

Modelo estocástico y algoritmos en-línea para el problema de gestión de saldo en caja en el Banco Central del Ecuador

MARÍA SOTO *

Resumen

El presente trabajo aborda el problema de la gestión adecuada de los fondos que la banca privada nacional mantiene en el Banco Central del Ecuador. Mediante la utilización de algoritmos en-línea, se busca dar soporte en la toma de decisiones al momento de determinar el monto de inversiones y retiros que deben realizarse diariamente sobre cuentas en el extranjero, para generar un nivel razonable de utilidad sin poner en riesgo la liquidez del sistema financiero nacional. Se formula un modelo de programación estocástica multietapa para el problema de gestión de saldo en caja en el Banco Central del Ecuador. Se observa que el tamaño del modelo determinístico equivalente para el problema estocástico crece exponencialmente al aumentar el número de períodos, lo que hace a los métodos de solución basados en el esquema de re-curso computacionalmente inaplicables en la práctica y por lo que se proponen nueve algoritmos en-línea para el problema; obteniéndose la mejor solución en uno de ellos. Se presentan los principales resultados numéricos de la calibración de los parámetros, así como de la evaluación del desempeño de los algoritmos. Como criterio de comparación se emplea en cada instancia el resultado obtenido por la política óptima del modelo estocástico, la cual está dada por la solución de un problema de flujo de costo mínimo. Finalmente, se presentan las conclusiones obtenidas y algunas recomendaciones para trabajos futuros respecto al problema de gestión de saldo en caja.

Abstract

This paper addresses the problem of adequate management of funds that the Ecuadorian private banks must keep in an account at the Central Bank to guarantee solvency of the Ecuadorian financial system. The idea is to invest a certain amount of these funds in foreign banks to increase profit with a very small risk. Through the use of online algorithms we look for decision-aid models to support the daily process of determining investments abroad.

We present a stochastic optimization model to manage the account balance in the Central Bank of Ecuador. It turns out that the deterministic equivalent of our multistage stochastic program is far too large to be computationally applicable, even for relatively small time periods. Therefore we investigate the use of heuristic techniques and propose online algorithms. We exhibit the results obtained in numerical simulations. Also, we

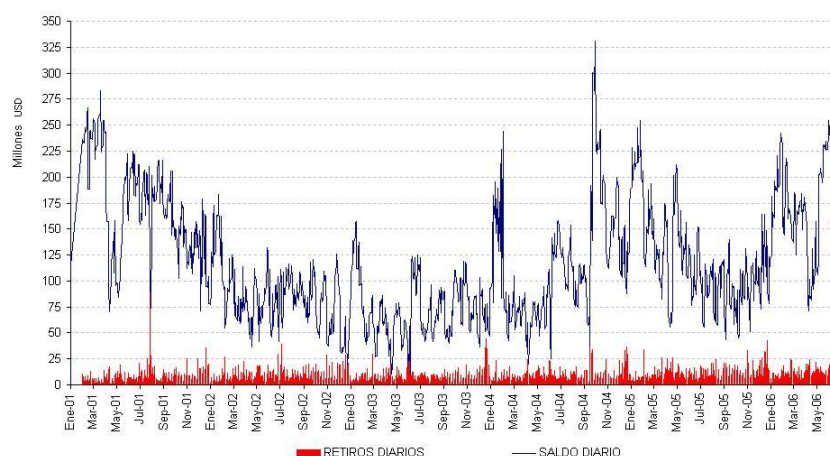
*La autora agradece por el apoyo y dirección del proyecto al Dr. Luis Miguel Torres. Este trabajo corresponde a un extracto de la tesis para obtener el título de Ingeniera Matemática, disponible en <http://www.math.epn.edu.ec/tesis/M-Soto>. Las opiniones vertidas en este documento son de exclusiva responsabilidad de la autora y no representa la posición oficial del BCE. Para cualquier comentario o sugerencia favor comunicarse con msoto@bce.ec.

compare the solutions obtained for the various algorithms to each other and to the optimal policy (i.e., the solution under perfect information), which is obtained by solving a minimum cost flow problem. Finally, we state some conclusions and observations that could be useful for future work.

1. 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 1.¹

Figura 1: Saldo en caja y retiros diarios. Enero 2001 a Junio 2006.



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,

¹La fuente de los datos es el Banco Central del Ecuador, a menos que se especifique lo contrario.

que se ajusta semanalmente. El saldo en caja diario no debe ser nunca inferior a este valor. 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.

1.1. Modelo de optimización para el PGSC

El problema de gestión de saldo en caja en el Banco Central del Ecuador (PGSC) 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:

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

s.r

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

$$\text{(POGSC)} \quad C_t = C_{t-1} + Y_t - I_t + M_{t-2} \quad t = 1, \dots, T \quad (3)$$

$$C_t \geq L_t \quad \forall t = 1, \dots, T \quad (4)$$

$$I_t, M_t, E_t \geq 0 \quad \forall t = 1, \dots, T. \quad (5)$$

donde: Y_t es el flujo neto en caja 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 son: 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) y (3). Por simplicidad, supondremos que $I_{-1} = M_{-1} = I_0 = M_0 = 0$.

La función objetivo (1) refleja la utilidad generada durante el horizonte de planificación (interés generado por el saldo diario en el extranjero menos los costos de envío y retiro). Se denota 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 .

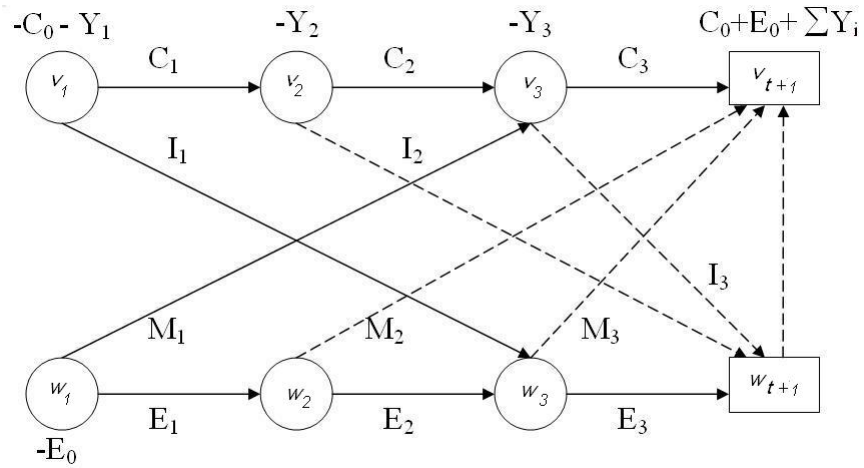
La ecuación (4) es la restricción de seguridad, donde L_t es el valor mínimo para el saldo en caja en el día t .

Notar que tanto el flujo neto en caja diario, como las tasas de interés y las tasas de cobro por transacción son variables aleatorias. 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, se suponen dichos parámetros conocidos de antemano. Así, la incertidumbre considerada en el modelo se refiere exclusivamente al flujo neto de dinero diario.

1.2. Solución bajo información perfecta

Dada una instancia del Problema Optimización de Gestión de Saldo Caja (POGSC) en la que 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 . En la Figura 2 se presenta el grafo para una instancia de tres días.

Figura 2: Grafo flujo de costo mínimo para una instancia de tres días del PGSC.



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 $A_3 := \{(v_t, w_{t+2}) : 1 \leq t \leq T-1\} \cup \{(w_T, w_{T+1})\}$ para representar los envíos al extranjero; y un conjunto $A_4 := \{(w_t, v_{t+2}) : 1 \leq t \leq T-1\} \cup \{(w_T, v_{T+1})\}$ para los retiros del extranjero. Adicionalmente, definimos un arco auxiliar de la forma (w_{T+1}, v_{T+1}) . La capacidad máxima de cada arco es ilimitada ($+\infty$). 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:

$$b(v_t) := \begin{cases} -C_0 - Y_1, & \text{si } t = 1 \\ -y_t, & \text{si } 2 \leq t \leq T \\ (C_0 + E_0 + \sum_{i=1}^T y_i), & \text{si } t = T+1 \end{cases} \quad (6)$$

$$b(w_t) := \begin{cases} -E_0, & \text{si } t = 1 \\ 0, & \text{en los demás casos.} \end{cases} \quad (7)$$

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:

$$c_a := \begin{cases} -r_t, & \text{si } a \in A_2, \\ c_t^I, & \text{si } a \in A_3, \\ c_t^M, & \text{si } a \in A_4, \\ 0, & \text{en los demás casos.} \end{cases} \quad (8)$$

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) y (3) son equivalentes a las restricciones de conservación de flujo sobre los nodos exteriores e interiores, respectivamente. Las restricciones (4) y (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 (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.

Como los valores del flujo de caja diario no pueden predecirse con exactitud, ni siquiera para horizontes de tiempo pequeños, el problema de gestión de saldo en caja del BCE es un problema de optimización bajo incertidumbre. 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.

1.3. Modelo estocástico multietapa para el PGSC

Cada escenario s se representa 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, se asume que el flujo de dinero para un día t puede tomar valores únicamente de un conjunto finito $\mathbb{Y}_t$. Es decir, $y_t \in \mathbb{Y}_t$, $\forall 1 \leq t \leq T$ con $|\mathbb{Y}_t| =: k_t \in \mathbb{N}$. 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 di-

ario 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. Se formula el siguiente programa multietapa con recurso:

$$\begin{aligned}
 \text{máx} \quad & \mathbf{E}[U] = \sum_{(y_1, \dots, y_T) \in \mathbb{Y}_1 \times \dots \times \mathbb{Y}_T} p(y_1, \dots, y_T) \\
 & \left[\sum_{t=1}^T [r_t E_t(y_1, \dots, y_t) - \right. \\
 & \left. c_t^I I_t(y_1, \dots, y_t) - c_t^M M_t(y_1, \dots, y_t)] \right] \\
 \text{s.t} \quad & E_t(y_1, \dots, y_t) = E_{t-1}(y_1, \dots, y_{t-1}) + \\
 & I_{t-2}(y_1, \dots, y_{t-2}) - M_t(y_1, \dots, y_t), \\
 & \forall (y_1, \dots, y_t) \in \mathbb{Y}_1 \times \dots \times \mathbb{Y}_t, \forall t = 1, \dots, T \\
 \text{(PEGSC)} \quad & C_t(y_1, \dots, y_t) = C_{t-1}(y_1, \dots, y_{t-1}) + \\
 & Y_t(y_1, \dots, y_t) - I_t(y_1, \dots, y_t) + \\
 & + M_{t-2}(y_1, \dots, y_{t-2}), \\
 & \forall (y_1, \dots, y_t) \in \mathbb{Y}_1 \times \dots \times \mathbb{Y}_t, \forall t = 1, \dots, T \\
 & C_t(y_1, \dots, y_t) \geq L_t, \\
 & \forall (y_1, \dots, y_t) \in \mathbb{Y}_1 \times \dots \times \mathbb{Y}_t, \forall t = 1, \dots, T \\
 & I_t(y_1, \dots, y_t), M_t(y_1, \dots, y_t), E_t(y_1, \dots, y_t) \geq 0, \\
 & \forall (y_1, \dots, y_t) \in \mathbb{Y}_1 \times \dots \times \mathbb{Y}_t, \forall t = 1, \dots, T
 \end{aligned}$$

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. 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}). Un ejemplo del modelo estocástico de gestión de saldo en caja para un horizonte corto se encuentra en la tesis completa.

2. Enfoque de solución mediante Algoritmos en-línea

La elevada complejidad del modelo estocástico multi-etapa propuesto 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 (Sven 2004). 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* (Borodin 1998).

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.

Se han propuesto y estudiado: 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.

Los algoritmos de bandas trabajan de manera muy similar: determinan niveles límite e inferior óptimos en los cuales debe fluctuar el saldo en caja; si el saldo en caja supera un valor límite conocido como *banda máxima* se transfiere dinero al exterior y cuando este cae por debajo del nivel inferior conocido como *banda mínima* se realiza un retiro del exterior. La diferencia de los algoritmos de bandas radica en la forma como establecen los valores de las bandas máxima y mínima y en el criterio utilizado para fijar la cantidad a enviar/retirar del exterior. Los algoritmos de bandas móviles establecen el valor de las bandas máxima y mínima en base a un saldo base el cual cambia semanalmente, mientras que los algoritmos de bandas fijas establecen los valores de las bandas constantes para todo el horizonte de planificación.

Respecto a los criterios utilizados para tomar las decisiones de cuánto enviar/retirar del exterior los algoritmos lineales lo realizan mediante una fun-

ción lineal del exceso/faltante de las bandas máxima/mínima; los algoritmos cuadráticos lo hacen mediante una función cuadrática sobre el mismo exceso/faltante; los algoritmos de 2 bandas establecen valores fijos a enviar y retirar y los algoritmos de 4 bandas establecen dos bandas intermedias para envia/retiran dinero.

Tanto los algoritmos fijos como móviles 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 lo tanto se propone el algoritmo Estocástico que emplea técnicas estocásticas que toma en cuenta la presencia de incertidumbre en el modelo. Para tomar las decisiones de cuánto y cuándo enviar/retirar dinero del exterior formula y resuelve cada día el PEGSC para un horizonte de tiempo manejable.

El desempeño de los algoritmos depende de los valores tomados para los parámetros de calibración como por ejemplo: bandas mínima y maxima, bandas intermedias, factores lineal y cuadrático de inversión/retiro. La búsqueda de estos valores se debe realizar durante una fase previa a la aplicación del algoritmo en un entorno determinado. Esta tarea se puede formular como un subproblema de optimización no lineal (presentado en el Anexo A) y además se consideran el método numérico de solución: método de Nelder-Mead (Lagarias 1965 y Nelder Mead 1998). 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 este estudio.

Para realizar la calibración se requiere conocer la función de distribución del flujo neto de dinero diario, para esto se realizó un estudio de la serie histórica mediante series de tiempo y análisis de intervención, encontrándose tres especificaciones para la variable: Modelo sarima (SA), Modelo de Impulso (IM) y modelo Sarima-Impulso (SI), además, con fines comparativos se especificó una distribución Normal (NO). Debido al uso de las diferentes distribuciones en la calibración de parámetros se obtuvo en total 35 algoritmos en-línea.

Para medir el desempeño de los algoritmos en-línea se realizó una simulación de los 35 algoritmos sobre instancias reales 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 10, 30 y 100 días. Se registra para cada uno de los algoritmos 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 míni-

mo aceptable: Llamaremos a estos días como “días malos”. A continuación, se resuelve el problema fuera-de-línea empleando un algoritmo para un 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*):

$$\text{gap} = \frac{u^* - \hat{u}}{u^*} \quad (9)$$

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 1 detalla para cada instancia los valores del *gap*, utilidad, utilidad corregida y número de días malos obtenido por el algoritmo que tuvo el mejor desempeño.

De las instancias consideradas en el 1 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 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

Cuadro 1: Algoritmos con el mejor desempeño, para instancias reales con horizontes de tiempo de 100 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

instancias, sin embargo en muchos casos estos algoritmos son los más riesgosos.

Con el fin de observar más detalladamente el comportamiento de los algoritmos de bandas, propuestos para el problema de gestión de saldo en caja, seleccionamos una instancia de 100 días y de igual manera aplicamos las 35 variantes de los algoritmos. La instancia está formada con los datos históricos desde el 3 de enero hasta el 25 de mayo de 2005.

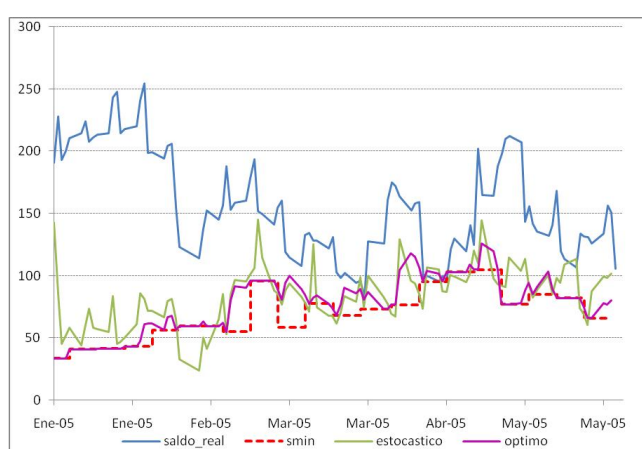
En el Cuadro 2 se presenta la utilidad corregida al final del período, el porcentaje de días en los cuales el saldo en caja descendió por debajo del nivel mínimo aceptable y el valor del gap obtenido para cada uno de los algoritmos y también para la política óptima.

Cuadro 2: Desempeño de los mejores algoritmos sobre la instancia real de enero-mayo 2005.

Algoritmo	Utilidad	% días malos	gap
2 Bandas Móvil Sarima-Impulso	3.2540	3	0.22
Lineal Móvil Sarima-Impulso	3.2983	2	0.21
Cuadrático Móvil Sarima-Impulso	3.2712	0	0.21
Estocástico Impulso	3.7518	28	0.10
Estocástico Sarima-Impulso	3.5920	29	0.14
Estocástico Sarima	3.4677	23	0.17
Óptimo	4.1596	0	0.00

Como se esperaba las variantes de Estocástico obtuvieron las mejores utilidades, sin embargo presentan los mayores porcentajes de días malos. Estocástico Impulso es el algoritmo con la mayor utilidad de 3,7518 millones de dólares y Estocástico Sarima-Impulso tiene la mayor cantidad de 29 días malos. A las utilidades obtenidas por las variantes de Estocástico le siguen los utilidades de los algoritmos móviles: Lineal Móvil SI, Cuadrático Móvil SI y 2-Bandas Móvil SI. Estos últimos algoritmos tienen menor porcentaje de días malos. En el Figura 3 se presentan los saldos en caja obtenidos por Estocástico Impulso, Por la política óptima y el saldo en caja real en ese período de tiempo.

Figura 3: Saldo en caja: Estocástico Impulso, Óptimo y Real



El saldo en caja generado al aplicar Estocástico Impulso cae por debajo del saldo mínimo en 28 días, presentándose los mayores problemas de liquidez en los primeros días del mes de febrero de 2005. Sin embargo, este saldo en caja es cerca al saldo en caja obtenido por la política óptima, razón por la cual obtiene la mayor utilidad, pero también tiene uno de los mayores riesgos de liquidez.

El saldo en caja obtenido por la política óptima se pega al saldo mínimo necesario en muchos de los días. Generalmente el saldo en caja real mantenido por el Banco es bastante superior a los saldo en caja obtenidos por el algoritmo y por la política óptima.

En la Figuras 4 y 5 se puede observar los saldo en caja obtenidos por los algoritmos Lineal Móvil SI y Cuadrático Móvil IM respectivamente, además

se puede observar los saldo en caja real y el óptimo.

Figura 4: Saldo en caja: Lineal Móvil SI, Óptimo y Real.

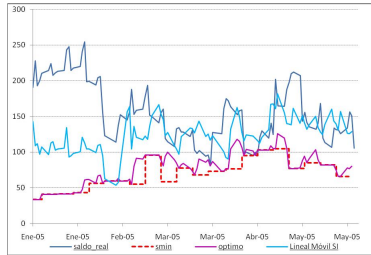
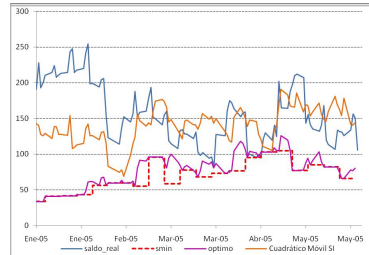


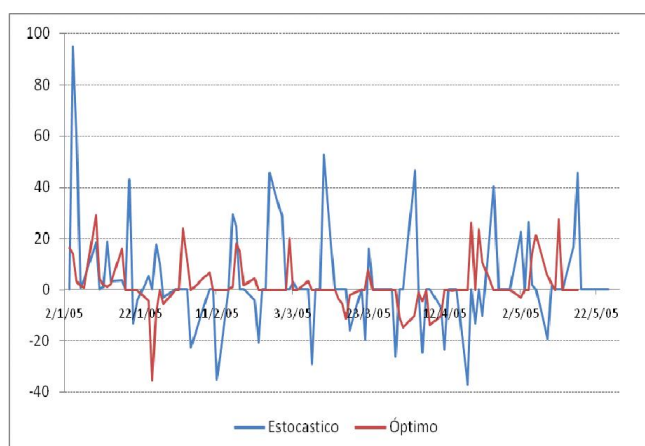
Figura 5: Saldo en caja: Cuadrático, Móvil SI Óptimo y Real.



Los saldos en caja obtenidos por estos algoritmos están más alejados del saldo en caja óptimo, sin embargo, son más bajos que el saldo en caja real del Banco Central y tiene un bajo riesgo de liquidez. Estocástico se acerca más al óptimo debido a utilizar modelos estadísticos para estimar la incertidumbre futura y así tomar las decisiones. Sin embargo las estimaciones están sujetas a cierto error, por lo que se pueden presentar problemas de liquidez. Los saldo en caja obtenidos por los algoritmos se desarrollan conforme se van tomando las decisiones de inversión y retiro de dinero.

En la Figura 6 se presentan las inversiones y retiros realizados por Estocástico y por la política óptima. Para poder visualizar se han graficado los retiros con signo negativo y las inversiones con signo positivo. La política óptima realiza menor cantidad de transacciones y sus montos de dinero son más bajos tanto en los retiros como en los depósitos.

Figura 6: Inversiones - retiros realizados por Estocástico IM y por el óptimo



En la Figuras 7 y 8 se puede comparar los retiros e inversiones de los algoritmos Lineal Móvil SI y Cuadrático Móvil con la política óptima respectivamente.

Figura 7: Inversiones-retiros por Lineal Móvil SI y el óptimo

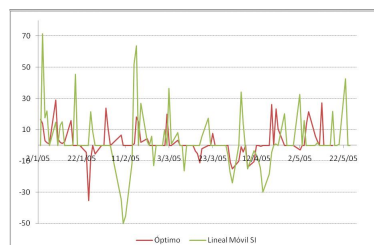
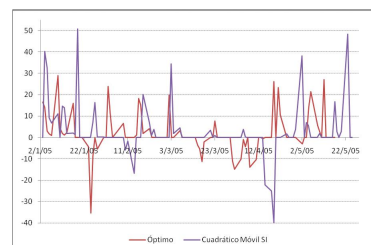


Figura 8: Inversiones-retiros por Cuadrático Móvil SI y el óptimo.



Generalmente los montos de las inversiones y retiros realizados por Estocásticos son mayores a los de los otros dos algoritmos. En las transacciones realizadas por Lineal se puede observar un inconveniente con los algoritmos de bandas. El 10 de febrero se retira un monto significativo debido al decaimiento del saldo, sin embargo en los siguientes días el flujo de caja se recupera y se debe realizar una inversión, se decir volver a enviar el dinero pedido en la

última transacción.

Se podría considerar aplicar los algoritmos Estocástico, Lineal móvil o 2-Bandas móvil que fueron los que tuvieron un mejor desempeño. A continuación, se detalla el funcionamiento de estos algoritmos.

Algoritmo Estocástico

Estocástico recibe como entrada, toda la información necesaria para formular el modelo estocástico para el PGSC 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 A describe el funcionamiento de Estocástico. Cabe notar que este algoritmo no tiene parámetros de calibración.

Algoritmo A: 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 (vectores de flujo en caja)

$p(s), \forall s \in S$: 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 PGSC para  $h$  días.
```

```

.   Asignar
.        $M := M_1$   $I := I_1$ 
.   fin si

```

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 la subsección 1.3, 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.

Algoritmo Lineal Móvil

Los valores de 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 deficit respecto a la banda mínima. Igualmente, los montos a retirar e 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 B describe los detalles.

Algoritmo B: LINEAL 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 : 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 2-Bandas Móvil

De igual manera 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 , el cuadro Algoritmo C describe los detalles.

Algoritmo C: 2-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 : 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

```

3. Conclusiones

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 es que a menudo se 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 decaer 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 para evitar realizar grandes saltos en el saldo en caja.

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 la 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 del algoritmo 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.

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 que si se va a utilizar dinero de la cuenta del extranjero los retornos de estos dineros 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.

Anexo A

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 un método numérico 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 este estudio.

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:

$$u(x, y, q) = \sum_{t=1}^T (r_t E_t - c_t^I I_t - c_t^M M_t) \quad (10)$$

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:

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

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 $f(x, q^*)$. En nuestra aplicación empleamos el método de Nelder-Mead (Lagarias 1965 y Nelder Mead 1998) para realizar esta tarea.

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.

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, $u(x)$. 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 (Montgomery 2002).

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

$$e = \frac{t_{\alpha/2, k-1} S_{u(x)}}{\sqrt{k}} \quad (11)$$

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 (11) se encuentra un tamaño k para la muestra mediante:

$$k = \left(\frac{t_{\alpha/2, k-1} S_{u(x)}}{e} \right)^2 \quad (12)$$

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 conforma 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 sea menor o igual que el limite 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 (12) se deduce:

$$k = \frac{\lceil 4S^2 \rceil}{e^2} \quad (13)$$

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.

Referencias

- [1] Birge John R. y Francois Louveaux. 1997. *Introduction to Stochastic Programming*. Springer.
- [2] G.B Datzing. 1955 “Linear programming under uncertainty”. *Management Science*. 1:197-206.
- [3] Dantzig G. B. 1999. “Planning under uncertainty”. *Annals of Operations Research*, 85:preface.
- [4] Beale E.M.L. 1995. “On minimizing a convex function subject to linear inequalities,” *J. Royal Statistical Society, Serie B*. 17:173-184.
- [5] Armijo L. 1996. *Minimization of functions having Lipschitz-continuous first partial derivatives*, Pacific J.Math.1-3.
- [6] Sven O. Krumke y Tjark Vredeveld. 2004. *Introduction to Online Optimization*.
- [7] Mathews John H. y Kurtis K. 2004. *Numerical Methods Using Matlab*, 4th Edition.
- [8] Borodin Allan y Ran El-Yaniv.1998. *Online Computation and Competitive Analysis*. Cambridge University.
- [9] Koutsoupias E. y C. Papadimitriou. 1994. “Beyond Competitive Analysis”. *Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science*,394-400.
- [10] Madasky A.1960.“Inequalities for stochastic linear programming problems,” *Management Science*. 6:197-204.
- [11] Sequeira Sebastián Eloy. *Real Time Evolution (RTE) for on-line optimization of continuous and semi-continuous chemical processes*. Universitat UPC. Enginyeria Química.
- [12] Ball M.O, T.L Magnanti, C.L. Monma y G.L Nemhauser.1995. “network models”, *Handbooks in operations research and management science*. 7.
- [13] Muñoz Martos María del Mar. 2003. *Programación estocástica : algunas aportaciones teóricas y computacionales*. Tesis Doctoral, Universidad Complutense de Madrid.
- [14] Nemhauser G.L, A.H.G. Rinnooy Kan M.J. y Todd Eds.1989. *Handbooks in OR & MS, Optimization*. 1:Chap VIII.
- [15] Heyman D.P. y M.J. Sobel, Eds. 1990. *Handbooks in OR & MS, Stochastic Models*. 2.

- [16] Charnes and W.W. Cooper. 1959. "Chance-constrained programming", *Management Science*. 5:73-79
- [17] Prékopa A. 1995. *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht, Netherlands.
- [18] Mgr. Šmid Martin. 2004. *On approximation of stochastic programming problems*, Charles University in Prague, Faculty of Mathematics and Physics, Department of Probability and Mathematical Statistics, Doctoral Thesis.
- [19] Kall P. y D. Stoyan. 1982. "Solving stochastic programming problems with recourse including error bounds". *Math. Operationsforsch. Statist. Ser. Optim.* 13:431-447.
- [20] Wald A. 1950. *Statistical Decision Functions*. John Wiley, Inc. New York, NY.
- [21] Ramos Andrés y Cerisola Santiago. 2007. *Optimización Estocástica*. Universidad Pontificia Comillas. Madrid-España.
- [22] Lagarias J.C., J.A. Reeds, M. H. Wright y P. E. Wright. 1998. "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions". *SIAM J. Optim.* 9:112-147.
- [23] Nelder J. A. y R. Mead. 1965. "A simplex method for function minimization". *Computer Journal*. 7: 308-313.
- [24] Press W. H, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1988. *Numerical Recipes in C*, Cambridge University Press.
- [25] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- [26] Löbel, Andreas. MCF, *A network simplex implementation*, version 1.3.
- [27] Montgomery D.C. y W. W. Hines. 2002. *Probabilidad y Estadística*, tercera edición, Wiley & Sons, Mexico.