [13]. Así, una estrategia posible para minimizar el valor de la función consiste en desplazarse de un vector inicial x_0 a otro x_1 , de acuerdo con:

$$x_1 = x_0 - \lambda \nabla f(x_0), \quad \lambda > 0.$$

La elección del parámetro λ , que se conoce como *longitud de paso*, es crucial para efectivamente reducir el valor de la función f . Una manera de escoger el paso es permitiendo que $\lambda = \beta^m$, donde $\beta \in (0, 1)$ y $m \geq 0$ es el entero positivo más pequeño que permite un suficiente decrecimiento en f . Es decir, con $\lambda = \beta^m$ se satisface:

$$f(x_0 - \lambda \nabla f(x_0)) - f(x_0) < -\alpha \lambda \|\nabla f(x_0)\|^2.$$

Esta estrategia, introducida en [7] es llamada la regla de Armijo. Usualmente el parámetro α se fija en 10^{-4} [5].

El método de descenso profundo ([6]) en cada iteración encuentra un nuevo punto como se indica en el cuadro Algoritmo 3.11.

ENTRADA:

Función $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

Punto inicial x_0 .

SALIDA:

Punto encontrado x_1 .

PARÁMETROS:

$\beta \in (0, 1)$ y α .

- Calcular el gradiente de f .
- Encontrar el menor $m \geq 0$, para $\lambda = \beta^m$, que satisfaga la condición de Armijo:

$$f(x_0 - \lambda \nabla f(x_0)) - f(x_0) < -\alpha \lambda \|\nabla f(x_0)\|^2.$$

- Hacer $x_1 = x_0 - \lambda \nabla f$.

Algoritmo 3.11: Descenso Profundo

Capítulo 4

IMPLEMENTACIÓN COMPUTACIONAL

Formular el programa estocástico para el PGSC no es una tarea fácil debido al número de escenarios, que crecen con el horizonte del problema, y hace que el tamaño del problema sea poco manejable. Por esto, se ha planteado y desarrollado algoritmos en-línea que ayudan a tomar decisiones conforme ingresa nueva información del problema. La aplicación y el desempeño de estos algoritmos depende de ciertos parámetros. Se ha planteado el problema de encontrar la mejor combinación de valores a estos parámetros como un modelo de optimización no-lineal. Para optimizar el modelo no-lineal de calibración de parámetros se ha empleado dos métodos numéricos de optimización: Nelder-Mead y Descenso Profundo. Estos métodos se han implementado en paquetes matemáticos como Matlab [36], sin embargo, por la forma de la función no lineal de calibración se ha programado estos métodos numéricos en el lenguaje C++. Para la implementación de los métodos numéricos junto con el cálculo de la brecha de optimalidad de los algoritmos (gap) se creó un programa llamado GSCaja, el cual permite: realizar simulaciones del flujo neto de dinero con los modelos descritos en el Apéndice A; aplicar los algoritmos en-línea a flujos de caja simulados como reales; calcular la utilidad esperada de los algoritmos en-línea para ciertos valores de parámetros; calibrar los parámetros de los algoritmos propuestos empleando los métodos numéricos y finalmente permite obtener el valor del gap como una medida para la calidad del algoritmo.

En este capítulo se presentan detalles concernientes a la implementación computacional del programa GSCaja.

4.1. ESTRUCTURA GENERAL DEL PROGRAMA GSCaja

Para su desarrollo el programa utiliza un fichero principal llamado main_GSCaja, ficheros secundarios y ficheros de cabecera. El fichero principal como los secundarios tienen extensión .cpp y los archivos de cabecera tienen extensión .h. En los archivos de cabecera se declaran las funciones y constantes que se utilizan en un archivo .cpp. En los archivos .cpp se debe incluir las librerías y archivos cabecera que se utilizarán en el desarrollo del código de las funciones. GSCaja contiene los siguientes ficheros secundarios y cabeceras:

- Ficheros secundarios

```

modelos.cpp  algoritmos.cpp  esperanza.cpp
lista.cpp    metodos.cpp    gap.cpp
archivos_programas.cpp  presentacion.cpp
estocastico.cpp

```

- Ficheros de Cabecera

```

constantes.h  constmodelos.h  constmetodos.h
modelos.h     algoritmos.h  esperanza.h
lista.h       metodos.h    gap.h
archivos_programas.h  presentacion.h

```

El archivo principal hace uso de las funciones de los archivos secundarios mediante la declaración de las respectivas cabeceras. De igual forma, los archivos secundarios pueden utilizar las funciones de otros archivos incluyendo las cabeceras donde se encuentran declaradas estas funciones. En la Figura 4.1 se presenta la relación que tienen los archivos dentro del programa.

Al fichero que la flecha apunta requiere de alguna función del fichero donde se

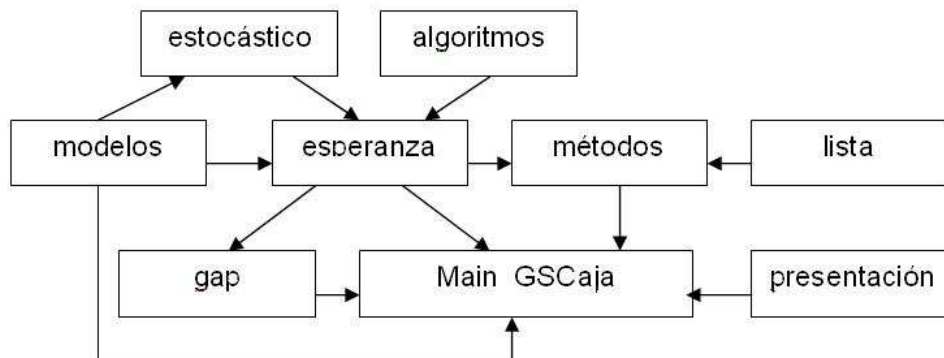


Figura 4.1: Estructura GSCaja.

origina la flecha. Por ejemplo, el fichero esperanza requiere de las funciones de modelos, y métodos requiere de las funciones de esperanza.

Tanto el fichero principal como los secundarios requieren de las funciones de archivos_programas.cpp para crear un archivo de texto con el desarrollo de sus funciones. En modelos.cpp se genera el flujo de caja neto para las funciones de evaluación de la utilidad de esperanza.cpp y para las funciones de estocástico.cpp. El

flujo es generado con las distribuciones de probabilidad Sarima-Impulso, Impulso, Sarima y Normal, o mediante valores históricos reales de flujo de caja. En algoritmos.cpp se realiza la aplicación de los algoritmos en-línea a un flujo de caja para un horizonte de planificación y unos parámetros dados. El fichero estocastico.cpp desarrolla el algoritmo Estocástico, el cual requiere para su evaluación hacer uso de las librerías SMI [41]. En esperanza.cpp se calcula el valor de la utilidad y la utilidad esperada al implementar un cierto algoritmo en-línea con un flujo de caja neto generado por una distribución de probabilidad. En lista.cpp están contenidas las diferentes funciones de la clase lista que ayudan al desarrollo del método de Nelder-Mead. En metodos.cpp se requieren de las funciones de esperanza.cpp y de lista.cpp para la implementación de los métodos de optimización de Nelder-Mead y de Descenso Profundo, utilizados para calibrar los parámetros de los algoritmos en-línea. En el fichero gap.cpp se desarrolla la instrucciones para evaluar la brecha de optimalidad de los algoritmos para lo cual se requiere la utilidad generada por los algoritmos y conocer la solución del problema fuera-de-línea, mediante el uso del programa mcf.exe [40]. El fichero archivos_programas.cpp permite crear o acceder a archivos .txt u otras extensiones, bien sea para guardar información acerca del desarrollo de GSCaja o para leer información requerida por funciones del programa, además, permite acceder al programa mcf.exe para obtener la solución óptima del problema fuera-de-línea. Finalmente, en el fichero presentacion.cpp se encuentran las funciones que muestran en pantalla la etiqueta y los menús de opciones, algoritmos y distribuciones de probabilidad del programa. En las siguientes secciones se detalla más sobre estos ficheros.

En la cabecera constantes.h están los parámetros iniciales para el desarrollo del PGSC como las tasa de retorno, de envió/retiro de dinero, los saldo iniciales en el exterior y en caja. Además, se encuentran las constantes necesarias para el cálculo de la utilidad esperada como: el valor asignado a infinito, el tamaño de la muestra piloto, el nivel de confianza y el error permitido. La cabecera constmodelos.h contiene los valores de los coeficientes necesarios para simular o generar flujos de caja neto con las funciones de distribución. En la cabecera constmetodos.h están los parámetros requeridos por los métodos numérico de optimización como los coeficientes de reflexión, expansion, contracción y criterios como de parada de longitud de paso, etc. En la cabecera modelos.h se declara las funciones utilizadas para generar los flujos de caja con las diferentes distribuciones. En la cabecera algoritmos.h están declaradas las funciones requeridas para el desarrollo de los algoritmos en-línea. En las cabeceras esperanza.h, lista.h, metodos.h, gap.h, archivos_programas.h y presentacion.h se declaran las funciones utilizadas

por sus respectivos ficheros secundarios.

El fichero principal main_GSCaja.cpp es el encargado de interactuar con el usuario y las diferentes funciones del programa. Presentando el menú de opciones de la Figura 4.2 el usuario puede escoger que actividad realizar.

```

////////////////////////////////////
                                PRESIONE EL NUMERO

Flujo de caja neto:
    1 Simulacion flujo de dinero diario
    2 Estimacion diaria de los flujo
Calibracion de parámetros
    3 Metodo de Nelder Mead
    4 Metodo del Descenso Profundo
Funcion de utilidad
    5 Calculo de la utilida esperada
Algoritmos en-linea
    6 Aplicar algoritmos en-linea
Brecha de optimalida
    7 Problema fuera-de-linea
    8 Calculo del Gap
0 Presentar nuevamente el menu de opciones
Otro numero para salir
////////////////////////////////////

```

Figura 4.2: Menú Principal del programa GSCaja.

Digitando el número 1 se accede a las funciones del fichero modelos y así se obtiene flujos de caja neto con las distintas distribuciones de probabilidad. Con el número 3 y 4 se accede a los métodos de Nelder-Mead y Descenso Profundo respectivamente, para la calibración de los parámetros de los algoritmos. Con la opción 5 se calcula la utilidad esperada para un horizonte de tiempo y el algoritmo requerido. Con la opción 6 se aplica los algoritmos en-línea. Con la opción 7 se accede a la solución óptima del problema fuera-de-línea para cualquier instancia.

Con la opción 8 se obtiene el valor del gap para cualquier algoritmo. Digitando el número cero se despliega nuevamente el menú de opciones y con cualquier número mayor a ocho se sale del programa.

A continuación se detallan con más detenimiento las funciones de los ficheros secundarios que permiten realizar las diferentes actividades del program GSCaja.

4.2. SIMULACIÓN DEL FLUJO DE CAJA

Una de las tareas principales es la generación del flujo de caja neto, para poder calcular la utilidad esperada y así, calibrar los parámetros de los algoritmos en-línea. Los flujos de caja generados deben mantener un comportamiento similar al del flujo real, para esto se ha estudiado el registro histórico de los dos últimos años. Del estudio estadístico se ha encontrado tres posibles distribuciones de probabilidad, las cuales se detallan en el Apéndice A. Para la implementación de estas distribuciones se ha creado funciones en el archivo modelo.cpp, el cual requiere de la inclusión de los archivos de cabecera: modelos.h, constmodelos.h, constantes.h y archivos_programas.h. Estas funciones se relacionan como se indica en la Figura 4.3.

De igual forma, a la función que apunta la flecha requiere utilizar la función don-

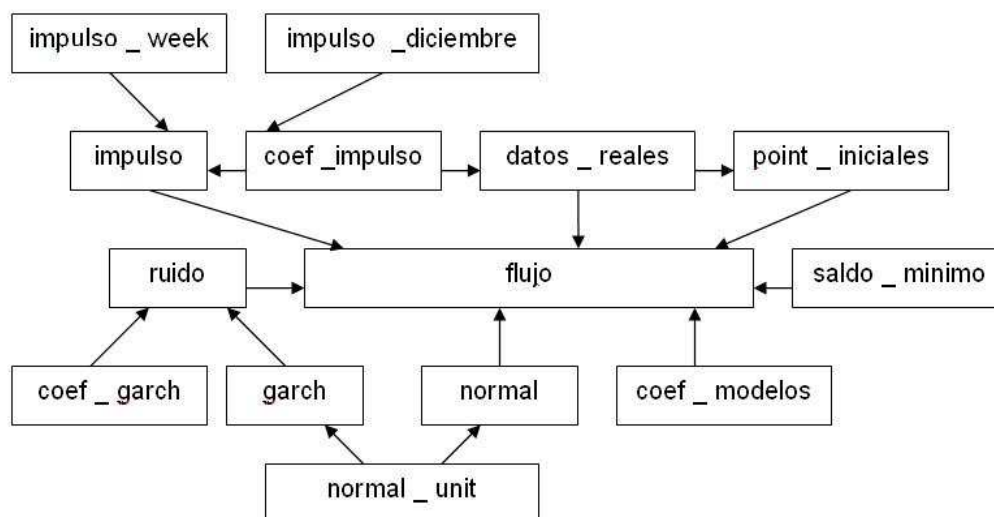


Figura 4.3: Estructura fichero modelos.cpp

de se origina la flecha, por ejemplo, la función `ruido` requiere de las funciones `coe_garch` y `garch`.

En la función `flujo` se generan los diferentes flujos de caja neto ya sean con las distribuciones de probabilidad o mediante datos históricos, esta función requiere principalmente de las funciones: `ruido`, `impulso`, `saldo_minimo`, `datos_reales`, `normal`, `point_iniciales` y `coef_modelos`. Estas funciones requieren de otras funciones como se indica en el diagrama. A continuación se detallan todas las funciones de `modelo.cpp`.

```
/*- void flujo(double * y,double*Smin,int M,int modelo,char e)
```

Esta función permite obtener las diferentes clases de flujos de dinero diario y saldo mínimo. Estos flujos pueden ser simulados con los distribuciones de probabilidad `Sarima-Impulso`, `Impulso` y `Sarima` o simplemente generados por medio de una distribución normal. También se permite acceder a un flujo de caja real.

Para acceder El flujo de dinero creado conforme el modelo solicitado y la cantidad de días `M` se almacena en el arreglo `y`, y saldo mínimo se almacena en `Smin`.

Para generar un flujo con las distribuciones el argumento `modelo` debe estar entre 1 y 3, así se ejecuta:

```
if(modelo<=3){
    int cant=6;                //cant: cantidad de puntos iniciales
    int T2=M+cant;
    double *ruido,*imp;
    ruido=new double [T2];
    imp=new double [T2];
    impulso(imp,modelo,T2);    //impulso(variables de intervención)
    funcion_ruido(ruido,T2,modelo); //ruido-garch(r,s)
    double *z,*aux_y;
    z=new double [T2];        //z=flujo-impulso.
    aux_y=new double[T2];
    double *coef,*coef_rui;  //coeficientes de los modelos
    coef=new double[3];
    coef_rui=new double[2];
    coef_modelos(coef,coef_rui,modelo);
    point_iniciales(aux_y,imp,z,modelo,cant); //puntos iniciales
    for(int t=cant;t<T2;t++){
        aux_y[t]=coef[0]*z[t-1]+coef[1]*z[t-5]-coef[2]*z[t-6]
        +coef_rui[0]*ruido[t-1]+coef_rui[1]*ruido[t-5]+ruido[t]+imp[t];
        z[t]=aux_y[t]-imp[t];
    }
}
```

```

        y[t-cant]=aux_y[t];}
    }

```

Para generar un flujo con una ley normal `modelo` debe ser igual a cuatro, y así se ejecuta la siguiente instrucción.

```

if(modelo==4){
    normal(y,mmean,ddesv_est,M);}

```

Para acceder a un flujo de caja histórico `modelo` debe ser igual a cinco y se ejecuta la instrucción.

```

if(modelo==5){
double *imp;
imp=new double[M];
datos_reales(Smin,imp,y,M);}

```

Mediante el valor `s` para el parámetro `e` se puede obtener un archivo de texto llamado `flujo_simulado` en la carpeta de datos, el cual contiene el flujo de dinero, el ruido, el impulso (dependiendo el flujo solicitado) y el saldo mínimo para los días requeridos.

```

-**- void saldo_minimo(double *Smin,int T2,double *y)

```

Mediante esta función se obtiene un arreglo `Smin` con el saldo mínimo para `T2`. El saldo mínimo se genera en función lineal al promedio del flujo de caja cada 5 días, como lo indican las siguientes instrucciones.

```

{
    int d=5;
    int i=0;
    while(i<T2)
    {
        double sy=0.0;
        int aux=min(T2,i+d);
        for(int j=i;j<aux;j++){
            sy=sy+y[j]; }
        double s_m= INTERCEPTO + COEFICIENTE*(sy/d);
        for(int k=i;k<aux;k++){
            Smin[k]=s_m; }
        i=i+d;
    }
}

```

INTERCETO y COEFICIENTE son constantes declaradas en la cabecera constmodelos.h.

```
**- void funcion_ruido(double *ruido,int T2,int modelo)
```

Esta función entrega un arreglo con una distribución para el ruido generado mediante un modelo Garch.

```
**- void garch(double*coef,double*ruido,int r,int s,int M)
```

Esta función crea un arreglo `ruido` de tamaño `M` con un modelo Garch(r, s). Los coeficientes necesarios para generar el modelo se almacenan en el arreglo `coef` de tamaño $r + s + 1$. La idea principal para generar el ruido Garch se refleja en las siguientes instrucciones:

```
{   double arch,garch;
    arch=garch=0.0;
    if(r!=0) {
        for(int i=1;i<=r;i++)
            arch=arch+coef[i]*pow(u[t-i],2); }
    if(s!=0) {
        for(int j=1;j<=s;j++)
            garch=garch+coef[r+j]*h[t-j]; }
    h[t]=coef[0]+arch+garch;
    ruido[t]=sqrt(h[t])*u[t];
}
```

Recordar que el modelo Garch ayuda a simular la varianza de la serie de flujo de caja neto.

```
**- void coef_garch(double *coefi_garch,int modelo)
```

Lee los coeficientes necesarios para generar el modelo Garch del fichero constmodelos.h, y los almacena en el arreglo `coef_garch`.

```
**- void normal_unit(double* x,int M)
```

Esta función genera un arreglo `x` de `M` números generados con una distribución Normal Estándar (media=0 y varianza=1). Los guarda y los almacena en el arreglo. Con las siguientes instrucciones se genera un número aleatorio de distribución Normal Estándar:

```

do{ double y,r;
    r=rand();
    y=(r)/RAND_MAX;
    v1[i]= 2*y-1;
    r=rand();
    y=(r)/RAND_MAX;
    v2[i]= 2*y-1;
    s[i]=pow(v1[i],2)+pow(v2[i],2); }
while (s[i]>1);
double aux;
aux=-2*log(s[i]);
aux=aux/s[i];
x[i]=sqrt(aux)*v1[i];

```

```

-- void normal(double* x,double mean,double desv_est,int M)

```

Genera un arreglo `x` de `M` componentes, con una distribución Normal de media y varianza dadas por los parámetros `mean` y `desvt_est` respectivamente. Para generar este arreglo utiliza la función `normal_unit`.

```

-- void impulso(double *imp,int modelo,int T2)

```

Crea un arreglo `imp` de tamaño `M` que contiene el impulso total para generar las distribuciones de probabilidad con intervención. Este impulso es la suma del impulso semanal más el impulso en el mes de diciembre.

```

-- void coef_impulso(double *day,int modelo)

```

Almacena en el arreglo `day` los coeficientes para crear el impulso para las diferentes distribuciones.

```

-- void imp_week(double *week,double *dias,int M)

```

Genera la variable de impulso semanal en el arreglo `week` de tamaño `M`. Esta variable es la combinación lineal de los impulsos diarios.

```

-- void imp_diciembre(double*dicie,int modelo,int M)

```

Genera un arreglo `dicie` de tamaño `M`, con la variable de impulso para el mes de diciembre.

```

-- void point_iniciales(double*flujo,double*imp,double*z,
int modelo,int M);

```

Permite obtener arreglos de tamaño M de flujo real, de impulso y un arreglo z que es la diferencia entre el flujo y el impulso.

```

-- void datos_reales(double *Smin,double *dia,double*flujo,
int M)

```

Esta función lee de un archivo de texto llamado `datos_reales.txt` los datos reales del saldo mínimo, el día y el flujo diario. Así abastece a los arreglos `Smin`, `dia` y `flujo`.

```

-- void coef_modelos(double *coef,double *coef_rui,int modelo)

```

Esta función proporciona los coeficientes para generar las funciones de distribución en los arreglos `coef` y `coef_rui`

4.3. IMPLEMENTACIÓN DE LOS ALGORITMOS EN-LÍNEA

En el fichero `algoritmos.cpp` se implementan los algoritmos en-línea para ayudar a manejar el saldo en caja. Estos algoritmos toman decisiones de cuándo y cuánto enviar y pedir del extranjero. Este archivo contiene las siguientes funciones.

```

-- void algoritmo_bandas(double *valor,int T2,int algorit,
double *y,double*s, double*se,double*I,double*M,
double*Smin,double*P,double &p,char opcion)

```

Esta función permite aplicar los algoritmos de bandas fijas y móviles a un flujo de caja dado en el arreglo `y` para un horizonte de T_2 días. Los parámetros del algoritmo se encuentran en el arreglo `valor`. El argumento `algorit` permite acceder a los distintos algoritmos, el cual está relacionado con `opcion`. Por ejemplo, para ejecutar 4-Bandas Móviles `algorit` toma el número 5 y `opcion` será `v`. Las decisiones tomadas por el algoritmo son guardadas en los arreglos `M` e `I` y los saldos en caja y en el exterior generados son almacenados en los arreglos `s` y `se` respectivamente.

```

-- double pedido(int algorit,double *s,int t,double b_min,
double factor_retiro)

```

Esta función ayuda a tomar las decisiones de cuánto pedir, de acuerdo a los saldos en caja y exterior, `b_min` y `factor_retiro`.

```

-- double invertir(int algorit,double *s,int t,double b_max,
double factor_inversion)

```

Esta función ayuda a tomar las decisiones de cuánto invertir en los diferentes algoritmos. El monto a invertir es realizado en base a los saldos en caja y exterior, `b_max` y `factor_inversion`.

4.3.1. IMPLEMENTACIÓN DEL ALGORITMO ESTOCÁSTICO

En el fichero `estocasticos.cpp` se implementa el algoritmo Estocástico, el cual toma las decisiones de inversión y pedido resolviendo el PEGSC para un horizonte de h días. Para lo cual se requiere de un conjunto de escenarios y datos iniciales como las tasas y saldos iniciales en el extranjero y en caja. Además, para resolver el programa estocástico generado para el horizonte de tiempo se utiliza las librerías SMI que aceptan un archivo escrito en el formato SMPSS del problema. Para crear el árbol de eventos se requiere primero la discretización de la variable en cada uno de los períodos de predicción para lo cual se hace uso de los intervalos de confianza. Luego se genera el árbol de eventos.

Funciones para crear el árbol de escenarios:

- `int number_point(int periodo)`

Esta función devuelve de acuerdo al período el número de subintervalos en los cuales se divide el rango del flujo de caja. Para el árbol de escenarios esta función indica el número de hojas que tiene una rama.

- `int numero_scenarios(int horizonte,int &TT)`

Retorna el número de escenarios para un cierto período. Además, esta función permite conocer mediante el parámetro `TT` el número de nodos o hojas que tiene el árbol de escenarios.

- `int recursiva_arbol(double *y,double *z,double *ruido, double *imp,int modelo,int MM,int period,double *estimates, double*probabilidad,int *ddia,int cont,double prob, int horizonte,int*escenario,int esc)`

Esta función genera el árbol de escenarios. Dentro de la función se estima el flujo y crea los puntos representantes de cada intervalo con la ayuda de las funciones `divi_interval`, `calc_si` y `EMC`.

Retorna el número de nodos generados más uno. Con la ayuda del parámetro `cont` se enumeran los nodos de acuerdo como son creados. Al terminar la función ha generado un arreglo con las estimaciones del flujo y un arreglo con las probabilidades de estas estimaciones para cada período durante el horizonte de tiempo. En el arreglo `ddia` se almacena el período al cual pertenece la estimación.

- `void divi_interval(double*medios,double *probab,int number, double *y,int t,int period,int modelo,double prob)`

Esta función calcula los intervalos de confianza para las predicciones en cada período de tiempo suponiendo que la historia del flujo disponible no es real sino estimada, requiriendo el cálculo del error en media cuadrática (EMC). Para obtener el EMC se requiere del cálculo de los coeficientes del modelo en la forma canónica. De esta forma los intervalos de confianza para las predicciones crecen conforme el período de estimación crezca. Divide cada intervalo máximo en 5 subintervalos. Se calcula los puntos medios de los subintervalos y luego calcula la predicción para el siguiente período a partir de los puntos medios. Este proceso se repite para el número de períodos deseado.

- `double EMC(int modelo,int M)`

Devuelve el error en media cuadrática que se presenta al realizar una predicción del flujo en un cierto período de tiempo. Esta función requiere del cálculo de los parámetros de la función canónica del modelo, que se obtienen mediante la función `calc_si`. El error de predicción crece conforme lo hace el período de estimación.

- `void calc_si(double *si,int modelo,int M)`

Esta función permite obtener los coeficientes de los modelos de las distribuciones de probabilidad Sarima, Impulso y Sarima-Impulso en forma canónica, los cuales ayudan a determinar el error en media cuadrática.

El SMI admite la escritura de los problemas en el formato SMPS para resolver los problemas de programación estocástica.

Para escribir un problema en el formato SMPS [20] se requieren tres archivos: archivo principal (core file), archivo de tiempo (time file) y archivo estocástico (stoch file).

Las funciones requeridas para poner el problema de gestión de saldo en caja en el formato SMPS son:

- *archivo principal*

```
void flujo_cor(int horizonte,double y0,double s_ext)
```

Crea el archivo principal para el problema de gestión de saldo en caja para el horizonte de tiempo requerido.

- *archivo de tiempo*

```
void flujo_time(int horizonte)
```

Crea el archivo de tiempo para el problema de gestión de saldo en caja para el horizonte requerido.

- *archivo estocástico*

```
void flujo_stoch_escenarios(double*y,double*z,double*ruido,
double*imp,double*Smin,int modelo,int t,int horizonte)
```

Genera el archivo estocástico para el problema de gestión de saldo en caja. Requiere de la ayuda de la función recursiva para generar los escenarios.

```
int ant(int &proba,int*ddia,int j,int horizonte)
```

Esta función ayuda a ordenar los valores estimados conforme el período de tiempo.

Las siguientes funciones junto con las anteriores implementan el algoritmo Estocástico.

- `void estocastico(double*y,int model_esc,int M,double*S, double*Se,double*Inv, double*Ped,double*Smin,double*P)`

Esta función aplica el algoritmo Estocástico, para un flujo almacenado en el arreglo `y` para un horizonte M días. En los arreglos `S`, `Se`, `Inv` y `Ped` almacena los saldo en caja, exterior, las inversiones y retiros realizados durante el horizonte de planificación.

- `void decisiones_stoch(double*y,double*aux_y,double*aux_ruido, double*imp,double*z,double*smin,int model_esc, int horizonte,int i)`

Dentro de esta función se solucionar un problema estocástico para un horizonte de h días fijado y así se toman las decisiones de enviar/retirar dinero. Para esto debe primero generar el formato SMPSS llamando a las funciones que lo crean. Luego llama a la función que resuelve el programa estocástico haciendo uso de las librerías del SMI. Finalmente llama a la función que lee las decisiones para el primer período.

- `void SmplsIO(const char* const)`

En esta función llama a las librerías del SMI para resolver el problema estocástico.

- `void lectura_solucion(double *Inv,
double *Ped,int t)`

Esta función lee la solución del problema estocástico para el primer período de un archivo de texto donde se encuentra la solución para todos los períodos del problema.

4.4. CALIBRACIÓN DE PARÁMETROS

4.4.1. EVALUACIÓN DE LA UTILIDAD ESPERADA

En el fichero `esperanza.cpp` principalmente se implementa la función que sirve para la calibración de parámetros de los algoritmos en-línea, a la cual se aplica los métodos numéricos de optimización. Además, se implementa la función para calcular la utilidad obtenida al aplicar un algoritmo a un flujo neto de caja (simulado o real) durante un horizonte de tiempo. A continuación, se detallan las funciones creadas para este propósito.

- `double expected(double a,double b,double c,double d,
int modelo,int algorit,int T)`

En esta función se calcula la utilidad esperada, obtenida al aplicar a flujos de caja generados por una función de distribución un algoritmo con ciertos parámetros durante un horizonte de tiempo. Esta utilidad esperada es calculada mediante simulaciones numéricas. Cada simulación consiste en generar valores para el flujo de caja y para el horizonte T empleando una distribución de probabilidad señalado por `modelo`. Luego se ejecuta sobre este flujo la función `algoritmo_linea_expect` para el algoritmo indicado por `algorit` y con los valores para los parámetros dados por a , b , c y d . Se registran la utilidad en $u0$ y el indicador de existencia de días malos en $p0$.

La cantidad de simulaciones a realizarse se determina por medio de técnicas simples de muestreo, detalles sobre el muestreo se aclaran en la sección 5.1. Se ejecutan N simulaciones iniciales (muestra piloto) para obtener un primer estimado del promedio y de la varianza de la utilidad. Conocidos estos valores se determina un número de simulaciones (tamaño de la muestra) que

permita estimar la utilidad esperada con el 95 % de confiabilidad y con un error permitido mediante la instrucción:

```
int k= int(ceil(4*varianza/pow(ERROR_MUESTRAL,2)));
```

similar a la ecuación 5.3. En cada simulación numérica se calcula u el promedio de la utilidad y p el promedio del indicador. Terminadas las k simulaciones numéricas se verifica si el valor promedio del indicador (riesgo) es inferior a al valor EPSILON. Si el riesgo supera este valor se asigna a la utilidad u el valor INFINITO, el cual se ha fijado igual a -500, que es una cantidad dos ordenes de magnitud más grande que los valores usuales de la utilidad para así evitar inestabilidades numéricas en los métodos de optimización. Al final, la función retorna el valor $-u$.

- `double util_algoritmo(double &uu,double *valor,int modelo,double*y, double *Smin,int T2,int algorit,double *P,char e,char*archivo)`

Esta función retorna la utilidad obtenida al aplicar el algoritmo `algorit` durante un horizonte `T2` a un flujo almacenado en el arreglo `y` generado por la distribución `modelo`. Mediante el parámetro `e` igual a `s` se obtiene un archivo de texto que detalla: el flujo generado, los saldos en el exterior y en caja, las inversiones, retiros, los costos, el beneficio, los días malos y la utilidad obtenidos con el algoritmo. Al final la función revisa los días en que el saldo en caja fue inferior al saldo mínimo y se realiza la penalización, mediante compras de dinero a un costo elevado, el valor del costo es constante y debe declararse en la cabecera `constantes.h`, así se obtiene una utilidad inferior a la primera.

- `double condicion_parametros(double*valor,int algorit, char &opcion)`

Esta función es la encargada de revisar que los valores asignados a los parámetros del algoritmo `algorit` sean razonables, es decir que los valores no sean negativos o por ejemplo la banda mínima no sea mayor a la banda máxima. Si esta función encuentra alguna incongruencia retorna el valor de INFINITO. Además asigna al parámetro `opcion` la letra `c` o `v` si el algoritmo es de bandas fijas o móviles respectivamente.

- `int cambio_algoritmo(int algorit)`

Por la similitud que tienen los algoritmos de bandas fijas con sus respectivas variantes de bandas móviles, la función que aplica los algoritmos en

línea admite valores para los algoritmos entre 1 y 4, y para diferenciar que clase de algoritmo es utiliza un parámetro `opcion` que indica si el algoritmo es de bandas fijas o móviles. Esta función es la encargada de cambiar los algoritmos de número 5, 6,7 y 8 a los números 1, 2, 3 y 4 respectivamente.

- `double calculo_utilidad(double *I,double *M,double *B, double *C, double *se,int T2)`

Esta es una función auxiliar que al entregarle arreglos con saldo exterior, inversiones, pedidos, costos y beneficio calcula y retorna el valor de la utilidad.

- `double calculo_utilidad_penal(double *I,double*M,double *B,double*C, double*se,double *penalizar,int T2)`

Esta función calcula la utilidad cuando se ha penalizado los días malos.

- `double risk(double*P,int T2)`

Esta es una función auxiliar que calcula y retorna el porcentaje de días en los cuales el saldo en caja es inferior al saldo mínimo.

- `void write_profit_algorit(double*y,double *valor,double *Smin, double *s,double *se,double *I,double *M,double *P, double *B, double*C,int algorit,double p,int T2,int modelo, double uti, char e,opcion,char* archivo)`

Esta función crea un archivo de texto con la información de la aplicación de un algoritmo en-línea.

4.4.2. IMPLEMENTACIÓN DE LOS MÉTODOS NUMÉRICOS DE OPTIMIZACIÓN

- `void nelder_mead(double *parametro,int modelo,int algorit, int T,char* archivo_iteracion,char*archivo_corrida)`

Esta función implementa el método de nelder mead, haciendo uso de los funciones de la clase lista. Terminada las iteraciones reporta dos archivos de texto uno con los puntos encontrados en cada iteración y el otro especifica la regla que se realizó en cada iteración.

- `void descenso(double*x0,int modelo,int algorit,int T, char* archivo_iter,char*archivo_punto)`

Esta función implementa el método de descenso profundo con la ayuda de las funciones `gradiente` y `producto_interno`.

- `double gradiente(double*x,double *grad,int modelo, int algorit,int T);`

Esta función almacena el gradiente de la función `expected` en un punto punto `x`.

- `double producto_interno(double*x,double *y,int M)`

Esta función calcula y retorna el producto interno entre dos vectores `x` y `y` de tamaño `M`.

Para el desarrollar el método de Nelder-Mead se crea la clase `lista`, la cual se encuentra en el fichero `lista.cpp`. Esta clase es una lista de doblemente enlazada que tiene: dos punteros uno a un nodo llamado `inicio` y el otro a un nodo llamado `fin`. Los nodos de la clase `lista` son estructuras que tiene dos punteros llamados `sig` y `ant` y cinco valores de tipo `double`. Además la clase `lista` tiene una variable de tipo entero llamada `tam`, la cual cuenta el número de nodos en la lista. La clase `lista` tiene las siguientes funciones:

- `void lista::lista();`

Esta función es el constructor de la lista, inicializa el valor `tam` con el valor cero y apunta los punteros `inicio` y `fin` hacia `NULL`.

- `struct nodo* crear(double a,double b,double c,double d, int modelo,int algorit,int T);`

Es la única función privada. Crea un nodo asignando a los cuatro campos numéricos de este los valores `a`, `b`, `c` y `d`, y en el quinto campo campo numérico almacena el valor entregado por la función `expected` calculada para un algoritmo `algorit`, cuyos parámetros son los valores de los cuatro primeros campos numéricos del nodo; para una distribución de probabilidad `modelo` y un horizonte de tiempo `T`. Hace que los punteros `sig` y `ant` apunten a `NULL` y retorna un puntero a este nuevo nodo creado.

- `void comb_afin (struct nodo *xx, struct nodo *xy, struct nodo *xz,double t,int modelo,int algorit,int T);`

Realiza una combinación afín entre los vectores formados por los cuatro valores numéricos de dos nodos `xx` y `xy`. El vector creado por la combinación afín es almacenado en el nodo `xz`.

- `void agregar_atras(double a,double b,double c,double d, int modelo,int algorit,int T)`

Esta función llamando a la función `crear` construye un nuevo nodo con los valores a , b , c y d recibidos como parámetros de esta función. Finalmente a este nuevo nodo creado lo agrega al final de la lista.

- `void puntos_iniciales(double a,double b,double c,double d, int modelo,int algorit,int T)`

Llama a la función `crear` para generar un primer nodo con los valores a , b , c y d . Luego llamando a la función `agregar_atras` se genera cinco puntos afinmente independientes, sumando al primer punto vectores (nodos) canónicos unitarios en todas las direcciones. Estos puntos son necesarios para comenzar las iteraciones del método de Nelder-Mead.

- `void sort()`

Esta función ordena ascendente la lista con respecto a los valores del quinto campo de los nodo. De esta manera el primer nodo de la lista tiene la utilidad esperada negativa más pequeña (para Nelder-Mead es el mejor punto) y el último nodo tiene el valor de utilidad esperada negativa más alto (para Nelder-Mead es el peor punto).

- `void promedio(struct nodo * xp)`

Obtiene los promedios entre los campos numéricos de los nodos de la lista y los almacena en los respectivos campos del nodo `xp`.

- `void accept (struct nodo *q)`

Inserta en la lista el nodo indicado por el puntero `q` en forma ordenada, comparando el valor del quinto campo con los valores de los quintos campos de los otros nodos de la lista. Finalmente la función elimina el último nodo de la lista (peor nodo) dejando únicamente los nodos que tiene las utilidad esperadas negativas más pequeñas.

- `void print_nodo(struct nodo *p, ostream& os)`

Imprime en la pantalla los campos numéricos de un nodo .

- `void print_nodo_bloque()`

Imprime los campos de todos los nodos de la lista en un archivo de texto

- `void shrink(int modelo,int algorit,int T)`

Esta función achica los valores de cada campo de los nodos de la lista.

4.5. CÁLCULO DE LA BRECHA DE OPTIMALIDAD

En el archivo `gap.cpp` están las funciones que calculan la brecha de optimalidad (`gap`) de los algoritmos. El `gap` es la diferencia entre la utilidad óptima y la utilidad del algoritmo dividida para la utilidad óptima. Para obtener el `gap` se requiere conocer la solución óptima de las instancias del problema, para lo cual se utiliza el software `mcf` el cual requiere que el programa este escrito en el formato DIMACS [42]. En este archivo se encuentran las funciones que ayudan a generar el formato del problema de flujo de costo mínimo para el PGSC.

- `void gap(int TT,int NN,double*flu_y,double*Smin, double *valor, int algorit, int model_esc, double*informacion, char*gap_archivo)`

Esta función calcula el valor del `gap` de un algoritmo para un flujo y saldo mínimo dados. Dentro de esta función se divide el arreglo de flujo en instancias de tamaño TT , a cada una de estas instancias se aplica el algoritmo escogido y se encuentra la utilidad óptimo, se calcula el `gap` de cada instancia y se obtiene el `gap` medio de las instancia.

- `void transfor_data(double*fflujo,double*flujo,double*Smin, int M)`

Los problemas de flujo de costo mínimo que son resueltos por el `mcf` requieren que las demandas, costos y las cotas para las capacidades sean enteras. Por esta razón se cambia los valores del flujo restando del flujo t el saldo mínimo del tiempo $t - 1$ más el flujo del tiempo t .

- `void dimacs(double*ddemand,double *iinf,int T2)`

Esta función crea el formato para la instancia del problema de flujo de costo mínimo, dado el flujo (`ddemand`) y el saldo mínimo (`iinf`).

- `double lectura(double*demand,double*inf,int TT,char e)`

Esta función lee la solución generada por el software y crea un archivo con las decisiones y saldos óptimos para la instancia dada.

Capítulo 5

RESULTADOS NUMÉRICOS

A continuación se presentan los principales resultados obtenidos durante una simulación computacional para medir el desempeño de los algoritmos en-línea descritos en el Capítulo 3, sobre instancias reales. Estas instancias fueron extraídas de una base de datos de registros históricos de flujo de caja en el Banco Central, para el período comprendido entre el 5 de Febrero del 2001 y el 4 de Agosto del 2005. Se consideraron horizontes de tiempo de diez, treinta y cien días. Como un paso previo, fue necesario calibrar los parámetros de los algoritmos empleando el esquema descrito en la sección 3.5.

5.1. CALIBRACIÓN DE PARÁMETROS

El procedimiento empleado para la calibración consistió en determinar numéricamente valores para los parámetros que permitan maximizar el valor esperado para la utilidad, $\bar{u}(x)$, obtenida por el algoritmo, suponiendo que el flujo diario en caja sigue una cierta distribución de probabilidad. Se requirió además que la probabilidad de que el saldo en caja descienda por debajo de un nivel mínimo (especificado por el Banco Central) no exceda un valor crítico.

La función a maximizar es $f : \mathbb{R}^n \rightarrow \mathbb{R}$ que asigna a cada combinación x de valores de los n parámetros de un algoritmo el valor medio de la utilidad, $\mu_u(x)$, para un horizonte de T días, asumiendo que el flujo diario de dinero sigue una distribución de probabilidad determinada. Si para alguna combinación de valores $\tilde{x} \in \mathbb{R}^n$, la probabilidad de que el saldo en caja descienda por debajo del nivel mínimo en al menos uno de los T días dentro del horizonte de tiempo supera un límite ϵ , entonces se define $f(\tilde{x}) := -\infty$. Llamaremos en adelante riesgo a esta probabilidad (ver sección 3.5 para más detalles en cuanto al modelo de calibración).

Se consideraron horizontes de tiempo de 10, 30 y 100 días. Por otra parte, luego de un análisis estadístico se seleccionaron tres distribuciones Sarima, Impulso y Sarima-Impulso. Se trabajó además, para fines comparativos, con una distribución Normal (NR). Detalles en cuanto a la selección del modelo estadístico se describen en el Apéndice A.

Evaluar la media $\mu_{u(x)}$ para cada combinación de parámetros de un algoritmo es difícil, debido a la forma de la función de utilidad para el PGSC y de la dependencia de la variable aleatoria flujo neto de caja, de la cual no se conoce su distribución de probabilidad. Por lo tanto, haremos uso de la estadística para inferir sobre la media de la variable aleatoria utilidad, $u(x)$.

La estadística extrae conclusiones acerca de una variable, en base a un análisis de muestreo. Las conclusiones que pueden extraerse acerca de una población dependen de la selección de la muestra. Una muestra es un conjunto de observaciones tomadas de una variable aleatoria [37].

Se debe seleccionar un tamaño de muestra lo bastante grande para obtener un estimación de la media de la utilidad, $\bar{u}(x)$, con un error especificado, e , a un nivel de confianza preestablecido, $(1 - \alpha)$. En este caso seleccionar una muestra de tamaño k , que brinde una estimación con las características mencionadas no es tan fácil, ya que no se conoce la varianza de la utilidad $\sigma_{u(x)}^2$. En estos casos cuando la varianza es desconocida el tamaño de la muestra se determina mediante ensayo y error [37].

Cuando la varianza es desconocida el error de precisión $e = |\bar{u}(x) - \mu_{u(x)}|$ a un nivel de confianza $(1 - \alpha)$ es:

$$e = \frac{t_{\alpha/2, k-1} S_{u(x)}}{\sqrt{k}} \quad (5.1)$$

donde $S_{u(x)}$ es la desviación estándar de la muestra y $t_{\alpha/2, k-1}$ es el punto porcentual superior a $\alpha/2$ de la distribución t con $k - 1$ grados de libertad ($P(-t_{\alpha/2, k-1} \leq t \leq t_{\alpha/2, k-1}) = 1 - \alpha$).

Para determinar el tamaño de la muestra por ensayo y error se selecciona una muestra piloto de tamaño $m > 30$ y se encuentra sus estadísticas: media, varianza y desviación estándar. A partir de las estadísticas muestrales y de la ecuación (5.1) se encuentra un tamaño k para la muestra mediante:

$$k = \left(\frac{t_{\alpha/2, k-1} S_{u(x)}}{e} \right)^2 \quad (5.2)$$

El valor de $t_{\alpha/2, k-1}$ se puede reemplazar por el punto porcentual superior a $(1 - \beta)$ de una distribución normal estándar Z .

Se calcula e para este tamaño k encontrado. El tamaño k se ajusta conforme al error calculado, aumentando o disminuyendo el número de observaciones. Con el tamaño ajustado se calcula nuevamente el error y se vuelve a ajustar el tamaño, este procedimiento se realiza hasta encontrar un tamaño que permita obtener un error menor o igual que el límite del error establecido.

Para muestrear la variable $u(x)$ se establece $1 - \beta = 0,9772$ y así $t_{\alpha/2, k-1} = 2$, con lo cual el tamaño de la muestra calculada satisface el error e . Así de la ecuación (5.2) se deduce:

$$k = \frac{\lceil 4S^2 \rceil}{e^2} \quad (5.3)$$

Para un algoritmo, una distribución de probabilidad del flujo de caja y un horizonte de tiempo específicos, cada simulación (observación) de la variable aleatoria $u(x)$ consiste en generar valores para el flujo neto de dinero en caja empleando la distribución de probabilidad y el horizonte de tiempo señalados. Se ejecuta luego el algoritmo sobre este flujo, con los valores para los parámetros dados por x , y se registran la utilidad $u(x)$ y un indicador $r(x) \in \{0, 1\}$ obtenidos al terminar el período. Este indicador se usa para calcular el riesgo y toma el valor de 1 si y sólo si el saldo en caja desciende por debajo del saldo mínimo en algún día.

Realizadas las k simulaciones para la variable aleatoria utilidad se registran los valores de la utilidad promedio \bar{u} y el riesgo promedio $\bar{r}(x)$. Finalmente, se estima el error de muestreo e_2 para el riesgo (al 95 % de confiabilidad) mediante la formula:

$$e_2 = \sqrt{\frac{1}{k}}.$$

Si $\bar{r}(x) + e_2$ supera al límite ϵ establecido, entonces se fija $f(x) := -\infty$, caso contrario se toma el valor promedio obtenido. Para evitar inestabilidades numéricas en los métodos de optimización se representó al infinito mediante el valor de 500, que es una cantidad dos ordenes de magnitud más grande que los valores usualmente calculados para la utilidad en las instancias consideradas.

Para los demás datos de entrada del problema, se trabajó con valores fijos durante la calibración: tasa de interés $r = 0,00015$, tasa por transferencias $c = 0,0002$, saldo en caja inicial $C_0 = 140,844$ y saldo exterior inicial de $E_0 = 20$ millones de dólares. El saldo mínimo se simula conjuntamente con el flujo neto de caja. El saldo mínimo se genera cada cinco días como función del flujo de caja neto promedio de cinco días, \bar{y} . De acuerdo a la siguiente relación lineal:

$$L = 75,615 - 3,3307Y$$

donde \bar{y} es el promedio semanal y L es el saldo mínimo semanal. Detalles en cuanto al saldo mínimo se describen en el Apéndice A.

La búsqueda de valores para los parámetros que maximicen la función f se realizó por medio de dos métodos numéricos: Nelder-Mead y descenso profundo,

los mismos que fueron expuestos en las subsecciones 3.5.2 y 3.5.3 respectivamente. En todos los casos, los valores obtenidos por el método de Nelder-Mead superan a aquellos encontrados por el algoritmo de descenso profundo, razón por la cual en lo que resta de esta sección, el análisis se concentrará únicamente en torno a los primeros.

En el Cuadro 5.1 se indican los valores iniciales utilizados en la aplicación del método. Para una descripción detallada del significado de cada parámetro, referirse a la página y cuadro señalados en la segunda columna.

Algoritmos de Bandas Fijas		
Algoritmo	Descripción	Parámetros iniciales
4-Bandas	Algoritmo 3.1 pág. 52	$b_{\text{máx}} = 160$ $b_{\text{high}} = 155$ $b_{\text{mín}} = 130$ $b_{\text{low}} = 140$
2-Bandas	Algoritmo 3.2 pág. 53	$b_{\text{máx}} = 160$ $a_1 = 20$ $b_{\text{mín}} = 130$ $a_2 = 20$
2-Bandas Lineal	Algoritmo 3.3 pág. 54	$b_{\text{máx}} = 160$ $a_1 = 1$ $b_{\text{mín}} = 130$ $a_2 = 1$
2-Bandas Cuadrático	Algoritmo 3.4 pág. 55	$b_{\text{máx}} = 150$ $a_1 = 0,05$ $b_{\text{mín}} = 100$ $a_2 = 0,05$
Algoritmos de Bandas Móviles		
Algoritmo	Descripción	Parámetros iniciales
4-Bandas móvil	Algoritmo 3.5 pág. 56	$A_1 = 130$ $B_1 = 110$ $A_2 = 80$ $B_2 = 100$
2-Bandas móvil	Algoritmo 3.6 pág. 57	$A_1 = 130$ $a_1 = 20$ $A_2 = 80$ $a_2 = 20$
2-B móvil Lineal	Algoritmo 3.7 pág. 58	$A_1 = 130$ $a_1 = 1$ $A_2 = 80$ $a_2 = 1$
2-B móvil Cuadrático	Algoritmo 3.8 pág. 59	$A_1 = 160$ $a_1 = 0,05$ $A_2 = 140$ $a_2 = 0,05$

Cuadro 5.1: Puntos iniciales.

Los restante puntos iniciales (el método de Nelder-Mead requiere de $n + 1$ pun-

tos iniciales afinmente independientes) se obtienen al sumar a los puntos iniciales señalados en la tabla vectores canónicos unitarios en todas las direcciones.

A continuación, se presentan los valores para los parámetros obtenidos por el método de Nelder-Mead para los diferentes algoritmos y distribuciones de probabilidad del flujo de caja, clasificados por horizonte de tiempo.

5.1.1. HORIZONTE DE 10 DÍAS

En el cuadro 5.2 se presentan los valores óptimos para los parámetros de cada algoritmo en-línea, encontrados por el método de Nelder-Mead, para un horizonte de 10 días con cada una de las cuatro distribuciones utilizadas para simular el flujo de caja. Se detallan además el valor esperado de la utilidad, el número de iteraciones del método, el número total de simulaciones numéricas efectuadas y el tiempo total requerido por el método para encontrar estos parámetros. Cabe señalar que para la calibración de estos parámetros se utilizó una muestra piloto de tamaño 300, un error e permitido menor a 5×10^{-4} y un $\epsilon = 0,05$.

Los valores calibrados para los parámetros de los algoritmos de bandas fijas para los horizontes de 10 y 30 días tratan de colocar la banda mínima alrededor del saldo mínimo promedio de 73,8 y a partir de esta establecen la banda máxima. Conforme el horizonte crece los algoritmos de bandas elevan los valores de sus parámetros en algunos casos estos parámetros son demasiado conservadores. Los parámetros calibrados para los algoritmos de bandas fijas tratan en su mayoría de mantener un monto pequeño sobre el saldo mínimo.

Se ha utilizado la notación Algoritmo-Distribución para indicar la distribución de probabilidad empleada para generar el flujo de dinero (SI=Sarima-Impulso, IM=Impulso y SA=Sarima).

Lineal Móvil Sarima obtiene la mejor utilidad esperada, igual a 0,121 millones de dólares seguido por Cuadrático Móvil SA con 0,115 millones de dólares y por 4-Bandas Móvil SA con 0,114 millones de dólares. 4-Bandas Móvil Normal y 4-Bandas Móvil Normal son los algoritmos que obtiene la peor utilidad esperada con 0,029 millones de dólares.

Con la distribución de probabilidad Sarima-Impulso el algoritmo que alcanza la mejor utilidad de 0,1 es Lineal Móvil SI. Para la distribución Impulso los algoritmo con la mejor utilidad de 0,06 son Cuadrático IM y 2-Bandas Móvil IM. Lineal Móvil

Algoritmo	Parámetros	\bar{u}	# it.	min.	# sim.
4-Bandas SI	$b_{\text{máx}} = 133,1$ $b_{\text{high}} = 89,7$ $b_{\text{mín}} = 84,61$ $b_{\text{low}} = 86,5$	0,084	110	14	2.096.8547
4-Bandas IM	$b_{\text{máx}} = 136,03$ $b_{\text{high}} = 118,45$ $b_{\text{mín}} = 78$ $b_{\text{low}} = 107,71$	0,059	153	8	1.198.2027
4-Bandas SA	$b_{\text{máx}} = 130,31$ $b_{\text{high}} = 121,32$ $b_{\text{mín}} = 73,99$ $b_{\text{low}} = 115,06$	0,094	126	19	28.457.314
4-Bandas NR	$b_{\text{máx}} = 146,6$ $b_{\text{high}} = 144,12$ $b_{\text{mín}} = 78,2$ $b_{\text{low}} = 139,99$	0,040	126	2	2.995.507
2-Bandas SI	$b_{\text{máx}} = 128,69$ $a_1 = 38,2$ $b_{\text{mín}} = 74,38$ $a_2 = 30,63$	0,084	140	17	25.461.807
2-Bandas IM	$b_{\text{máx}} = 137,68$ $a_1 = 26,56$ $b_{\text{mín}} = 75,17$ $a_2 = 12,49$	0,058	99	6	8.986.520
2-Bandas SA	$b_{\text{máx}} = 149,87$ $a_1 = 44,36$ $b_{\text{mín}} = 98,54$ $a_2 = 3,29$	0,089	103	17	25.461.756
2-Bandas NR	$b_{\text{máx}} = 158,65$ $a_1 = 25,16$ $b_{\text{mín}} = 82,14$ $a_2 = 28,87$	0,038	101	2	2.995.456
Lineal SI	$b_{\text{máx}} = 109,63$ $a_1 = 1,61$ $b_{\text{mín}} = 85,91$ $a_2 = 1,08$	0,089	92	14	20.968.409
Lineal IM	$b_{\text{máx}} = 126,69$ $a_1 = 1,33$ $b_{\text{mín}} = 97,75$ $a_2 = 1,13$	0,058	91	6	8.986.475
Lineal SA	$b_{\text{máx}} = 146,63$ $a_1 = 1,24$ $b_{\text{mín}} = 74,32$ $a_2 = 1,07$	0,094	113	22	32.950.574
Lineal NR	$b_{\text{máx}} = 130,84$ $a_1 = 1,37$ $b_{\text{mín}} = 82,5$ $a_2 = 1,08$	0,039	76	1	1.497.639
Cuadrático SI	$b_{\text{máx}} = 112,69$ $a_1 = 0,055$ $b_{\text{mín}} = 74,89$ $a_2 = 0,104$	0,089	140	21	31.452.821
Cuadrático IM	$b_{\text{máx}} = 118,54$ $a_1 = 0,036$ $b_{\text{mín}} = 74,08$ $a_2 = 0,076$	0,06	130	8	11.981.962
Cuadrático SA	$b_{\text{máx}} = 143,09$ $a_1 = 0,057$ $b_{\text{mín}} = 74,87$ $a_2 = 0,071$	0,086	99	16	23.964.054

Algoritmo	Parámetros	\bar{u}	# it.	min.	# sim.
Cuadrático NR	$b_{\text{máx}} = 147,35$ $a_1 = 0,064$ $b_{\text{mín}} = 77,78$ $a_2 = 0,056$	0,038	83	1	1.497.753
4-Bandas M. SI	$A_1 = 48,01$ $B_1 = 38,22$ $A_2 = 5,92$ $B_2 = 29,18$	0,094	111	11	16.475.287
4-Bandas M. IM	$A_1 = 178,27$ $B_1 = 140,54$ $A_2 = 1,25$ $B_2 = 120,13$	0,03	95	2	2.995.507
4-Bandas M. SA	$A_1 = 66,85$ $B_1 = 56,25$ $A_2 = 1,72$ $B_2 = 5,44$	0,114	119	28	41.937.094
4-Bandas M. NR	$A_1 = 223,86$ $B_1 = 177,04$ $A_2 = 1,01$ $B_2 = 163,69$	0,029	91	1	1.497.753
2-Bandas M. SI	$A_1 = 69,15$ $a_1 = 40,68$ $A_2 = 1,02$ $a_2 = 33,01$	0,087	85	11	16.475.185
2-Bandas M. IM	$A_1 = 67,904$ $a_1 = 36,44$ $A_2 = 7,94$ $a_2 = 38,15$	0,060	114	7	10.484.274
2-Bandas M. SA	$A_1 = 91,72$ $a_1 = 62,3$ $A_2 = 25,66$ $a_2 = 10,24$	0,107	124	29	43.434.848
2-Bandas M. NR	$A_1 = 194,35$ $a_1 = 31,1$ $A_2 = 0,74$ $a_2 = 25,34$	0,029	99	1	1.497.803
Lineal M. SI	$A_1 = 44,59$ $a_1 = 1,69$ $A_2 = 9,74$ $a_2 = 1,93$	0,1	105	17	25.461.608
Lineal M. IM	$A_1 = 203,77$ $a_1 = 1,34$ $A_2 = 0,75$ $a_2 = 1,13$	0,03	87	2	2.995.463
Lineal M. SA	$A_1 = 67,27$ $a_1 = 1,59$ $A_2 = 7,97$ $a_2 = 0,94$	0,121	126	38	56.914.628
Lineal M. NR	$A_1 = 199,3$ $a_1 = 1,24$ $A_2 = 3,25$ $a_2 = 1,21$	0,03	61	1	1.497.662
Cuadrático M. SI	$A_1 = 73,71$ $a_1 = 0,073$ $A_2 = 0,94$ $a_2 = 0,07$	0,074	126	15	22.466.301
Cuadrático M. IM	$A_1 = 236,74$ $a_1 = 0,062$ $A_2 = 0,94$ $a_2 = 0,06$	0,03	79	1	1.497.714

Cuadrático M. SA	$A_1 = 30,52$	$a_1 = 0,008$	0,115	125	32	47.928.108
	$A_2 = 25,65$	$a_2 = 0,1$				
Cuadrático M. NR	$A_1 = 223,02$	$a_1 = 0,06$	0,03	105	1	1.497.911
	$A_2 = 0,88$	$a_2 = 0,064$				

Cuadro 5.2: Parámetros óptimos para un horizonte de tiempo de 10 días.

SA alcanza la mejor utilidad de 0,121 con la distribución Sarima y 4-Bandas NR obtiene la mejor utilidad de 0,04 para la distribución Normal.

El tiempo que cada algoritmo tarda en encontrar los parámetros depende de la distribución de probabilidad. Los algoritmos simulados con Sarima y Sarima-Impulso requieren de mayor tiempo para encontrar los parámetros óptimos. Debido a que las distribuciones de probabilidad SA y SI explican mayor variabilidad del flujo neto. Las utilidades obtenidas con flujos de caja neto de distribución SA o SI tiene mayor variabilidad y por lo tanto requieren de una muestra más grande para obtener una estimación de la media con el error permitido y al nivel de confianza establecido.

El número promedio de iteraciones requeridas para encontrar los valores óptimos de los parámetros de los algoritmos es de 107, con un tiempo promedio de 12 minutos. Durante la optimización, se ejecutan en total un promedio de 17'364.555 simulaciones numéricas.

5.1.2. HORIZONTE DE 30 DÍAS

En el Cuadro 5.3 se presentan los valores óptimos de los parámetros encontrados por el método de Nelder-Mead para un horizonte de 30 días. En la calibración de estos parámetros se utilizó una muestra piloto de tamaño 300, un error e permitido menor a 5×10^{-3} y un $\epsilon = 0,05$.

Lineal Móvil Sarima obtiene la mejor utilidad esperada, igual a 0,64 millones de dólares seguido por 4-Bandas Móviles Sarima con 0,62 millones de dólares y por 2-Bandas Móviles Sarima con 0,61 millones de dólares. Cuadrático Móvil Normal es el algoritmos que obtiene la peor utilidad esperada de 0,01, seguido de 4-Bandas Móvil Normal, 2-Bandas Móvil Normal y Lineal Móvil Normal con una utilidad de 0,02 millones de dólares.

Con la distribución de probabilidad Sarima-Impulso el algoritmo que alcanza la mejor utilidad de 0,46 es Lineal Móvil SI. Para la distribución Impulso los algoritmo con la mejor utilidad de 0,29 es Lineal Móvil IM. Lineal Móvil SA alcanza la mejor utilidad de 0,64 con la distribución Sarima y 4-Bandas NR obtiene la mejor utilidad de 0,06 para la distribución Normal.

De igual manera que para el horizonte de 10 días los algoritmos que requieren mayor tiempo de maquina son los simulados con las distribuciones Sarima y Sarima-Impulso. Debido a que requieren de una muestra más grande para obtener una estimación de la media con el error permitido y al nivel de confianza establecido.

El número promedio de iteraciones requeridas para encontrar los valores óptimos de los parámetros de los algoritmos es de 86, con un tiempo promedio de 7 minutos. Durante la optimización, se ejecutan en total un promedio de 10'161.889 simulaciones numéricas.

Algoritmo	Parámetros	\bar{u}	# it.	min.	# sim.
4-Bandas SI	$b_{max} = 135,43$ $b_{high} = 130,35$ $b_{min} = 103,03$ $b_{low} = 116,45$	0,40	129	9	13.479.780
4-Bandas IM	$b_{max} = 155,58$ $b_{high} = 139,09$ $b_{min} = 75,46$ $b_{low} = 136,75$	0,24	130	3	4.732.901
4-Bandas SA	$b_{max} = 182,46$ $b_{high} = 172,36$ $b_{min} = 141,83$ $b_{low} = 165,91$	0,52	89	14	20.968.547
4-Bandas NR	$b_{max} = 193,02$ $b_{high} = 185,54$ $b_{min} = 170,72$ $b_{low} = 177,38$	0,06	31	1	199.700
2-Bandas SI	$b_{max} = 145,19$ $a_1 = 25,29$ $b_{min} = 101,49$ $a_2 = 16,98$	0,40	80	7	10.484.274
2-Bandas IM	$b_{max} = 155,63$ $a_1 = 22,5$ $b_{min} = 104,65$ $a_2 = 26,46$	0,23	87	3	4.867.698
2-Bandas SA	$b_{max} = 176,4$ $a_1 = 19,8$ $b_{min} = 146,89$ $a_2 = 15,74$	0,52	84	13	19.470.794
2-Bandas NR	$b_{max} = 219,11$ $a_1 = 19,87$ $b_{min} = 168,78$ $a_2 = 20,4$	0,04	30	1	149.775
Lineal SI	$b_{max} = 133,37$ $a_1 = 1,14$ $b_{min} = 105,03$ $a_2 = 1,13$	0,40	77	5	7.488.767
Lineal IM	$b_{max} = 139,79$ $a_1 = 1,31$ $b_{min} = 116,36$ $a_2 = 1,06$	0,24	89	3	5.332.002
Lineal SA	$b_{max} = 166,73$ $a_1 = 0,97$ $b_{min} = 151,15$ $a_2 = 1,02$	0,53	80	14	21.322.017
Lineal NR	$b_{max} = 204,47$ $a_1 = 0,84$ $b_{min} = 181,8$ $a_2 = 1,11$	0,04	32	1	1.677.484
Cuadrático SI	$b_{max} = 133,31$ $a_1 = 0,065$ $b_{min} = 74$ $a_2 = 0,042$	0,42	79	6	9.170.744
Cuadrático IM	$b_{max} = 130,72$ $a_1 = 0,027$ $b_{min} = 118,4$ $a_2 = 0,067$	0,24	91	3	5.391.912
Cuadrático SA	$b_{max} = 155,413$ $a_1 = 0,011$ $b_{min} = 154,39$ $a_2 = 0,05$	0,52	91	13	19.021.468

Algoritmo	Parámetros	\bar{u}	# it.	min.	# sim.
Cuadrático NR	$b_{max} = 192,883$ $a_1 = 0,008$ $b_{min} = 188,56$ $a_2 = 0,052$	0,03	47	1	174.738
4-Bandas M. SI	$A_1 = 71,59$ $B_1 = 63,3$ $A_2 = 9,75$ $B_2 = 63,01$	0,42	107	6	9.255.891
4-Bandas M. IM	$A_1 = 57,36$ $B_1 = 53,85$ $A_2 = 23,55$ $B_2 = 45,8$	0,28	122	4	5.991.013
4-Bandas M. SA	$A_1 = 90,22$ $B_1 = 88,99$ $A_2 = 56,37$ $B_2 = 71,65$	0,62	129	28	41.937.094
4-Bandas M. NR	$A_1 = 167,36$ $B_1 = 147,8$ $A_2 = 117,38$ $B_2 = 132,52$	0,02	40	1	1.497.753
2-Bandas M. SI	$A_1 = 74,4$ $a_1 = 32,98$ $A_2 = 17,77$ $a_2 = 26,83$	0,43	96	7	10.908.632
2-Bandas M. IM	$A_1 = 76,6$ $a_1 = 31,01$ $A_2 = 21,89$ $a_2 = 25,59$	0,27	76	3	3.994.014
2-Bandas M. SA	$A_1 = 111,88$ $a_1 = 32,17$ $A_2 = 52,62$ $a_2 = 19,34$	0,61	89	15	22.815.726
2-Bandas M. NR	$A_1 = 168$ $a_1 = 19$ $A_1 = 112,5$ $a_2 = 21,5$	0,02	34	1	124.813
Lineal M. SI	$A_1 = 56,96$ $a_1 = 1,73$ $A_2 = 23,56$ $a_2 = 1,22$	0,46	87	8	12.606.096
Lineal M. IM	$A_1 = 59,16$ $a_1 = 1,24$ $A_2 = 25,15$ $a_2 = 1,55$	0,29	130	5	7.888.173
Lineal M. SA	$A_1 = 95,31$ $a_1 = 1,59$ $A_2 = 63,85$ $a_2 = 0,8$	0,64	93	20	30.279.631
Lineal M. NR	$A_1 = 160,26$ $a_1 = 1,02$ $A_2 = 111,33$ $a_2 = 1,02$	0,02	51	1	174.728
Cuadrático M. SI	$A_1 = 42,84$ $a_1 = 0,018$ $A_2 = 25,75$ $a_2 = 0,084$	0,45	142	11	16.699.950
Cuadrático M. IM	$A_1 = 47,64$ $a_1 = 0,022$ $A_2 = 30,54$ $a_2 = 0,087$	0,28	144	5	7.938.093

Cuadrático M. SA	$A_1 = 91,55$	$a_1 = 0,014$	0,59	108	6	9.036.395
	$A_2 = 59,8$	$a_2 = 0,074$				
Cuadrático M. NR	$A_1 = 195,5$	$a_1 = 0,04$	0,01	43	1	99.850
	$A_2 = 110$	$a_2 = 0,053$				

Cuadro 5.3: Parámetros óptimos para un horizonte de tiempo de 30 días.

5.1.3. HORIZONTE DE 100 DÍAS

En el Cuadro 5.4 se detallan los valores óptimos encontrados para cada algoritmo. En la calibración de estos parámetros se utilizó una muestra piloto de tamaño 100, un error e permitido menor a 5×10^{-3} y un $\epsilon = 0,05$.

Lineal Móvil Sarima obtiene la mejor utilidad esperada, igual a 3,35 millones de dólares seguido por 2-Bandas Móviles Sarima con 3,32 millones de dólares y por 4-Bandas Sarima con 3,14 millones de dólares. 4-Bandas Móvil Normal es el algoritmos que obtiene la peor utilidad esperada de 1,04, seguido de Cuadrático Móvil Normal, 2-Bandas Móvil Normal con una utilidad de 1,09 millones de dólares. Con la distribución de probabilidad Sarima-Impulso el algoritmo que alcanza la mejor utilidad de 2,72 es Lineal Móvil SI. Para la distribución Impulso el algoritmo con la mejor utilidad de 1,9 es Lineal Móvil IM. Lineal Móvil SA alcanza la mejor utilidad de 3,35 con la distribución Sarima y Lineal Móvil NR obtiene la mejor utilidad de 1,42 para la distribución Normal.

El número promedio de iteraciones requeridas para encontrar los valores óptimos de los parámetros de los algoritmos es de 93, con un tiempo promedio de 37 minutos. Durante la optimización, se ejecutan en total un promedio de 17'853.041 simulaciones numéricas.

Algoritmo	Parámetros	\bar{u}	# it.	min	# sim.
4-Bandas SI	$b_{max} = 146,81$ $b_{high} = 137,68$ $b_{min} = 122,01$ $b_{low} = 122,51$	2,55	97	53	26.142.611
4-Bandas IM	$b_{max} = 152,90$ $b_{high} = 147,00$ $b_{min} = 126,91$ $b_{low} = 128,82$	1,73	112	78	38.376.959
4-Bandas SA	$b_{max} = 214,42$ $b_{high} = 195,24$ $b_{min} = 184,50$ $b_{low} = 190,11$	3,14	67	4	2.117.010
4-Bandas NR	$b_{max} = 208,47$ $b_{high} = 189,23$ $b_{min} = 105,80$ $b_{low} = 189,15$	1,19	79	1	295.397
2-Bandas SI	$b_{max} = 152,84$ $a_1 = 22,34$ $b_{min} = 117,45$ $a_2 = 17,81$	2,50	89	51	25.297.432
2-Bandas IM	$b_{max} = 162,07$ $a_1 = 20,88$ $b_{min} = 121,48$ $a_2 = 19,65$	1,69	88	64	31.500.763
2-Bandas SA	$b_{max} = 227,68$ $a_1 = 10,09$ $b_{min} = 184,31$ $a_2 = 20,69$	2,80	99	5	2.486.256
2-Bandas NR	$b_{max} = 192,02$ $a_1 = 28,95$ $b_{min} = 130,25$ $a_2 = 15,04$	1,32	106	1	500.535
Lineal SI	$b_{max} = 143,11$ $a_1 = 1,88$ $b_{min} = 125,72$ $a_2 = 0,72$	2,55	63	36	17.732.026
Lineal IM	$b_{max} = 149,94$ $a_1 = 1,40$ $b_{min} = 132,40$ $a_2 = 0,58$	1,74	101	74	36.243.555
Lineal SA	$b_{max} = 191,96$ $a_1 = 0,11$ $b_{min} = 191,45$ $a_2 = 0,93$	2,83	87	6	2.789.860
Lineal NR	$b_{max} = 174,47$ $a_1 = 1,05$ $b_{min} = 138,50$ $a_2 = 1,08$	1,34	88	1	254.370
Cuadrático SI	$b_{max} = 150,22$ $a_1 = 0,057$ $b_{min} = 118,84$ $a_2 = 0,040$	2,45	89	76	37.523.607
Cuadrático IM	$b_{max} = 154,86$ $a_1 = 0,051$ $b_{min} = 132,06$ $a_2 = 0,027$	1,68	86	68	33.716.239
Cuadrático SA	$b_{max} = 229,09$ $a_1 = 0,008$ $b_{min} = 193,17$ $a_2 = 0,047$	2,84	92	6	3.134.489

Algoritmo	Parámetros	\bar{u}	# it.	min.	# sim.
Cuadrático NR	$b_{max} = 165,06$ $a_1 = 0,013$ $b_{min} = 139,10$ $a_2 = 0,049$	1,31	79	1	320.013
4-Bandas M. SI	$A_1 = 114,46$ $B_1 = 61,25$ $A_2 = 18,43$ $B_2 = 59,20$	2,46	111	59	28.907.869
4-Bandas M. IM	$A_1 = 76,81$ $B_1 = 61,92$ $A_2 = 31,62$ $B_2 = 52,28$	1,86	133	104	51.300.566
4-Bandas M. SA	$A_1 = 273,54$ $B_1 = 196,13$ $A_2 = 146,74$ $B_2 = 194,35$	2,33	95	5	2.666.778
4-Bandas M. NR	$A_1 = 179,09$ $B_1 = 135,60$ $A_2 = 15,12$ $B_2 = 115,56$	1,04	92	1	270.780
2-Bandas M. SI	$A_1 = 81,02$ $a_1 = 28,70$ $A_2 = 26,18$ $a_2 = 25,43$	2,64	91	54	26.413.391
2-Bandas M. IM	$A_1 = 82,99$ $a_1 = 27,78$ $A_2 = 31,40$ $a_2 = 22,56$	1,86	90	75	36.711.266
2-Bandas M. SA	$A_1 = 136,39$ $a_1 = 22,18$ $A_2 = 100,09$ $a_2 = 15,44$	3,32	58	7	3.232.955
2-Bandas M. NR	$A_1 = 112,75$ $a_1 = 38,73$ $A_1 = 47,77$ $a_2 = 27,39$	1,37	99	1	402.068
Lineal M. SI	$A_1 = 64,30$ $a_1 = 1,58$ $A_2 = 31,96$ $a_2 = 1,33$	2,72	113	69	33.814.704
Lineal M. IM	$A_1 = 72,22$ $a_1 = 1,70$ $A_2 = 36,48$ $a_2 = 1,26$	1,90	96	74	36.465.102
Lineal M. SA	$A_1 = 128,20$ $a_1 = 0,91$ $A_2 = 91,17$ $a_2 = 1,11$	3,35	82	8	3.807.334
Lineal M. NR	$A_1 = 95,88$ $a_1 = 1,01$ $A_2 = 46,33$ $a_2 = 1,21$	1,42	86	1	369.246
Cuadrático M. SI	$A_1 = 80,24$ $a_1 = 0,047$ $A_2 = 23,23$ $a_2 = 0,092$	2,48	105	91	44.662.362
Cuadrático M. IM	$A_1 = 87,83$ $a_1 = 0,045$ $A_2 = 31,12$ $a_2 = 0,076$	1,70	102	80	39.205.695

Cuadrático M. SA	$A_1 = 188,27$	$a_1 = 0,009$	2,78	81	8	4.012.472
	$A_2 = 78,50$	$a_2 = 0,063$				
Cuadrático M. NR	$A_1 = 146,45$	$a_1 = 0,081$	1,09	99	1	623.617
	$A_2 = 39,79$	$a_2 = 0,045$				

Cuadro 5.4: Parámetros óptimos para un horizonte de tiempo de 100 días.

5.2. RESULTADOS PARA INSTANCIAS REALES

Compararemos en esta sección el desempeño de los algoritmos en-línea sobre instancias basadas en datos reales, obtenidos del registro histórico de flujo de caja del Banco Central. Para cada uno de los ocho algoritmos de bandas consideraremos cuatro variantes, determinadas por la distribución de probabilidad del flujo empleada durante la calibración de los parámetros. Usaremos la notación Algoritmo-Distribución para distinguir cada una de estas variantes (por ejemplo: Lineal SA).

Estocástico requiere de un conjunto de escenarios. Para generar los escenarios se ha utilizado las tres distribuciones de probabilidad. Así, se obtiene tres variantes para el algoritmo Estocástico: Estocástico SA, Estocástico IM y Estocástico SI.

Estocástico requiere formular el problema PEGSC para un horizonte h para tomar sus decisiones. Para la aplicación de Estocástico en estas instancias se ha establecido un horizonte $h = 6$ días.

5.2.1. HORIZONTE DE 10 DÍAS

La primera simulación con instancias reales consistió en trabajar con los datos históricos del 2001 hasta el 2005. Los valores diarios de flujo en caja en este período se agruparon para formar 115 instancias con horizontes de 10 días. Sobre cada instancia se corren las 35 variantes de los algoritmos en-línea, registrando para cada una la utilidad u obtenida al final del período y el número de días (de haberlos) en los cuales el saldo en caja descendió por debajo del nivel mínimo aceptable: Llamaremos como antes a estos días como “días malos”. A continuación, se resuelve el problema fuera-de-línea empleando el programa mcf [40] para encontrar el flujo de costo mínimo. Se obtiene así la utilidad óptima u^* bajo el supuesto de que toda la información del flujo de caja es conocida de antemano. Este valor se usa con el fin de evaluar la calidad de las soluciones de los distintos algoritmos. Existe, sin embargo, un inconveniente: debido a la presencia de días

malos, las soluciones de los algoritmos en-línea pueden no ser factibles para el problema fuera-de-línea y por lo tanto alcanzar valores de utilidad superiores a u^* . Fue necesario, por lo tanto, *corregir* las soluciones en-línea realizando “compras” de dinero a un costo alto, por el monto bajo el saldo mínimo. Esto tiene como consecuencia que la utilidad u descienda a un valor corregido \hat{u} que refleja esta compra de dinero.

Como una medida para evaluar la calidad de una solución en-línea se empleó la brecha de optimalidad (*gap*):

$$gap = \frac{u^* - \hat{u}}{u^*}$$

Notar que este valor mide la diferencia relativa entre la utilidad (corregida) obtenida mediante la aplicación de un algoritmo y la utilidad que podría obtenerse si fuera posible anticipar cómo va a ser el flujo de caja en el futuro. En otras palabras, es un indicador del costo relativo de la incertidumbre.

El Cuadro 5.5 detalla estos valores para el algoritmo con el mejor desempeño en cada instancia. Por otra parte, el Cuadro 5.6 resume el desempeño de cada algoritmo sobre todas las instancias: su brecha promedio de optimalidad (\overline{gap}), su brecha máxima ($gap_{\text{máx}}$), el número total de días malos y el número de instancias que presentaron al menos un día malo (“instancias malas”).

En el 5.5 se puede observar que en las instancias consideradas el algoritmo Estocástico Sarima-Impulso obtuvo la mejor solución en 36 instancias; Lineal y 2-Bandas Sarima-Impulso obtuvieron las mejores soluciones en 14 instancias; 4-Bandas Sarima-Impulso y Estocástico Impulso fueron los mejores en 12 instancias; Lineal Móvil Sarima-Impulso, obtuvo la mejor utilidad en 11 instancias; Cuadrático Sarima-Impulso obtuvo la mejor solución en 5 instancias; 2-Bandas Impulso obtuvo la mejor solución en 3 instancias; Estocástico Sarima-Impulso obtuvo la mejor solución en 2 instancias y los algoritmos 4-Bandas Impulso, 4-Bandas Sarima, Cuadrático Impulso, 2-Bandas Móvil Sarima-Impulso, Cuadrático Móvil Normal y Cuadrático Móvil Sarima-Impulso obtuvieron las utilidades más altas en una instancia.

El algoritmo Cuadrático IM en la instancia 19 tiene el valor del $gap = 0,56$ más alto, seguido Lineal Móvil SI que obtiene el $gap = 0,38$ en la instancias 57 y 68. Por otro lado, el algoritmo Cuadrático Móvil Normal tiene el valor más bajo del $gap = 0,004$ en la instancia 87, seguido de Estocástico Impulso con un $gap = 0,03$ obtenido en la instancia 60. 4-Bandas SA y 4-Bandas IM también obtienen un gap pequeño

Nº	Instancia	Algoritmo	u	\hat{u}	u^*	gap	días malos
1	05-02-01	Estocástico SI	0,081	0,081	0,106	0,24	0
2	19-02-01	Estocástico IM	0,093	0,093	0,116	0,20	0
3	12-03-01	Estocástico SI	0,112	0,112	0,138	0,18	0
4	26-03-01	Estocástico SI	0,085	0,085	0,099	0,14	0
5	16-04-01	Estocástico SI	0,118	0,118	0,149	0,21	0
6	07-05-01	Estocástico SI	0,082	0,079	0,103	0,23	1
7	28-05-01	Estocástico IM	0,098	0,098	0,121	0,19	0
8	11-06-01	Estocástico SI	0,090	0,090	0,115	0,22	0
9	25-06-01	Lineal SI	0,073	0,071	0,079	0,10	1
10	09-07-01	Lineal SI	0,077	0,075	0,085	0,13	1
11	13-08-01	Estocástico SI	0,115	0,115	0,143	0,20	0
12	27-08-01	Estocástico IM	0,083	0,082	0,098	0,16	1
13	10-09-01	Estocástico SI	0,095	0,095	0,121	0,21	0
14	24-09-01	Estocástico IM	0,087	0,083	0,095	0,12	2
15	15-10-01	Estocástico SI	0,102	0,102	0,127	0,20	0
16	05-11-01	Estocástico SI	0,086	0,086	0,112	0,23	0
17	19-11-01	Estocástico IM	0,091	0,091	0,117	0,23	0
18	03-12-01	4-Bandas SA	0,040	0,035	0,037	0,05	1
19	17-12-01	Cuadrático IM	0,042	0,025	0,056	0,56	2
20	14-01-02	Estocástico SI	0,114	0,114	0,142	0,20	0
21	28-01-02	4-Bandas IM	0,050	0,036	0,038	0,05	2
22	18-02-02	Estocástico IM	0,108	0,108	0,138	0,22	0
23	04-03-02	2-Bandas SI	0,062	0,062	0,074	0,16	0
24	18-03-02	Estocástico SI	0,087	0,087	0,105	0,17	0
25	08-04-02	Estocástico SI	0,074	0,074	0,089	0,17	0
26	22-04-02	4-Bandas SI	0,075	0,070	0,077	0,09	1
27	13-05-02	Estocástico SI	0,091	0,091	0,117	0,22	0
28	03-06-02	2-Bandas SI	0,062	0,062	0,073	0,14	0
29	17-06-02	Lineal Móvil SI	0,078	0,069	0,080	0,14	1
30	08-07-02	Cuadrático SI	0,047	0,047	0,069	0,32	0

Nº	Instancia	Algoritmo	u	\hat{u}	u^*	gap	días malos
31	22-07-02	Estocástico SI	0,094	0,094	0,118	0,21	0
32	05-08-02	Estocástico SI	0,073	0,072	0,092	0,21	1
33	19-08-02	Estocástico SI	0,089	0,089	0,117	0,24	0
34	02-09-02	2-Bandas SI	0,062	0,061	0,077	0,21	1
35	16-09-02	Estocástico SI	0,094	0,094	0,119	0,22	0
36	30-09-02	2-Bandas SI	0,062	0,059	0,079	0,24	1
37	21-10-02	Lineal SI	0,079	0,077	0,092	0,17	1
38	11-11-02	Estocástico SI	0,083	0,079	0,100	0,21	1
39	25-11-02	2-Bandas IM	0,053	0,037	0,044	0,15	2
40	06-01-03	Estocástico SI	0,129	0,129	0,162	0,20	0
41	20-01-03	Estocástico SI	0,101	0,101	0,130	0,22	0
42	03-02-03	2-Bandas SI	0,062	0,062	0,073	0,15	0
43	17-02-03	4-Bandas SI	0,075	0,060	0,067	0,10	2
44	10-03-03	Estocástico SI	0,100	0,100	0,127	0,21	0
45	24-03-03	Lineal SI	0,079	0,079	0,091	0,14	1
46	07-04-03	Estocástico SI	0,082	0,073	0,098	0,26	2
47	05-05-03	Cuadrático SI	0,074	0,074	0,095	0,22	0
48	19-05-03	Lineal Móvil SI	0,079	0,078	0,094	0,17	1
49	16-06-03	Estocástico SI	0,088	0,088	0,117	0,25	0
50	30-06-03	Lineal SI	0,066	0,066	0,078	0,15	0
51	14-07-03	Estocástico SI	0,064	0,064	0,098	0,35	0
52	28-07-03	Lineal SI	0,078	0,078	0,095	0,18	0
53	11-08-03	4-Bandas SI	0,072	0,072	0,105	0,31	0
54	25-08-03	Lineal Móvil SI	0,080	0,080	0,100	0,20	0
55	08-09-03	Lineal SI	0,073	0,073	0,091	0,20	0
56	22-09-03	Lineal Móvil SI	0,085	0,085	0,120	0,29	0
57	13-10-03	Lineal Móvil SI	0,082	0,082	0,133	0,38	0
58	27-10-03	Cuadrático SI	0,077	0,075	0,095	0,21	1
59	17-11-03	Lineal SI	0,072	0,072	0,106	0,31	0
60	01-12-03	Estocástico IM	0,030	0,024	0,024	0,03	1

Nº	Instancia	Algoritmo	u	\hat{u}	u^*	gap	días malos
61	15-12-03	Estocástico IM	0,028	0,028	0,042	0,34	0
62	12-01-04	Estocástico SA	0,104	0,104	0,143	0,27	0
63	26-01-04	2-Bandas SI	0,089	0,083	0,099	0,17	2
64	09-02-04	4-Bandas SI	0,070	0,064	0,071	0,09	1
65	01-03-04	2-Bandas SI	0,089	0,089	0,122	0,27	0
66	15-03-04	Estocástico SI	0,087	0,087	0,120	0,28	0
67	29-03-04	Lineal SI	0,072	0,072	0,095	0,25	0
68	19-04-04	Lineal Móvil SI	0,076	0,076	0,121	0,38	0
69	03-05-04	Cuadrático SI	0,046	0,045	0,065	0,32	1
70	17-05-04	2-Bandas SI	0,062	0,052	0,058	0,10	2
71	07-06-04	2-Bandas SI	0,057	0,057	0,080	0,29	0
72	21-06-04	Lineal Móvil SI	0,087	0,087	0,121	0,27	0
73	05-07-04	4-Bandas SI	0,066	0,066	0,083	0,21	0
74	19-07-04	Estocástico SI	0,084	0,084	0,126	0,33	0
75	02-08-04	2-Bandas SI	0,062	0,060	0,082	0,26	1
76	23-08-04	Lineal SI	0,075	0,075	0,109	0,31	0
77	20-09-04	Estocástico IM	0,023	0,023	0,031	0,27	0
78	04-10-04	4-Bandas SI	0,069	0,069	0,099	0,30	0
79	18-10-04	4-Bandas SI	0,072	0,069	0,081	0,14	1
80	08-11-04	Lineal Móvil SI	0,130	0,130	0,177	0,26	0
81	22-11-04	Cuadrático SI	0,084	0,079	0,100	0,21	1
82	10-01-05	Estocástico SI	0,110	0,110	0,149	0,27	0
83	24-01-05	4-Bandas SI	0,075	0,066	0,077	0,14	1
84	14-02-05	Estocástico SA	0,100	0,100	0,146	0,31	0
85	28-02-05	Lineal SI	0,064	0,064	0,090	0,29	0
86	14-03-05	Estocástico SI	0,079	0,079	0,118	0,33	0
87	04-04-05	Cuadrático M. NO	0,030	0,026	0,026	0,004	1
88	18-04-05	2-Bandas SI	0,062	0,062	0,092	0,32	0
89	02-05-05	2-Bandas SI	0,062	0,062	0,089	0,30	0
90	16-05-05	2-Bandas M. SI	0,087	0,087	0,133	0,35	0

Nº	Instancia	Algoritmo	u	\hat{u}	u^*	gap	días malos
91	06-06-05	4-Bandas SI	0,069	0,069	0,105	0,34	0
92	20-06-05	Estocástico SI	0,094	0,093	0,135	0,31	1
93	04-07-05	2-Bandas SI	0,057	0,057	0,084	0,32	0
94	18-07-05	Estocástico SI	0,101	0,101	0,126	0,20	0
95	01-08-05	Estocástico SI	0,079	0,067	0,101	0,33	2
96	22-08-05	Lineal Móvil SI	0,092	0,092	0,116	0,20	0
97	05-09-05	4-Bandas SI	0,061	0,061	0,094	0,35	0
98	19-09-05	Estocástico SI	0,077	0,077	0,113	0,32	0
99	03-10-05	2-Bandas SI	0,062	0,061	0,081	0,24	1
100	17-10-05	Estocástico IM	0,099	0,099	0,133	0,26	0
101	07-11-05	Cuadrático M. SI	0,1093	0,109	0,140	0,22	0
102	21-11-05	Estocástico IM	0,088	0,088	0,112	0,21	0
103	09-01-06	Lineal Móvil SI	0,117	0,117	0,156	0,25	0
104	23-01-06	Lineal SI	0,088	0,088	0,119	0,26	0
105	06-02-06	Estocástico SI	0,083	0,080	0,109	0,27	1
106	20-02-06	2-Bandas IM	0,053	0,050	0,072	0,31	1
107	13-03-06	Estocástico SI	0,080	0,080	0,109	0,26	0
108	27-03-06	2-Bandas IM	0,053	0,048	0,052	0,08	1
109	17-04-06	Lineal Móvil SI	0,113	0,113	0,144	0,22	0
110	08-05-06	Lineal SI	0,070	0,070	0,096	0,27	0
111	29-05-06	Lineal SI	0,078	0,078	0,098	0,21	0
112	12-06-06	Estocástico SI	0,088	0,088	0,115	0,24	0
113	26-06-06	4-Bandas SI	0,073	0,068	0,078	0,12	2
114	10-07-06	Estocástico IM	0,078	0,075	0,099	0,25	1
115	24-07-06	4-Bandas SI	0,069	0,069	0,088	0,21	0

Cuadro 5.5: Algoritmos con el mejor desempeño, para instancias reales con horizontes de tiempo de 10 días.

Algoritmo	\overline{gap}	$gap_{m\acute{a}x}$	instancias malas	días malos
4-Bandas SI	0,37	0,91	36	78
4-Bandas IM	0,44	0,87	14	24
4-Bandas SA	0,44	0,83	13	24
4-Bandas NR	0,58	0,74	9	12
2-Bandas SI	0,35	0,83	33	79
2-Bandas IM	0,43	0,71	14	26
2-Bandas SA	0,48	0,75	14	28
2-Bandas NR	0,59	0,75	8	12
Lineal SI	0,33	0,98	34	75
Lineal IM	0,45	0,82	12	19
Lineal SA	0,47	0,73	12	20
Lineal NR	0,58	0,72	9	12
Cuadrático SI	0,37	0,99	22	50
Cuadrático IM	0,44	0,65	12	23
Cuadrático SA	0,58	0,72	8	11
Cuadrático NR	0,61	0,75	6	8
4-Bandas M. SI	0,39	0,91	22	29
4-Bandas M. IM	0,69	0,83	5	7
4-Bandas M. SA	0,48	0,86	12	16
4-Bandas M. NR	0,69	0,83	5	7
2-Bandas M. SI	0,43	0,90	14	21
2-Bandas M. IM	0,45	0,99	12	16
2-Bandas M. SA	0,55	0,88	10	18
2-Bandas M. NR	0,69	0,83	5	7
Lineal M. SI	0,36	0,94	20	29
Lineal M. IM	0,69	0,83	5	7
Lineal M. SA	0,51	0,88	11	14
Lineal M. NR	0,69	0,83	5	7

Cuadrático M. SI	0,56	0,80	8	10
Cuadrático M. IM	0,68	0,83	5	7
Cuadrático M. SA	0,50	0,94	7	9
Cuadrático M. NR	0,68	0,83	5	7
Estocástico SI	0,28	0,93	59	111
Estocástico IM	0,27	0,93	52	94
Estocástico SA	0,40	0,97	37	57

Cuadro 5.6: Desempeño promedio de los algoritmos sobre instancias reales con horizontes de tiempo de 10 días.

de 0,05 en las instancias 18 y 21 respectivamente. El valor del gap en la instancia 60 indica que la utilidad podría ser mejorada sólo en un 3% si se conociera de antemano el flujo de caja neto.

10 instancias tienen 2 días malos, 19 instancias tiene un día malo y las 76 instancias restantes no tienen días malos.

En el Cuadro 5.6 se presentan los valores promedios del desarrollo de los algoritmos en todas las instancias reales disponibles de 10 días. Estocástico IM tiene el *gap* promedio más bajo, $\overline{gap} = 0,27$, y significa que en promedio el costo por no conocer el flujo neto real y utilizar el algoritmo Estocástico SA es del 26% de la utilidad óptima. Estocástico SI obtiene un $\overline{gap} = 0,28$, sin embargo, es el algoritmo que tiene la mayor cantidad de instancias malas 59 con 111 días malos, esto lo convierte en el algoritmo con más riesgo. Lineal SI y 2-Bandas SI obtienen valores de \overline{gap} iguales a 0,33 y 0,35 con 34 y 33 instancias malas respectivamente. El resto de algoritmos tiene valores de \overline{gap} más elevados, aunque tienen menor número de instancias malas. Por otro lado, 4-Bandas M. IM, 4-Bandas M. NR, 2-Bandas M. NR, Lineal M. IM y Lineal M. NR con 5 instancias malas y 7 días malos son los algoritmos más seguros pero con el valor del gap promedio más elevado de 0,69, lo cual significa que la utilidad perdida por no conocer el flujo de caja real y utilizar estos algoritmos es del 69%.

5.2.2. HORIZONTE DE 30 DÍAS

En la segunda simulación con instancias reales de igual forma se trabaja con los datos históricos del 2001 hasta el 2005. Los valores diarios de flujo en ca-

ja en este período se agruparon para formar 34 instancias con horizontes de 30 días. Igualmente sobre cada instancia se corren las 35 variantes de los algoritmos en-línea, registrando para cada una la utilidad u obtenida al final del período y el número de días en los cuales el saldo en caja descendió por debajo del nivel mínimo aceptable.

El Cuadro 5.7 detalla estos valores para el algoritmo con el mejor desempeño en cada instancia. Por otra parte, el Cuadro 5.8 resume el desempeño de cada algoritmo sobre todas las instancias: su brecha promedio de optimalidad (\overline{gap}), su brecha máxima ($gap_{\text{máx}}$), el número total de días malos y el número de instancias malas.

De las instancias consideradas en el 5.7 se observa que las variantes de Estocástico obtuvieron la mejor solución en casi todas las instancias. Estocástico Impulso fue el mejor en 18 instancias, Estocástico Sarima-Impulso obtuvo la mejor solución en 14 instancias, Estocástico Sarima obtuvo la mejor solución en 2 instancias y Cuadrático Sarima-Impulso fue el mejor en una instancia.

El algoritmo Estocástico Sarima-Impulso en la instancia 34 tiene el valor del $gap = 0,40$ más alto y Estocástico Impulso en la instancia 25 tiene el segundo valor alto de $gap = 0,38$. Por otro lado, el valor más bajo del gap es de $0,12$ alcanzado por Estocástico IM en las instancias 3 y 7 y por Estocástico SI en la instancia 5, esto significa que la utilidad en dichas instancias podría ser mejorada en un 12% si se conociera de antemano el flujo de caja neto.

La instancia 25 tiene el mayor número de días malos (10) al aplicar el algoritmo Estocástico IM y sólo 3 instancias no tiene días malos.

En el Cuadro 5.8 se presentan los valores promedios del desarrollo de los algoritmos en todas las instancias reales de 30 días. Dos variantes de Estocástico obtienen los mejores valores para el gap: Estocástico IM con un $\overline{gap} = 0,21$ es el valor más bajo, sin embargo, es el algoritmo que tiene la mayor cantidad de 32 instancias malas con 107 días malos, esto lo convierte en el algoritmo con más riesgo. Estocástico SI obtiene un $\overline{gap} = 0,22$ con 30 instancias y 97 días malos. El resto de algoritmos tiene valores de \overline{gap} más elevados, aunque tienen menor número de instancias malas.

Nº	Instancia	Algoritmo	u	\hat{u}	u^*	gap	días malos
1	05-02-01	Estocástico SI	0,49	0,49	0,58	0,14	0
2	26-03-01	Estocástico SI	0,47	0,47	0,55	0,15	1
3	28-05-01	Estocástico IM	0,48	0,47	0,54	0,12	3
4	27-08-01	Estocástico IM	0,40	0,40	0,47	0,15	3
5	15-10-01	Estocástico SI	0,55	0,55	0,62	0,12	1
6	28-01-02	Estocástico SI	0,19	0,18	0,27	0,33	4
7	18-03-02	Estocástico IM	0,35	0,35	0,39	0,12	2
8	13-05-02	Estocástico SI	0,32	0,31	0,41	0,25	3
9	08-07-02	Estocástico IM	0,29	0,28	0,34	0,17	1
10	19-08-02	Estocástico SI	0,35	0,34	0,43	0,19	3
11	30-09-02	Estocástico IM	0,23	0,22	0,32	0,31	5
12	20-01-03	Estocástico SI	0,36	0,35	0,43	0,19	3
13	10-03-03	Estocástico SI	0,44	0,43	0,51	0,15	2
14	16-06-03	Estocástico IM	0,32	0,32	0,43	0,24	0
15	28-07-03	Estocástico IM	0,32	0,32	0,42	0,24	0
16	08-09-03	Estocástico IM	0,35	0,35	0,46	0,25	1
17	27-10-03	Estocástico IM	0,24	0,21	0,28	0,28	6
18	15-12-03	Estocástico IM	0,44	0,42	0,57	0,26	5
19	09-02-04	Estocástico SI	0,31	0,26	0,40	0,35	9
20	29-03-04	Estocástico IM	0,34	0,34	0,41	0,18	3
21	17-05-04	Cuadrático SI	0,20	0,17	0,24	0,26	7
22	05-07-04	Estocástico IM	0,29	0,28	0,39	0,28	2
23	04-10-04	Estocástico SI	0,41	0,38	0,50	0,25	4
24	10-01-05	Estocástico IM	0,61	0,58	0,70	0,16	3
25	28-02-05	Estocástico IM	0,27	0,23	0,37	0,38	10
26	18-04-05	Estocástico SI	0,46	0,42	0,52	0,19	7
27	06-06-05	Estocástico IM	0,45	0,44	0,52	0,15	5
28	18-07-05	Estocástico IM	0,49	0,45	0,62	0,27	7
29	05-09-05	Estocástico SI	0,34	0,32	0,41	0,20	4
30	17-10-05	Estocástico IM	0,62	0,61	0,70	0,14	4

31	23-01-06	Estocástico SA	0,33	0,32	0,48	0,33	3
32	13-03-06	Estocástico IM	0,31	0,30	0,47	0,36	2
33	08-05-06	Estocástico SI	0,34	0,34	0,47	0,28	1
34	26-06-06	Estocástico SA	0,21	0,21	0,35	0,40	1

Cuadro 5.7: Algoritmos con el mejor desempeño, para instancias reales con horizontes de tiempo de 30 días.

5.2.3. HORIZONTE DE 100 DÍAS

En esta simulación con instancias reales se trabaja con los datos históricos del 2001 hasta el 2005. Los valores diarios de flujo en caja en este período se agruparon para formar 8 instancias con horizontes de 100 días. Igualmente sobre cada instancia se corren las 35 variantes de los algoritmos en-línea, registrando para cada una la utilidad u obtenida al final del período y el número de días en los cuales el saldo en caja descendió por debajo del nivel mínimo aceptable.

El Cuadro 5.9 detalla estos valores para el algoritmo con el mejor desempeño en cada instancia. Por otra parte, el Cuadro 5.10 resume el desempeño de cada algoritmo sobre todas las instancias: su brecha promedio de optimalidad (\overline{gap}), su brecha máxima ($gap_{\text{máx}}$), el número total de días malos y el número de instancias malas.

De las instancias consideradas en el 5.9 se observa que Estocástico Impulso obtuvo la mejor solución 5 instancias, Estocástico Sarima-Impulso fue el mejor en 2 instancia y Estocástico Sarima obtuvo la mejor solución en una instancia. El algoritmo Estocástico SI en la instancia 5 tiene el valor del $gap = 0,23$ más alto y Estocástico IM en la instancia 1 tiene el valor más bajo del $gap = 0,07$, esto significa que la utilidad para dicha instancia podría ser mejorada sólo en un 7% si se conociera de antemano el flujo de caja neto.

La instancia 7 tiene el mayor número de días malos (30) al aplicar el algoritmo Estocástico IM y las instancia 1 presenta el menor número de días malos (5).

En el Cuadro 5.10 se presentan los valores promedios del desarrollo de los algoritmos en todas las instancias reales de 100 días. Las variantes de Estocástico obtienen los mejores valores de gap promedio, aunque todas sus instancias son malas. Estocástico IM con un $\overline{gap} = 0,11$ es el valor más bajo y significa que en promedio el costo por no conocer el flujo neto real y utilizar el algoritmo Estocásti-

Algoritmo	\overline{gap}	$gap_{m\acute{a}x}$	instancias malas	días malos
4-Bandas SI	0,47	0,71	10	16
4-Bandas IM	0,50	0,65	7	11
4-Bandas SA	0,81	0,96	1	2
4-Bandas NR	0,90	1,00	1	2
2-Bandas SI	0,47	0,74	9	15
2-Bandas IM	0,54	0,78	6	10
2-Bandas SA	0,81	1,00	1	2
2-Bandas NR	0,90	0,98	1	2
Lineal SI	0,45	0,71	10	16
Lineal IM	0,52	0,77	7	11
Lineal SA	0,79	0,99	1	2
Lineal NR	0,92	1,00	1	2
Cuadrático SI	0,40	0,57	8	13
Cuadrático IM	0,52	0,77	7	11
Cuadrático SA	0,80	0,99	1	2
Cuadrático NR	0,94	1,00	1	2
4-Bandas M. SI	0,46	0,72	7	11
4-Bandas M. IM	0,50	0,76	3	5
4-Bandas M. SA	0,73	0,97	1	2
4-Bandas M. NR	0,92	0,98	1	2
2-Bandas M. SI	0,46	0,78	6	10
2-Bandas M. IM	0,49	0,80	3	5
2-Bandas M. SA	0,71	0,97	1	2
2-Bandas M. NR	0,92	0,98	1	2
Lineal M. SI	0,41	0,66	6	10
Lineal M. IM	0,45	0,71	4	6
Lineal M. SA	0,71	0,97	1	2
Lineal M. NR	0,93	0,99	1	2

Cuadrático M. SI	0,43	0,68	6	10
Cuadrático M. IM	0,47	0,72	5	8
Cuadrático M. SA	0,73	0,95	2	3
Cuadrático M. NR	0,96	1,00	1	2
Estocástico SI	0,22	0,38	30	97
Estocástico IM	0,21	0,42	32	107
Estocástico SA	0,37	0,66	18	30

Cuadro 5.8: Desempeño promedio de los algoritmos sobre instancias reales con horizontes de tiempo de 30 días.

Nº	Instancia	Algoritmo	u	\hat{u}	u^*	gap	días malos
1	05-02-01	Estocástico IM	3,69	3,68	3,95	0,07	5
2	14-01-02	Estocástico IM	2,04	1,99	2,27	0,13	15
3	08-07-02	Estocástico SI	1,54	1,49	1,72	0,14	14
4	19-04-04	Estocástico IM	1,28	1,22	1,52	0,20	12
5	20-09-04	Estocástico SI	1,70	1,46	1,90	0,23	19
6	28-02-05	Estocástico IM	2,10	2,02	2,45	0,18	22
7	01-08-05	Estocástico IM	2,73	2,49	2,88	0,14	30
8	09-01-06	Estocástico SA	3,46	3,43	3,95	0,13	8

Cuadro 5.9: Algoritmos con el mejor desempeño, para instancias reales con horizontes de tiempo de 100 días.

Algoritmo	\overline{gap}	$gap_{m\acute{a}x}$	instancias malas	días malos
4-Bandas SI	0,37	0,49	5	15
4-Bandas IM	0,41	0,54	5	12
4-Bandas SA	0,71	0,92	1	1
4-Bandas NR	0,57	0,71	2	7
2-Bandas SI	0,39	0,51	4	17
2-Bandas IM	0,42	0,58	5	16
2-Bandas SA	0,73	0,92	1	1
2-Bandas NR	0,51	0,67	3	10
Lineal SI	0,36	0,48	4	18
Lineal IM	0,40	0,53	4	14
Lineal SA	0,74	0,93	1	1
Lineal NR	0,51	0,67	3	5
Cuadrático SI	0,38	0,49	6	19
Cuadrático IM	0,42	0,56	4	9
Cuadrático SA	0,80	0,98	1	1
Cuadrático NR	0,52	0,68	3	6
4-Bandas M. SI	0,42	0,55	3	11
4-Bandas M. IM	0,39	0,51	2	4
4-Bandas M. SA	0,80	0,88	0	0
4-Bandas M. NR	0,67	0,83	2	5
2-Bandas M. SI	0,34	0,46	3	11
2-Bandas M. IM	0,37	0,50	3	10
2-Bandas M. SA	0,69	0,92	1	1
2-Bandas M. NR	0,48	0,66	2	2
Lineal M. SI	0,33	0,43	3	8
Lineal M. IM	0,35	0,46	3	6
Lineal M. SA	0,65	0,87	1	1
Lineal M. NR	0,45	0,61	3	4

Cuadrático M. SI	0,38	0,50	5	14
Cuadrático M. IM	0,41	0,55	4	11
Cuadrático M. SA	0,81	0,97	1	1
Cuadrático M. NR	0,63	0,86	2	6
Estocástico SI	0,12	0,20	8	116
Estocástico IM	0,11	0,16	8	140
Estocástico SA	0,23	0,42	8	58

Cuadro 5.10: Desempeño promedio de los algoritmos sobre instancias reales con horizontes de tiempo de 100 días.

co IM es del 11 % de la utilidad óptima, sin embargo, es el algoritmo que tiene el mayor número de 140 días malos, esto lo convierte en el algoritmo con más riesgo. Estocástico SI obtiene un $\overline{gap} = 0,12$ con 116 días malos. Estocástico SA obtiene un $\overline{gap} = 0,23$ 58 días malos. El resto de algoritmos tiene valores de \overline{gap} más elevados, aunque con menor número de días malos.

En todos los horizontes de tiempo analizados las variantes del algoritmo Estocástico obtienen los mejores valores de gap y son los mejores en la mayoría de instancias, sin embargo en muchos casos estos algoritmos son los más riesgosos.

5.2.4. REGISTRO HISTÓRICO DE 500 DÍAS

En esta simulación con instancias reales se trabaja con los datos históricos del 2001 hasta el 2003, que forman una sola instancia con horizonte de 500 días. Igualmente sobre cada instancia se corren las 35 variantes de los algoritmos en línea, registrando para cada una la utilidad u obtenida al final del período, la brecha de optimalidad (gap) y el número de días en los cuales el saldo en caja descendió por debajo del nivel mínimo aceptable. El Cuadro 5.11 resume el desempeño de cada algoritmo sobre la instancia.

La utilidad óptima de la instancia es 35,756. El algoritmo que obtiene el valor del gap más pequeño de 0,04 es Estocástico SI con una utilidad corregida 34,17 y 61 días malos. Estocástico IM obtiene el segundo mejor valor de gap de 0,05 con una utilidad corregida de 34,14, y 79 días malos. Estocástico SA obtiene el tercer mejor valor del gap de 0,07 con una utilidad corregida de 33,30 y 31 días malos. Los algoritmos que tiene la mayor utilidad son las variantes de Estocástico, sin

Algoritmo	u	\hat{u}	gap	días malos
4-Bandas SI	31,07	30,96	0,13	13
4-Bandas IM	30,58	30,58	0,14	9
4-Bandas SA	26,12	26,12	0,27	0
4-Bandas NR	27,88	27,86	0,22	6
2-Bandas SI	31,10	31,01	0,13	13
2-Bandas IM	30,61	30,55	0,15	9
2-Bandas SA	25,84	25,84	0,28	0
2-Bandas NR	29,52	29,48	0,18	7
Lineal SI	31,22	31,11	0,13	13
Lineal IM	30,88	30,79	0,14	10
Lineal SA	25,35	25,35	0,29	0
Lineal NR	29,32	29,28	0,18	4
Cuadrático SI	30,71	30,59	0,14	8
Cuadrático IM	30,27	30,18	0,16	7
Cuadrático SA	24,40	24,40	0,32	0
Cuadrático NR	28,80	28,74	0,20	6
4-Bandas M. SI	30,61	30,58	0,14	5
4-Bandas M. IM	30,97	30,97	0,13	3
4-Bandas M. SA	20,12	20,12	0,44	0
4-Bandas M. NR	26,71	26,71	0,25	1
2-Bandas M. SI	31,39	31,37	0,12	7
2-Bandas M. IM	31,17	31,17	0,13	0
2-Bandas M. SA	26,59	26,59	0,26	0
2-Bandas M. NR	29,54	29,54	0,17	0
Lineal M. SI	31,67	31,66	0,11	5
Lineal M. IM	31,31	31,30	0,12	2
Lineal M. SA	27,16	27,16	0,24	0
Lineal M. NR	29,98	29,98	0,16	1

Cuadrático M. SI	29,85	29,73	0,17	15
Cuadrático M. IM	28,90	28,80	0,19	11
Cuadrático M. SA	22,10	22,09	0,38	2
Cuadrático M. NR	27,03	27,01	0,24	4
Estocástico SI	34,62	34,17	0,04	61
Estocástico IM	34,61	34,14	0,05	79
Estocástico SA	33,48	33,30	0,07	31

Cuadro 5.11: Desempeño de los algoritmos sobre la instancia real con horizonte de tiempo de 500 días.

embargo, tienen el mayor número de instancias malas. Los algoritmos Cuadrático SA, 4-Bandas M. SA y Cuadrático M. SA son los algoritmos con menor número de días malos, pero tienen los valores de gap más altos.

Capítulo 6

CONCLUSIONES

Entre las funciones del Banco Central del Ecuador está el manejo de cierta cantidad de dinero formado por depósitos que la banca privada realiza en esta identidad. El fin de mantener este saldo es el de asegurar la suficiente liquidez monetaria para garantizar el correcto funcionamiento del sistema financiero. Si bien la función del banco no es hacer producir este dinero, se ha observado que los montos de saldo superan considerablemente a los requerimientos (retiros) del sistema, manteniendo así, enormes cantidades de dinero inactivas. Por esto, se ha planteado la idea de invertir el excedente de dinero en el extranjero, de una manera segura, como una alternativa para generar recursos adicionales para el fisco.

El problema de gestión de saldo en caja del Banco Central consiste en decidir respecto a cuándo y cuánto dinero enviar o retirar del extranjero, de tal forma que el saldo en caja neto permita cubrir los requerimientos del sistema financiero. Para mantener la suficiente liquidez el Banco ha establecido un valor de saldo mínimo ajustado semanalmente, como una medida de seguridad. Para tomar estas decisiones el Banco debe tomar en cuenta principalmente los saldos en caja y en el extranjero; el flujo de caja neto resultante de las transacciones del sistema financiero, las tasas de inversión y las tasas (costos) por transferencia de dinero. En base a estos parámetros, se ha planteado un modelo de optimización POGSC en el Capítulo 2 para optimizar las utilidades generadas por las inversiones, mientras se respetan los valores de saldo mínimo.

POGSC se convierte en un programa estocástico cuando las tasas de retorno y/o el flujo de caja son desconocidos. En esta tesis se abordó el problema cuando el flujo de caja neto es aleatorio y se supuso el conocimiento de las tasas de retorno. Se consideraron distribuciones de probabilidad para este flujo y se obtuvo un árbol de escenarios con el cual se formuló el modelo estocástico (PEGSC) para el problema de gestión de saldo en caja del Capítulo 2. Sin embargo, resolver el programa estocástico correspondiente al PGSC no es computacionalmente practicable, aún para un horizonte de planificación mediano (15 días), debido a que los horizontes de tiempo considerados en instancias reales conducirían a programas lineales con una cantidad astronómica de variables (en el orden de 10^{20}).

Sin embargo, la programación estocástica puede aún ser empleada para el desarrollo de algoritmos en-línea que alcanzan buenos resultados. Se propusieron y desarrollaron en el Capítulo 3 algoritmos en-línea: algoritmos de bandas fijas, bandas móviles que son modificaciones de los primeros algoritmos y finalmente un algoritmo que emplea técnicas de programación estocástica para la toma de decisiones. Estos algoritmos toman decisiones conforme ingresa nueva información del problema y son fácilmente aplicables. Tanto los algoritmos de bandas fijas como móviles toman sus decisiones en base a los saldos en caja y exterior, con la ayuda de ciertos parámetros. Dada la forma de los algoritmos propuestos estos pueden alcanzar en algunas instancias utilidades superiores a las del óptimo violando la restricción de seguridad. Cuando en un día no se cumple con la restricción de seguridad se lo llama *día malo*. Para penalizar a un día malo se supone que se puede realizar la “compra” de dinero a un costo alto, por un monto igual al faltante para cubrir el valor del saldo mínimo impuesto.

Estos algoritmos requieren de ciertos parámetros para su aplicación, los cuales fueron calibrados planteando un modelo de optimización no lineal para el PGSC y utilizando los métodos numéricos descritos en el Capítulo 3. El tiempo que el método tarda en encontrar los parámetros óptimos de un algoritmo depende de la distribución de probabilidad con la cual se este calibrando y del nivel de error permitido para la muestra. Así, cuando se trabaja con las distribuciones Sarima y Sarima-Impulso se requieren de mayor tiempo para encontrar los parámetros, ya que estas explican mayor variabilidad del flujo de caja neto, por lo tanto requieren de una muestra más grande para obtener una estimación de la media con el error permitido y al nivel de confianza establecido. Si el valor para la probabilidad permitido para el riesgo (numero de días malos) es muy pequeño, el método para algunas distribuciones y algoritmos no encontrara una combinación de parámetros factible.

Calibrados los parámetros se aplicó los algoritmos a instancias reales de 10, 30, 100 y 500 días. Los valores de utilidad obtenidos por estos algoritmos fueron comparados con la utilidad óptima, para encontrar la brecha de optimalidad *gap* en cada caso. La solución óptima de los algoritmos (solución del problema fuera-de-línea) se la obtiene una vez terminado el período de planificación cuando todos los valores de flujo de caja neto son ya conocidos. Se demostró en el Capítulo 2 que en este caso el PGSC se puede reducir a un problema de flujo de costo mínimo. El valor del *gap* mide la diferencia relativa entre la utilidad (corregida) obtenida mediante la aplicación de un algoritmo y la utilidad que podría obtenerse si fuera posible anticipar cómo va a ser el flujo de caja en el futuro, es decir, es un indicador del costo relativo de la incertidumbre. Así, el *gap* de un algoritmo significa que

el costo por no conocer información perfecta sobre el flujo de caja futuro y aplicar dicho algoritmo es de $gap\%$ de la utilidad óptima; o lo que es lo mismo, significa el % en que podría ser mejorada la utilidad si se conociera el flujo de caja futuro.

A continuación, se resaltan algunas conclusiones importantes:

- La ventaja de los algoritmos en-línea está en la forma sencilla de tomar sus decisiones y en no requerir de información acerca de eventos futuros, haciendo su implementación más fácil.
- Un problema común, que se ve reflejado en el ejemplo del Capítulo 3 es que a menudo los algoritmos de bandas fijas como móviles realizan retiros en base al decaimiento del saldo en caja, pero mientras las transacciones se efectivizan puede recuperarse el saldo debido a depósitos locales. Como consecuencia, se tendrá un saldo en caja muy alto y deberá volver a enviarse el dinero de vuelta, causando únicamente costos de transferencia inútiles. Aún peor, cuando se realizan inversiones en base al crecimiento esporádico del saldo en caja, mientras se efectiviza el dinero el saldo en caja puede caer bajo el saldo mínimo, poniendo en riesgo la liquidez del sistema, debido a retiros en los días siguientes. Por este motivo, deben realizarse retiros del extranjero, es decir, volver a pedir el dinero invertido; y así sólo se generan costos por transferencia.
- Aparentemente, una buena estrategia consiste en enviar/retirar cantidades moderadas de dinero de una sola vez que mediante envíos/retiros sucesivos, para así evitar realizar grandes saltos en el saldo en caja y disminuir el número de transacciones realizadas, además, en el caso de las inversiones el dinero permanece más tiempo en el extranjero, generando más utilidades.
- Los algoritmos de bandas fijas resultan ser poco flexibles ante los cambios del saldo mínimo. Con los mismos parámetros, un algoritmo puede resultar demasiado riesgoso en algunos días y en otros simplemente invertir cantidades de dinero pequeñas y pedir cantidades enormes. Lo que no sucede con los algoritmos de bandas móviles que se van adaptando conforme las exigencias del saldo mínimo.
- Generalmente las variantes del Algoritmo Estocástico resultaron obtener los mejores valores para el gap , sin embargo también resultaron ser los algoritmos más riesgosos. Las variantes de los algoritmo de 4-Bandas y Lineal le

siguen a las variantes del algoritmo estocástico en el valor del *gap*, con un menor valor de riesgo. Generalmente los algoritmos de bandas fijas resultaron ser más seguros pero sus valores de *gap* son más grandes. Se puede decidir aplicar un algoritmo de acuerdo a prioridades de riesgo o ganancia. Si se desea obtener mayor utilidad se expone a un mayor riesgo, y si se desea ganar en seguridad se pierde en utilidad.

- La solución óptima para el problema en todas las instancias probadas, refleja un comportamiento particular: se puede observar que al realizar una inversión I_t en el día t no se deben realizar retiros de dinero durante los

$$k = \lceil \frac{c_t^I + c_t^M}{r_t} \rceil + 1$$

días siguientes, es decir $M_s = 0$, $s = t + 1, \dots, t + k$. A partir del día $t + k + 1$ se pueden realizar retiros de dinero por cualquier cantidad de dinero disponible. Es decir, siempre que se realice una inversión de dinero se deben dejar pasar k días para retirar algún monto de dinero. La razón es debido a los flujos de caja neto bastante bajos (negativos) que requieren dinero extra, estos tienen dos alternativas que el dinero sea del saldo en caja disponible y/o el dinero sea de un retiro solicitado hace dos días. Si se va a utilizar dinero de la cuenta del extranjero los retornos de este dinero deberían por lo menos pagar los costos de transferencia, y esto sólo se consigue realizando retiros después de k o más días de la última inversión.

Recomendaciones

En trabajos futuros podrían ser considerados los siguientes tópicos adicionales:

- Analizar el caso cuando las tasas de retorno son también parámetros estocásticos.
- Considerar diferentes formas de invertir el dinero en el extranjero como por ejemplo: Fondos a plazo fijo o intereses capitalizables.
- Considerar diferentes funciones para el costo o tasas por transferencia de dinero.

- Considerar los valores de saldo mínimo establecidos, están realizados en base a retiros acumulados semanalmente, lo cual hace que los saldos mínimo sean demasiados seguros respecto a los retiros diarios.

Apéndice A

DISTRIBUCIONES DE PROBABILIDAD PARA EL FLUJO DE CAJA NETO

Para resolver el programa estocástico de gestión de saldo en caja se requiere del conocimiento de la función de probabilidad del flujo de caja neto diario. No se conoce a ciencia cierta cuál es dicha función de probabilidad, sin embargo se tiene los datos históricos acerca de los depósitos y retiros diarios, proporcionados por el Banco Central. Mediante la utilización de métodos estadísticos se desea extraer toda la información posible contenida en los datos, acerca del patrón de comportamiento de la variable. En las siguientes secciones se realiza el ajuste de modelos para explicar la naturaleza del flujo de caja neto.

A.1. MODELOS UNIVARIANTES

Una serie temporal esta formada por las observaciones de una variable a lo largo de un tiempo en intervalos regulares de tiempo, ya sea días, semanas, meses, trimestres, años, etc. Los modelos de series temporales univariantes se dieron a conocer en 1970 con el trabajo de los estadísticos Box y Jenkins [15]. Estos modelos se basan únicamente en la información pasada de la misma variable para explicar su comportamiento. Las predicciones obtenidas con estos modelos univariantes se basan en la hipótesis de que las condiciones futuras serán análogas a las pasadas y son especialmente útiles en la previsión a corto plazo. Estos modelos consideran que la serie temporal estudiada ha sido generada por un proceso estocástico, que es una familia de variables aleatorias que corresponden a momentos sucesivos de tiempo. De esta forma, una serie temporal se puede ver como una realización de un proceso estocástico ([14], [13]).

El objetivo en esta sección es ajustar un modelo de series temporales para el flujo de caja neto diario en base a las observaciones de enero hasta diciembre del 2005.

El gráfico A.1 muestra que la serie de flujo neto parece oscilar alrededor de un valor constante cercano a cero sin alejarse de forma permanente. Esto sugiere

que la serie es estacionaria en media. Además, se observa pequeñas oscilaciones cada 5 días (flujos altos seguidos de flujos pequeños), lo que indica que podría existir estacionalidad de orden 5. Para realizar una mejor apreciación de la esta-

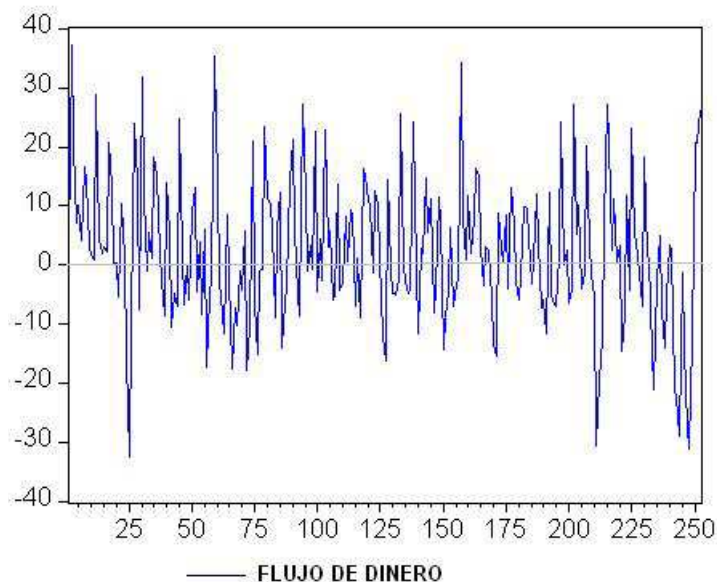


Figura A.1: Flujo de caja neto enero-diciembre del 2005

cionariedad y de la relación que la serie de flujo mantiene con sí mismo en la Figura A.2 se presentan las funciones de autocorrelación simple y parcial de la serie. Se observa que los coeficientes de la función de autocorrelación simple tienen

Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
1	0.257	0.257	16.799	0.000	
2	-0.036	-0.109	17.122	0.000	
3	-0.067	-0.031	18.264	0.000	
4	0.065	0.096	19.367	0.001	
5	0.277	0.247	39.265	0.000	
6	-0.046	-0.208	39.819	0.000	
7	-0.159	-0.070	46.395	0.000	
8	-0.097	-0.011	48.842	0.000	
9	0.052	0.046	49.560	0.000	
10	0.236	0.152	64.328	0.000	
11	-0.049	-0.110	64.958	0.000	
12	-0.123	-0.017	68.967	0.000	
13	-0.099	-0.058	71.568	0.000	
14	0.042	0.027	72.039	0.000	
15	0.172	0.063	80.034	0.000	

Figura A.2: Correlograma de la serie flujo de dinero

un decrecimiento rápido hacia cero, lo que indica que la serie es estacionaria en media. Además, se observa en la función de autocorrelación una alta correlación

(dependencia lineal) que existe con los rezagos 5, 10 y 15; esto junto con lo observado en el gráfico de la serie indica que existe la presencia de estacionalidad de orden 5. Como los rezagos de orden 5 comienzan a decaer rápidamente a partir del rezago 15 se concluye que la componente estacional de la serie es estacionaria.

Del examen visual de la serie se observa que no existe ningún comportamiento evolutivo de la varianza, al parecer se mantiene la dispersión de la serie. Para confirmar esta impresión visual se realiza un gráfico entre una medida de variabilidad, como la desviación estándar o el rango y una medida de nivel como la media local. Para hacer comparaciones homogéneas se toma para esta serie intervalos de observaciones semanales (5 días) y se calcula la media y la desviación estándar para cada intervalo. En la Figura A.3 se representa a cada par de media y desviación estándar.

Los puntos graficados no muestra ningún esquema claro, los puntos están aline-

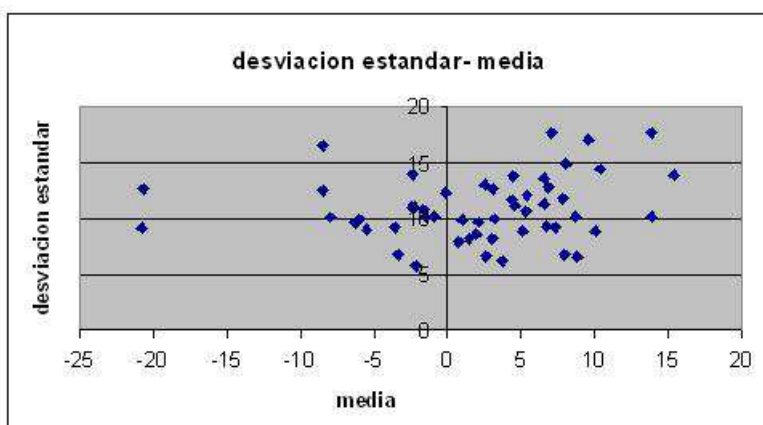


Figura A.3: Desviación estándar vs la media

ados entorna a una línea paralela al eje de las abscisas, por lo tanto, la serie es estacionaria en varianza.

En el correlograma de la Figura A.2 se observa que la función de autocorrelación parcial se anula a partir del retardo 11, los coeficientes de la función de autocorrelación parcial de los retardo 5 y 10 son altos, al igual que el coeficiente del retardo 1; esto sugiere que el valor de la serie en el día t depende de los valores del flujo tomados el día anterior y hace dos semana atrás. Así, se tiene dos posibles modelos: $SARIMA(1, 0, 0)(2, 0, 0)_5$ o $SARIMA(1, 0, 0)(1, 0, 1)$; estimando los coeficientes de estos modelos con la ayuda de un paquete estadístico, se obtienen los Cuadros de información A.1 y A.2.

En ambos modelos se observa que los coeficientes son estadísticamente significativos ($\text{Prob} < 0,05$). No hay problemas en las funciones de autocorrelación simple y autocorrelación parcial (no presentadas por espacio), los coeficientes de los residuos están dentro de las bandas y por el valor de la probabilidad del estadístico Q mayor que 0,05 se dice que los residuos no están correlacionados.

Los dos modelos anteriores son adecuados para la previsión de la serie, ya que los residuos no tienen una estructura de dependencia, es decir no contienen información para explicar el comportamiento del modelo y siguen un proceso de ruido blanco. Sin embargo se debe escoger el modelo que mejor explique la conducta de la serie observada. Para la selección se compara los criterios de ajuste

Dependent Variable: FLUJO				
Method: Least Squares				
Date: 04/05/07 Time: 00:54				
Sample (adjusted): 12 252				
Included observations: 241 after adjustments				
Convergence achieved after 7 iterations				
Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	0.330124	0.062971	5.242442	0.0000
SAR(5)	0.262100	0.064913	4.037733	0.0001
SAR(10)	0.255688	0.064964	3.935827	0.0001
R-squared	0.219506	Mean dependent var		1.761930
Adjusted R-squared	0.212947	S.D. dependent var		12.15909
S.E. of regression	10.78706	Akaike info criterion		7.606941
Sum squared resid	27693.82	Schwarz criterion		7.650320
Log likelihood	-913.6364	Durbin-Watson stat		1.938519

Cuadro A.1: Información $SARIMA(1, 0, 0)(2, 0, 0)_5$

del modelo que se dan en los respectivos cuadros de información. Estos son el criterio de máxima verosimilitud (likelihood), la suma de residuos al cuadrado (SRC), el criterio de Akaike (AIC) y el criterio de Schwarz (BIC). Se escoge el modelo en el cual los valores de los criterios sean los más pequeños, en especial el valor de BIC debe ser el mínimo de los modelos. En el Cuadro A.3 se resumen los criterios de los dos modelos. $SARIMA(1, 0, 0)(1, 0, 1)_5$ tiene los valores de los criterios más

Dependent Variable: FLUJO				
Method: Least Squares				
Date: 04/05/07 Time: 01:12				
Sample (adjusted): 7 252				
Included observations: 246 after adjustments				
Convergence achieved after 13 iterations				
Backcast: 26				
Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	0.325009	0.062546	5.196338	0.0000
SAR(5)	0.800960	0.071643	11.17994	0.0000
MA(5)	-0.538806	0.104029	-5.179387	0.0000
R-squared	0.233025	Mean dependent var		1.875720
Adjusted R-squared	0.226712	S.D. dependent var		12.09698
S.E. of regression	10.63769	Akaike info criterion		7.578804
Sum squared resid	27498.00	Schwarz criterion		7.621552
Log likelihood	-929.1929	Durbin-Watson stat		1.944177

Cuadro A.2: Información $SARIMA(1, 0, 0)(1, 0, 1)_5$

pequeños.

CRITERIO	<i>SARIMA</i> (1, 0, 0)(2, 0, 0) ₅	<i>SARIMA</i> (1, 0, 0)(1, 0, 1) ₅
Suma de residuos al cuadrado	27693.82	27498.00
Máxima Verosimilitud	-913.6364	-929.1929
Información de Akaike	7.606941	7.578804
Criterio de Schwarz	7.650320	7.621552

Cuadro A.3: Criterios para la selección del modelo

Se concluye que la serie temporal de flujo de dinero diario pudo haber sido generada por un proceso estocástico $SARIMA(1, 0, 0)(1, 0, 1)_5$. Este proceso se especifica como:

$$\begin{aligned}
 \Phi_1(B^5)\phi_1(B)\nabla_5^0\nabla^0Y_t &= \theta_0(B)\Theta_1(B^5)\mu_t, \\
 (1 - \Phi_1B^5)(1 - \phi_1B)Y_t &= (1 - \Theta_1B^5)\mu_t, \\
 (1 - \phi_1B - \Phi_1B^5 + \Phi_1\phi_1B^6)Y_t &= (1 - \Theta_1B^5)\mu_t, \\
 Y_t - \phi_1Y_{t-1} - \Phi_1Y_{t-5} + \Phi_1\phi_1Y_{t-6} &= \mu_t - \Theta_1\mu_{t-5} \quad (A.1)
 \end{aligned}$$

A partir (A.1) se obtiene la serie de interés Y_t , sustituyendo los valores de los coeficientes estimados se obtiene el modelo $SARIMA(1, 0, 0)(1, 0, 1)_5$ para la serie. A este modelo lo llamaremos *Sarima* y se especifica como:

$$\begin{aligned}
 Y_t &= \phi_1Y_{t-1} + \Phi_1Y_{t-5} - \Phi_1\phi_1Y_{t-6} - \Theta_1\mu_{t-5} + \mu_t, \\
 Y_t &= 0,325009Y_{t-1} + 0,800960Y_{t-5} - 0,260319Y_{t-6} - 0,538806\mu_{t-5} + \mu_t. \quad (A.2)
 \end{aligned}$$

El modelo detallado (A.2) indica que el valor que tome la variable de flujo de dinero en el día t depende de los valores tomados por esta el día anterior, el valor de la semana pasada y el valor de la variable 6 días atrás. Además, esta serie depende del valor del ruido tomado la semana anterior y de el ruido para el día t .

Este modelo sirve para la predicción a corto plazo del valor medio de la serie. Estableciendo intervalos de confianza a una confiabilidad α para la predicción de la serie al tiempo $T + l$ utilizando la información disponible hasta el día T , Y_{T+l} de la siguiente manera.

$$Prob[\tilde{Y}_{T+l/T} - Z_{\frac{\alpha}{2}}\sqrt{EMC[\tilde{Y}_{T+l/T}]} < Y_{T+l} < \tilde{Y}_{T+l/T} + Z_{\frac{\alpha}{2}}\sqrt{EMC[\tilde{Y}_{T+l/T}]}] = \alpha$$

donde $\tilde{Y}_{T+l/T}$ es el predictor para el período $T+l$. Como el predictor se construye como una función lineal de los valores μ_t conocidos. Se especifica Y_t como:

$$Y_t = \mu_t + \psi_1\mu_{t-1} + \psi_2\mu_{t-2} + \dots + \psi_k\mu_{t-k} + \dots$$

Los coeficientes de esta combinación lineal se obtienen de la relación (A.3).

$$\begin{aligned} (1 - \phi_1 B - \Phi_1 B^5 + \Phi_1 \phi_1 B^6) \psi(B) &= (1 - \Theta_1 B^5), \\ (1 - \phi_1 B - \Phi_1 B^5 + \Phi_1 \phi_1 B^6)(1 + \psi_1 B^1 + \psi_2 B^2 + \dots) &= (1 - \Theta_1 B^5) \end{aligned} \quad (\text{A.3})$$

Operando e igualando los factores de las partes de la ecuación valor de los coeficientes ψ 's:

$$\begin{aligned} \psi_k &= \phi_1 \psi_{k-1}, \text{ para } k = 1, 2, 3, 4, \\ &= \text{con } \psi_0 = 1, \\ \psi_k &= \phi_1 \psi_{k-1} + \Phi_1 + \Theta_1, \text{ para } k = 5, \\ \psi_k &= \phi_1 \psi_{k-1} + \Phi_1 \psi_{k-5} - \phi_1 \Phi_1 \psi_{k-6} \text{ para } k = 6, 7, \dots \end{aligned}$$

Así, el predictor para el día t es igual al valor Y_t de la serie menos el ruido para el este día t , es decir será

$$\tilde{Y}_{T+l/T} = \psi_1 \mu_{T+l-1} + \psi_2 \mu_{T+l-2} + \dots + \psi_k \mu_{T+l-k} + \dots$$

$EMC[\tilde{Y}_{T+l/T}]$ es el error en media cuadrática del predictor. Se define como la varianza del error de predicción, $EMC[\tilde{Y}_{T+l/T}] = E[Y_{T+l} - \tilde{Y}_{T+l/T}]^2$. El EMC del predictor para la serie de flujo de caja es

$$\begin{aligned} EMC[\tilde{Y}_{T+1/T}] &= E[\mu_{T+1}]^2 \\ EMC[\tilde{Y}_{T+k/T}] &= (1 + \sum_1^{k-1} \psi_k^2) \sigma, \quad k = 2, 3, 4, \dots \end{aligned}$$

Una vez seleccionado el modelo que será utilizado en la predicción de la media se busca un modelo para explicar la variabilidad de la serie. Se ajusta un modelo $SARIMA(1, 0, 0)(1, 0, 1)_5 - GARCH(0, 1)$ para explicar esta variabilidad. El Cuadro de información A.4 indica que los coeficientes de este modelo son estadísticamente significativos. Los correlogramas de los residuos y residuos al cuadrado de la Figura A.4 no muestran problemas en la adecuación de los residuos. Los ruidos se asemejan a una ley normal de media 0,019 y desviación estándar 1,000.

Correlogram of Standardized Residuals						Correlogram of Standardized Residuals Squared					
Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob	Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
		1 -0.008	-0.008	0.0179				1 -0.010	-0.010	0.0257	
		2 -0.024	-0.024	0.1647				2 -0.028	-0.028	0.2238	
		3 -0.048	-0.048	0.7336				3 0.081	0.080	1.8706	
		4 0.001	-0.001	0.7337	0.392			4 -0.008	-0.007	1.8871	0.170
		5 -0.012	-0.014	0.7693	0.681			5 0.099	0.104	4.3528	0.113
		6 -0.064	-0.066	1.8011	0.615			6 -0.020	-0.026	4.4578	0.216
		7 -0.069	-0.072	3.0250	0.554			7 0.010	0.017	4.4819	0.345
		8 -0.016	-0.023	3.0942	0.685			8 -0.060	-0.080	5.4142	0.367
		9 0.017	0.006	3.1687	0.787			9 -0.021	-0.016	5.5319	0.478
		10 0.043	0.035	3.6399	0.820			10 -0.028	-0.048	5.7362	0.571
		11 -0.044	-0.047	4.1374	0.845			11 -0.068	-0.054	6.9447	0.543
		12 0.003	-0.002	4.1394	0.902			12 0.005	0.000	6.9518	0.642
		13 -0.072	-0.081	5.4937	0.856			13 0.002	0.019	6.9526	0.730
		14 0.012	-0.001	5.5346	0.903			14 0.014	0.026	7.0039	0.799
		15 0.020	0.016	5.6442	0.933			15 0.041	0.052	7.4573	0.826
		16 -0.036	-0.039	5.9945	0.946			16 -0.087	-0.081	9.4470	0.738
		17 -0.029	-0.030	6.2203	0.961			17 -0.004	-0.009	9.4504	0.801

Figura A.4: Modelo $SARIMA(1, 0, 0)(1, 0, 1)_5 - GARCH(0, 1)$

El modelo será:

$$Y_t = 0,323931Y_{t-1} + 0,804255Y_{t-5} - 0,260521Y_{t-6} - 0,543514\mu_{t-5} + \mu_t.$$

$$\mu_t = h_t\xi_t$$

$$h_t^2 = 208,0037 - 0,858935h_{t-1}^2$$

(A.4)

donde ξ es un ruido blanco, Normal de media 0 y varianza 1.

En la Figura A.5 se puede observar la serie ajustada, la serie real para 200 observaciones del año 2005 y los límites de confianza al 95 %. Se observa que la serie estimada se ajusta bastante bien a la serie real de flujo de caja neto.

A.2. MODELOS DE INTERVENCIÓN

Las series reales se ven con frecuencia afectadas por sucesos puntuales conocidos. Si se incluye estos efectos en la serie se puede mejorar la precisión de la estimación de los parámetros y de las previsiones. Un tipo de suceso conocido es la estacionalidad. Cuando la estacionalidad es debida a un efecto constante y sistemático, como sucede por ejemplo en algunas series climatológicas, produce un efecto determinista de período s . Por ejemplo, una estacionalidad mensual, $s = 12$, puede modelarse con $s - 1$ variables impulso, que tomen el valor uno en un mes y cero en el resto.

Dependent Variable: FLUJO				
Method: ML - ARCH (Marquardt)				
Date: 04/05/07 Time: 16:49				
Sample (adjusted): 7 252				
Included observations: 246 after adjustments				
Convergence achieved after 14 iterations				
Backcast: 26, Variance backcast: ON				
GARCH = C(4) + C(5)GARCH(-1)				
Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	0.323931	5.145209	0.062958	0.0000
SAR(5)	0.804255	0.071095	11.31241	0.0000
MA(5)	-0.543514	0.106011	-5.126956	0.0000
Variance Equation				
C	208.0037	39.87701	5.216132	0.0000
GARCH(-1)	-0.858935	0.315288	-2.724287	0.0064
R-squared	0.233014	Mean dependent var		1.875720
Adjusted R-squared	0.226712	S.D. dependent var		12.09698
S.E. of regression	10.68182	Akaike info criterion		7.593783
Sum squared resid	27498.38	Schwarz criterion		7.665030
Log likelihood	-929.0353	Durbin-Watson stat		1.942650

Cuadro A.4: Información $SARIMA(1, 0, 0)(1, 0, 1)_5 - GARCH(0, 1)$

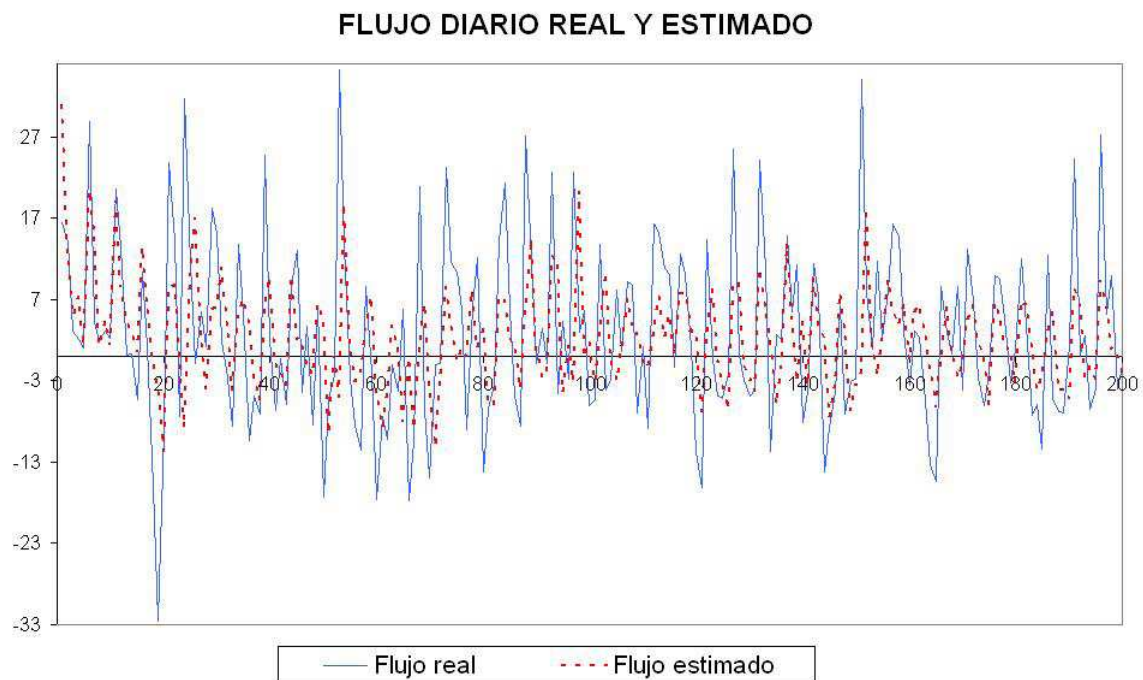


Figura A.5: Flujo real versus el flujo estimado por el modelo

Box y Tiao [38] denominaron análisis de intervención a la inclusión en un modelo de series temporales de sucesos específicos que producen efectos deterministas mediante variables ficticias.

Las variables ficticias más utilizadas para representar sucesos cualitativos que afectan a la serie son de dos tipos: variables impulso y variables escalón. Las variables impulso representan sucesos que ocurren únicamente en un instante. Las variables escalón representan acontecimientos que comienzan en un instante conocido y se mantienen a partir de ese instante.

Uno mejor estudio de la serie de flujo de caja neto es mediante el análisis de intervención.

Introducimos dentro del modelo Sarima variables explicativas para los días laborales. Este segundo enfoque tiene la ventaja que ayuda a investigar el efecto de los días de la semana; ya que la actividad es distinta, por ejemplo, el flujo neto en los días lunes es inferior al del día miércoles. Además, para estudiar si el comportamiento de la serie cambia en los diferentes meses del año, se incluye variables explicativas para los meses, ya que la serie podría estar afectada por festividades que ocurren de manera irregular.

Se ajustó un primer modelo introduciendo al modelo Sarima cinco variables explicativas para los días laborables y 11 variables explicativas para los meses del año. En este modelo resultó que las variables explicativas para los días de la semana laborable y para el mes de diciembre son estadísticamente significativas mientras que las variables explicativas para el resto de meses no son significativas¹. De esta forma, se ajusta un modelo Sarima solo con las variables significativas, el cual no presenta problemas en los correlogramas y sus coeficientes son significativos. A este modelo lo llamaremos *Sarima-Impulso* y puede formularse como:

$$\begin{aligned} Z_t &= \phi_1 Z_{t-1} + \Phi_1 Z_{t-5} - \phi_1 \Phi_1 Z_{t-6} - \theta_1 \mu_{t-1} - \theta_5 \mu_{t-5} + \mu_t \\ Z_t &= Y_t - I_t \\ I_t &= a_1 Il_t + a_2 Im_t + a_3 Imi_t + a_4 Ij_t + a_5 Iv_t + a_6 Id_t \end{aligned}$$

donde

Z : variable de flujo más el impulso diario

I : impulso total en el tiempo

Il : variable indicatriz de impulso para los días lunes. Esto es $Il = 1$ si es lunes y $Il = 0$ caso contrario.

Im : variable indicatriz de impulso para los días martes

Imi : variable indicatriz de impulso para los días miércoles

Ij : variable indicatriz de impulso para los días jueves

Iv : variable indicatriz de impulso para los días viernes y

Id : variable indicatriz de impulso para los meses de diciembre

$a_i, i = 1, 2, \dots, 6$ coeficientes de intervención

Las variable de impulso son variable de valor 0 o 1, 1 si es el día o mes de interés o 0 caso contrario. De esta manera el modelo es de la forma:

$$Y_t = [\phi_1 Z_{t-1} + \Phi_1 Z_{t-5} - \phi_1 \Phi_1 Z_{t-6} - \theta_1 \mu_{t-1} - \theta_5 \mu_{t-5} + \mu_t] + I_t$$

o

$$\begin{aligned} Y_t &= [\phi_1 Y_{t-1} + \Phi_1 Y_{t-5} - \phi_1 \Phi_1 Y_{t-6}] + [I_t - \phi_1 I_{t-1} - \Phi_1 Y_{t-5} + \phi_1 \Phi_1 Y_{t-6}] \\ &\quad - \theta_1 \mu_{t-1} - \theta_5 \mu_{t-5} + \mu_t] + I_t \end{aligned}$$

El modelo Sarima-impulso con los coeficientes estimados por el paquete Eviews es:

$$\begin{aligned} Y_t &= 0,686Z_{t-1} + 0,252Z_{t-5} - 0,173Z_{t-6} - 0,399\mu_{t-1} - 0,287\mu_{t-5} + \mu_t] + I_t \\ I_t &= -3,240Il_t + 14,734Im_t + 4,895Imi_t - 2,508Ij_t - 2,151Iv_t - 6,359Id_t \end{aligned}$$

¹por motivos de espacio de la tesis no se detalla minuciosamente el proceso de selección de los modelos

Para la varianza se utiliza el modelo GARCH(1,1) esto es:

$$\begin{aligned}\mu_t &= h_t \varepsilon_t \\ h_t^2 &= \alpha_0 + \alpha_1 \mu_{t-1}^2 + \beta_1 h_{t-1}^2\end{aligned}$$

donde:

ε es un ruido blanco, es decir, una variable normal de media cero y varianza 1 $N(0, 1)$.

α_0 , α_1 , β_1 son los coeficientes de estimación.

La ecuación con coeficientes estimados es:

$$\begin{aligned}\mu_t &= h_t \varepsilon_t \\ h_t^2 &= 11,675 + 0,264 \mu_{t-1}^2 + 0,568 h_{t-1}^2\end{aligned}$$

En la Figura A.6 se expone la predicción del flujo con sus intervalos de confianza obtenidos con Sarima-Impulso para 10 días.

Se puede observar que el flujo real se encuentra entre los límites de predicción y

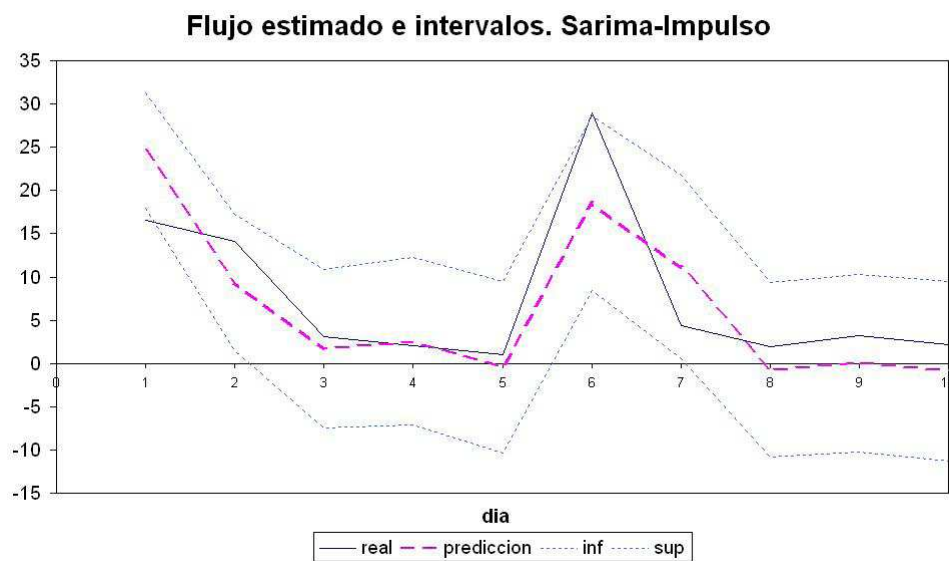


Figura A.6: predicción diaria realizada con Sarima-Impulso.

la serie estimada tiene un comportamiento bastante similar al flujo real.

También se ajustó un modelo que solo depende del retardo anterior y de las variables de impulso para diciembre, lunes, martes y jueves. Los coeficientes de las variables de este modelo son estadísticamente significativos y no se presentan problemas en la adecuación de los residuos. Para la varianza se emplea un

modelo GARCH(1,1) de igual manera que el modelo anterior. A este modelo lo llamaremos *Impulso* y se especifica de la siguiente forma:

$$Y_t = \gamma Y_{t-1} + I_t + \mu_t$$

$$I_t = a_1 I_{t-1} + a_2 I_{m,t} + a_4 I_{j,t} + a_6 I_{d,t}$$

Reemplazando los valores de los coeficientes estimados por el paquete se tiene:

$$Y_t = 0,380Y_{t-1} + I_t + \mu_t$$

$$I_t = -2,855I_{t-1} + 17,179I_{m,t} - 4,195I_{j,t} - 5,806I_{d,t}$$

Para la varianza :

$$\mu_t = h_t \varepsilon_t$$

$$h_t^2 = 14,817 + 0,291\mu_{t-1}^2 + 0,510h_{t-1}^2$$

Por fines de comparación también se trabajo con una distribución Normal de media 1,9 y desviación estándar 11,0.

A.3. SALDO MÍNIMO

Se busca una relación entre el flujo de caja y el saldo mínimo. Se dispone de datos del saldo mínimo desde septiembre del 2005 hasta agosto del 2006. El saldo mínimo se ajusta cada semana, por esto se busca una relación con el flujo promedio calculado cada cinco días y el saldo mínimo semanal.

Se ajusta un modelo de regresión lineal con estas dos variables y se obtiene el Cuadro A.5 con la información del ajuste. Por el valor de la $Prob < 0,05$ se concluye que tanto el intercepto como el flujo promedio son estadísticamente significativos, además el valor de la probabilidad ($< 0,05$) del estadístico F indica que al menos una de variable es significativa. El valor del $R^2 = 0,64$ indica el porcentaje de variabilidad explicada por el ajuste.

En la Figura A.7 indica los puntos y la recta ajustada. La relación que se utilizará para el saldo mínimo será:

$$L = 75,615 - 3,3307Y$$

donde L es el saldo mínimo y Y el flujo promedio. De esta forma, el valor del saldo mínimo crece conforme el flujo promedio disminuye, es decir cuando los requerimientos del sistema sean mayores el saldo mínimo crecerá.

Dependent Variable: SALDO_MINIMO				
Method: Least Squares				
Date: 08/23/07 Time: 19:41				
Sample: 1 38				
Included observations: 38				
Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	75.61529	2.764748	27.34980	0.0000
FLUJO_PROMEDIO	-3.330744	0.417411	-7.979541	0.0000
R-squared	0.638819	Mean dependent var	70.65094	
Adjusted R-squared	0.628786	S.D. dependent var	27.25535	
S.E. of regression	16.60595	Akaike info criterion	8.508595	
Sum squared resid	9927.271	Schwarz criterion	8.594784	
Log likelihood	-159.6633	F-statistic	63.67307	
Durbin-Watson stat	1.738124	Prob(F-statistic)	0.000000	

Cuadro A.5: Información ajuste del saldo mínimo y el flujo promedio cada 5 días.

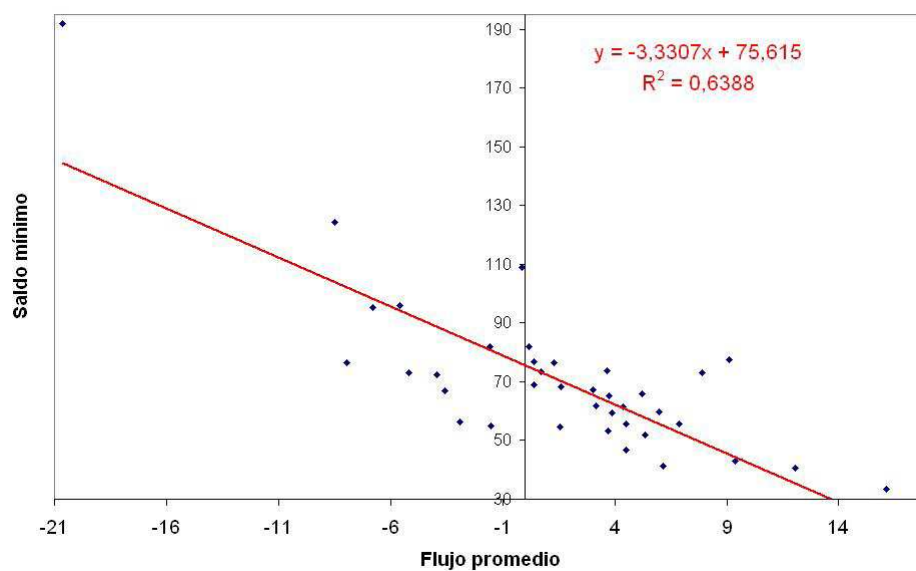


Figura A.7: predicción diaria realizada con Sarima-Impulso.

Apéndice B

SOLUCIÓN DEL PGSC

En este apéndice se formula el modelo extenso de programación estocástico para el PGSC del ejemplo del Capítulo 2. Además, se detalla las herramientas utilizadas para resolver los problemas estocástico para el PGSC y bajo información perfecta (FCM).

B.1. MODELO EXTENSO DEL PEGSC PARA EL EJEMPLO

La formulación del modelo extenso para el PGSC con 16 escenarios y bajo los supuestos descritos en la Sección 2.4 es de la siguiente forma:

$$\begin{aligned} \text{máx} \quad & \sum_{y_1=1}^2 \sum_{y_2=3}^6 \sum_{y_3=7}^{14} \sum_{y_4=15}^{30} \sum_{y_5=31}^{46} p(y_1, y_2, y_3, y_4, y_5) \{ rE_1 - c[I_1 + M_1] + \\ & rE_2(y_1) - c[I_2(y_1) + M_2(y_1)] + \\ & rE_3(y_1, y_2) - c[I_3(y_1, y_2) + M_3(y_1, y_2)] + \\ & rE_4(y_1, y_2, y_3) - c[I_4(y_1, y_2, y_3) + M_4(y_1, y_2, y_3)] + \\ & rE_5(y_1, y_2, y_3, y_4) - c[I_5(y_1, y_2, y_3, y_4) + M_5(y_1, y_2, y_3, y_4)] + \\ & rE_6(y_1, y_2, y_3, y_4, y_5) - c[I_6(y_1, y_2, y_3, y_4, y_5) + M_6(y_1, y_2, y_3, y_4, y_5)] \} \end{aligned}$$

s.r

$$C_1 + I_1 = E_0 + y_1 = 80 + 2,12 = 82,12$$

$$E_1 + M_1 = E_0 = 250$$

$$C_2(1) - C_1 + I_2(1) = y_2(1)$$

$$E_2(1) - E_1 + M_2(1) = 0$$

$$C_2(2) - C_1 + I_2(2) = y_2(2)$$

$$E_2(2) - E_1 + M_2(2) = 0$$

$$\begin{aligned}
& C_3(1, 3) - C_2(1) + I_3(1, 3) - M_1 = y_3(1, 3) \\
& E_3(1, 3) - E_2(1) - I_1 + M_3(1, 3) = 0 \\
& C_3(1, 4) - C_2(1) + I_3(1, 4) - M_1 = y_3(1, 4) \\
& E_3(1, 4) - E_2(1) - I_1 + M_3(1, 4) = 0 \\
& C_3(2, 5) - C_2(2) + I_3(2, 5) - M_1 = y_3(2, 5) \\
& E_3(2, 5) - E_2(2) - I_1 + M_3(2, 5) = 0 \\
& C_3(2, 6) - C_2(2) + I_3(2, 6) - M_1 = y_3(2, 6) \\
& E_3(2, 6) - E_2(2) - I_1 + M_3(2, 6) = 0 \\
& C_4(1, 3, 7) - C_3(1, 3) + I_4(1, 3, 7) - M_2(1) = y_4(1, 3, 7) \\
& E_4(1, 3, 7) - E_3(1, 3) - I_2(1) + M_4(1, 3, 7) = 0 \\
& C_4(1, 3, 8) - C_3(1, 3) + I_4(1, 3, 8) - M_2(1) = y_4(1, 3, 8) \\
& E_4(1, 3, 8) - E_3(1, 3) - I_2(1) + M_4(1, 3, 8) = 0 \\
& C_4(1, 4, 9) - C_3(1, 4) + I_4(1, 4, 9) - M_2(1) = y_4(1, 4, 9) \\
& E_4(1, 4, 9) - E_3(1, 4) - I_2(1) + M_4(1, 4, 9) = 0 \\
& C_4(1, 4, 10) - C_3(1, 4) + I_4(1, 4, 10) - M_2(1) = y_4(1, 4, 10) \\
& E_4(1, 4, 10) - E_3(1, 4) - I_2(1) + M_4(1, 4, 10) = 0 \\
& C_4(2, 5, 11) - C_3(2, 5) + I_4(2, 5, 11) - M_2(2) = y_4(1, 4, 11) \\
& E_4(2, 5, 11) - E_3(2, 5) - I_2(2) + M_4(2, 5, 11) = 0 \\
& C_4(2, 5, 12) - C_3(2, 5) + I_4(2, 5, 12) - M_2(2) = y_4(2, 5, 12) \\
& E_4(2, 5, 12) - E_3(2, 5) - I_2(2) + M_4(2, 5, 12) = 0 \\
& C_4(2, 6, 13) - C_3(2, 6) + I_4(2, 6, 13) - M_2(2) = y_4(2, 5, 13) \\
& E_4(2, 6, 13) - E_3(2, 6) - I_2(2) + M_4(2, 6, 13) = 0 \\
& C_4(2, 6, 14) - C_3(2, 6) + I_4(2, 6, 14) - M_2(2) = y_4(2, 5, 14) \\
& E_4(2, 6, 14) - E_3(2, 6) - I_2(2) + M_4(2, 6, 14) = 0 \\
& C_5(1, 3, 7, 15) - C_4(1, 3, 7) - M_3(1, 3) = y_5(1, 3, 7, 15) \\
& E_5(1, 3, 7, 15) - E_4(1, 3, 7) - I_3(1, 3) = 0 \\
& C_5(1, 3, 7, 16) - C_4(1, 3, 7) - M_3(1, 3) = y_5(1, 3, 7, 16) \\
& E_5(1, 3, 7, 16) - E_4(1, 3, 7) - I_3(1, 3) = 0 \\
& C_5(1, 3, 8, 17) - C_4(1, 3, 8) - M_3(1, 3) = y_5(1, 3, 8, 17)
\end{aligned}$$

$$\begin{aligned}
& E_5(1, 3, 8, 17) - E_4(1, 3, 8) - I_3(1, 3) = 0 \\
& C_5(1, 3, 8, 18) - C_4(1, 3, 8) - M_3(1, 3) = y_5(1, 3, 8, 18) \\
& E_5(1, 3, 8, 18) - E_4(1, 3, 8) - I_3(1, 3) = 0 \\
& C_5(1, 4, 9, 19) - C_4(1, 4, 9) - M_3(1, 4) = y_5(1, 4, 9, 19) \\
& E_5(1, 4, 9, 19) - E_4(1, 4, 9) - I_3(1, 4) = 0 \\
& C_5(1, 4, 9, 20) - C_4(1, 4, 9) - M_3(1, 4) = y_5(1, 4, 9, 20) \\
& E_5(1, 4, 9, 20) - E_4(1, 4, 9) - I_3(1, 4) = 0 \\
& C_5(1, 4, 10, 21) - C_4(1, 4, 10) - M_3(1, 4) = y_5(1, 4, 10, 21) \\
& E_5(1, 4, 10, 21) - E_4(1, 4, 10) - I_3(1, 4) = 0 \\
& C_5(1, 4, 10, 22) - C_4(1, 4, 10) - M_3(1, 4) = y_5(1, 4, 10, 22) \\
& E_5(1, 4, 10, 22) - E_4(1, 4, 10) - I_3(1, 4) = 0 \\
& C_5(2, 5, 11, 23) - C_4(2, 5, 11) - M_3(2, 5) = y_5(2, 5, 11, 23) \\
& E_5(2, 5, 11, 23) - E_4(2, 5, 11) - I_3(2, 5) = 0 \\
& C_5(2, 5, 11, 24) - C_4(2, 5, 11) - M_3(2, 5) = y_5(2, 5, 11, 24) \\
& E_5(2, 5, 11, 24) - E_4(2, 5, 11) - I_3(2, 5) = 0 \\
& C_5(2, 5, 12, 25) - C_4(2, 5, 12) - M_3(2, 5) = y_5(2, 5, 12, 25) \\
& E_5(2, 5, 12, 25) - E_4(2, 5, 12) - I_3(2, 5) = 0 \\
& C_5(2, 5, 12, 26) - C_4(2, 5, 12) - M_3(2, 5) = y_5(2, 5, 12, 26) \\
& E_5(2, 5, 12, 26) - E_4(2, 5, 12) - I_3(2, 5) = 0 \\
& C_5(2, 6, 13, 27) - C_4(2, 6, 13) - M_3(2, 6) = y_5(2, 6, 13, 27) \\
& E_5(2, 6, 13, 27) - E_4(2, 6, 13) - I_3(2, 6) = 0 \\
& C_5(2, 6, 13, 28) - C_4(2, 6, 13) - M_3(2, 6) = y_5(2, 6, 13, 28) \\
& E_5(2, 6, 13, 28) - E_4(2, 6, 13) - I_3(2, 6) = 0 \\
& C_5(2, 6, 14, 29) - C_4(2, 6, 14) - M_3(2, 6) = y_5(2, 6, 14, 29) \\
& E_5(2, 6, 14, 29) - E_4(2, 6, 14) - I_3(2, 6) = 0 \\
& C_5(2, 6, 14, 30) - C_4(2, 6, 14) - M_3(2, 6) = y_5(2, 6, 14, 30) \\
& E_5(2, 6, 14, 30) - E_4(2, 6, 14) - I_3(2, 6) = 0 \\
& C_6(1, 3, 7, 15, 31) - C_5(1, 3, 7, 15) - M_4(1, 3, 7) = y_6(1, 3, 7, 15, 31) \\
& E_6(1, 3, 7, 15, 31) - E_5(1, 3, 7, 15) - I_4(1, 3, 7) = 0 \\
& C_6(1, 3, 7, 16, 32) - C_5(1, 3, 7, 16) - M_4(1, 3, 7) = y_6(1, 3, 7, 15, 32) \\
& E_6(1, 3, 7, 16, 32) - E_5(1, 3, 7, 16) - I_4(1, 3, 7) = 0 \\
& C_6(1, 3, 8, 17, 33) - C_5(1, 3, 8, 17) - M_4(1, 3, 8) = y_6(1, 3, 8, 17, 33)
\end{aligned}$$

$$\begin{aligned}
& E_6(1, 3, 8, 17, 33) - E_5(1, 3, 8, 17) - I_4(1, 3, 8) = 0 \\
& C_6(1, 3, 8, 18, 34) - C_5(1, 3, 8, 18) - M_4(1, 3, 8) = y_6(1, 3, 8, 18, 34) \\
& E_6(1, 3, 8, 18, 34) - E_5(1, 3, 8, 18) - I_4(1, 3, 8) = 0 \\
& C_6(1, 4, 9, 19, 35) - C_5(1, 4, 9, 19) - M_4(1, 4, 9) = y_6(1, 4, 9, 19, 35) \\
& E_6(1, 4, 9, 19, 35) - E_5(1, 4, 9, 19) - I_4(1, 4, 9) = 0 \\
& C_6(1, 4, 9, 20, 36) - C_5(1, 4, 9, 20) - M_4(1, 4, 9) = y_6(1, 4, 9, 20, 36) \\
& E_6(1, 4, 9, 20, 36) - E_5(1, 4, 9, 20) - I_4(1, 4, 9) = 0 \\
& C_6(1, 4, 10, 21, 37) - C_5(1, 4, 10, 21) - M_4(1, 4, 10) = y_6(1, 4, 10, 21, 37) \\
& E_6(1, 4, 10, 21, 37) - E_5(1, 4, 10, 21) - I_4(1, 4, 10) = 0 \\
& C_6(1, 4, 10, 22, 38) - C_5(1, 4, 10, 22) - M_4(1, 4, 10) = y_6(1, 4, 10, 22, 38) \\
& E_6(1, 4, 10, 22, 38) - E_5(1, 4, 10, 22) - I_4(1, 4, 10) = 0 \\
& C_6(2, 5, 11, 23, 39) - C_5(2, 5, 11, 23) - M_4(2, 5, 11) = y_6(2, 5, 11, 23, 39) \\
& E_6(2, 5, 11, 23, 39) - E_5(2, 5, 11, 23) - I_4(2, 5, 11) = 0 \\
& C_6(2, 5, 11, 24, 40) - C_5(2, 5, 11, 24) - M_4(2, 5, 11) = y_6(2, 5, 11, 24, 40) \\
& E_6(2, 5, 11, 24, 40) - E_5(2, 5, 11, 24) - I_4(2, 5, 11) = 0 \\
& C_6(2, 5, 12, 25, 41) - C_5(2, 5, 12, 25) - M_4(2, 5, 12) = y_6(2, 5, 12, 25, 41) \\
& E_6(2, 5, 12, 25, 41) - E_5(2, 5, 12, 25) - I_4(2, 5, 12) = 0 \\
& C_6(2, 5, 12, 26, 42) - C_5(2, 5, 12, 26) - M_4(2, 5, 12) = y_6(2, 5, 12, 26, 42) \\
& E_6(2, 5, 12, 26, 42) - E_5(2, 5, 12, 26) - I_4(2, 5, 12) = 0 \\
& C_6(2, 6, 13, 27, 43) - C_5(2, 6, 13, 27) - M_4(2, 6, 13) = y_6(2, 6, 13, 27, 43) \\
& E_6(2, 6, 13, 27, 43) - E_5(2, 6, 13, 27) - I_4(2, 6, 13) = 0 \\
& C_6(2, 6, 13, 28, 44) - C_5(2, 6, 13, 28) - M_4(2, 6, 13) = y_6(2, 6, 13, 28, 44) \\
& E_6(2, 6, 13, 28, 44) - E_5(2, 6, 13, 28) - I_4(2, 6, 13) = 0 \\
& C_6(2, 6, 14, 29, 45) - C_5(2, 6, 14, 29) - M_4(2, 6, 14) = y_6(2, 6, 14, 29, 45) \\
& E_6(2, 6, 14, 29, 45) - E_5(2, 6, 14, 29) - I_4(2, 6, 14) = 0 \\
& C_6(2, 6, 14, 30, 46) - C_5(2, 6, 14, 30) - M_4(2, 6, 14) = y_6(2, 6, 14, 30, 46) \\
& E_6(2, 6, 14, 30, 46) - E_5(2, 6, 14, 30) - I_4(2, 6, 14) = 0 \\
& I_1, M_1, I_2(y_1), M_2(y_1), I_3(y_1, y_2), M_3(y_1, y_2), I_4(y_1, y_2, y_3), M_4(y_1, y_2, y_3), \\
& I_5(y_1, y_2, y_3, y_4), M_5(y_1, y_2, y_3, y_4), \\
& I_6(y_1, y_2, y_3, y_4, s_5), M_6(y_1, y_2, y_3, y_4, y_5) \geq 0 \quad \forall y_1, y_2, y_3, y_4, y_5, \\
& C_1, C_2(y_1), C_3(y_1, y_2), C_4(y_1, y_2, y_3), C_5(y_1, y_2, y_3, y_4) \geq 33,472, \\
& C_6(y_1, y_2, y_3, y_4, y_5) \geq 40,739. \quad \forall y_1, y_2, y_3, y_4, y_5, \tag{B.1}
\end{aligned}$$

Para resolver problemas de programación estocástica como el formulado existen algunos métodos. Los métodos más utilizados están basados en la *outer linearization*, esta técnica de planos cortantes se conoce como *method L-Shaped*. Algunas variantes y extensiones del método de L-Shaped han sido diseñadas, por ejemplo, añadiendo términos de regularizaciones no lineales [1]. Además ya existen un conjunto de librerías llamadas SMI [41] las cuales permiten resolver problemas de programación estocástica con recurso y con restricciones probabilísticas. Estas librerías aceptan los problemas escritos en el formato SMPS. En la siguiente sección se detalla como describir un problema en este formato.

B.2. FORMATO SMPS

El formato SMPS [20] es usado para describir programas lineales estocásticos. El SMPS ayuda a convertir los programas lineales deterministas a programas lineales estocásticos añadiendo información acerca de la estructura de las variables dinámicas y estocásticas. Este formato requiere de tres archivos: archivo principal (*core file*), archivo de tiempo (*time file*) y archivo estocástico (*stoch file*). Cada archivo contiene registros de cabecera y de datos.

El *registro de cabecera* contiene tres campos para letras en la siguiente posición:

Primer campo	columnas 1-14
Segundo campo	columnas 15-24
Tercer campo	columnas 40-49

El *registro de datos* contiene tres campos para letras, dos campos para números y campo de código, que deben ser escritos en la siguiente posición:

Campo de código	columnas 2 y 3
Primer campo de letras	columnas 5-12
Segundo campo de letras	columnas 15-22
Primer campo numérico	columnas 25-36
Tercer campo de letras	columnas 40-47
Segundo campo numérico	columnas 50-61

Los campos de nombre o letras aceptan hasta ocho caracteres y admiten los símbolos ASCII. Los campos numéricos admiten hasta veinte caracteres incluido dígitos, signos, exponentes y puntos decimales.

Archivo Principal

En el archivo principal se detalla toda la información determinista del problema. En este archivo se incluye el nombre y tipo de cada restricción, los nombres de las filas y columnas. Este archivo también provee las variables que son reemplazados por otras variables (variables que cambian en el tiempo) para de esta forma crear todos los elementos estocásticos. Estos elementos deben ser mencionados en este archivo y se debe asignarles un valor preliminar que puede o no ser significativo. Este archivo tiene las siguientes secciones:

- **NAME.** Se da el nombre del problema en el segundo campo de letras. Este nombre es verificado en los demás archivos. Sólo la sección NAME pertenece al registro de cabecera las demás secciones pertenecen al registro de datos.
- **ROWS.** Se da el nombre y tipo de cada restricción y función objetivo. El tipo de cada fila se escribe en el campo de código utilizando las letras:
 L para las restricciones de la forma $a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b$
 G para las restricciones de la forma $a_1x_1 + a_2x_2 + \dots + a_nx_n \geq b$
 E para las restricciones de igualdad $a_1x_1 + a_2x_2 + \dots + a_nx_n = b$
 N indica una fila libre(sin restricciones) o para la función objetivo. Por defecto la primera fila de clase 'N' es tomada como la función objetivo.
 El nombre de cada fila se da en el segundo campo de letras.
- **COLUMNNS.** En el primer campo de letras se escribe el nombre de la columna, en el segundo campo de letras se debe enlazar el nombre de una fila, en el primer campo numérico se da el valor del correspondiente coeficiente en la matriz de restricciones. En el tercer campo de letras se puede enlazar con otra fila y así, colocar el valor del coeficiente correspondiente en el segundo campo numérico. En esta sección el campo de código es vacío (se debe dejar el espacio de este campo).
- **RHS.** (Right-hand side) es el valor a la derecha de cada restricción, es el valor de restricción de cada fila. En el primer campo de letras se asigna un nombre. en el segundo campo de letras se escribe el nombre de la fila la cual tiene la restricción. En el primer campo numérico se escribe el valor de la restricción derecha. En el tercer campo de letras y segundo campo numérico pueden contener información acerca de otra restricción. Las restricciones que son igual a cero no requieren una entrada en esta sección.

- **RANGES.** Esta sección permite obtener restricciones del tipo L y G. En el primer campo numérico se da un nombre de identificación, en el segundo campo numérico el nombre de la fila y en el primer campo numérico contiene el rango de variación. Asumiendo que el valor del lado derecho de una fila i es b_i , que ya han sido detallados en la sección RHS y que el valor de rango es r_i , se tiene la siguiente tabla que detalla la acción del rango de acuerdo con el tipo de fila.

Tipo	Efecto			
G	$b_i \leq$	$a_{i1}x_{i1} + a_{i2}x_{i2} + \dots + a_{in}x_{in}$	$\leq b_i + r_i $	
L	$b_i - r_i \leq$	$a_{i1}x_{i1} + a_{i2}x_{i2} + \dots + a_{in}x_{in}$	$\leq b_i$	
E	$b_i \leq$	$a_{i1}x_{i1} + a_{i2}x_{i2} + \dots + a_{in}x_{in}$	$\leq b_i + r_i $	si $r_i > 0$
E	$b_i - r_i \leq$	$a_{i1}x_{i1} + a_{i2}x_{i2} + \dots + a_{in}x_{in}$	$\leq b_i$	si $r_i < 0$

Por ejemplo si la fila F2 es del tipo G y se define el lado derecho de 50 en la sección RHS:

```
*23456789 123456789 123456789
RHS
      RHS1      F2      50
```

entonces la declaración de rango:

```
*23456789 123456789 123456789
RANGE
      RANGE1      F2      10
```

define un rango entre [50,60].

- **BOUNDS.** Esta sección permite acotar las variables de decisión. En el primer campo de letra se da un nombre de identificación, el segundo campo de letras contiene el nombre de la columna, el primer campo numérico contiene el valor de la cota y en el campo de código se clarifica el tipo de cota. En la siguiente tabla se indica el código y las clases de cotas.

Código	Interpretación
LO	Cota inferior
Up	Cota superior
FX	Valor fijo
FR	Variable libre
MI	Variable negativa (cota inferior $-\infty$)
PL	Variable positiva (cota superior $+\infty$)
BV	Variable Binaria (0 ó 1)
UI	Cota superior si las variables fueron declaradas como enteras en la sección de COLUMNS.

En los códigos FX, MI, PL o BV, el valor en el primer campo numérico es ignorado, ya que por defecto las cotas son 0 e $+\infty$ para las variables continuas y 0 y 1 para variables enteras. Por ejemplo,

```
*23456789 123456789 123456789
BOUNDS
UP BOUND1 COL4 15
```

define una cota superior de 15 para la variable de decisión de la columna COL4.

- ENDATA. Se finaliza la descripción del archivo principal

El archivo principal del formato SMPS para describir PEGSC de (B.1) se detalla a continuación:

```
*Core file problema Flujo de Caja
*23456789 123456789 123456789 123456789 123456789
NAME Flujo_Caja ROWS
N UTIL
E INT1
E EXT1
E INT2
E EXT2
E INT3
E EXT3
```

E INT4
 E EXT4
 E INT5
 E EXT5
 E INT6
 E EXT6

COLUMNS

C1	INT1	1	INT2	-1
E1	EXT1	1	EXT2	-1
E1	UTIL	0.00015		
M1	EXT1	1	INT3	-1
M1	UTIL	-0.0002		
I1	INT1	1	EXT3	-1
I1	UTIL	-0.0002		
C2	INT2	1	INT3	-1
E2	EXT2	1	EXT3	-1
E2	UTIL	0.00015		
M2	EXT2	1	INT4	-1
M2	UTIL	-0.0002		
I2	INT2	1	EXT4	-1
I2	UTIL	-0.0002		
C3	INT3	1	INT4	-1
E3	EXT3	1	EXT4	-1
E3	UTIL	0.00015		
M3	EXT3	1	INT5	-1
M3	UTIL	-0.0002		
I3	INT3	1	EXT5	-1
I3	UTIL	-0.0002		
C4	INT4	1	INT5	-1
E4	EXT4	1	EXT5	-1
E4	UTIL	0.00015		
M4	EXT4	1	INT6	-1
M4	UTIL	-0.0002		
I4	INT4	1	EXT6	-1
I4	UTIL	-0.0002		
C5	INT5	1	INT6	-1
E5	EXT5	1	EXT6	-1
E5	UTIL	0.00015		

C6	INT6	1
E6	EXT6	1
E6	UTIL	0.00015
RHS		
RHS	INT1	109.492
RHS	EXT1	20
RHS	INT2	109.492
RHS	INT3	109.492
RHS	INT4	109.492
RHS	INT5	109.492
RHS	INT6	109.492
ENDATA		

Archivo de tiempo

El propósito de este archivo es descomponer el archivo principal en los diferentes escenarios y etapas del problema. Este archivo tiene las siguientes secciones:

- **TIME.** Se detalla el nombre del problema en el segundo campo de letras. Este nombre debe ser igual al del archivo principal y estocástico. Esta sección pertenece al archivo de cabecera.
- **PERIODS.** El archivo de cabecera contiene la palabra PERIODS en el primer campo de letras. Si el archivo principal tiene un orden de tiempo entonces en el segundo campo de letras se escribe IMPLICIT o se puede dejar en blanco. Si el archivo principal no tiene un orden temporal se pone la palabra EXPLICIT.

Cada registro de datos contiene el nombre de un período en el tercer campo de letras. Si el archivo de tiempo es un formato implicit entonces el primer campo de letras contiene el nombre de la primera columna en este período, el segundo campo de letras contiene el nombre de la primera fila. Si el archivo de tiempo es explicit, los primeros dos campos de nombre no son usados. La fila de la función objetivo debe siempre pertenecer al primer período.

- **ROWS.** Esta sección es opcional, especifica para cada fila el período al cual pertenece.
- **COLUMNS.** Esta sección también es opcional, especifica para cada columna el período al cual pertenece.

- ENDATA. Se finaliza la descripción del archivo de tiempo

El problema estocástico (B.1) tiene 5 períodos y su archivo de tiempo es:

```
*Time file problema Flujo de Caja
*23456789 123456789 123456789 123456789 123456789
NAME          Flujo_Caja
PERIODS       IMPLICIT
  S0          INT1          PERIOD01
  S1          INT2          PERIOD02
  S2          INT3          PERIOD03
  S3          INT4          PERIOD04
  S4          INT5          PERIOD05
  S5          INT6          PERIOD06
ENDATA
```

Archivo estocástico

Las *etapas* ayudan a descomponer un problema estocástico en piezas más pequeñas. Estas indican el tiempo cuando nueva información esta disponible. El archivo estocástico permite resolver el problema determinista equivalente de un problema estocástico. Para esto se requiere de información acerca de las variables aleatorias. Si todas las variables son finitamente distribuidas se puede construir un árbol de eventos. El árbol de eventos puede ser descrito en tres diferentes formas: escenario por escenario, nodo por nodo e implícitamente usando distribuciones marginales. La descripción implícita son utilizadas con distribuciones continuas. Para el problema de gestión de saldo en caja se utiliza la forma explícita escenario por escenario, razón por la cual se explicará su formato, para conocer los otras formas se puede revisar en [20].

Secciones para la forma de escenarios.

- STOCH. Se da el nombre del problema en el segundo campo de letras. Este nombre debe ser igual al del archivo principal y de tiempo.
- SCENARIOS. Esta sección describe un árbol de eventos explícitos escenario por escenario. Para el primer período se tiene un único nodo, muchas veces este nodo suele ser llamado nodo raíz. Un escenario es un camino desde el nodo raíz hasta una de los hojas (nodo) del árbol de eventos. Cada rama de un período es compartido entre un escenario y su escenario padre, sólo

después de ocurrida una rama habrá información duplicada. Esto permite la reducción redundancia en el árbol, así comprimiendo el tamaño del archivo estocástico.

Esta sección usa dos clases de registros de datos. La primera para indicar que comienza un nuevo escenario, tiene el código SC, se indica el nombre del escenario en el primer campo de letras, en el primer campo numérico se especifica la probabilidad que el escenario ocurra, en el segundo campo de letras se escribe el nombre del escenario del cual se ramifica y en el último campo de letras se escribe el período en el cual sucede la ramificación. El escenario que se origina en la primera etapa y no tiene un escenario padre se debe indicar con la palabra ROOT en el segundo campo de letras.

En el segundo registro de datos da los valores de las columnas y filas asumidas por el escenario.

El archivo estocástico para el árbol de la Sección 2.4 con el cual se formula (B.1) se describe como:

```
*Stoch escenarios file problema flujo de Caja
*23456789 123456789 123456789 123456789 123456789
STOCH          Flujo_Caja
SCEN          DISCRETE      REPLACE
SC SCEN_1     'ROOT'        0.0530842  PERIOD01
  RHS         INT2          17.6274
  RHS         INT3          17.1252
  RHS         INT4          22.816
  RHS         INT5          11.8574
  RHS         INT6          22.2558
SC SCEN_2     SCEN_1        0.0530842  PERIOD05
  RHS         INT5          -10.0941
  RHS         INT6          15.1214
SC SCEN_3     SCEN_1        0.0530842  PERIOD04
  RHS         INT4          0.876122
  RHS         INT5          4.7267
  RHS         INT6          19.9383
SC SCEN_4     SCEN_3        0.0530842  PERIOD05
  RHS         INT5          -17.2248
  RHS         INT6          12.8039
SC SCEN_5     SCEN_1        0.0530842  PERIOD03
  RHS         INT3          -4.7049
```

	RHS	INT4	15.7211	
	RHS	INT5	9.55145	
	RHS	INT6	21.5064	
SC	SCEN_6	SCEN_5	0.0530842	PERIOD05
	RHS	INT5	-12.4001	
	RHS	INT6	14.372	
SC	SCEN_7	SCEN_5	0.0530842	PERIOD04
	RHS	INT4	-6.21884	
	RHS	INT5	2.42078	
	RHS	INT6	19.1889	
SC	SCEN_8	SCEN_7	0.0530842	PERIOD05
	RHS	INT5	-19.5307	
	RHS	INT6	12.0544	
SC	SCEN_9	SCEN_1	0.0530842	PERIOD02
	RHS	INT2	-3.13368	
	RHS	INT3	10.3776	
	RHS	INT4	20.623	
	RHS	INT5	11.1446	
	RHS	INT6	22.0242	
SC	SCEN_10	SCEN_9	0.0530842	PERIOD05
	RHS	INT5	-10.8069	
	RHS	INT6	14.8898	
SC	SCEN_11	SCEN_9	0.0530842	PERIOD04
	RHS	INT4	-1.31689	
	RHS	INT5	4.01395	
	RHS	INT6	19.7067	
SC	SCEN_12	SCEN_11	0.0530842	PERIOD05
	RHS	INT5	-17.9375	
	RHS	INT6	12.5722	
SC	SCEN_13	SCEN_9	0.0530842	PERIOD03
	RHS	INT3	-11.4524	
	RHS	INT4	13.5281	
	RHS	INT5	8.8387	
	RHS	INT6	21.2747	
SC	SCEN_14	SCEN_13	0.0530842	PERIOD05
	RHS	INT5	-13.1128	
	RHS	INT6	14.1403	
SC	SCEN_15	SCEN_13	0.0530842	PERIOD04


```

RHS      INT4      -8.41185
RHS      INT5      1.70803
RHS      INT6      18.9572
SC SCEN_16  SCEN_15      0.0530842  PERIOD05
RHS      INT5      -20.2435
RHS      INT6      11.8228
ENDATA

```

B.3. FORMATO DIMACS

Como ya se mencionó en el Capítulo 2 si se conocen todos los valores del flujo de caja neto el POGSC se puede reducir a un problema de flujo de costo mínimo. Un algoritmo combinatorio que permite resolver problemas de flujo de costo mínimo es el método del simplex en redes, que está implementado en el programa mcf [40]. Este programa lee los problemas descritos en el formato DIMACS el cual tiene la siguiente estructura:

- *Líneas de comentarios* cada línea de comentario comienza con la letra *c* y pueden aparecer en cualquier parte del archivo.
- *La línea del problema* esta aparece solo una vez en el archivo, comienza con la letra *p*, *min* para identificar que es un problema de MCF, el número de nodos y el de arcos en la red.
- *Descripción de los nodos* cada línea de descripción de los nodos comienza con la letra *n*, luego el número de identificación del nodo y la cantidad de demanda o suministro, los nodos de transbordo no necesitan ser declarados.
- *Descripción de los arcos* estas líneas comienzan con la letra *a* a continuación esta el número del nodo de salida y del nodo de destino, luego las cotas inferior y superior de la capacidad para el arco y finalmente el costo de dicho arco.

Las demandas, costos y las cotas para las capacidades deben ser enteros para poder resolver en el programa de mcf, además la cota inferior de la capacidad debe ser cero, para lo cual se realiza una transformación en las cotas de las capacidades y en la demanda de los arcos. Para transformar la cota inferior a cero de un arco $(i, j) \in A$ se disminuye la demanda del nodo i en el valor de la cota inferior de capacidad de dicho arco l_{ij} , se incrementa este valor al nodo j y se resta el

valor de la cota inferior de la cota superior de la capacidad del arco.

El formato DIMACS para una instancia de cinco días con un saldo en caja inicial de 140,844; saldo exterior inicial de 20; tasa por enviar/retirar dinero de $c\% = 0,02\%$; tasa de retorno $r\% = 0,015\%$; y flujo de caja, saldo mínimo y flujo transformado como se indican a continuación:

Día	Saldo Mínimo	Flujo	Flujo Transformado
1	65,191	-1,961	5498
2	65,191	9,892	9892
3	65,191	9,491	9491
4	65,191	4,64	4640
5	65,191	-3,288	-11400
	73,303		

es de la siguiente forma:

c Formato DIMACS-MCF para un flujo de caja de 5 dias

c

c linea del problema:

p min 11 16

c

c descripcion de los nodos:

c

n 1 7369199

n 2 989199

n 3 949099

n 4 463999

n 5 -1139999

n 6 2000000

n 11 -10631497

c descripcion de los arcos:

c

c descripcion de los arcos dia interior entre si:

c

a 1 2 0 1000000000 0

a 2 3 0 1000000000 0

a 3 4 0 1000000000 0

a 4 5 0 1000000000 0

```

a 5 11 0 1000000000 0
c descripcion de los arcos dia interior con dia exterior:
c
a 1 8 0 1000000000 20
a 2 9 0 1000000000 20
a 3 10 0 1000000000 20
c descripcion de los arcos exterior entre si:
c
a 6 7 0 1000000000 -15
a 7 8 0 1000000000 -15
a 8 9 0 1000000000 -15
a 9 10 0 1000000000 -15
a 10 11 0 1000000000 -15
c descripcion de los arcos dia exterior con dia interior:
c
a 6 3 0 1000000000 20
a 7 4 0 1000000000 20
a 8 5 0 1000000000 20

```

En la escritura de este formato se ha multiplicado por 1000 el flujo de dinero y por 1×10^5 a las tasas de retorno y costo para transformarlas en cantidades enteras.

Al resolver un problema de FCM el programa mcf retorna los resultados en el siguiente formato:

- *Líneas de comentarios* que de igual manera comienzan con la letra *c*
- *Línea de solución* esta comienza con la letra *s* e indica el mínimo valor de costo del flujo factible.
- *Líneas de asignación de flujo* estas empiezan con la letra *f* luego indica los nodos del arco origen y destino y finalmente el valor del flujo por dicho arco.

La solución arrojada por el programa para la instancia de ejemplo es:

```

c Output to minimum-cost flow problem C:\GFCaja\datos\fcm
c The problem was solved with a network simplex code
c
c need 10 iteration(s) in 0 second(s).
s -344121965
f 3 4 949099

```

f 4 5 1413098
 f 5 11 273099
 f 1 8 7369199
 f 2 9 989199
 f 6 7 2000000
 f 7 8 2000000
 f 8 9 9369199
 f 9 10 10358398
 f 10 11 10358398

c

c All other variables are zero

En la siguiente tabla se indica los valores del flujo, saldo mínimo y los saldo en caja/externo, inversiones, retiros y utilidad obtenidos con la solución óptima.

Día	Flujo	Saldo mínimo	Saldo caja	Saldo exterior	Inversiones	Retiros
1	-1.961	65.191	65.191	20	73.692	0
2	9.892	65.191	65.191	20	9.89199	0
3	9.491	65.191	74.682	93.692	0	0
4	4.64	65.191	79.322	103.584	0	0
5	-3.288	65.191	76.034	103.584	0	0
utilidad =0.03441						

- [1] John R. Birge & Francois Louveaux. *Introduction to Stochastic Programming*. Springer(1997).
- [2] G.B Dantzig, "Linear programming under uncertainty," *Managment Sciencie* 1 (1955) pp. 197-206.
- [3] G. B. Dantzig. Planning under uncertainty. *Annals of Operations Research*, 85:preface, 1999.
- [4] E.M.L Beale, "On minimizing a convex function subject to linear inequalities," *J. Royal Statistical Society, Serie B* 17 (1955) pp. 173-184.
- [5] Kelley C.T, *Iterative Methods for optimization*. Society for Industrial and Applied Mathematics.SIAM.(1999).
- [6] A.Cauchy, *Methode generale pour la resolution des systemes d'equations simultanees*, Comp. Rend.Acad.SCI.Paris,(1847),pp 536-538.
- [7] L.Armijo, *Minimization of functions having Lipschitz-continuous first partial derivatives*, Pacific J.Math,(1966),pp 1-3.
- [8] Sven O. Krumke & Tjark Vredeveld. *Introduction to Online Optimizatio*. March 2004
- [9] John H. Mathews & Kurtis K. *Numerical Methods Using Matlab*, 4th Edition.(2004)
- [10] Allan Borodin & Ran El-Yaniv, *Online Computation and Competitive Analysis*. Cambridge University, 1998.
- [11] E. Koutsoupias and C. Papadimitriou, *Beyond Competitive Analysis*, Proceedings of the 35th Annual IEEE Sympocium on the Foundations of Computer Sciencie,1994,pp.394-400.
- [12] R. Motwani and P. Raghavan. *Randomized Algorithms*
- [13] Alfonso Novales Cinca. *Econometria*, segunda edición. McGraw-Hill (1993).
- [14] Ezequiel Uriel Jimenez. *Análisis de Series Temporales modelos ARIMA*. España (1985).
- [15] Box, G. E. P.,y Jenkins, G. M. *Time series Analysis. Forecasting and control*. Holden Day, segunda edición (1970).
- [16] Raiffa, *Desición Analysis* (Addison-Wesley, Reading, MA,1968).

- [17] Raiffa and R. Schlaifer, *Applied Statistical Decision Theory* (Harvard University, Boston, MA, 1961).
- [18] A. Madasky, "Inequalities for stochastic linear programming problems," *Management Science* 6 (1960) pp.197-204.
- [19] Sequeira, Sebastián Eloy. *Real Time Evolution (RTE) for on-line optimisation of continuous and semi-continuous chemical processes*. Universitat UPC. Enginyeria Química.
- [20] H.I. Gassmann, *The SMPSS format for stochastic linear programs* School of Business Administration, Dalhousie University, Canada (<http://myweb.dal.ca/gassmann/smpps2.htm>)
- [21] M.O. Ball, T.L Magnanti, C.L. Monma G.L Nemhauser, *network models*, Handbooks in operations research and management science. volumen 7, (1995) Elsevier Science B.V.All, Amsterdam, The Netherlands.
- [22] Muñoz Martos, María del Mar (2003) Programación estocástica : algunas aportaciones teóricas y computacionales. Tesis Doctoral, Universidad Complutense de Madrid.
- [23] G.L. Nemhauser A.H.G. Rinnooy Kan M.J. Todd Eds., *Handbooks in OR & MS, Vol 1, Optimization*. Elsevier Science Publishers B.V (North-Holland) 1989. Chap VIII, Roger Roger J. -B.Wets. 573-574.
- [24] D.P. Heyman, M.J. Sobel, Eds., *Handbooks in OR & MS, Stochastic Models*. Vol 2, Elsevier Science Publishers B.V (North-Holland) 1990.
- [25] Charnes and W.W. Cooper, "Chance-constrained programming", *Management Science* 5 (1959) pp. 73-79
- [26] A. Prékopa, *Stochastic Programming* (Kluwer Academic Publishers, Dordrecht, Netherlands, 1995).
- [27] Mgr. Martin Šmid, *On approximation of stochastic programming problems*, Charles University in Prague, Faculty of Mathematics and Physics, Department of Probability and Mathematical Statistics, 2004, Doctoral Thesis, 2004.
- [28] Andy Philpott, "Stochastic Programming Introduction". internet. www.stopro.org. Acceso: 20 julio 2007.

- [29] P. Kall and D. Stoyan, "Solving stochastic programming problems with recourse including error bounds," *Math. Operationsforsch. Statist. Ser. Optim.* 13(1982) pp.431-447.
- [30] A. Wald, *Statistical Decision Functions* (John Wiley, Inc. New York, NY, 1950).
- [31] Ramos Andrés y Cerisola Santiago, *Optimización Estocástica*. Universidad Pontificia Comillas. Madrid-España(2007).
- [32] J.C. Lagarias, J.A. Reeds, M. H. Wright, P. E. Wright, *Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions*, *SIAM J. Optim.*, Vol. 9 (1998), 112-147.
- [33] J. A. Nelder and R. Mead, *A simplex method for function minimization*, *Computer Journal* 7 (1965), 308-313.
- [34] F. H. Walters, L. R. Parker, S. L. Morgan, and S. N. Deming, *Sequential Simplex Optimization*, CRC Press, Boca Raton, FL, 1991.
- [35] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, UK, 1988.
- [36] Math Works, Matlab, The Math Works, Natick, MA, 1994.
- [37] W. W. Hines y D.C. Montgomery, *Probabilidad y Estadística*, tercera edición, Wiley & Sons, Mexico (2002). ç
- [38] Box G. E. and Tiao G.C. *Intervention Analysis with Applications to Economic and environment Problems*. *Journal of the American Statistical Association*, (1975)
- [39] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- [40] Löbel, Andreas. MCF, *A network simplex implementation*, version 1.3.
- [41] Descargable en la pagina web de la COIN-OR, <http://www.coin-or.org/index.html>
- [42] DIMACS(1993). The first DIMACS international algorithm implementation challenge. Available via WWW at URL <ftp://dimacs.rutgers.edu/pub/netflow>.

[]