



102
201
UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE CIENCIAS

Optimización del Sistema de Pagos en el
Mercado de Títulos mediante la
Descomposición Dantzig-Wolfe

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

A C T U A R I O

P R E S E N T A

ALEJANDRO DE LOS SANTOS SANTOS



FACULTAD DE CIENCIAS
UNAM

MÉXICO

DIRECTOR DE TESIS:
DR. DAVID ROMERO VARGAS.

1997

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AVENIDA DE
MÉXICO

M. en C. Virginia Abrin Batule
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
P r e s e n t e

Comunicamos a usted que hemos revisado el trabajo de Tesis:
"Optimización del Sistema de Pagos en el Mercado de Títulos
Mediante la Descomposición Danzig-Wolfe"

realizado por Alejandro de los Santos Santos
con número de cuenta 8825476-0 , pasante de la carrera de ACTUARIA

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis
Propietario
Propietario
Propietario
Suplente
Suplente

DR. DAVID ROMERO VARGAS / *[Signature]*
DR. GILBERTO CALVILLO VIVES - *[Signature]*
M. en C. AGUSTIN CANO GARCES
DR. FRANCISCO SOLIS ROBLEDA *[Signature]*
DR. ABDON SANCHEZ ARROYO *[Signature]*

Consejo Departamental de Matemáticas

[Signature]

Optimización del Sistema de Pagos en el
Mercado de Títulos mediante la
Descomposición Dantzig-Wolfe

Alejandro de los Santos Santos

*A mis madres,
especialmente a Pancha.*

GRACIAS *por todo.*

Resumen

En este trabajo presentamos el desarrollo, y la aplicación, de algunos modelos y métodos especializados de la Investigación de Operaciones para atacar un problema que se presenta en el mercado de valores mexicano. En particular, abordamos el problema de liquidación de las operaciones del mercado de títulos, utilizando el *algoritmo de descomposición de Dantzig-Wolfe* para problemas de programación lineal.

Indice General

1	Introducción	1
2	Antecedentes financieros	6
3	Antecedentes matemáticos	13
3.1	Investigación de Operaciones	16
3.2	Historia de la Investigación de Operaciones	18
3.3	La investigación de operaciones actualmente	20
4	Modelación del problema	22
4.1	Notación	25
4.2	Construcción	26
5	Metodología	31
5.1	Lectura de datos	35
5.2	Reducción del crédito especial	36

5.3 Optimización de las operaciones a liquidar	39
5.4 Solución óptima	42
5.5 Solución entera	42
5.6 Formulación del problema de flujos	42
5.7 Resolución del problema de flujos	44
5.8 Determinación de créditos especiales	45
5.9 Liquidación	46
6 La descomposición Dantzig-Wolfe	47
6.1 Modificaciones cuando los subproblemas son no acotados	54
6.2 Inicialización	55
6.3 Resumen del algoritmo	58
7 Implantación en computadora	62
7.1 Entrada	64
7.2 Salida	67
8 Descripción del sistema computacional	68
9 Ejemplos y resultados	75
9.1 Ejemplo hipotético	76
9.2 Ejemplos reales	83

Indice General

iii

10 Conclusiones**87****A Aplicaciones financieras****91****B Código de la descomposición Dantzig-Wolfe****97****Referencias Bibliográficas****107**

Índice de Figuras

3.1	Clasificación de modelos matemáticos	15
4.1	Modelo para optimizar la liquidación	29
5.1	Reducción del crédito especial (problema entero)	37
5.2	Reducción del crédito especial (problema relajado)	38
5.3	Optimización de las operaciones a liquidar (programa entero)	40
5.4	Optimización de las operaciones a liquidar (programa relajado)	41
5.5	Problema de Flujos	44
7.1	Flujo de información	65
8.1	Diagrama de Flujo del PROCOVAL	70
8.2	Diagrama de Estructura	73
9.1	Operaciones pendientes de liquidación	78
9.2	Operaciones con títulos tipo a	79

9.3	Movimientos de saldos en títulos tipo <i>a</i> , al liquidar las operaciones	80
9.4	Movimientos de efectivo al liquidar las operaciones de títulos tipo <i>b</i>	81
9.5	Liquidación óptima	82

Capítulo 1

Introducción

Diariamente, las instituciones financieras de nuestro país realizan operaciones de compra-venta de títulos del gobierno federal (CETES, BONDES, etc.) y de deuda bancaria (bonos, pagarés, etc.). Estos títulos se encuentran depositados en cuentas del Banco de México y del INDEVAL¹, respectivamente, por lo que la institución vendedora debe solicitar que se haga el traspaso de los títulos desde alguna de sus cuentas hacia alguna cuenta de la institución compradora.

Actualmente, la única restricción que hay para poder hacer este traspaso es que el vendedor tenga los títulos correspondientes en alguna de sus cuentas. No es necesario verificar que el comprador tenga el dinero, puesto que todas las instituciones participantes cuentan con una línea de crédito muy amplia por parte del Banco de México. Si al final del día, los fondos de algún participante no alcanzan para cubrir el importe de sus compras, el Banco de México le prestará el dinero que haga falta. Con este esquema, existe la oportunidad, para los participantes en el mercado de compra-venta de títulos gubernamentales y de deuda bancaria, de comprar cantidades de títulos que tal vez no puedan pagar. Como consecuencia, es posible que alguno de los participantes termine el ejercicio con un saldo de efectivo negativo y que el banco central deba desembolsar, de sus fondos, el dinero necesario para liquidar las operaciones del mercado de títulos.

De esta forma, el Banco de México asume los riesgos de liquidación en que incurre el mercado de compra-venta de títulos. Debido a esto, ha decidido implantar algunas modificaciones en las reglas de liquidación de las operaciones de este mercado.

El problema que abordamos en el presente trabajo, consiste, básicamente, en determinar, de manera óptima, qué operaciones pueden liquidarse, respetando las nuevas reglas de liquidación para el mercado de títulos, propuestas por el Banco de México. Para resolverlo, desarrollamos una metodología que hace uso de algunas herramientas de la *Investigación de Operaciones*.

¹Instituto para el Depósito de Valores

La Investigación de Operaciones (o, también llamada, Ciencia de la Administración) se refiere a un planteamiento científico para tomar decisiones, que busca determinar cómo diseñar y operar mejor un sistema, normalmente bajo condiciones que requieren la asignación de recursos escasos. La rápida evolución de las computadoras, unida a la teoría matemática desarrollada, ha permitido que la Investigación de Operaciones pueda aplicarse exitosamente en la resolución de problemas reales de la industria, el comercio y el gobierno.

El objetivo de esta tesis es hacer una aportación al acervo de aplicaciones de la Investigación de Operaciones en el ámbito financiero. En particular, nuestro trabajo puede ubicarse dentro de las aplicaciones que hacen uso de algoritmos altamente especializados de la *Programación Lineal*, como lo es el algoritmo de descomposición de Dantzig-Wolfe.

Conceptualmente, podemos dividir este documento en tres grandes secciones. Cada sección está constituida por uno o más capítulos.

En la primera sección, formada por el capítulo 1, explicamos las reformas al sistema de pagos de nuestro país que dan origen al problema en cuestión. Los tópicos de esta sección son de tipo financiero.

La segunda sección de esta tesis es de tipo matemático y está compuesta por los capítulos 2 al 5. A su vez, podemos dividir esta sección en 3 subsecciones. En la primera subsección, correspondiente al capítulo 2, tratamos la Investigación de Operaciones en general. En los capítulos 3 y 4 modelamos matemáticamente el problema real y exponemos una metodología de solución al problema. En la tercera subsección (capítulo 5), exponemos un algoritmo especializado que permite solucionar eficientemente el modelo planteado.

La tercera y última sección comprende los capítulos 6, 7 y 8; y presenta un sistema computacional que permite resolver el problema real. Esta sección se enfoca, pues, a cuestiones de programación de sistemas.

Enseguida, hacemos una descripción breve de los capítulos en que está dividido el presente trabajo.

Capítulo 1. Planteamos el problema financiero que atacaremos a lo largo de nuestro trabajo.

Capítulo 2. Hacemos una introducción de los conceptos matemáticos que utilizaremos, incluyendo una breve reseña de la Investigación de Operaciones.

Capítulo 3. Exponemos la forma en que "tradujimos" el problema real a términos matemáticos.

Capítulo 4. Proponemos una metodología que permite resolver satisfactoriamente el modelo matemático y, consecuentemente, el problema real.

Capítulo 5. Explicamos el algoritmo que nos permitirá resolver los modelos planteados de manera eficiente.

Capítulo 6 y 7. Presentamos al lector la forma en que programamos el sistema computacional que nos ayuda a resolver los modelos, basándonos en la metodología propuesta.

Capítulo 8. Ejemplificamos de manera sencilla la utilización de la metodología expuesta, y presentamos algunos resultados obtenidos con datos reales.

Capítulo 9. Exponemos algunas conclusiones derivadas del período de pruebas al que sometimos una versión del sistema.

Al final de esta tesis se encuentran dos apéndices.

Apéndice A. Listamos algunos artículos y conferencias presentados recientemente, que exponen aplicaciones de la Investigación de Operaciones en la resolución de problemas financieros.

Apéndice B. Presentamos la parte del código del sistema que implanta el algoritmo de descomposición de Dantzig-Wolfe.

En la última página listamos las referencias bibliográficas empleadas a lo largo de todo el documento.

Capítulo 2

Antecedentes financieros

En las economías de mercado es común que el dinero de sus agentes económicos esté depositado en las cuentas de las instituciones financieras, de forma que para liquidar el importe de una operación de compra-venta, es posible que el comprador retire dinero de alguna de sus cuentas y lo deposite en una cuenta del vendedor. En estos casos, el sistema financiero en que se concerta dicha operación debe contar con un mecanismo que transfiera fondos entre cuentas de sus distintos participantes.

El *sistema de pagos* de un país es el mecanismo con el que se transfiere dinero entre sus distintos agentes económicos. En otras palabras, el sistema de pagos es el mecanismo empleado para liquidar operaciones de compra-venta en las que se transfieren fondos de comprador a vendedor [5].

La forma en que se hace el pago de un producto puede ser en efectivo, con cheque, con tarjeta de crédito, etc. Estas formas tienen distintos riesgos asociados [3]. Por ejemplo, si se usa un cheque para liquidar una operación, existe la posibilidad de que este documento no tenga fondos y por tanto el vendedor no podrá transferir el dinero del comprador a sus cuentas. Este es un riesgo que debe asumir el vendedor y que no se presentaría si el pago se hiciera en efectivo. Sin embargo, con los pagos en efectivo el vendedor tiene el dinero en su poder y corre el riesgo de robo, lo cual no sucede si sólo acepta cheques o tarjetas de crédito. Por lo tanto, las economías donde el manejo de cheques, o tarjetas de crédito, es más común que el pago en efectivo presentan mayor riesgo de liquidación para los vendedores; aunque, por otro lado, los compradores no tienen que llevar consigo el dinero, lo cual es más práctico y menos peligroso.

En el sistema de pagos deben estar comprendidas las diferentes formas de pago, tratando de disminuir y controlar los riesgos propios de cada una.

En los países donde existe un banco central la liquidación de las operaciones más importantes, y por tanto la transferencia final del dinero, se hace con cuentas que tienen

los principales agentes económicos (generalmente los bancos) en dicho banco, por lo que este último debe poner especial atención en el riesgo del sistema de pagos. En nuestro país el banco central es Banco de México.

Hasta ahora, los créditos que tienen los bancos comerciales en sus cuentas con Banco de México son prácticamente ilimitados. Esto provoca que el riesgo de liquidación en que incurre el sistema de pagos mexicano sea absorbido por Banco de México, ya que si un banco no tiene fondos suficientes para saldar una deuda, Banco de México le prestará el dinero que haga falta.

Desde su autonomía, en abril de 1994, Banco de México inició un estudio del sistema de pagos mexicano, evaluando los riesgos en que incurre como banco central y ha decidido implantar algunas modificaciones tendientes a disminuir estos riesgos.

En particular, se ha propuesto modificar las reglas de operación del mercado de títulos del gobierno federal y deuda bancaria (*mercado de valores*), con políticas que disminuyan el riesgo de liquidación del mismo. Enseguida expondremos la forma en que se están implantando dichas políticas.

Así como Banco de México es el administrador de los títulos gubernamentales, el INDEVAL S. D.¹ es la institución encargada de administrar el papel bancario en nuestro país. Esta institución desarrolló un sistema denominado SIDV con el que se pueden transferir títulos de cualquier tipo entre instituciones financieras participantes (casas de bolsa y bancos), de forma que si dos de estas instituciones quieren hacer una operación de compra-venta de títulos gubernamentales o de papel bancario, el vendedor hace una solicitud al SIDV para que transfiera los títulos desde su cuenta hacia alguna cuenta del comprador.

Actualmente, el SIDV realiza la transferencia siempre que haya suficientes títulos en la

¹Instituto para el Depósito de Valores. Sociedad de Depósito.

cuenta del vendedor, si no es así, se retiene la solicitud de la operación para ser procesada en el transcurso del día. Por lo tanto, para cerrar una operación, la institución financiera que solicita vender algún tipo de título, debe comprobar que tiene los títulos en su poder. El SIDV también podría verificar que el comprador tenga suficientes fondos para liquidar la operación, sin embargo, esto no se hace debido a que las instituciones participantes tienen crédito ilimitado en el sistema. Este crédito también es respaldado por Banco de México.

Al final del día, los movimientos de efectivo y las solicitudes de transferencia en las cuentas de títulos gubernamentales se envían a Banco de México, donde se verifica que los participantes tengan suficientes títulos para realizar todas las operaciones solicitadas. Los traspasos de efectivo derivados de las transferencias realizadas, en papel bancario y títulos gubernamentales, no sufren ninguna verificación, pues cuando los fondos de algún participante no alcanzan para cubrir el importe de sus operaciones, Banco de México le presta el dinero que haga falta. Hasta ahora, los bancos que reciben estos préstamos no dan ninguna garantía de pago, por lo que existe la posibilidad de que no salden esta deuda. Así, Banco de México garantiza la liquidación de las operaciones en el mercado de valores a través de créditos ilimitados para sus participantes, corriendo el riesgo de que no le paguen. Esto define un sistema de pagos en el que existe la oportunidad, para los participantes en el mercado de valores, de comprar cantidades de títulos que tal vez no pueden pagar.

Dos de las reformas más importantes que se están haciendo en el sistema de pagos, particularmente en el mercado de títulos del gobierno federal y deuda bancaria, son:

- Limitar y controlar el crédito que extiende Banco de México a cada participante.
- Establecer una política de entrega contra pago para cerrar las operaciones del mercado.

En general, la *entrega contra pago* es una política con la que operan algunos sistemas de liquidación de valores, en la que el comprador recibirá sus valores si, y sólo si, hace el pago por los mismos.

En nuestro caso, esto quiere decir que cada operación se cerrará sólo si el participante que compra tiene el dinero para liquidar títulos que realmente tenga el participante que vende. De esta forma, si el comprador no cuenta con el dinero o el vendedor no posee los títulos, entonces la operación no puede cerrarse y se dice que *no pasa*.

La forma de operar en los sistemas que hacen entrega contra pago se puede dar de diversas formas. Hay sistemas en los que se liquidan las operaciones sólo si el vendedor tiene los títulos y el comprador el dinero. En otros sistemas, para agilizar la operación, los participantes cuentan con acuerdos de crédito, considerando que quien compra títulos los da en garantía. Es decir, si el comprador no tiene la liquidez suficiente para cerrar una operación, habrá alguien más que le preste el dinero, tomando como garantía de pago algunos títulos que aquel posea.

En el sistema de entrega contra pago que se propone para el mercado de títulos mexicano, Banco de México, como institución de banca central, continuará dando crédito (ahora limitado) a los participantes que lo requieran, pero tomará en garantía de pago algunos de sus títulos. Para lograr esto último, Banco de México tomará en garantía aquellos títulos cuyo valor cubra el importe del préstamo. Este tipo de crédito se denomina *crédito con garantía colateral*, o simplemente, colateralizado.

Los títulos de cada participante están repartidos en diferentes cuentas y una de éstas, llamada *cuenta de tenencias propias*, será la que se use para que Banco de México tome los títulos que considere necesarios para garantizar el pago de los préstamos.

La participación de Banco de México, otorgando créditos colateralizados, y la política de entrega contra pago dan las siguientes reglas de operación en el mercado de valores:

1. No se permiten sobregiros en las cuentas de títulos. Ningún participante podrá vender títulos que no tiene.
2. El sobregiro en efectivo que se permite a los participantes no será mayor a un límite que le asigne Banco de México previamente. En otras palabras, el crédito que se otorgue a cada participante estará limitado.
3. Los títulos que tiene cada participante, en sus cuentas de tenencias propias, podrán ser tomados por Banco de México como garantía de pago, por lo que no se permitirá que los sobregiros de efectivo de cada participante sean mayores al valor total de los títulos que posee en dichas cuentas.

Estas reglas de operación son, en realidad, restricciones a las operaciones que se pueden liquidar en el mercado, ya que para que se lleve a cabo el cierre de una operación, el participante que vende debe tener los títulos en sus cuentas (pues no se puede sobregirar) y el participante que compra debe tener el efectivo, o en su defecto, línea de crédito y suficientes títulos en sus cuentas propias para cubrir el monto correspondiente. Por tanto, para liquidar una operación, es necesario hacer diferentes verificaciones previas, lo cual podría retrasar, e inclusive detener, el *modus operandi* actual.

Además de estas reglas, Banco de México propone que cualquier operación del mercado pueda liquidarse *parcialmente*. Esto es, si en alguna operación el vendedor no tiene todos los títulos que solicita transferir o el comprador sólo puede pagar una cantidad menor que la solicitada, dicha operación podrá cerrarse por la cantidad que ambos puedan cubrir. En este caso, se hace la liquidación de esa cantidad y se elabora una nueva solicitud por los títulos que no se vendan. Hasta ahora, la posibilidad de liquidar las operaciones de manera parcial sólo existía para operaciones con títulos gubernamentales; se propone que se haga lo mismo con operaciones de papel bancario. Cuando pueden liquidarse todos los títulos solicitados en una operación se dice que ésta *pasa totalmente*; si sólo se puede

liquidar una cantidad menor se dice que *pasa parcialmente*.

En seguida se presenta una metodología, basada en un modelo de programación lineal, cuyo objetivo es determinar las operaciones que se pueden liquidar bajo las reglas expuestas, maximizando el valor total de las que pasen, cuidando que se mantenga la agilidad de la operación actual y otorgando montos de crédito razonables por parte de Banco de México.

La principal característica de la metodología que se presenta es que no se atenderá la solicitud de operaciones una por una, sino que se pondrán todas en una cola de espera mientras se decide qué operaciones se pueden liquidar. A lo largo del ejercicio se irá formando la cola con las operaciones que esperan ser atendidas. En determinado momento se hará un corte, es decir, se detendrá el registro de nuevas solicitudes y un sistema computacional, mediante un modelo de programación lineal, determinará óptimamente cuáles de las que están en cola es conveniente cerrar. Las operaciones elegidas se liquidarán todas a la vez. En los casos en que la solución óptima sea pasar parcialmente alguna operación, ésta se cerrará por la cantidad encontrada y los títulos restantes constituirán una nueva solicitud que volverá a la cola de espera, junto con las operaciones que no puedan cerrarse. A esta cola se irán incorporando las nuevas operaciones hasta el siguiente corte, en el cual el procedimiento será repetido. Los cortes en el registro de solicitudes se harán tantas veces al día como se considere necesario.

Capítulo 3

Antecedentes matemáticos

En la vida cotidiana se nos presentan problemas que tenemos que resolver o decisiones que debemos tomar. La forma en que resolvemos estos problemas depende de la naturaleza de los mismos y de nuestras características propias. Así, a veces tomamos decisiones de acuerdo con nuestra experiencia, nuestros conocimientos, etc.

En algunos casos es posible utilizar técnicas matemáticas para resolver problemas reales o para tomar decisiones. Para lograr esto, primero debemos observar que cualquier situación, por sencilla que parezca, generalmente está inmersa en un sistema más complejo. Cuando se presenta un problema, éste se relaciona con un conjunto de variables del sistema que lo afectan directa o indirectamente, así que antes de tomar una decisión hay que considerar muchos factores del sistema. Si los factores, o variables, que intervienen en el problema son cuantificables, es muy probable que la utilización de las técnicas matemáticas apropiadas nos permita resolver dicho problema, o bien, tomar una decisión acertada.

Para aplicar dichas técnicas, el primer paso es hacer una *modelación matemática* del sistema en estudio. Esta modelación consiste en traducir a símbolos matemáticos las variables del sistema que influyen en el problema que queremos atacar, así como las principales relaciones que guardan estas variables entre sí. A veces, para decidir cuál es la mejor solución al problema, o la mejor decisión, el objetivo buscado se expresa en términos matemáticos, formando con esto una *función objetivo*. El conjunto de símbolos, sus relaciones y la función objetivo recibe el nombre de *modelo matemático*. Este modelo deberá representar, en lo fundamental, el sistema en estudio; de forma tal que sus soluciones sean fácilmente traducidas a soluciones del problema real.

Los modelos matemáticos se pueden clasificar en: *determinísticos* y *estocásticos*. Los primeros se caracterizan por que pueden predecirse con certeza los resultados obtenidos de cada acción que se tome. En los segundos, los resultados derivados de cada acción siempre son inciertos.

A su vez, los modelos determinísticos pueden dividirse en *lineales* y *no lineales*. En los modelos del primer tipo, tanto la función objetivo como las relaciones que hay entre las variables del modelo se pueden expresar con ecuaciones lineales.

En un modelo lineal, es posible que las ecuaciones que lo forman tengan ciertas características entre sí. Esto otorga al modelo una estructura particular, en cuyo caso hay que estudiarlo de manera independiente, creando un nuevo tipo de modelos matemáticos lineales: los modelos con estructuras especiales.

A lo largo de este trabajo trabajaremos con modelos matemáticos lineales con estructuras especiales.

En el diagrama de la figura 3.1 presentamos una clasificación de los modelos matemáticos propuesta por G. Dantzig [4].

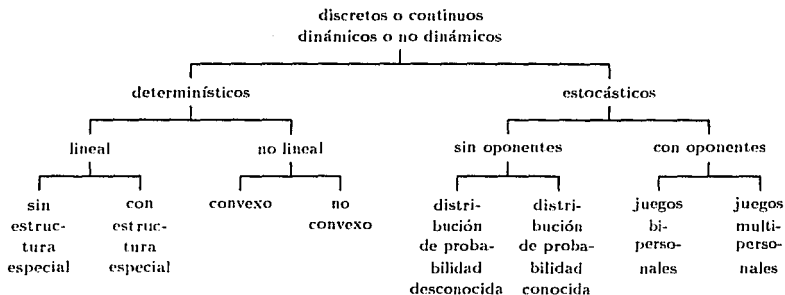


Figura 3.1: Clasificación de modelos matemáticos.

Una vez construido un modelo matemático que refleje el sistema real a nuestra satisfacción, es indispensable aplicar metodologías especiales para resolverlo.

Cuando el problema que queremos resolver se refiere a la conducción y coordinación de operaciones o actividades de una organización, o a la toma de alguna decisión dentro de la misma, las técnicas utilizadas para construir los modelos matemáticos correspondientes y los algoritmos de resolución de estos modelos, se estudian dentro de un campo denominado *Investigación de Operaciones*.

3.1 Investigación de Operaciones

Una de las definiciones más aceptadas de Investigación de Operaciones es la siguiente:

“La Investigación de Operaciones es la aplicación del método científico, por grupos interdisciplinarios, a problemas que comprenden el control de sistemas organizados hombre-máquina, para dar soluciones que sirvan mejor a los propósitos de la organización como un todo” [1]

Para aplicar el método científico mencionado en esta definición se requiere, precisamente, de un modelo matemático. Por lo que podemos decir que *“la Investigación de Operaciones es la aplicación de la metodología científica a través de modelos, primero para representar al problema real que se quiere resolver en un sistema y segundo para resolverlo”* [8]. Además, siempre que es posible, la investigación de operaciones busca la mejor solución del problema en estudio, por medio de una o varias funciones objetivo. En consecuencia, la investigación de operaciones se puede definir como la aplicación de modelos y métodos matemáticos para ayudar a la utilización óptima de los recursos de un sistema.

Dependiendo del problema que se quiera resolver, la investigación de operaciones aplica

técnicas diferentes en la construcción y resolución del modelo. Estas técnicas se pueden dividir en: simulación, programación lineal, programación no lineal, programación dinámica, programación heurística, teoría de inventarios, teoría de colas, análisis estadístico y técnica de revisión y evaluación de proyectos (PERT). El análisis estadístico comprende la teoría de la probabilidad, análisis de regresión, muestreo y pruebas de hipótesis.

En el presente trabajo hacemos uso de técnicas especiales de programación lineal para resolver un problema financiero real.

En resumen, la contribución del enfoque de investigación de operaciones proviene principalmente de [6]:

- 1. La estructuración de una situación de la vida real como un modelo matemático, con lo que se logra la abstracción de los elementos esenciales para que pueda buscarse una solución que concuerde con los objetivos del tomador de decisiones. Esto implica tomar en cuenta el problema dentro del contexto del sistema completo.*
- 2. El análisis de la estructura de tales soluciones y el desarrollo de procedimientos sistemáticos para obtenerlas.*
- 3. El desarrollo de una solución, incluyendo la teoría matemática, si es necesario, que lleve al valor óptimo de la medida de lo que se espera del sistema (o quizá, que compare los cursos de acción alternativos, evaluando esta medida para cada uno).*

En la práctica, la instrumentación de un proyecto de investigación de operaciones en la solución de un problema real en una organización incrementa la posibilidad de tomar mejores decisiones, ayuda a coordinar eficazmente las múltiples componentes de la organización, permite mayor control y, en general, puede mejorar el sistema completo.

3.2 Historia de la Investigación de Operaciones

Podemos pensar que los orígenes de la investigación de operaciones se remontan a mediados del siglo antepasado, cuando el economista Quesnay empieza a utilizar modelos matemáticos para expresar sistemas económicos en términos matemáticos. En 1874, el también economista Walras hace uso de técnicas similares para construir modelos más complejos. Jordan, Minkowski y Farkas empiezan a desarrollar teorías para los modelos lineales alrededor de 1900. Los modelos dinámicos probabilísticos tienen su origen con Markov a finales del siglo pasado. Los modelos de colas se originan con Erlang, a principios del siglo XX. El desarrollo de los modelos de inventarios se lleva a cabo en los años veinte, y al mismo tiempo, los húngaros König y Egerváry estudian los problemas de asignación, usando modelos matemáticos. En la década de los treinta, Von Neuman ciñenta lo que se conoce como Teoría de Juegos. Sin embargo, es hasta la Segunda Guerra Mundial cuando realmente se inicia el desarrollo de la investigación de operaciones como tal.

La escasez de recursos materiales y económicos que se presentó durante la guerra orilló a los gobiernos de los diferentes países involucrados a buscar la mejor manera de asignar estos pocos recursos a sus diferentes campañas. Para lograr esto, primero el gobierno británico y después el de los Estados Unidos, recurrieron a un grupo de investigadores, quienes aplicaron diferentes procedimientos científicos para solucionar problemas de tipo táctico y estratégico. De hecho, le pidieron a los investigadores que estudiaran las operaciones militares. Estos equipos de científicos fueron los primeros de investigación de operaciones y, según se afirma, contribuyeron a ganar importantes batallas, como la Batalla de Inglaterra y la Batalla del Atlántico Norte.

Al finalizar la guerra, algunos de estos científicos continuaron desarrollando la investigación de operaciones, logrando avances importantes. Por ejemplo, la fuerza aérea

norteamericana, a través de su centro de investigación Rand Corporation, comisionó a un grupo de matemáticos para resolver problemas de distribución de recursos militares entre los aliados que se encontraban dispersos por todo el mundo. Como resultado, el doctor George Dantzig presentó, a finales del verano de 1947 el método *Simplex*, dando inicio a la programación lineal. Antes de que finalizara la década de los 50, muchas de las herramientas de la investigación de operaciones ya se habían desarrollado relativamente bien. Tal es el caso de la programación lineal, programación dinámica, teoría de colas y teoría de inventarios.

Debido a la explosión industrial de la posguerra, los problemas originados por la complejidad y especialización en las organizaciones aumentaron significativamente. En la búsqueda de las soluciones de sus problemas, las empresas recurrieron nuevamente a los científicos, descubriendo que, en realidad, los problemas a los que se enfrentaban eran básicamente los mismos que habían resuelto los militares, pero en diferentes contextos. Gracias a la experiencia adquirida y a los logros obtenidos durante la guerra, la investigación de operaciones se filtró, poco a poco, en la industria, los negocios y el gobierno civil. En la actualidad, las doce corporaciones más grandes del mundo, y un gran número de las organizaciones industriales pequeñas, tienen equipos de investigación de operaciones bien establecidos. Muchas industrias, incluyendo la de aviación y proyectiles teleguiados, la automovilística, la de comunicaciones, la de computadoras, la de energía eléctrica, la electrónica, la alimentaria, la metalúrgica, la minera, la de papel, la de petróleo y la de transporte han hecho un amplio uso de la investigación de operaciones. Incluso instituciones financieras, oficinas gubernamentales y hospitales han incrementado el uso de tal ciencia.

Además del rápido avance en el desarrollo de la teoría de la investigación de operaciones, cabe mencionar otro factor que ha dado gran ímpetu al crecimiento del campo: la embestida de la revolución de las computadoras.

Como hemos mencionado, en tan sólo dos décadas gran parte de las técnicas de la investigación de operaciones presentaron avances teóricos importantes. Estos avances permiten, en teoría, plantear y solucionar una gran variedad de modelos matemáticos. Sin embargo, la cantidad de cálculos que se requieren para resolver los modelos es, generalmente, muy grande y requiere del manejo de muchos datos a la vez. Por esto, los avances tecnológicos presentados en el campo computacional, son los que realmente han hecho posible que la investigación de operaciones pueda aplicarse en la vida real. Sin estos avances, a nivel hardware y software, los alcances de la investigación de operaciones serían muy limitados.

3.3 La investigación de operaciones actualmente

Existen varias asociaciones mundiales que agrupan a miembros que se dedican a la investigación de operaciones en todo el mundo. En Estados Unidos, la Operations Research Society of America (ORSA), establecida en 1952, y The Institute of Management Sciences (TIMS), fundado en 1953, cuentan con más de 6000 miembros cada una. ORSA publica, bimestralmente, la revista *Operations Research* y TIMS, *Management Science*, cada mes. Además, las dos sociedades publican conjuntamente *Interfaces* y *Mathematics of Operations Research*. Estas cuatro revistas contienen más de 3000 páginas al año que informan sobre nuevas investigaciones y aplicaciones en el campo. También existen asociaciones Canadienses, Europeas, Latinoamericanas y Asiáticas, que editan más de dos docenas de revistas mensualmente, con temas relacionados a los tópicos de investigación de operaciones. De hecho, hay más de 33 países miembros en la International Federation of Operational Research Societies (IFORS), en los que cada país cuenta con una sociedad nacional dedicada a este campo [6].

En México, la investigación de operaciones se utiliza dentro del sector de servicios

públicos, entre otros en el Banco de México, la Secretaría de Comunicaciones y Transportes, la Comisión Federal de Electricidad, Petróleos Mexicanos, etc. Además, las empresas privadas más grandes de nuestro país se han servido de las principales herramientas de la investigación de operaciones, logrando mejorar la utilización de sus recursos en diferentes contextos. Tal es el caso de la industria automotriz, la metal mecánica, la textil, la de bebidas embotelladas, la de transportación aérea, etc.

Sin embargo, debemos mencionar que, como sucede en muchos otros campos, el desarrollo de la investigación de operaciones en México aún es muy pobre. Existen distintos problemas, en el gobierno y en la industria, que podrían resolverse con la aplicación de técnicas de investigación de operaciones. El camino por andar es largo, pero confiamos en que los resultados obtenidos, dentro y fuera del país, alienten el uso de estas técnicas.

La investigación de operaciones también ha tenido un impacto considerable en las universidades. Hoy en día se han diseñado programas profesionales, y de posgrado, en la especialidad de investigación de operaciones en universidades de todo el mundo. Es precisamente en las universidades donde se desarrolla gran parte de la investigación básica en este campo.

También existen aplicaciones importantes de la investigación de operaciones en el ámbito financiero a nivel mundial. En el apéndice A listamos algunos artículos referentes a este tipo de aplicaciones, publicados en diferentes partes del mundo entre el mes de enero de 1990 y diciembre de 1996. Hemos agrupado los artículos de acuerdo a la revista en la que se publicaron. Las revistas consultadas fueron: *Operations Research*, *Management Science*, *Journal of the Operational Research Society*, e *Interfaces*. Los artículos de cada revista están ordenados por año de publicación. En los últimos puntos del mismo apéndice incluimos la nota bibliográfica de algunas conferencias impartidas en Alemania, dentro del *Symposium de Investigación de Operaciones de 1996*, que presentan aplicaciones de la investigación de operaciones al ámbito financiero.

Capítulo 4

Modelación del problema

Como mencionamos en capítulos anteriores, el problema que abordamos en esta tesis consiste, básicamente, en determinar, de manera óptima, las operaciones del mercado de títulos que pueden liquidarse durante el día, cumpliendo con ciertas reglas. Para atacar este problema, utilizaremos técnicas de Investigación de Operaciones. En esta sección planteamos un modelo de programación matemática que ayuda a determinar cuáles operaciones es conveniente liquidar en cada corte.

Recordemos que para liquidar una operación hay que estar seguros de que:

1. No se incurra en sobregiros en las cuentas de títulos, por parte del vendedor.
2. El saldo de efectivo del comprador sea mayor que un límite de sobregiro previamente definido por Banco de México.
3. Cuando el saldo en efectivo de un participante sea negativo, el valor absoluto de este saldo deberá ser menor que el valor total de los títulos que tenga el participante en sus cuentas de tenencias propias.

Para plantear un modelo que refleje apropiadamente la manera de operar en el mercado y que, por tanto, permita encontrar el conjunto óptimo de operaciones que se pueden cerrar en el mismo, también se debe considerar que:

- En el mercado pueden hacerse operaciones de compra-venta de cualquier tipo de títulos, por cualquier cantidad, a cualquier precio y por cualesquiera dos participantes.
- Las operaciones pueden cerrarse por la cantidad total solicitada o por una cantidad menor. Es decir, pueden *pasar* de manera *total* o *parcial*.
- No todos los títulos se pueden tomar en garantía. Cada participante tiene depósitos de sus títulos en diferentes tipos de cuentas: cuentas propias y cuentas de terceros.

De las primeras se pueden obtener los títulos que se necesiten como garantía de pago, de las segundas no.

- Un participante puede hacer una operación para depositar o retirar títulos desde y/o hacia cualquiera de sus cuentas.
- Llamaremos *posición* de un participante al valor de todos los títulos que se le pueden tomar como garantía, es decir, al valor total de los títulos que tiene en sus cuentas propias.
- El valor de cada título lo determina Banco de México, diariamente, este valor es único y es independiente del precio al que se acuerde comprar o vender el título en cada operación. Al proceso de asignación del valor al título, lo llamaremos *valuación* del título, y al valor obtenido lo llamaremos *valor como garantía* del título.
- No hay una única operación de compra o de venta entre dos instituciones. Puede hacerse más de una operación entre dos participantes, del mismo tipo de título, con las mismas cuentas y por las mismas cantidades, y cada operación debe ser tratada de modo independiente.
- El saldo en efectivo de cualquier participante puede ser negativo. En ese caso, se dice que el participante incurre en un *sobregiro* de su cuenta de efectivo. El monto del sobregiro alcanzado es igual al valor absoluto del saldo en efectivo, cuando este último es negativo.

4.1 Notación

Sean:

I : número total de instituciones participantes. El conjunto de instituciones participantes estará indizado por i, j , es decir, $1 \leq i \leq I; 1 \leq j \leq I$.

T : cantidad de los diferentes tipos de títulos que se operaron. El conjunto de tipos de títulos estará indizado por t , es decir, $1 \leq t \leq T$.

u, v : tipo de cuenta = $\begin{cases} 1 & \text{cuenta de tenencias propias} \\ 2 & \text{cuenta de terceros} \end{cases}$

$K(i, j, t, u, v)$: conjunto de todas las operaciones en las que el participante i vende títulos tipo t de su cuenta u a la cuenta v del participante j . Este conjunto estará indizado por k , quien nos permite identificar cada operación de manera única.

$K(i, j, t, u, v)$: número total de operaciones en las que el participante i vende títulos tipo t de su cuenta u a la cuenta v del participante j . $K(i, j, t, u, v)$ es el número de elementos $K(i, j, t, u, v)$ y es, obviamente, un entero positivo. ($0 \leq k \leq K(i, j, t, u, v)$)

V_t : valor como garantía de los títulos tipo t . Este será un número real determinado por Banco de México, con un valor próximo al precio comercial de los títulos.

T_{it} : saldo inicial del participante i en su cuenta u de títulos tipo t . Es el número de títulos que hay en una cuenta, por lo que es una cantidad entera no negativa.

E_i : saldo inicial de efectivo del participante i . Es un número real con dos decimales.

P_i : posición inicial del participante i . Evaluación inicial de valor (en tipos de títulos que están en las cuentas propias del participante i , es decir, $P_i = \sum_{t=1}^T T_{it} \cdot V_t$). La posición de un participante es un número real positivo con dos decimales.

\hat{P}_i : posición final del participante i . Es un número real no negativo con dos decimales.

\mathcal{L}_i : límite de crédito, en efectivo, permitido al participante i . \mathcal{L}_i es un número real no negativo con dos decimales. Observa que $-\mathcal{L}_i$ es el límite de sobregiro del participante i , es decir, el número negativo más pequeño que puede valer el saldo en efectivo de i .

\mathcal{C}_{ijkluv} : cantidad de títulos tipo l que el participante i pretende transferir de su cuenta u a la cuenta v del participante j en la transacción k . Es un número entero positivo.

\mathcal{P}_{ijkluv} : precio al que el participante i desea vender títulos tipo l de su cuenta u a la cuenta v del participante j en la transacción k . El precio estará expresado en pesos y por tanto, será un número real no negativo con dos decimales.

.

Variables:

x_{ijkluv} : número de títulos l que el participante i puede vender desde su cuenta u a la cuenta v del participante j en la transacción k .

4.2 Construcción

El objetivo del modelo de programación lineal entera que plantearemos es maximizar el valor total de las operaciones que se cierren, esto es, si el valor como garantía del título l es \mathcal{V}_l pesos, el valor de cada operación se obtiene multiplicando el número de títulos que se pueden vender (x_{ijkluv}) por el valor de los mismos. Por tanto, la función objetivo del modelo es:

$$(\max) Z = \sum_{i=1}^I \sum_{j=1}^I \sum_{l=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^{K(i,j,l,u,v)} (\mathcal{V}_l x_{ijkluv})$$

Necesitamos un primer conjunto de restricciones que evite sobregiros en el saldo de títulos de cada participante, en cada una de sus cuentas. Hay que asegurar que el saldo

en títulos de cada cuenta con la que se pretenda hacer una operación sea mayor o igual a cero. Es decir, la cantidad de títulos que tiene una cuenta inicialmente más el número de éstos que entran a la misma, debe ser mayor o igual a la cantidad de títulos que sale. Usando la notación de la sección 4.1, toda cuenta u del participante i en títulos l debe cumplir con:

$$(1) \quad \bar{T}_{itu} + \sum_{j=1}^I \sum_{v=1}^2 \sum_{k=1}^2 K_{(j,i,t,v,u)} x_{jiktuv} - \sum_{j=1}^I \sum_{v=1}^2 \sum_{k=1}^2 K_{(i,j,t,u,v)} x_{ijktuv} \geq 0$$

Enseguida, para cada participante i plantearemos una restricción que impide que su saldo final de efectivo sea menor que el límite de sobregiro fijado por Banco de México. El saldo inicial de efectivo más el dinero que entra por una operación de venta menos el dinero que sale por concepto de compras da el saldo final de efectivo del participante, por lo que la restricción es:

$$(2) \quad \bar{E}_i + \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K_{(j,i,t,u,v)} (\mathcal{P}_{jiktuv} x_{jiktuv}) \\ - \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K_{(j,i,t,u,v)} (\mathcal{P}_{jiktuv} x_{jiktuv}) \geq -\mathcal{L}_i$$

La posición de los participantes se calcula valuando los títulos que tiene cada uno de ellos en sus cuentas propias. Entonces, para calcular la posición final del participante i se suma su posición inicial más el valor de los títulos que entran a alguna de sus cuentas propias y de esto se resta el valor de los títulos que hayan salido de cualquiera de estas cuentas.

$$(3) \quad \hat{P}_i = \bar{P}_i + \sum_{j=1}^I \sum_{t=1}^T \sum_{v=1}^2 \sum_{k=1}^2 K_{(j,i,t,v,1)} (\mathcal{V}_t x_{jikt(v+1)}) - \sum_{j=1}^I \sum_{t=1}^T \sum_{v=1}^2 \sum_{k=1}^2 K_{(i,j,t,1,v)} (\mathcal{V}_t x_{ijkt(v)})$$

Esta última ecuación no es, propiamente, una restricción del modelo. La presentamos únicamente para simplificar la exposición del mismo. En los programas lineales que se

presenten, se seguirá escribiendo esta ecuación; sin embargo, en la implantación computacional de la metodología usada se hace la sustitución de \hat{P}_i y se simplifican las ecuaciones correspondientes.

Si algún participante termina el ejercicio con saldo de efectivo negativo, el sobregiro alcanzado podría ser cubierto por su posición final a manera de garantía. Por esto, el saldo de efectivo de cada participante deberá estar respaldado por su posición final, es decir, si hay sobregiro, su monto debe ser menor que \hat{P}_i . Así, para cada participante i tenemos:

$$(4) \quad \bar{E}_i + \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^{K(i,j,t,u,v)} (P_{ijktuv} x_{ijktuv}) \\ - \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^{K(j,i,t,u,v)} (P_{jiktuv} x_{jiktuv}) \geq -\hat{P}_i$$

Para que el modelo encuentre soluciones congruentes con la operación real es importante que el valor que puedan tomar las variables de decisión esté acotado por la cantidad original de títulos que se solicita transferir. En otras palabras, la cantidad de títulos que se podrá cerrar en cada operación estará restringida por:

$$(5) \quad C_{ijktuv} \geq x_{ijktuv} \geq 0$$

Por último, hay que recordar que las variables de decisión del problema corresponden a la cantidad de títulos que se pueden liquidar en cada operación. Por tanto, cada variable tiene la restricción:

$$(6) \quad x_{ijktuv} \text{ entera}$$

Tomando la función objetivo, reacomodando las ecuaciones (1)-(6) y agrupándolas, se plantea el modelo de programación lineal entera, P_0 , de la figura 4.1.

$$\begin{aligned}
 (\max) Z &= \sum_{i=1}^I \sum_{j=1}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,u,v) (V_t x_{ijktuv}) \\
 \text{s.a.} \\
 (1) \quad \bar{T}_{stu} &- \sum_{j=1}^I \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,u,v) x_{ijktuv} + \sum_{j=1}^I \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,v,u) x_{jiktuv} \geq 0 \quad \forall i, t, u \\
 (2) \quad \bar{E}_i + \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,u,v) (P_{ijktuv} x_{ijktuv}) &- \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,u,v) (P_{jiktuv} x_{jiktuv}) \geq -\mathcal{L}_i \quad \forall i \\
 P_0 = (3) \quad \bar{P}_i &- \sum_{j=1}^I \sum_{t=1}^T \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,1,v) (V_t x_{ijkt1v}) + \sum_{j=1}^I \sum_{t=1}^T \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,v,1) (V_t x_{jikt1v}) = \hat{P}_i \quad \forall i \\
 (4) \quad \bar{E}_i + \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,u,v) (P_{ijktuv} x_{ijktuv}) &- \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,u,v) (P_{jiktuv} x_{jiktuv}) \geq -\hat{P}_i \quad \forall i \\
 (5) & C_{ijktuv} \geq x_{ijktuv} \geq 0 \quad \forall i, j, k, t, u, v \\
 (6) & x_{ijktuv} \text{ entera} \quad \forall i, j, k, t, u, v
 \end{aligned}$$

Figura 4.1: Modelo para optimizar la liquidación

Si, al resolver P_0 , el valor óptimo de alguna variable es igual a su cota superior, la operación correspondiente pasa de manera total. Si dicho valor está entre cero y su cota superior, la operación pasa parcialmente. Por último, si el valor óptimo es cero, la operación no pasa.

Debemos observar que cuando el saldo en efectivo inicial de cada participante (\bar{E}_i) es mayor o igual que su sobregiro máximo permitido ($-\mathcal{L}_i$), y el valor de los títulos que tiene en sus cuentas propias (\bar{P}_i) cubre los sobregiros que haya alcanzado (o sea, $\bar{P}_i > -\bar{E}_i$), entonces una solución factible de P_0 es hacer todas las variables iguales a cero, o dicho de otra manera, P_0 tiene al vector de coordenadas iguales a cero como solución factible.

Capítulo 5

Metodología

Hasta ahora, hemos presentado un programa lineal entero, P_0 , que modela las operaciones en el mercado de títulos mexicano y nos permite decidir óptimamente cuáles de ellas pueden liquidarse sin violar las reglas de operación. Sin embargo, es muy importante que el tiempo invertido en tomar esta decisión sea suficientemente pequeño, para que la utilización del modelo matemático no entorpezca el *modus operandi* del mercado. En este capítulo, presentamos una metodología que nos lleva a resolver satisfactoriamente P_0 .

Cada variable de decisión de P_0 corresponde a una operación del mercado de valores. Actualmente, el número diario de operaciones registradas en dicho mercado es de varios cientos y se espera que esta cantidad aumente considerablemente, por lo que el modelo podría tener miles de variables. Debido a la restricción de integridad y al número de las variables involucradas, ningún algoritmo conocido nos permite encontrar una solución óptima de P_0 con recursos de cómputo razonables o en tiempos suficientemente cortos.

Las dificultades mencionadas en el párrafo anterior no presentan ningún obstáculo en realidad, ya que para el mercado de valores es satisfactoria cualquier solución entera que respete las reglas de operación y que tienda al óptimo teórico, por lo que podemos utilizar métodos que encuentren soluciones aproximadas al óptimo del modelo entero, a través de modelos que no tengan restricciones de integridad (normalmente llamados *relajados*).

Partiendo de que el modelo entero presentado, P_0 , es acotado y básicamente representa un problema de flujos (de títulos y de efectivo) con restricciones adicionales, se desarrolló un procedimiento basado en la hipótesis de que el punto óptimo de P_0 está muy cerca del punto óptimo de un problema similar sin restricciones de integridad, o mejor dicho, se espera que el óptimo del problema de programación entera esté en una vecindad cercana al punto óptimo del problema relajado. Por tanto, si encontramos el óptimo del problema relajado sólo hay que buscar una solución entera, apropiada, cercana a dicho punto.

Considérese P_0 sin la restricción (6), es decir, P_0 relajado. Como este programa se

puede resolver eficientemente y su solución puede no ser entera debido a las restricciones (2)-(4) de P_0 , se truncarán los valores óptimos encontrados, se verificará que el truncamiento no provoque sobregiros en las cuentas de títulos y, de ser necesario, Banco de México otorgará a los participantes el crédito adicional (llamado aquí *crédito especial*) que fuese necesario para lograr la factibilidad de P_0 .

El crédito especial a otorgar debe estar controlado para que no aumente indefinidamente. Esto es, al truncar y obtener una solución entera, las restricciones que pueden violarse son las de efectivo o las de posición, y en ese caso, se dará un poco más de crédito a cada participante que haya sobrepasado su límite. Sin embargo, como dicho crédito sólo sirve para lograr factibilidad, la intención no es dar crédito especial siempre; por lo que al inicio de cada corte se tratará de disminuir el crédito total otorgado hasta el momento.

Por otro lado, P_0 relajado es un programa lineal cuya matriz de restricciones tiene una estructura especial de la cual nos valdremos para abatir los tiempos de resolución.

Como cada variable de decisión de P_0 corresponde a una operación del mercado y en cada una de éstas sólo se opera un tipo de título, agruparemos las variables de acuerdo al tipo t de título que involucran, generando subvectores de decisión X_t .

Con las restricciones (1) de P_0 se controla la cantidad de títulos que entra y sale de cada cuenta, es decir, su flujo. Los participantes tienen cuentas únicas para cada tipo de título, consecuentemente, las restricciones de un tipo de título son independientes de las de cualquier otro. Debido a esto, las restricciones (1) son independientes de las demás y podemos agruparlas, por cuenta u , participante i y tipo t de título, generando submatrices de restricciones que llamaremos a_{it} .

Las restricciones (2)-(4) de P_0 se plantean para cada participante, incorporando todos los movimientos de efectivo y posición originados en cualquier operación en que él intervinga. De esta forma, si agrupamos las variables por tipo t de título como habíamos dicho,

la matriz de restricciones correspondiente a (2)-(4) se verá particionada en submatrices A_i .

Por lo anterior, P_0 tiene la siguiente estructura:

$$\begin{aligned} \text{Min } Z &= C_1 X_1 + C_2 X_2 + \dots + C_T X_T \\ \text{s.a.} \\ (1) \quad & a_1 X_1 && \geq b_1 \\ (1) \quad & & a_2 X_2 && \geq b_2 \\ & & & \ddots & \\ (1) \quad & & & & a_T X_T \geq b_T \\ \\ (2)-(4) \quad & A_1 X_1 + A_2 X_2 + \dots + A_T X_T \geq b \end{aligned}$$

Donde, para cada tipo t de título, $t = 1, \dots, T$: X_t es el vector de variables de decisión; C_t es el vector de coeficientes en la función objetivo del vector X_t ; a_t es la submatriz de restricciones del flujo en títulos, por cuenta u y participante i ; b_t es el lado derecho de la submatriz a_t ; A_t es la submatriz correspondiente a las restricciones (2)-(4) de P_0 ; b es el vector de lados derechos de las restricciones de efectivo y posición.

En el siguiente capítulo expondremos un algoritmo (llamado *algoritmo de descomposición de Dantzig-Wolfe*) que hace uso de esta estructura para economizar recursos computacionales y reducir los tiempos invertidos en resolver este tipo de programas.

En resumen, el procedimiento para decidir qué operaciones se deben liquidar en cada corte consiste en los siguientes pasos:

1. Lectura de datos.
2. Reducción del crédito especial.

3. Optimización de las operaciones a liquidar.
4. Solución óptima.
5. Solución entera.
6. Formulación del problema de flujos.
7. Resolución del problema de flujos.
8. Determinación de créditos especiales.
9. Liquidación.

5.1 Lectura de datos

Se hace un corte en el sistema que captura la solicitud de operaciones en el mercado de títulos, o sea, se detiene la solicitud de nuevas operaciones.

Se lee y se verifica la información relevante del problema, es decir, las operaciones pendientes en la cola de espera, los saldos iniciales en títulos (\bar{T}_{iu}), los saldos iniciales en efectivo (\bar{E}_i), los límites de crédito (\mathcal{L}_i) y la valuación de la cartera de cada participante (\bar{P}_i).

Cada operación pendiente en la cola de espera se registra con: identificación de la institución vendedora (i) y de la compradora (j), cuenta desde la que se vende (u), cuenta en la que se compra (v), tipo de título (t), cantidad (\mathcal{C}_{ijktuv}), precio (\mathcal{P}_{ijktuv}) y folio (k).

La valuación \mathcal{V}_i de cada tipo t de títulos es conocida.

5.2 Reducción del crédito especial

Si algún participante se sobregiró en el corte anterior, es posible que en el nuevo corte tenga pendientes algunas operaciones de venta, con lo que aumentaría su saldo de efectivo y ya no usaría crédito especial. Por tanto, es importante determinar si pasando algunas operaciones pendientes se puede disminuir el monto total del crédito especial utilizado. En este paso, se usan los datos leídos para formular el problema P_1 de programación lineal entera-mixta (ver figura 5.1). Las restricciones de este programa son análogas a las de P_0 y la función objetivo busca disminuir el monto del crédito especial otorgado hasta el presente corte.

Nuevamente, las dimensiones de P_1 no nos permiten resolver este programa directamente, por lo que descartaremos las restricciones de integridad (6); obteniendo el problema relajado P'_1 (ver figura 5.2).

Se resuelve P'_1 . Como las variables y_i son no negativas, los argumentos usados para mostrar la factibilidad de P_0 son válidos para ver que P'_1 tiene al *cero* como solución factible.

Como la estructura de P'_1 es similar a la de P_0 , podemos aplicar el algoritmo de descomposición de Dantzig-Wolfe para resolver P'_1 de manera eficiente.

$$\begin{aligned}
 (\min) Y &= \sum_{i=1}^I y_i \\
 \text{s.a.} \\
 (1) \quad \bar{T}_{itu} &- \sum_{j=1}^I \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,u,v) x_{ijktuv} + \sum_{j=1}^I \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,u,v) x_{jiktuv} \geq 0 \quad \forall i, t, u \\
 (2') \quad \bar{E}_i &+ \sum_{j \neq i} \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,u,v) (P_{ijktuv} x_{ijktuv}) - \sum_{j \neq i} \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,u,v) (P_{jiktuv} x_{jiktuv}) \geq -\mathcal{L}_i - y_i \quad \forall i \\
 P_i = (3) \quad \bar{P}_i &- \sum_{j=1}^I \sum_{t=1}^T \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,t,v) (V_i x_{ijktv}) + \sum_{j=1}^I \sum_{t=1}^T \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,t,v) (V_i x_{jiktv}) = \hat{P}_i \quad \forall i \\
 (4') \quad \bar{E}_i &+ \sum_{j=1}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,u,v) (P_{ijktuv} x_{ijktuv}) - \sum_{j \neq i} \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,u,v) (P_{jiktuv} x_{jiktuv}) \geq -\hat{P}_i - y_i \quad \forall i \\
 (5) \quad &C_{ijktuv} \geq x_{ijktuv} \geq 0 \quad \forall i, j, k, t, u, v \\
 (6) \quad &x_{ijktuv} \text{ entera} \quad \forall i, j, k, t, u, v \\
 (7) \quad &y_i \geq 0 \quad \forall i
 \end{aligned}$$

Figura 5.1: Reducción del crédito especial (problema entero)

$$\begin{aligned}
 (\min) Y &= \sum_{i=1}^I y_i \\
 \text{s.a.} \\
 (1) \quad \bar{T}_{itu} &- \sum_{j=1}^I \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,u,v) x_{ijktuv} + \sum_{j=1}^I \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,v,u) x_{jiktuv} \geq 0 \quad \forall i, t, u \\
 (2') \quad \bar{E}_i &+ \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,u,v) (P_{ijktuv} x_{ijktuv}) - \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,v,u) (P_{jiktuv} x_{jiktuv}) \geq -\mathcal{L}_i - y_i \quad \forall i \\
 (3) \quad \bar{P}_i &- \sum_{j=1}^I \sum_{t=1}^T \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,t,v) (V_i x_{ijktuv}) + \sum_{j=1}^I \sum_{t=1}^T \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,t,v) (V_i x_{jiktuv}) = \hat{P}_i \quad \forall i \\
 (4') \quad \bar{E}_i &+ \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(i,j,t,u,v) (P_{ijktuv} x_{ijktuv}) - \sum_{\substack{j=1 \\ j \neq i}}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^2 K(j,i,t,u,v) (P_{jiktuv} x_{jiktuv}) \geq -\hat{P}_i - y_i \quad \forall i \\
 (5) \quad &C_{ijktuv} \geq x_{ijktuv} \geq 0 \quad \forall i, j, k, t, u, v \\
 (7) \quad &y_i \geq 0 \quad \forall i
 \end{aligned}$$

Figura 5.2: Reducción del crédito especial (problema relajado)

5.3 Optimización de las operaciones a liquidar

Sea Y^* el valor óptimo de la función objetivo de P'_1 . Entonces, Y^* es el monto mínimo de crédito especial que debe otorgar Banco de México, hasta el momento, para liquidar el mercado sin que los participantes violen las reglas de operación. En otras palabras, Y^* es el valor mínimo de crédito especial requerido para que la solución entera propuesta en el corte inmediato anterior sea factible en P_1 . Una vez que se conoce este valor, se determinan las operaciones pendientes que pueden liquidarse, sin aumentar el crédito especial Y^* a otorgar; para lograr esto, se plantea el programa entero-mixto P_2 (ver figura 5.3). El planteamiento de este programa es similar a los anteriores y su objetivo es encontrar las operaciones de valor máximo que puedan liquidarse sin otorgar más crédito que Y^* .

Se simplifica P_2 , descartando las restricciones de integridad, con lo que obtenemos el programa lineal relajado P'_2 (figura 5.4).

Como la cota de las variables y_i se obtuvo de resolver P'_1 , el punto óptimo de P'_1 es solución factible de P'_2 . Por su estructura, P'_2 también puede resolverse con el algoritmo de descomposición de Dantzig-Wolfe.

5.4 Solución óptima

La solución óptima de P'_2 indica, en principio, la cantidad de títulos que pueden liquidarse hasta el corte actual, por operación, utilizando el menor crédito especial posible.

Si en la solución óptima de P'_2 hay al menos una variable cuyo valor óptimo no sea entero (en principio, el valor óptimo de las variables de un problema de programación lineal puede ser cualquier número real), ir al paso 5.6 para reformular el problema, puesto que es inaceptable cerrar operaciones por una cantidad de títulos que no sea un número entero.

En cambio, si la solución óptima de P'_2 es entera ir al paso 5.5.

5.5 Solución entera

Si en la solución óptima de P'_2 cada variable tiene un valor entero que está comprendido entre cero y la cantidad total de títulos de la operación correspondiente, entonces el óptimo de P'_2 también es óptimo de P_2 . El crédito especial otorgado por Banco de México al participante i , será igual al valor óptimo de y_i en P'_2 , para $i = 1, \dots, I$. Ir al paso 5.9 para liquidar las operaciones.

5.6 Formulación del problema de flujos

La nueva formulación reconoce implícitamente que no es posible encontrar el óptimo de P_2 con recursos razonables de cómputo, y entonces buscaremos una solución (entera) cercana al óptimo de P'_2 , aún cuando esta solución viole las restricciones (2')-(4') de P_2 , pero tratando de que la violación sea mínima. La búsqueda se realiza en el espacio de

soluciones enteras próximo al punto óptimo de P_2' , con la hipótesis tácita de que éste está muy próximo al punto óptimo de P_2 . La forma de replantear el problema es la siguiente:

Supondremos, en lo sucesivo, que la cantidad por la que se pretende hacer cada operación es igual al entero más grande, encontrado a través de P_2' , que puede liquidarse. Es decir, se supondrá que cada operación pendiente es por la cantidad que ya se sabe que puede pasar sin problemas; dicha cantidad puede ser, incluso, cero.

En otras palabras, sea $x_{ijkltuv}^*$ la cantidad óptima a liquidar en cada operación, según P_2' . De aquí en adelante supondremos que la cantidad por la que se quiere hacer cada operación será $C_{ijkltuv}'$, la cual se obtiene truncando $x_{ijkltuv}^*$. Esto es, $C_{ijkltuv}' = [x_{ijkltuv}^*]$, donde $[x]$ indica el mayor entero que es menor o igual a x .

Este recorte a las cantidades operadas es tal que las operaciones cuya liquidación óptima en P_2' es entera, no deben tener problemas para pasar de manera total, pues la cantidad supuesta de las mismas es igual a la cantidad óptima a liquidar. Sin embargo, como las cantidades originales de las operaciones con liquidación óptima no entera se truncaron al menor entero más próximo, es posible que la fracción de título cancelada deje corto, en la tenencia de éstos, al participante que compra y hay que cancelar, parcialmente, alguna otra operación en la que éste último podía vender pues tenía los títulos suficientes.

Por tanto, hay que determinar cuáles operaciones con cantidades supuestas pasan sin problemas y cuáles no pueden pasar completamente pues provocarían sobregiros en las cuentas de títulos. Para lograr esto, simplificaremos P_2' considerando sólo las restricciones (1) de títulos y reemplazando con $C_{ijkltuv}'$ las cotas superiores $C_{ijkltuv}$ en (5), lo que da lugar al problema de flujo en redes P_3 (ver figura 5.5).

Es importante hacer notar que aunque en P_3 consideramos todas las operaciones originales de P_2 , la cantidad operada en cada una es $C_{ijkltuv}'$. Además, P_3 tiene *cero* como

solución factible pues, como ya sabemos, $\overline{T}_{itu} \geq 0$.

$$P_3 = \left\{ \begin{array}{l} \text{(max)} \quad Z = \sum_{i=1}^I \sum_{j=1}^I \sum_{t=1}^T \sum_{u=1}^2 \sum_{v=1}^2 \sum_{k=1}^{K(i,j,t,u,v)} (V_i \times x_{ijktuv}) \\ \text{s.a.} \\ \text{(1)} \quad \sum_{j=1}^I \sum_{u=1}^2 \sum_{k=1}^{K(i,j,t,u,v)} x_{ijktuv} - \sum_{j=1}^I \sum_{v=1}^2 \sum_{k=1}^{K(j,i,t,v,u)} x_{jiktuv} \leq \overline{T}_{itu} \quad \forall i, t, u \\ \text{(5)} \quad C'_{ijktuv} \geq x_{ijktuv} \geq 0 \quad \forall i, j, k, t, u, v \end{array} \right.$$

Figura 5.5: Problema de Flujos

5.7 Resolución del problema de flujos

Se resuelve P_3 . Debido a que P_3 es un problema de flujos, su solución óptima es entera y obtenerla es relativamente fácil.

Con las nuevas cotas C'_{ijktuv} , esperamos que la solución óptima de P_3 esté muy cerca del óptimo de P'_2 . De hecho, por la función objetivo que tiene el problema de flujos, las variables se "pegarán" a su cota superior y en el punto óptimo de P_3 el valor de cada variable estará por debajo de su valor óptimo en P'_2 , sólo si es necesario para lograr factibilidad en las cuentas de títulos.

La solución óptima de P_3 indica la cantidad de títulos que se deben liquidar en cada operación.

5.8 Determinación de créditos especiales

Los saldos finales de cada participante se determinan sustituyendo el valor óptimo de P_3 en las ecuaciones (2')-(4') de P_2 . Como P_3 fue planteado sin considerar restricciones de efectivo ni de posición, es posible que alguno de los participantes se sobregire en cualquiera de estas dos cuentas. El valor absoluto de este sobregiro será igual al monto del crédito *especial* que Banco de México le otorgará al participante, con objeto de simplificar el proceso de optimización. Esperamos que el monto total del crédito especial no sea mucho mayor que el valor óptimo Y^* de P_1' .

El crédito especial otorgado a cada participante sólo servirá para cubrirlo en caso de que se sobregire, en efectivo o posición, al aplicar las operaciones parciales sugeridas por la solución óptima del problema P_3 y su origen es la falta de recursos (computacionales y algorítmicos) eficientes para resolver P_0 . Por lo tanto, no se otorgará crédito especial a los participantes que no resulten sobregirados con la solución óptima de P_3 .

En el siguiente corte del día, el sistema buscará, a través de P_1 , que el monto del crédito especial se mantenga lo más pequeño posible. Si al final del ejercicio aún hay créditos especiales diferentes de cero, Banco de México cubrirá el importe de dicho crédito. Las operaciones del día siguiente iniciarán con crédito especial igual a cero para todos los participantes.

5.9 Liquidación

El sistema cerrará las operaciones que indica la solución óptima encontrada, según sea el caso.

- Se liquidan totalmente las operaciones que corresponden a las variables cuyo valor óptimo es igual a la cantidad total de títulos en la operación original. Estas operaciones desaparecen de la cola después de liquidarse.
- Se liquidan parcialmente las operaciones que corresponden a las variables cuyo valor óptimo es menor a la cantidad de títulos acordada originalmente. Por cada operación parcial (en caso de haber) se crea una nueva operación por la cantidad que no pasó y el conjunto de éstas regresa a la cola para ser reprocesadas en el siguiente corte del ejercicio. La parte de la operación que sí pasó se considera como una operación cerrada, por esa cantidad, y desaparece de la cola.
- Las operaciones que corresponden a variables cuyo valor óptimo es cero no pasan, es decir, regresan a la cola para ser reprocesadas en el siguiente corte del ejercicio.

Capítulo 6

La descomposición Dantzig-Wolfe

Sabemos que para tratar de resolver problemas de programación lineal de dimensiones grandes es más conveniente, desde el punto de vista computacional, implementar el método Simplex Revisado que el algoritmo Simplex normal, ya que por las características del primero, se explotan y aprovechan mejor los recursos de las computadoras [8].

Así como el método Simplex Revisado es una mejora del algoritmo Simplex para problemas grandes que requieren gran cantidad de almacenamiento de información, debido al problema computacional que éste representa, se puede pensar que la descomposición Dantzig-Wolfe es una mejora del Simplex Revisado para problemas aún más grandes que tengan cierta estructura.

La idea principal del algoritmo de descomposición es que la estructura del programa lineal por resolver permite dividir dicho problema en varios subproblemas independientes y un problema "maestro" que conservará las relaciones comunes de los subproblemas, con la ventaja de que los programas así obtenidos serán de menor tamaño que el original y por tanto, más fáciles de resolver.

Consideremos un problema de programación lineal con la siguiente estructura:

$$\begin{aligned} \text{Min } Z &= C_1 X_1 + C_2 X_2 + \cdots + C_t X_t \\ \text{s.a.} & \\ & a_1 X_1 \leq b_1 \\ & \quad a_2 X_2 \leq b_2 \\ & \quad \quad \quad \vdots \\ & \quad \quad \quad a_t X_t \leq b_t \\ & A_1 X_1 + A_2 X_2 + \cdots + A_t X_t \leq b \\ & X_i \geq 0 \quad \forall i = 1, \dots, t \end{aligned}$$

donde, $X_i \in \mathbb{R}^{n_i}$, a_i es una matriz de $m_i \times n_i$, $b_i \in \mathbb{R}^{m_i}$, $C_i \in \mathbb{R}^{n_i}$, A_i es una matriz de $m \times n_i$ y $b \in \mathbb{R}^m$; para $i = 1, \dots, t$.

El punto de partida de la descomposición es que el problema anterior se puede reescribir como:

$$\text{Min } Z = C_1 X_1 + C_2 X_2 + \dots + C_t X_t$$

s. a.

$$A_1 X_1 + A_2 X_2 + \dots + A_t X_t \leq b$$

donde cada X_i debe cumplir con:

$$a_i X_i \leq b_i$$

$$X_i \geq 0 \quad \forall i = 1, \dots, t$$

Estas últimas restricciones definen regiones convexas y obligan a las X_i a ser puntos interiores de las mismas.

Suponiendo que tales regiones son acotadas (más adelante eliminaremos ese supuesto), sabemos por el Teorema de Representación que toda solución factible de los subproblemas se puede expresar como una combinación lineal convexa de los puntos extremos de la región [2]. Es decir, si suponemos que la región i tiene k_i puntos extremos, x_{ij} , entonces toda solución factible de dicho subproblema se puede escribir como:

$$X_i = \sum_{j=1}^{k_i} \lambda_{ij} x_{ij}$$

donde, $\sum_{j=1}^{k_i} \lambda_{ij} = 1$, $\lambda_{ij} \geq 0 \quad \forall i = 1, \dots, t; j = 1, \dots, k_i$.

Con tales representaciones, el problema original es equivalente a:

$$\text{Min } Z = C_1 \sum_{j=1}^{k_1} \lambda_{1j} x_{1j} + C_2 \sum_{j=1}^{k_2} \lambda_{2j} x_{2j} + \dots + C_t \sum_{j=1}^{k_t} \lambda_{tj} x_{tj}$$

s.a.

$$\begin{aligned} A_1 \sum_{j=1}^{k_1} \lambda_{1j} x_{1j} + A_2 \sum_{j=1}^{k_2} \lambda_{2j} x_{2j} + \dots + A_t \sum_{j=1}^{k_t} \lambda_{tj} x_{tj} &\leq b \\ \sum_{j=1}^{k_1} \lambda_{1j} &= 1 \\ \sum_{j=1}^{k_2} \lambda_{2j} &= 1 \\ &\vdots \\ \sum_{j=1}^{k_t} \lambda_{tj} &= 1 \\ \lambda_{ij} &\geq 0 \quad \forall i, j \end{aligned}$$

Si denotamos $S_{ij} = A_i x_{ij}$ y $C'_{ij} = C_i x_{ij}$, para $i = 1, \dots, t$; $j = 1, \dots, k_i$; el problema se puede reescribir como:

$$\text{Min } Z = \sum_{j=1}^{k_1} \lambda_{1j} C'_{1j} + \sum_{j=1}^{k_2} \lambda_{2j} C'_{2j} + \dots + \sum_{j=1}^{k_t} \lambda_{tj} C'_{tj}$$

s.a.

$$\begin{aligned} \sum_{j=1}^{k_1} \lambda_{1j} S_{1j} + \sum_{j=1}^{k_2} \lambda_{2j} S_{2j} + \dots + \sum_{j=1}^{k_t} \lambda_{tj} S_{tj} &\leq b \\ \sum_{j=1}^{k_1} \lambda_{1j} &= 1 \\ \sum_{j=1}^{k_2} \lambda_{2j} &= 1 \\ &\vdots \\ \sum_{j=1}^{k_t} \lambda_{tj} &= 1 \\ \lambda_{ij} &\geq 0 \quad \forall i, j \end{aligned}$$

A esta última representación del problema, le llamaremos Programa Maestro y es la que se usará para desarrollar, propiamente, el Algoritmo de Descomposición [4].

Observemos que toda base del Programa Maestro debe tener $m + t$ vectores mientras que las del programa original tienen $m + \sum_{i=1}^l m_i$. Como la principal dificultad al solucionar un problema lineal depende del tamaño de su base, la transformación que se hizo al programa original es, en principio, una reducción en la complejidad al resolver.

Ahora bien, supongamos que de alguna manera (que se explicará más adelante) encontramos una solución inicial que sea básica factible para el Programa Maestro, entonces, siguiendo la metodología del Algoritmo Simplex, hay que determinar si esta solución es óptima y, si no lo es, mejorarla. Recuerde-se que para saber si hay una solución mejor que la actual, debemos calcular para toda columna j no básica: $\bar{C}_j = Z_j - C_j'$; donde Z_j se obtiene multiplicando el vector de soluciones duales Π correspondiente por la columna j y C_j' es el coeficiente de esta última en la función objetivo [2]. Si el máximo de los \bar{C}_j es menor o igual que cero la solución encontrada es óptima, si no, ésta puede mejorarse.

Esta prueba de optimalidad debe hacerse con todas las columnas no básicas j del Programa Maestro, pero, como las columnas de éste se obtienen a partir de puntos extremos de los subproblemas, entonces, cada \bar{C}_j corresponde a un punto extremo de algún subproblema i . Por tanto, para encontrar el $\max\{\bar{C}_j\}$ podemos determinar al máximo de éstos en cada subproblema y posteriormente elegir el mayor de éstos. En otras palabras, primero se buscará el punto extremo de la región i que más contribuya a mejorar la función objetivo y después hay que encontrar al mejor para todos los subproblemas, o sea, el $\max_i\{\bar{C}_j\}$.

Antes de iniciar la prueba de optimalidad consideraremos varias cosas. Dado que \bar{C}_j estará asociado a un subproblema i , lo denotaremos como \bar{C}_{ij} y, por tanto, nos centraremos en determinar, $\max_i\{\bar{C}_{ij}\} = \max_i\{Z_{ij} - C_{ij}'\}$. Además, si la solución actual tiene $j_i - 1$ columnas básicas asociadas al subproblema i , el punto extremo buscado será el j_i -ésimo asociado a dicha región y lo denotaremos como x_{ij} . Como Π puede separarse en dos vectores, π' de dimensión m y π de dimensión t , entonces:

$$\begin{aligned} Z_{ij} &= \Pi \times \begin{pmatrix} S_{ij} \\ e_i \end{pmatrix} \\ &= (\pi', \pi) \times \begin{pmatrix} S_{ij} \\ e_i \end{pmatrix} \\ &= \pi' S_{ij} + \pi e_i \end{aligned}$$

dado que πe_i es constante para el subproblema i :

$$\begin{aligned} \max_i \{\bar{C}_{ij}\} &= \max_i \{Z_{ij} - C'_{ij}\} \\ &= \max_i \{\pi' S_{ij} + \pi e_i - C'_{ij}\} \\ &= \max_i \{\pi' S_{ij} - C'_{ij}\} \end{aligned}$$

Por último, sustituyendo el valor de S_{ij} y C'_{ij} para el punto extremo x_{ij} de la región i :

$$\begin{aligned} \max_i \{\bar{C}_{ij}\} &= \max_i \{\pi' A_i x_{ij} - C_i x_{ij}\} \\ &= \max_i \{(\pi' A_i - C_i) x_{ij}\} \\ &= \max_i \{\hat{C}_i x_{ij}\} \end{aligned}$$

donde, $\hat{C}_i = \pi' A_i - C_i$ y x_{ij} tiene que ser solución factible, es decir, deberá cumplir: $a_i x_{ij} \leq b_i$ para $i = 1, \dots, l$; $j = 1, \dots, k_i$.

En resumen, para decidir qué columna puede entrar a la base, hay que determinar x_{ij} que maximice $\hat{C}_i x_{ij}$ con la restricción de que x_{ij} sea punto extremo de la región i .

Por tanto, el siguiente paso del Simplex, en este caso, consiste en encontrar para todo subproblema i :

$$\begin{aligned} \max Z_i &= \widehat{C}_i x_{ij} \\ \text{s.a.} & \\ a_i x_{ij} &\leq b_i \\ x_{ij} &\geq 0 \end{aligned}$$

Como supusimos que la región i es acotada y el anterior es un programa lineal con coeficientes obtenidos a partir del vector dual Π , el óptimo se alcanzará en algún punto extremo (x_{ij}^*) de la región y corresponderá a alguna columna no básica del Programa Maestro.

Sea Z_i^* el máximo del subproblema i , entonces, si

$$\max_i \{\overline{C}_{ij}\} = Z_i^* + \pi e_i = Z_i^* + \pi(i)$$

es mayor que cero, la columna $(A_i x_{ij}^*) = (s_{ij}^*)$ debe entrar a la base, con coeficiente de costos $C'_{ij} = C_i x_{ij}^*$, para lo cual se le multiplica por la inversa de la matriz básica y se pivotea sobre ella, obteniendo una nueva base y por tanto un nuevo vector Π con el cual podemos hacer otra prueba de optimalidad y repetir el ciclo anterior, hasta encontrar que no hay i tal que $Z_i^* + \pi(i) > 0$ y se tenga el óptimo.

En este punto, $Z_i^* + \pi(i) \leq 0$ toda $i = 1, \dots, t$ y tenemos una base óptima del Programa Maestro que se obtuvo a través del cálculo de puntos extremos de cada región, sin necesidad de conocer de antemano todos los puntos extremos del problema sino generando sólo aquel que debiera entrar a la base. Esto, junto con la reducción en la tamaño de la base, hace más atractivo tratar de resolver el problema original con la metodología presentada que con otro algoritmo.

Es importante notar que el algoritmo presentado es sólo una implantación del Simplex Revisado, con la única diferencia de que el cálculo de $\overline{C}_{ij} = Z_{ij} - C'_{ij}$ se hace resolviendo

los subproblemas. Por tanto, si se tiene presente una regla de prevención de ciclos, tanto para el Programa Maestro como para los subproblemas, el algoritmo converge en un número finito de iteraciones.

Debemos recordar que para plantear el Programa Maestro usamos una transformación, a partir del Teorema de Representación, por lo que si al final en la base óptima se tienen k'_i puntos extremos de la región i , es posible generar las soluciones óptimas de cada subproblema original utilizando las soluciones óptimas λ_{ij}^* del Programa Maestro con la transformación:

$$X_1^* = \sum_{j=1}^{k'_1} \lambda_{1j}^* x_{1j}$$

$$X_2^* = \sum_{j=1}^{k'_2} \lambda_{2j}^* x_{2j}$$

$$\vdots$$

$$X_i^* = \sum_{j=1}^{k'_i} \lambda_{ij}^* x_{ij}$$

6.1 Modificaciones cuando los subproblemas son no acotados

Al construir el Programa Maestro supusimos que las restricciones de los subproblemas definían regiones acotadas y por tanto, por el Teorema de Representación, cada X_i factible podría expresarse como una combinación lineal convexa de puntos extremos. Sin embargo, es posible que alguno de los subproblemas tenga una región factible no acotada, en tal caso, el mismo teorema dice que las soluciones factibles se pueden expresar como una combinación lineal convexa de puntos extremos más una combinación lineal no negativa de direcciones extremas [2], es decir, todo X_i factible deberá cumplir:

$$x_i = \sum_{j=1}^{k_i} \lambda_{ij} x_{ij} + \sum_{l=0}^{r_i} \mu_{il} d_{il}$$

donde, $\lambda_{ij} \geq 0$, $\sum_{j=1}^{k_i} \lambda_{ij} = 1$, x_{ij} es punto extremo de la región i , $\mu_{il} \geq 0$ y d_{il} es dirección extrema de la región, para $i = 1, \dots, l$; $j = 1, \dots, k_i$; $l = 1, \dots, r_i$ (se supone que hay r_i direcciones extremas).

A partir de aquí, la construcción y la solución del Programa Principal son análogas a las desarrolladas previamente y, por tanto, el algoritmo presentado seguirá siendo válido, salvo que es posible que la columna S_{ij}^* candidata a entrar a la base esté asociada a una dirección extrema y no a un punto extremo [4]. En este caso, la columna entrante sería:

$$\begin{pmatrix} S_{ij}^* \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} A_i d_{ij}^* \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

con coeficiente $C'_{ij} = C_i d_{ij}^*$.

6.2 Inicialización

El Algoritmo de Descomposición necesita una solución inicial factible del Programa Maestro, lo cual implica que se deben encontrar $m + l$ columnas básicas (asociadas a puntos extremos de los Subproblemas o a variables de holgura) tales que al aplicarles la transformación S_{ij} cumplan con las restricciones de dicho programa. En esta sección expondremos un procedimiento que permite encontrar esas columnas por medio de un Programa Maestro Fase I.

Para construir el Programa Maestro Fase I primero se busca una solución factible de cada subproblema. En particular, podemos tomar la solución x_{i1}^* que se obtiene al resolver:

$$\begin{aligned} \text{Max } Z_i &= C_i x_i \\ \text{s.a.} & \\ & a_i x_i \leq b_i \end{aligned}$$

donde C_i es el vector de coeficientes originales asociados a la región i , para $i = 1, \dots, t$.

Si a las soluciones x_{i1}^* (o d_{i1}^* en caso de que el subproblema i sea no acotado) se les aplica la transformación $S_{i1} = A_i x_{i1}^*$, se tienen las primeras t columnas del que será el Programa Maestro Fase I.

Supongamos, para generalizar, que en el programa original hay m_1 restricciones de tipo menor o igual, m_2 de tipo mayor o igual, y m_3 igualdades ($m = m_1 + m_2 + m_3$). Si agregamos $m_1 + m_2$ variables de holgura, y_j , y $m_2 + m_3$ variables artificiales, a_k , se obtiene el Programa Maestro Fase I:

$$\begin{aligned} \text{Min } W &= \sum_{k=1}^{m_2+m_3} a_k \\ \text{s.a.} & \\ & \sum_{i=1}^t \lambda_i S_{i1} + \sum_{j=1}^{m_1} y_j e_j - \sum_{j=1}^{m_2} y_{m_1+j} e_{m_1+j} + \sum_{k=1}^{m_2+m_3} a_k e_{m_1+k} = b \\ & \lambda_{i1} = 1 \quad \forall i = 1, \dots, t \\ & y_j \geq 0 \quad \forall j = 1, \dots, m_1 + m_2 \\ & a_k \geq 0 \quad \forall k = 1, \dots, m_2 + m_3 \end{aligned}$$

Este último programa tiene una base inicial B formada por variables a_k , y_j y λ_{i1} , a partir de la cual se puede aplicar el Algoritmo de Descomposición previamente expuesto, pero que ahora buscará columnas del tipo $\begin{pmatrix} S_i^* \\ y_i \end{pmatrix}$ que deban entrar a la base con objeto de minimizar la suma de las a_k .

En otras palabras, con la base inicial encontrada y la nueva función objetivo (W) se puede determinar $\Pi = C_B B^{-1}$ con el cual se forma $\hat{C}_i = \pi' A_i - C_i$, se resuelven los subproblemas con estos nuevos coeficientes y se busca $Z_i^* + \pi(i) > 0$ que permita entrar a la base a la columna $\begin{pmatrix} S_i^* \\ y_i \end{pmatrix}$ generando una nueva solución y por tanto una nueva base, con la cual se reinicia el proceso.

Si al término de este procedimiento el valor óptimo de W es cero, entonces la base tiene $m + t$ columnas del tipo $\begin{pmatrix} S_i^* \\ y_i \end{pmatrix}$ que cumplen con las restricciones del Programa Maestro, lo cual quiere decir que se tiene la solución básica inicial buscada.

Si, por el contrario, el valor óptimo de W es diferente de cero, entonces no hay solución factible del Programa Maestro (pues se requiere que alguna a_j sea mayor que cero para satisfacer las restricciones) y por tanto, no hay solución factible del problema original, pues no es posible hacer una combinación convexa de puntos extremos de los subproblemas que cumpla con las restricciones iniciales.

6.3 Resumen del algoritmo

1. Plantear y resolver:

$$\begin{aligned} \text{Max } Z_i &= C_i x_i \\ \text{s.a.} & \\ & a_i x_i \leq b_i \end{aligned}$$

para todo subproblema $i = 1, \dots, t$. Sea x_{i1}^* la solución óptima del subproblema i .

2. Aplicar la transformación

$$S_{i1} = A_i x_{i1}^*$$

El coeficiente asociado a la columna S_{i1} es: $C'_{i1} = C_i x_{i1}^*$.

3. Construir el Programa Maestro Fase I:

$$\begin{aligned} \text{Min } W &= \sum_{k=1}^{m_2+m_3} a_k \\ \text{s.a.} & \\ & \sum_{i=1}^t \lambda_{i1} S_{i1} + \sum_{j=1}^{m_1} y_j e_j - \sum_{j=1}^{m_2} y_{m_1+j} e_{m_1+j} + \sum_{k=1}^{m_2+m_3} a_k e_{m_1+k} = b \\ & \lambda_{i1} = 1 \quad \forall i = 1, \dots, t \\ & y_j \geq 0 \quad \forall j = 1, \dots, m_1 + m_2 \\ & a_k \geq 0 \quad \forall k = 1, \dots, m_2 + m_3 \end{aligned}$$

4. Se tiene una base del Programa Maestro Fase I formada por variables a_k , y_j y λ_{i1} .

Calcular el vector de las variables duales correspondiente ($\Pi = C_B B^{-1}$).

5. Iniciar el Ciclo Principal siguiente:

- Con el vector $\Pi = (\pi', \pi)$ asociado a las variables duales de la solución actual generar las funciones objetivo de cada bloque:

$$\hat{C}_i = \pi' A_i - C_i$$

- Resolver para todo bloque i :

$$\text{Max } Z_i = \hat{C}_i x_{ij}$$

s.a.

$$a_{ij} x_{ij} \leq b_i$$

$$x_{ij} \geq 0$$

y determinar Z_i^* .

- Calcular $\bar{C}_i = Z_i^* + \pi(i)$. Hay dos posibilidades:
 - Si $\bar{C}_i \leq 0$ para todo bloque i , estamos en el óptimo. Fin del ciclo.
 - Si $\bar{C}_i > 0$ para algún bloque i , se debe continuar el ciclo.
- Sean i' el bloque con que se alcanza el máximo de los \bar{C}_i y $x_{i'j}^*$ la solución óptima asociada.
- Generar la columna

$$S_{i'j}^* = \begin{pmatrix} A_{i'} x_{i'j}^* \\ \epsilon_{i'}$$

que tendrá coeficiente de costos $C_{i'j}' = C_{i'} x_{i'j}^*$.

- Agregar esa columna al Programa Maestro.
- Resolver el nuevo Programa Maestro, usando la solución inmediata anterior para ahorrar iteraciones.
- Obtener un nuevo vector Π con el cual se repite el ciclo.

6. Al término del Ciclo Principal hay dos posibilidades:

- Si el valor óptimo de la función objetivo es cero, entonces tenemos una solución factible inicial del Programa Maestro. (ir a 7)
- Si el valor óptimo de la función objetivo es diferente de cero, entonces no hay solución factible del Programa Maestro y por tanto el problema original es infactible. Fin.

7. Si se tiene una base del Programa Maestro formada por k_i columnas asociadas al bloque i (columnas del tipo S_{ij}), se determinan nuevos coeficientes de costos: $C'_{ij} = C_i x_{ij}$, para $i = 1, \dots, t$; $j = 1, \dots, k_i$.

En la construcción del Programa Maestro fase I (paso 3), se agregaron $m_1 + m_2$ variables de holgura; sea $L \subseteq \{1, \dots, m_1 + m_2\}$ el conjunto de índices de esas variables que son básicas en la solución actual.

Formar el Programa Maestro:

$$\text{Min } Z = \sum_{j=1}^{k_1} \lambda_{1j} C'_{1j} + \sum_{j=1}^{k_2} \lambda_{2j} C'_{2j} + \dots + \sum_{j=1}^{k_t} \lambda_{tj} C'_{tj}$$

s.a.

$$\begin{aligned} \sum_{j=1}^{k_1} \lambda_{1j} S_{1j} + \sum_{j=1}^{k_2} \lambda_{2j} S_{2j} + \dots + \sum_{j=1}^{k_t} \lambda_{tj} S_{tj} \pm \sum_{l \in L} y_l e_l &= b \\ \sum_{j=1}^{k_1} \lambda_{1j} &= 1 \\ \sum_{j=1}^{k_2} \lambda_{2j} &= 1 \\ &\vdots \\ \sum_{j=1}^{k_t} \lambda_{tj} &= 1 \\ \lambda_{ij} &\geq 0 \quad \forall i, j \\ y_l &\geq 0 \quad \forall l \in L \end{aligned}$$

8. Como se conoce la base de este programa, se puede determinar $\Pi = C_B B^{-1}$ para los nuevos coeficientes.
9. Con el vector Π calculado en 8 y la función objetivo calculada en 7 entrar a un ciclo como el explicado en 5.
10. Al término de este nuevo ciclo, suponiendo que en la base óptima se tienen k'_i variables básicas asociadas al subproblema i , la solución óptima del programa original para cada subproblema es:

$$x_i^* = \sum_{j=1}^{k'_i} \lambda_{ij}^* x_{ij}$$

Capítulo 7

Implantación en computadora

En capítulos anteriores presentamos una metodología que permite optimizar la liquidación de operaciones en el mercado de valores mexicano. Dicha metodología fue implantada en un sistema computacional llamado *PROCOVAL* (Procesador de Colas de Operaciones Pendientes en el Módulo de Valores del INDEVAL) que fue desarrollado de la siguiente manera.

Debido a que la parte medular de la metodología de solución consiste en resolver varios problemas de programación lineal cuya estructura es conocida, y más aún, dado que la estructura de dichos programas nos permite aplicar el algoritmo de descomposición de Dantzig-Wolfe, el *PROCOVAL* tiene un módulo que resuelve problemas de programación lineal a través de dicho algoritmo.

Recordemos que la idea básica del algoritmo de descomposición de Dantzig-Wolfe es dividir un programa lineal con cierta estructura en varios subproblemas más pequeños. Cada subprograma obtenido tiene los mismos componentes y características, y se puede resolver con las mismas técnicas; por tal motivo, podemos pensar que cada subprograma pertenece a un mismo tipo de dato. Para aprovechar esto, decidimos desarrollar el *PROCOVAL* con técnicas de programación orientada a objetos, ya que la esencia de esta programación es, precisamente, diseñar y construir tipos de datos convenientes, a los cuales se asocian operaciones propias [9]. Este tipo de programación permite ahorrar recursos computacionales sustanciales, ya que se manejan estructuras de datos adecuadas, además de que permite manipular los objetos construidos de la manera que más convenga al programador. En nuestro caso, se construyó una clase llamada "ProgramaLineal", cuyos objetos son problemas de programación lineal. Al aplicar la descomposición, el *PROCOVAL* crea uno de estos objetos por cada subprograma. Además, los objetos de la clase "ProgramaLineal" se pueden resolver con el método Simplex Revisado y se les puede cambiar la función objetivo, el lado derecho, etc.

El *PROCOVAL* fue programado en C++ y compilado con *Borland*, *gnu* y *cygnus*,

para sistemas operativos *DOS* y *UNIX*.

Por estar programado en *C++*, los diferentes módulos que componen el PROCOVAL son en realidad funciones enteras [7]. Todas las funciones tienen como valor de retorno *erro* si no hubo ningún error y un valor diferente de *erro* si lo hubo. Este valor es único y nos permite rastrear e identificar el tipo de error en que se incurrió.

En la figura 7.1 representamos el flujo de información del sistema. En las secciones posteriores explicamos los componentes de este diagrama.

7.1 Entrada

Los datos de entrada del PROCOVAL son:

- Archivo de operaciones pendientes. En este archivo están registradas todas las operaciones pendientes de liquidación. Por cada operación pendiente hay un registro compuesto por: fecha, institución vendedora, cuenta desde la que se vende, institución compradora, cuenta hacia la que se compra, tipo de título operado, cantidad de títulos, precio por título y folio de la operación.

Cuando una operación tiene datos incorrectos, el sistema pregunta al usuario si desea continuar sin esa operación o si se detiene el proceso.

- Archivo de saldos iniciales de títulos. Este archivo tiene un registro por cada cuenta participante y expresa su saldo en títulos. Los registros de este archivo están formados por: fecha, institución, cuenta, tipo de título, y saldo de la cuenta al momento de detener el registro de operaciones.
- Archivo de valores de garantía. Este archivo es copia de una tabla en la que se registra el valor que tiene cada tipo de título para Banco de México. El archivo

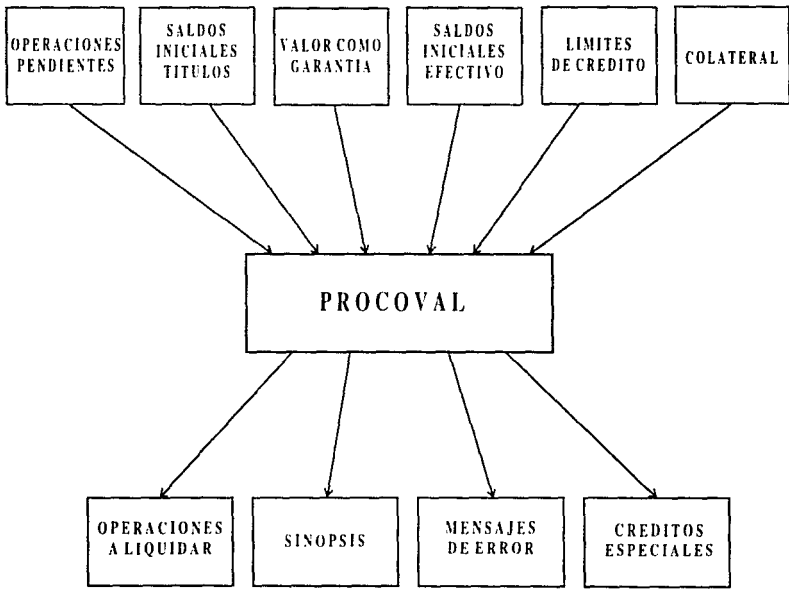


Figura 7.1: Flujo de información

contiene un registro por cada tipo de título. En cada registro aparece el tipo de título y su valor como garantía.

Cuando no se encuentra el valor como garantía de un tipo de título, se supone que este valor es igual al precio con que se operó por primera vez dicho título.

- Archivo de saldos iniciales de efectivo, límites de crédito y colateral. En un único archivo, se guarda un registro por cada institución participante, conteniendo: fecha, institución, límite de crédito para ese día, saldo de efectivo en el módulo de valores al momento de detenerse las operaciones, y valor de los títulos en cuentas propias del participante (colateral).

En la lectura de estos archivos el sistema verifica que los formatos sean los acordados con el INDEVAL. Además, se verifica que no haya incongruencias en los datos, tales como fechas diferentes en cada archivo, cantidades operadas negativas o cero, precios negativos, saldos en títulos menores que cero, etc. En caso de que se detecte alguno de estos errores, el PROCOVAL manda el mensaje correspondiente.

Aún cuando las instituciones deben dar garantías colaterales a los créditos que les sean otorgados, es posible que sólo tengan que garantizar que pueden pagar un porcentaje del monto total de su préstamo. En consecuencia, el sistema requiere como dato de entrada el porcentaje a garantizar con colateral, para la fecha que se desea procesar.

Hasta ahora, los nombres de los archivos de entrada y el porcentaje a garantizar se alimentan manualmente. Una vez que el PROCOVAL se incorpore al sistema que maneja las operaciones en el INDEVAL, y que tanto el porcentaje a garantizar como los nombres de los archivos de entrada sean fijos, el sistema podrá funcionar de manera automática.

7.2 Salida

Como salida, el PROCOVAL genera lo siguiente:

- Archivo con las operaciones que pasan. Contiene un registro en el que se guarda, por cada operación procesada: fecha, folio de la operación y cantidad de títulos que pueden liquidarse en esa operación. Este archivo se llama "indeal.txt".
- Archivo de sinopsis. Contiene una sinopsis del proceso realizado por el PROCOVAL en la que se incluyen: número de operaciones procesadas, monto total asociado a dichas operaciones, cantidad de títulos operados, número de operaciones que pueden pasar total y parcialmente, porcentaje de operaciones en las que se liquida al menos un título, cantidad de títulos que se pueden liquidar y monto asociado a dichos títulos. Este archivo se llama "resumen.doc".
- Archivo de información. Contiene una reseña del proceso realizado en la que aparecen: fecha, nombres de los archivo de entrada, mensajes para saber si se planteó el problema de flujos y estado de la solución o soluciones encontradas. El nombre de este archivo es "dvp.log" y se diseñó para que el usuario conozca los detalles del proceso que siguió el PROCOVAL.

En caso de ocurrir alguna eventualidad durante el proceso, el sistema produce mensajes de advertencia, o error, según sea el caso. Cuando se generan estos mensajes, el PROCOVAL escribe en el archivo "dvp.log" una descripción detallada de la eventualidad detectada para que el usuario identifique qué ocurrió exactamente, la parte del proceso en que detectó el problema y, cuando es posible, ofrece alternativas para resolverlo.

Si al término del proceso algunas instituciones requieren créditos especiales, el sistema genera una lista con sus claves y el monto de los créditos requeridos.

Capítulo 8

Descripción del sistema computacional

En una primera etapa, el PROCOVAL solamente procesará las operaciones que estén pendientes de liquidación al final del día. Es decir, durante el ejercicio, el INDEVAL irá liquidando las operaciones de fácil verificación y sólo las que no se hayan liquidado al cierre del mercado serán procesadas con el PROCOVAL. Esto provoca que la metodología expuesta en el capítulo 5 sufra algunas modificaciones. Debido a que no se hacen cortes durante el día, no hay créditos especiales y, por tanto, no es necesario plantear los programas P_1 y P'_1 presentados en dicho capítulo. Así, la metodología de solución se simplifica de la siguiente manera:

1. Lectura de datos al cierre de operaciones.
3. Optimización de operaciones a liquidar.
4. Solución óptima.
5. Solución entera.
6. Formulación del problema de flujos.
7. Resolución del problema de flujos.
8. Determinación de créditos especiales.
9. Liquidación.

La descripción de estos puntos es igual a la presentada en el capítulo 4 y la metodología asociada se esquematiza con la figura 8.1. Es importante notar la desaparición del paso 2 de la metodología original.

En lo sucesivo, cuando hagamos referencia al PROCOVAL, debe entenderse que hablamos de una versión del sistema que implanta esta nueva metodología.

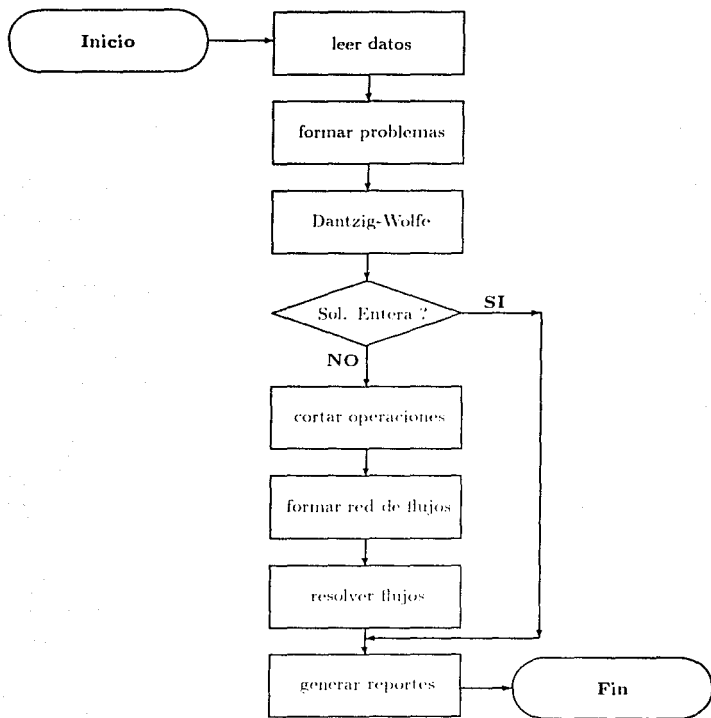


Figura 8.1: Diagrama de Flujo del PROCIVAL

Para correr el PROCIVAL se necesitan tres archivos: "procoval", "precios.txt" y "coderr.dat". El primer archivo es el que activa el sistema. El segundo archivo contiene el valor como garantía de los títulos vigentes a la fecha de ejecución. El último archivo contiene una tabla que relaciona los códigos de error del sistema con mensajes que los describen. Estos tres archivos deben encontrarse en el mismo subdirectorio.

En una corrida normal, el PROCIVAL pedirá al usuario los siguientes datos: fecha de ejecución, nombre del archivo de operaciones pendientes, nombre del archivo de saldos en títulos de las cuentas participantes, porcentaje de crédito que se debe garantizar colateralmente y, por último, nombre del archivo que contiene el saldo de efectivo, el límite de crédito y la posición inicial de cada institución participante.

Cuando los archivos de entrada se encuentren en el mismo subdirectorio que el archivo ejecutable, basta con escribir sus nombres para que el sistema los encuentre. Si los archivos están en otros subdirectorios, hay que introducir su trayectoria completa.

Al finalizar el proceso, los archivos de salida del PROCIVAL se encontrarán en el mismo subdirectorio que el archivo ejecutable.

Como programa computacional, PROCIVAL está compuesto por diferentes funciones, desarrolladas con C++, que permiten implantar los pasos de la metodología. El código del sistema está repartido en diferentes archivos, con objeto de agrupar las diferentes funciones de acuerdo a sus características.

1. Archivo de lectura. En este archivo ("lectura.cc") están las funciones que permiten leer todos los datos de entrada y cargarlos en memoria. Para lograr esto, el sistema realiza el siguiente procedimiento. Primero, lee las operaciones pendientes de liquidación y construye estructuras especiales para manejar confiablemente cada operación. Enseguida, accesa los saldos iniciales de títulos en cada cuenta participante. Finalmente, lee de un mismo archivo los límites de crédito, saldos de efectivo

y posiciones, de las instituciones. En los archivos de saldos solamente aparecen datos de cuentas e instituciones que hayan tenido operaciones pendientes.

2. Archivo de modelación. El nombre de este archivo es "modela.cc". Contiene las funciones que plantean el modelo de programación lineal y el modelo de flujos.
3. Archivo de optimización. En este archivo, llamado "optimiza.cc", se encuentran las funciones que permiten optimizar los modelos construídos. En particular, contiene una función que implanta el algoritmo de descomposición de Dantzig-Wolfe, construyendo dos clases (PROGMAESTRO y BLOQUES) y utilizando sus métodos.
4. Archivo de resultados. Las funciones que generan los archivos de salida del sistema y verifican la factibilidad de los resultados obtenidos están en el archivo "saldos.cc".
5. Archivo principal. En el archivo "procoval.cc" está el programa principal, junto con algunas funciones que controlan el flujo del sistema.
6. Archivo de métodos. Este archivo contiene la implantación de los métodos de cada clase y se llama "metodos.cc".
7. Archivo de simplex. La implantación del método simplex revisado, con todas las funciones necesarias, se encuentra en un archivo independiente de los demás, llamado "simplex.cc".

La unión de estos archivos, y de los archivos de encabezados, produce un código de más de 5.000 líneas. Por razones prácticas, hemos decidido que esta tesis no incluya la impresión de todo el código del sistema. Sin embargo, en el apéndice B, imprimimos un listado de las funciones que implantan el algoritmo de descomposición de Dantzig-Wolfe.

En la figura 8.2 presentamos un diagrama con las principales funciones del sistema y las relaciones que hay entre ellas.

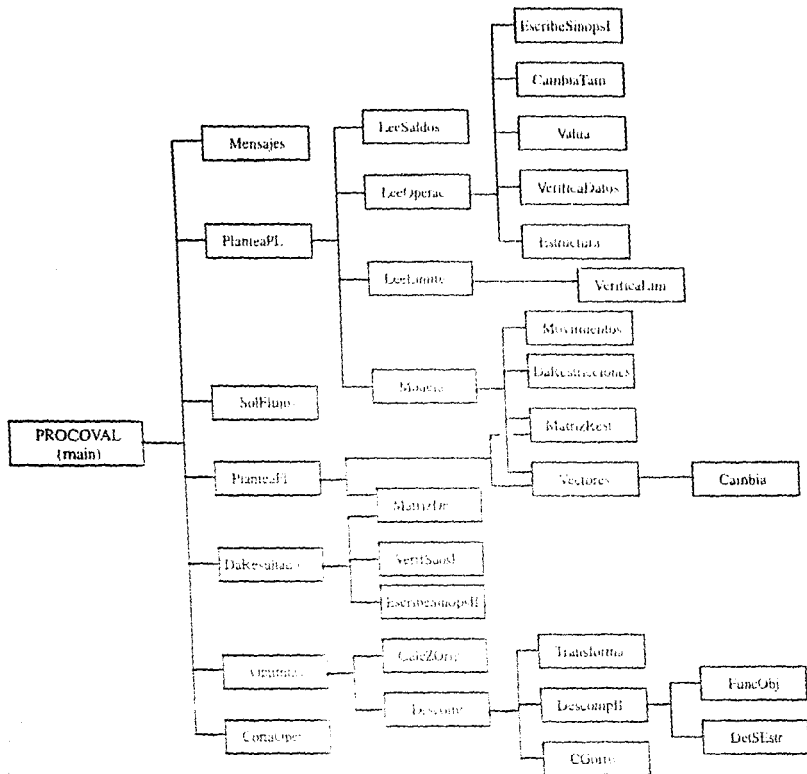


Figura 8.2: Diagrama de Estructura

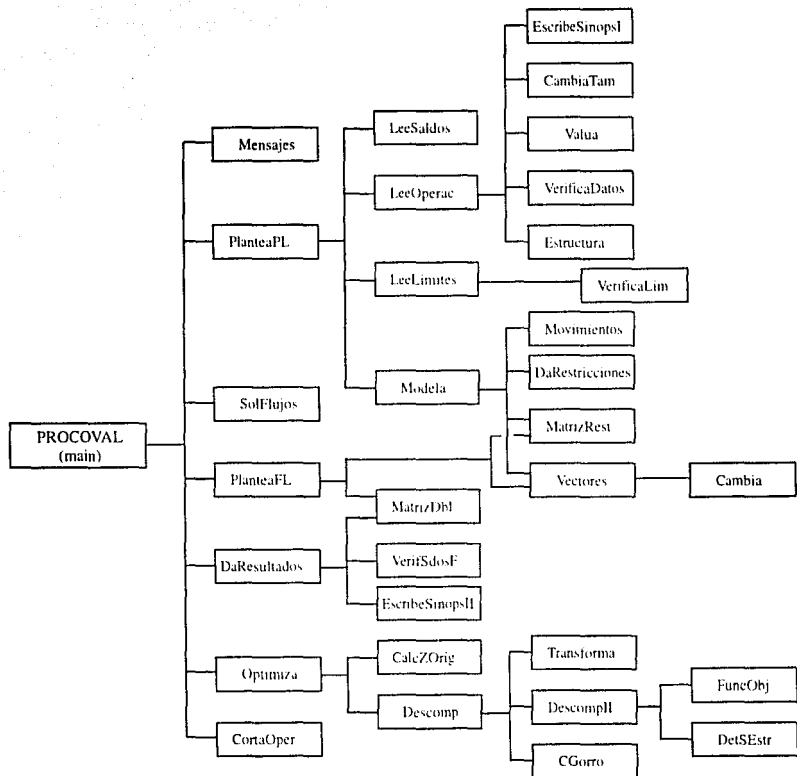


Figura 8.2: Diagrama de Estructura

De cualquier forma, los archivos fuentes del sistema, y algunos archivos de datos, pueden obtenerse solicitándolos al autor de esta tesis en la dirección de e-mail:

sasa@lya.fciencias.unam.mx.

Capítulo 9

Ejemplos y resultados

En este capítulo presentamos algunos ejemplos de la utilización del PROCOVAL para determinar, a partir de un conjunto de operaciones pendientes, cuáles pueden liquidarse. Primero presentamos un ejemplo hipotético muy sencillo para tratar de explicar la esencia del método de solución. En un segundo ejemplo, tomamos las operaciones que terminaron pendientes en el módulo de valores del INDEVAL, en un día típico de fines de 1996, para analizar el proceso realizado por el PROCOVAL sobre las mismas. Finalmente, presentamos una tabla que muestra los logros obtenidos al usar este sistema.

9.1 Ejemplo hipotético

Supongamos que hay tres bancos (Banco *A*, Banco *B* y Banco *C*) que realizan operaciones con dos tipos de títulos diferentes (títulos tipo *a* y tipo *b*). Supongamos, además, que al final del día hay 6 (seis) operaciones pendientes de liquidación y que no se otorga crédito a los participantes.

En las siguientes tablas presentamos, de manera detallada, las operaciones pendientes y los saldos que tiene cada banco al cierre del ejercicio.

Número de Operación	Banco Vendedor	Banco Comprador	Cantidad de Títulos	Tipo de Título	Precio por Título (\$)
1	<i>A</i>	<i>B</i>	10	<i>a</i>	9
2	<i>A</i>	<i>C</i>	15	<i>a</i>	9
3	<i>C</i>	<i>B</i>	17	<i>a</i>	9
4	<i>B</i>	<i>A</i>	20	<i>b</i>	8
5	<i>B</i>	<i>C</i>	30	<i>b</i>	8
6	<i>A</i>	<i>C</i>	10	<i>b</i>	8

	Banco A	Banco B	Banco C
Saldo en títulos tipo <i>a</i>	13	10	4
Saldo en títulos tipo <i>b</i>	10	50	0
Saldo de Efectivo (\$)	80	0	50

Una forma de representar estas operaciones es con una red de flujos [2], construida de la siguiente manera: se usa un nodo por cada banco participante y una arista dirigida por cada operación. Para cada operación pendiente en la que el banco i vende títulos al banco j se usa una arista que va de i a j . La cantidad máxima de flujo que puede ir por esa arista es igual a la cantidad de títulos de la operación. Como hay operaciones con dos distintos tipos de títulos, se requieren dos tipos diferentes de aristas, en la gráfica de la figura 9.1, las aristas oscuras representan operaciones con títulos tipo a y las claras operaciones con títulos tipo b .

Si intentáramos liquidar las operaciones, una por una, verificando que se cumplan las reglas vigentes, resultaría que ninguna de ellas puede liquidarse. La razón principal por la cual las operaciones están pendientes de liquidación, es que ninguno de los bancos participantes tiene el efectivo suficiente para comprar todos los títulos que le quieren vender.

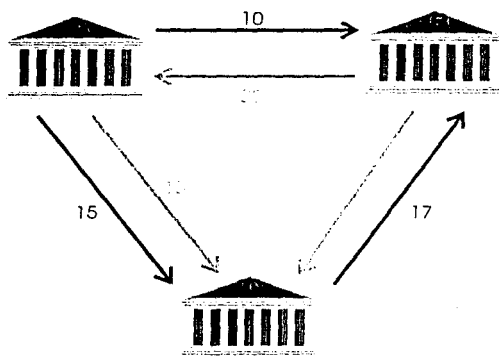


Figura 9.1: Operaciones pendientes de liquidación

Al observar las operaciones 2 y 3, vemos que el banco *C* quiere comprar 15 títulos tipo *a* y, al mismo tiempo, quiere vender 17 de esos títulos. En caso de que estas operaciones pudieran liquidarse, el dinero que gaste el banco *C* en la operación 2 es menor que el dinero que recibiría en la operación 3 y por tanto, no solamente no se sobregiraría, sino que su saldo de efectivo aumentaría. En la figura 9.2 se presentan estas operaciones, junto con el saldo en títulos tipo *a* de cada banco.

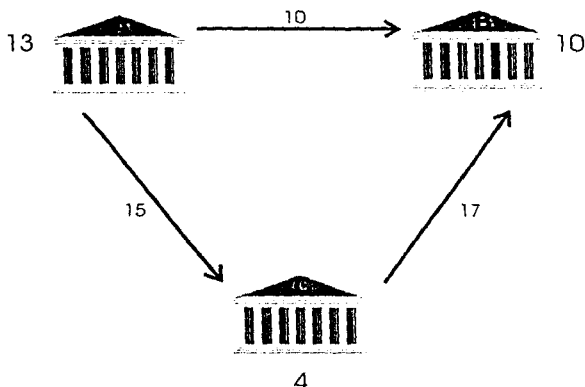


Figura 9.2: Operaciones con títulos tipo *a*

El problema, al intentar pasar estas dos operaciones, es que el banco *A* no tiene el saldo suficiente en títulos tipo *a* para venderlos en la operación 2, ya que sólo dispone de 13 títulos.

Con las modificaciones que propone Banxico para liquidar el mercado, el banco *A* puede vender sus 13 títulos tipo *a* al banco *C*, quien sumará estos títulos a los 4 que tiene en sus cuentas y obtendrá así los 17 que requiere para liquidar la operación 3. Con esto, el banco *C* incrementará en \$36 su saldo de efectivo, terminando así con \$86 y 0 (cero) títulos tipo *a* (ver figura 9.31).

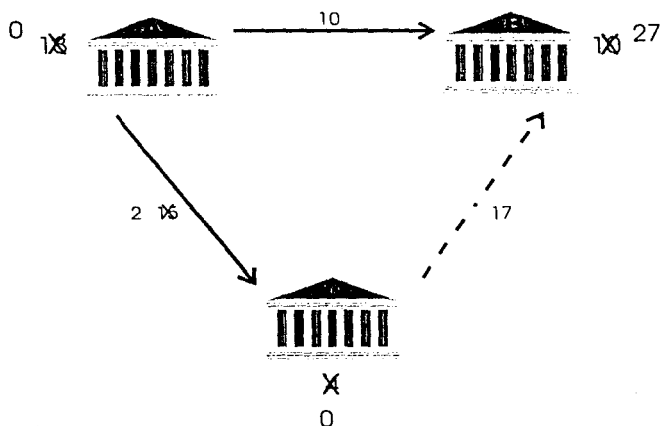


Figura 9.3: Movimientos de saldos en títulos tipo *a*, al liquidar las operaciones

Hasta aquí hemos visto que sería deseable liquidar parcialmente (con 13 títulos) la operación 2 y totalmente la 3, pues esto aumentaría el saldo en efectivo del banco *C*, mejorando su situación y liquidando dos operaciones más en el mercado. Sin embargo, observa que el banco *B* no tiene los \$153 que se requieren para liquidar los 17 títulos de la operación 3. ¿Se puede hacer algo para que el saldo en efectivo del banco *B* sea, al menos, de \$153 y podamos así liquidar las operaciones propuestas?

Sabemos que en caso de pasar las operaciones 2 y 3 de la manera sugerida, el banco *C* tendría \$86. Esta cantidad le permite comprar los 10 títulos tipo *b* que el banco *A* le vende en la operación 6, con la ventaja de que el banco *A* no tiene problema para hacer esta venta, pues su saldo inicial en este tipo de títulos es, precisamente, 10.

Esta operación daría \$80 al banco *A* que, sumados a los \$80 que tiene como saldo inicial de efectivo, darían \$160. Esos \$160 los puede utilizar el banco *A* para comprar los 20 títulos tipo *b* que le vende el banco *B*. Dado que el saldo inicial en títulos tipo *b* del banco *B* es 50, este banco puede hacer la venta sin ningún problema, obteniendo así \$160. De esta manera, al liquidar totalmente las operaciones 6 y 4 se podría aumentar el saldo en efectivo del banco *B* a \$160, que es mayor que los \$153 que se requerían (ver figura 9.4).

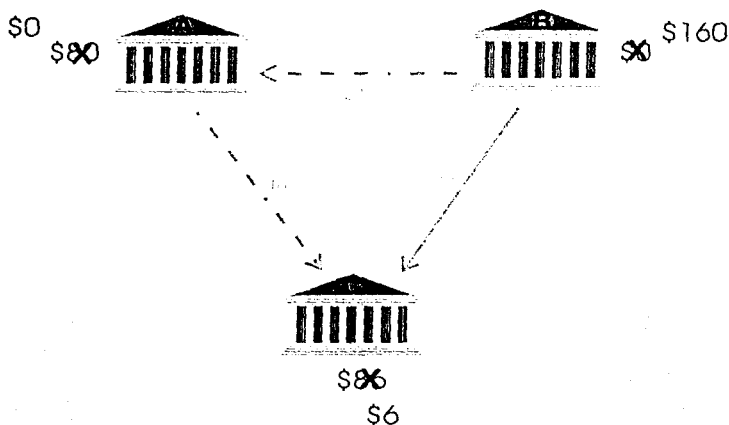


Figura 9.4: Movimientos de efectivo al liquidar las operaciones de títulos tipo *b*

Por lo tanto, la manera óptima de liquidar las operaciones pendientes es: pasar las operaciones 2, 4 y 6 de manera total, y la 3 de manera parcial. Logrando, en consecuencia, transferir 60 de los 102 títulos que estaban atorados al principio, despejando además, 4 de las 6 operaciones pendientes (ver la figura 9.5).

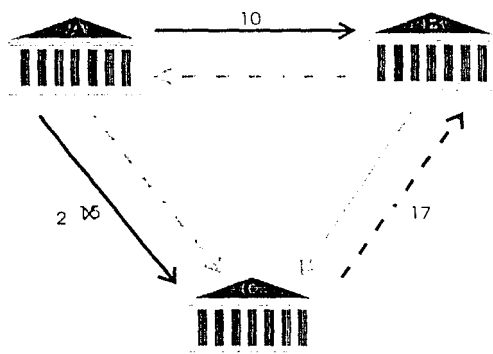


Figura 9.5: Liquidación óptima

Una de las mayores ventajas de modelar el problema de la liquidación de operaciones pendientes con un esquema como el presentado, es que la verificación de las reglas se hace globalmente para todo el conjunto de las operaciones, y no para cada operación en particular. La ganancia es muy clara si consideras la liquidación que acabamos de proponer, observando lo siguiente. Para liquidar las operaciones 2 y 3 de nuestro ejemplo, hay que aumentar el saldo del banco C. Sin embargo, la manera de aumentar su saldo es liquidar las operaciones 4 y 6. Esto implica que la liquidación de estas últimas operaciones requiere que se liquiden las primeras, y éstas no pueden liquidarse hasta que lo hagan aquellas. Si se verifica cada liquidación de manera independiente no puede pasar ninguna

de las operaciones, pero al liquidar todas las operaciones juntas, no se violan las reglas de operación.

Debemos notar que en este ejemplo no nos hemos preocupado de que los bancos hayan tenido suficientes títulos para garantizar el pago de los posibles créditos que recibieran, ya que su límite de crédito era cero. Esto simplificó el problema pues nos olvidamos de las posiciones iniciales de los bancos, del valor como garantía de los títulos, de que los títulos a trapasar estuvieran en cuentas propias, etc.

9.2 Ejemplos reales

Dentro de las pruebas a las que se ha sometido el PROCOVAL, la parte más importante consiste en procesar las operaciones que terminan pendientes de liquidación en el módulo de valores del INDEVAL. Al final del día, el INDEVAL genera los archivos de entrada necesarios y los envía a Banxico para ser procesados con el PROCOVAL.

Es muy importante recordar que, tal y como se expuso en el capítulo anterior, la versión del PROCOVAL utilizada en las pruebas es una modificación del sistema originalmente propuesto.

A manera de ejemplo, expondremos la manera en que el PROCOVAL procesa las operaciones de un día típico. Tomaremos un día de finales de 1996, en el que la cantidad de operaciones pendientes recibidas es la más cercana al promedio de operaciones que se registraron en el PROCOVAL durante todo un mes.

De acuerdo a la metodología de solución, el primer paso consiste en leer los archivos de entrada para conocer, detalladamente, las operaciones pendientes, los saldos y los límites de crédito de los participantes involucrados en estas operaciones. Después de leer los archivos se tiene la siguiente información:

Número de operaciones pendientes:	57
Número de instituciones participantes:	19
Diferentes tipos de títulos operados:	26
Cantidad de títulos pendientes:	1,764'411,146
Monto pendiente de liquidación:	\$ 1,555'073,905
Porcentaje de préstamo que se debe garantizar colateralmente:	70 %

Para hacer una representación gráfica de estas operaciones, tendríamos que formar una red con 19 nodos y 57 aristas de 26 tipos diferentes. Obviamente, sería muy difícil tratar de analizar una gráfica como ésta de la misma manera que hicimos con el ejemplo anterior.

En este caso, cada participante tiene una línea de crédito asignada por Banxico, lo cual modifica un poco la manera de determinar las operaciones que pueden liquidarse. Si el saldo en efectivo de los participantes es suficiente para liquidar sus operaciones de compra, la determinación de las operaciones que pasan se hace de la misma forma que en el ejemplo anterior. En cambio, si algún participante no tuviera para pagar una operación, pero su línea de crédito es suficientemente amplia, se puede utilizar ésta para extenderle un préstamo, verificando que el valor de los títulos que tiene el participante en sus cuentas propias permita garantizar el pago del 70% del dinero que se le preste.

Después de leer las operaciones, se plantea el problema de programación lineal correspondiente (ver el paso 3 de la metodología) y se resuelve. La solución óptima de este ejemplo no es entera, lo que obliga a plantear el problema de flujos. Al resolver este último problema, se obtiene la cantidad entera de títulos que debe liquidarse en cada operación. Haciendo una sinopsis de la solución obtenida encontramos lo siguiente:

Número de operaciones que pasan totalmente:	35
Número de operaciones que pasan parcialmente:	6
Cantidad de títulos que se liquidan:	719'212,099
Monto liquidado en las operaciones:	\$ 738'381,279

En términos porcentuales, con la solución obtenida por el PROCOVAL es posible liquidar al menos un título en 71.93% de las operaciones y se pueden traspasar el 40.76% de títulos pendientes de liquidación. Además, el valor de las operaciones que se liquidan con esta solución, representa el 47.48% del monto total pendiente de liquidación.

En conclusión, utilizando la metodología propuesta se puede liquidar casi la mitad del monto pendiente. Hay que señalar que durante toda la etapa de pruebas a la que se sometió esta versión del PROCOVAL los resultados obtenidos fueron similares.

En la siguiente tabla presentamos un resumen del proceso realizado por el sistema con las operaciones pendientes de un mes completo.

PROCESAMIENTO DE LAS OPERACIONES PENDIENTES DE OCTUBRE DE 1996

COLATERAL A GARANTIZAR: 50% del 1 al 6, 60% del 7 al 20 y 70% del 21 al 31 de octubre.

DIA	OPERACIONES PROCESADAS	TITULOS OPERADOS	MONTO OPERADO	OPERACIONES QUE PASAN		PORCENTAJES QUE PASAN		
				COMPLETAS	PARCIALES	OPERACIONES	TITULOS	MONTO
1	25	2,191,008,499	2,093,874,675	5	6	41.00	35.80	34.40
2	31	189,337,132	1,056,316,806	9	3	38.71	56.37	12.39
3	87	3,290,607,944	3,224,677,773	60	12	82.76	90.51	91.13
4	62	954,164,094	1,547,342,083	28	11	62.90	25.63	14.51
7	71	1,055,821,995	1,786,263,150	44	10	76.06	23.74	4.47
8	60	1,347,206,569	988,509,856	29	4	55.00	13.40	2.02
9	53	153,510,160	332,176,855	36	3	73.58	91.11	100.00
10	14	407,009,837	546,357,491	31	4	79.55	19.14	54.41
11	65	5,347,628,048	5,696,610,400	30	9	60.00	82.81	80.88
14	32	228,632,679	293,459,654	14	1	46.88	45.74	34.08
15	45	1,554,973,619	1,338,636,675	31	3	75.36	61.52	53.63
16	60	432,765,004	1,039,808,045	26	8	56.67	34.09	23.90
17	64	1,523,351,250	1,958,164,844	41	3	68.75	4.83	15.48
18	84	7,101,563,745	2,923,612,572	60	5	77.38	80.80	79.85
21	52	7,409,476,068	5,540,347,686	32	6	73.08	35.66	26.78
22	37	2,451,797,667	1,800,272,427	28	3	83.78	74.77	87.22
23	32	962,204,783	1,033,774,325	24	3	84.38	82.92	79.00
24	117	1,444,274,721	3,023,484,853	36	21	48.72	8.74	33.05
25	37	1,764,411,436	1,555,073,995	35	6	71.93	40.76	47.48
28	45	1,333,858,593	1,233,136,561	37	1	84.44	14.03	3.14
29	42	23,822,396	0	41	0	97.62	98.35	0.00
30	40	1,694,368,801	1,379,772,195	34	2	90.00	29.72	14.55
31	63	4,867,753,783	2,832,078,257	20	10	47.62	19.36	39.36
PROMEDIO	55	2,079,654,196	1,878,859,916	32	6	68.22	49.92	46.32

Capítulo 10

Conclusiones

A lo largo de este trabajo, propusimos una metodología que nos permite decidir qué operaciones se pueden liquidar en el mercado de títulos mexicano. Dicha metodología fue probada diariamente, durante cuatro meses, y los resultados obtenidos son contundentes.

La prueba principal consistió en utilizar una versión simplificada del PROCOVAL para procesar las operaciones pendientes de liquidación en el módulo de valores del INDEVAL, al final del día. Con las soluciones obtenidas por el sistema, en promedio, fue posible liquidar operaciones cuyo valor representa más del 40% del monto pendiente. En otras palabras, el importe de las operaciones que pudieron liquidarse gracias al PROCOVAL, correspondió al 40% del total del monto pendiente de liquidación. Además, estas soluciones permitieron liquidar, en promedio, al menos un título en 60% de las operaciones pendientes y se logró pasar el 50% de los títulos pendientes de liquidación.

En resumen, aplicando la metodología que propusimos, se pudieron hacer liquidaciones en más de la mitad de las operaciones, con lo que se saldó aproximadamente la mitad del monto pendiente y se traspasó la mitad de los títulos, en promedio.

Estos resultados representan una mejora considerable en la liquidación del mercado de títulos si se comparan con los obtenidos por otros procedimientos que se han estado utilizando. Más aún, creemos que si usa el PROCOVAL en el transcurso del día, se presentarán mejoras similares en cada corte, con lo que el número de operaciones que terminen pendientes de liquidación al cierre del ejercicio será mucho menor.

Los tiempos totales de corrida del sistema nunca excedieron de un par de minutos, es decir, el tiempo que tomó al PROCOVAL leer los archivos de entrada, determinar las operaciones que pueden liquidarse y escribir los archivos de salida fue menor que 2 minutos, siempre. Cuando el sistema tome los datos de entrada de manera automática estos tiempos pueden reducirse aún más, por lo cual, no existe el riesgo de entorpecer el mercado al utilizar el PROCOVAL para procesar las operaciones pendientes de liquidación.

Por otro lado, observamos que usando el PROCOVAL los créditos requeridos por los participantes fueron menores que los necesitados hasta ahora. Esto es una consecuencia directa de que en la metodología propuesta todas las operaciones se procesan al mismo tiempo, lo cual permite que el dinero obtenido al liquidar un operación sea utilizado, inmediatamente, por el participante que vende para respaldar una operación de compra, y dado que el objetivo de los programas que se plantean es maximizar el valor de las operaciones que pasen, ese dinero se puede invertir en la operación que más convenga a todo el mercado en su conjunto.

En conclusión, se alcanzaron los dos objetivos principales de la metodología: uno, encontrar el conjunto óptimo de operaciones que pueden liquidarse, sin entorpecer el mercado; y dos, reducir el crédito que otorga Banco de México en el mercado de títulos.

Aparentemente, la exigencia de títulos para garantizar los créditos que otorga Banco de México no fue un factor determinante en la liquidación de las operaciones del mercado, aun cuando esta política es nueva en nuestro país y a pesar de que el porcentaje de préstamo que se debe garantizar es cada vez más alto. Analizando las operaciones recibidas durante el período de pruebas observamos que, cuando los participantes creen que necesitarán crédito para liquidar sus compras, hacen operaciones de traspaso a sus cuentas propias para cubrir las garantías requeridas.

Según creemos, la modificación que más afectará a las instituciones financieras es la que permite liquidar operaciones, en papel bancario, de manera parcial. Sin embargo, estamos seguros de que estas instituciones no pondrán objeciones a la implantación de esta nueva regla, puesto que los resultados obtenidos son claramente ventajosos para todos.

El PROCOVAL es un sistema altamente transportable, ya que fue desarrollado y compilado en diferentes plataformas.

La capacidad del sistema permite procesar varias decenas de miles de operaciones por corrida. No obstante, es posible que cuando el mercado mexicano crezca, esta capacidad sea insuficiente. Previendo esta situación, se ha sugerido desarrollar una versión del PROCOVAL que funcione de manera distribuida, o sea, que corra en distintas máquinas. La principal ventaja de una implantación como ésta es que se podrán resolver problemas muy grandes distribuyendo el trabajo entre distintos procesadores. El tiempo efectivo de uso de cada procesador será muy breve, y en casos extremos, para cada tipo de título operado se usará uno de estos procesadores.

Es importante subrayar que la aplicación de técnicas de investigación de operaciones puede ser muy útil para resolver problemas financieros, aún cuando las técnicas empleadas no sean demasiado sofisticadas y los problemas sean bastante complejos, como en nuestro caso.

A manera de reflexión final, aconsejamos a quien desee desarrollar aplicaciones de tipo financiero, como la presentada, que no inicie esta labor sin conocimientos sólidos de las técnicas de investigación de operaciones, de finanzas y de computación, o sin la ayuda de un grupo de expertos en estas materias. Estos conocimientos son indispensables pues, como se ha visto, es necesario entender bien el problema que se quiere abordar; después, hay que modelarlo correctamente y, finalmente, hay que hacer una buena implantación para resolverlo. Si se puede entender el problema sin poder modelarlo, o se puede modelar sin poder resolverlo, o se puede resolver sin entender la solución, será muy difícil obtener un resultado satisfactorio. Es necesaria la conjunción y el dominio de todas las etapas del proceso de resolución.

Apéndice A

Aplicaciones financieras

Operations Research

1. Zipkin, P. *The Structure of Structured Bond Portfolio Models*. Operations Research. Supplement 1 jan.-feb., Vol. 40, 1992. Baltimore, U.S.A.
2. Cyert, R. *Knowledge and Economic Development*. Operations Research. No. 1 jan.-feb., Vol. 39, 1991. Baltimore, U.S.A.
3. Scarf, H. *Mathematical Programming and Economic Theory*. Operations Research. No. 3 may-june, Vol. 38, 1990. Baltimore, U.S.A.
4. Dada, M.; Srikanth K. *Monopolistic Pricing and the Learning Curve: An Algorithmic Approach*. Operations Research. No. 4 july-aug., Vol. 38, 1990. Baltimore, U.S.A.

Management Science

1. Bell, D. *Risk, Return, and Utility*. Management Science, Journal of the Institute of Management Sciences. No. 1 jan., Vol. 41, 1995. U.S.A.
2. Chowdhry, B. *Corporate hedging of Exchange Risk When Foreign Currency Cash Flow is Uncertain*. Management Science, Journal of the Institute of Management Sciences. No. 6 june, Vol. 41, 1995. U.S.A.
3. Glynn, P.; Iglehart, D. *Trading Securities Using Trailing Stops*. Management Science, Journal of the Institute of Management Sciences. No. 6 june, Vol. 41, 1995. U.S.A.
4. Lee, S.; Chang, K. *Mean-Variance-Instability Portfolio Analysis: A Case of Taiwan's Stock Market*. Management Science, Journal of the Institute of Management Sciences. No. 7 july, Vol. 41, 1995. U.S.A.

5. Golany, B.; Tamir, E. *Evaluating Efficiency-Effectiveness-Equality Trade-offs: A Data Envelopment Analysis Approach*. Management Science, Journal of the Institute of Management Sciences. No. 7 july, Vol. 41, 1995. U.S.A.
6. Barranquand, J. *Numerical Valuation of High Dimensional Multivariate European Securities*. Management Science, Journal of the Institute of Management Sciences. No. 12 dec., Vol. 41, 1995. U.S.A.
7. Hilliard, J.; Kau, J.; Keenan D.; Muller, W. *Pricing a Class of American and European Path Dependent Securities*. Management Science, Journal of the Institute of Management Sciences. No. 12 dec., Vol. 41, 1995. U.S.A.
8. Eun, C; Resnick, B. *International, Diversification of Investment Portfolios: U.S. and Japanese's Perspectives*. Management Science, Journal of the Institute of Management Sciences. No. 1 jan., Vol. 40, 1994. U.S.A.
9. Board, J.; Sutcliffe, C. *Estimation Methods in Portfolio Selection and the Effectiveness of Short Sales Restrictions: U. K. Evidence*. Management Science, Journal of the Institute of Management Sciences. No. 4 apr., Vol. 40, 1994. U.S.A.
10. Mean, P; Curran, M. *Valuing Asian and Portfolio Options by Conditioning on the Geometric*. Management Science, Journal of the Institute of Management Sciences. No. 12 dec., Vol. 40, 1994. U.S.A.
11. Simaan, Y. *Portfolio Selection And Asset Pricing. Three-parameter Framework*. Management Science, Journal of the Institute of Management Sciences. No. 5 may, Vol. 39, 1993. U.S.A.

Journal of the Operational Research Society

1. Yeh, Q. *The Application of Data Envelopment Analysis in Conjunction with Financial Ratios for Bank Performance Evaluation*. Journal of the Operational Research Society. No. 8 aug., Vol. 47, 1996. U.K.
2. Li, S. *A Satisficing Chance Constrained Model in the Portfolio Selection of Insurance Lines and Investments*. Journal of the Operational Research Society. No. 9 sep., Vol. 46, 1995. U.K.
3. Cabral, J; Guimaraes R. *Dynamic Calibration of Price-Trend Parameters for Commodity Futures Trading*. Journal of the Operational Research Society No. 8 aug., Vol. 45, 1994. U.K.
4. Hui, T.; Kwan, E.; Lee, C. *Optimal Portfolio Diversification: Empirical Bayes Versus Classical Approach*. Journal of the Operational Research Society. No. 11 nov., Vol. 44, 1993. U.K.
5. Edmond, E.; Bedri, F. *A Study of Risk-Exposed Cotton Trading*. Journal of the Operational Research Society. No. 5 may, Vol. 44, 1993. U.K.

Interfaces

1. Mulvey, J. *Generating Scenarios for the Towers Perrin Investment System*. Interfaces. No. 2 mar.-apr., Vol. 26, 1996. U.S.A.
2. Sodhi, M. *Development of a DSS for Fixed-Income Securities using OOP*. Interfaces. No. 2 mar.-apr., Vol. 26, 1996. U.S.A.
3. Cari no, D.; Kent, T.; Myers, D.; Stacy, C.; Sylvanus, M.; Turner, A.; Watanabe, K.; Ziemba, W. *The Russell-Yasuda Kasai Model: An Asset/Liability Model for a*

- Japanese Insurance Company using Multistage Stochastic Programming.* Interfaces. No. 1 jan.-feb., Vol. 24, 1994. U.S.A.
4. Holmer, M. *The Asset-Liability Management Strategy System at Fannie Mae.* Interfaces. No. 3 may-june, Vol. 24, 1994. U.S.A.
 5. Mulvey, J. *An Asset-Liability Investment system.* Interfaces. No. 3 may-june, Vol. 24, 1994. U.S.A.
 6. Lilly, C. *Optimal Lease Structuring at GE Capital.* Interfaces. No. 3 may-june, Vol. 24, 1994. U.S.A.
 7. Ziemba, W. *Investing in the Turn-of-the-year Effect in the U.S. Futures Markets.* Interfaces. No. 3 may-june, Vol. 24, 1994. U.S.A.
 8. Weigel, E. *Super Units and Super Shares.* Interfaces. No. 3 may-june, Vol. 24, 1994. U.S.A.
 9. Golub, B.; Pohlman, L. *Mortgage Prepayments and an Analysis of the Wharton Prepayment Model.* Interfaces. No. 3 may-june, Vol. 24, 1994. U.S.A.
 10. Huysmans, J. *Using the Systems Approach to Increase Management Science Impact on Business.* Interfaces. No. 5 sept.-oct., Vol. 24, 1994. U.S.A.

Symposium über operations research

1. Kort, P. *Dynamic Investment Policies of the Firm.* Symposium über operations research. sept. 5, 1996. Braunschweig, Deutschland.
2. Neck, R. *Optimal Fiscal Policies for Austria.* Symposium über operations research. sept. 5, 1996. Braunschweig, Deutschland.

3. Bauerova, D. *Fuzzy Mathematics of Finance*. Symposium über operations research. sept. 6, 1996. Braunschweig, Deutschland.
4. Wu, Y. *A Forecast Study of Applying the Neural Network in the Optimal Investment and Hedging Combination Model by Taking the Interest Rate as the Main Axis*. Symposium über operations research. sept. 6, 1996. Braunschweig, Deutschland.

Apéndice B

Código de la descomposición

Dantzig-Wolfe

Pag: 1

```

1 #include "dvpsmplx01.h"
2
3 void Transforma(WORD m, WORD n, wVECTOR &ColAnt, WORD k, dMAT &a,
4 PdVECTOR xt, dVECTOR &St)
5 {
6     /* esta rutina obtiene St del bloque k */
7
8     // St = (at * xt)
9
10    pElemento ap;
11    WORD i, j;
12
13    // inicializa
14
15    for (i = 0; i < m; i++)
16        St[i] = 0;
17
18    // barre las columnas de la matriz conjunta, para el bloque k
19
20    for (i = 0; i < n; i++)
21    {
22        ap = a.columna(ColAnt[k] + i);
23        while (ap != NULL)
24        {
25            // St = at * xt
26
27            j = ap->indice;
28            St[j] += ap->valor * (*xt)[i];
29            ap = ap->siguiente;
30        }
31    }
32 } /* end Transforma() */
33
34 void FuncObj(WORD nn, WORD k, pBLOQUE *bloque, dMAT &a,
35 wVECTOR &ColAnt, dVECTOR &pi, PdVECTOR *cprit)
36 {
37     /* esta rutina calcula la nueva funcion objetivo (cprit)
38     del bloque k */
39
40     // cprit = (pi * a) - ct
41
42     WORD i, j;
43     pElemento ap;
44     PdVECTOR apc;
45
46     // obtiene la funcion objetivo del bloque
47
48     apc = (*bloque[k]).DameC();
49
50     for (j = 0; j < nn; j++)
51     {
52         // cprit = - ct
53
54         (*cprit[k])[j] = -(*apc)[j];
55
56         // obtiene la columna de la matriz (a)
57
58         ap = a.columna(j + ColAnt[k]);
59         while (ap != NULL)
60

```

Pag: 2

```

61         (
62             // cprit = (pi * a)
63         )
64         i = ap->indice;
65         (*cprit[k])[j] += (pi[i] * ap->valor);
66         ap = ap-> siguiente;
67     }
68 }
69 ) /* end FuncObj() */
70
71
72 double CGorro(WORD k, WORD n, pBLOQUE *bloque, dVECTOR &xt)
73 {
74     /* esta funcion obtiene el valor de c_gorro.
75     para la funcion objetivo del PROGMAESTRO.
76     a partir de el vector solucion del bloque k */
77
78     // cgorro = ct * xt
79
80     WORD          i;
81     PdVECTOR      apc;
82     double        cgorro;
83
84     // inicializa
85
86     cgorro = 0;
87
88     // obtiene la funcion objetivo del bloque k
89
90     apc = (*bloque[k]).DameC();
91     for (i = 0; i < n; i++)
92     {
93         // cgorro = ct * xt
94
95         if (fabs(xt[i]) > EPS)
96             cgorro += (*apc)[i] * xt[i];
97     }
98
99     return (cgorro);
100 } /* end CGorro */
101
102
103 void DetSEstr(WORD M, WORD m, WORD max, pBLOQUE *bloque,
104             WORD &ndif0, dMAT &a, PdVECTOR x, dVECTOR &SEstr,
105             double &cEstr, wVECTOR &InDif0, wVECTOR &ColAnt)
106 {
107     /* esta rutina determina S_estrella
108     que formara parte del programa maestro */
109
110     // SEstr = a * xt
111
112     WORD          i, j, n;
113     Elemento      *ap;
114     PdVECTOR      apc;
115
116     // max es el bloque asociado a la columna que debe entrar
117
118     n = (*bloque[max]).N();
119
120     // inicializa

```

Pag: 3

```

121
122     for (i = 0; i < M; i++)
123         SEstr[i] = 0;
124
125     // barre las columnas del bloque max en la matriz conjunta
126
127     for (i = 0; i < n; i++)
128     {
129         ap = a.columna(i + ColAnt[max]);
130         while (ap != NULL)
131         {
132             // SEstr = a * xt
133
134             j = ap->indice;
135             SEstr[j] += ap->valor * (*x)[i];
136             ap = ap->siguiente;
137         }
138     }
139
140     // pone 1 en la restriccion del bloque max
141
142     if ((*bloque[max]).CoraSup())
143         SEstr[m + max] = 1;
144
145     // determina los elementos diferentes de cero que entraran
146     // al programa maestro
147
148     NDif0 = 0;
149     for (i = 0; i < M; i++)
150         if (fabs(SEstr[i]) > EPS)
151         {
152             InDif0[NDif0] = i;
153             NDif0++;
154         }
155
156     /* calcula c_estrella asociado a la columna s_estrella */
157
158     // cEstr= ct * xt
159
160     cEstr = 0;
161
162     // obtiene ct del bloque max
163
164     apc = (*bloque[max]).DameC();
165     for (i = 0; i < n; i++)
166     {
167         // cEstr = ct * xt
168
169         if (fabs((*x)[i]) > EPS)
170             cEstr += (*apc)[i] * (*x)[i];
171     }
172     } /* end DetSEstr */
173
174
175 int descompII(WORD M, WORD m, pBLOQUE *bloque, PROGEQRED &ProgMaestro,
176             WORD NumTit, wVECTOR &ColAnt, dMAT &a)
177 {
178     /*
179     esta funcion realiza el ciclo principal
180     del algoritmo de descomposicion Dantzig-Wolfe */

```


Pag: 4

```
181
182 double          CBarEs, cEstr, Zt;
183 long double     CMax;
184 WORD            k, nn, mm;
185 WORD            max, NDif0;
186 BYTE           optimo;
187 dVECTOR         SEstr(M, "SEstr");
188 wVECTOR         InDif0(M, "InDif0");
189 PdVECTOR        *cprit, *x;
190 PdVECTOR        Pi;
191
192 Pi = ProgMaestro.DamePi();
193
194 // inicializa cprimas
195
196 cprit = new PdVECTOR(NumTit);
197
198 if(cprit == NULL)
199 {
200     // no hay memoria para cprit "
201     return (1600);
202 }
203
204 // inicializacion de las soluciones de cada iteracion
205
206 x = new PdVECTOR(NumTit);
207
208 if(x == NULL)
209 {
210     // no hay memoria para x
211     return (1601);
212 }
213
214 // inicia el ciclo principal
215
216 do
217 {
218     CMax = -INFINITO;
219     // para cada bloque
220     for (k = 0; k < NumTit; k++)
221     {
222         // obtiene las dimensiones de cada bloque
223         mm = (*bloque[k]).M();
224         nn = (*bloque[k]).N();
225         // aparta memoria
226         cprit[k] = new dVECTOR((2 * mm) + nn, "cprit");
227         // genera nuevas funciones objetivo para el bloque k
228         FuncObj(nn, k, bloque, a, ColAnt, *Pi, cprit);
229         // resuelve los bloques con la nueva funcion objetivo
230         // y guarda las soluciones
231     }
232 }
233
234
235
236
237
238
239
240
```

Pag: 5

```
241         x[k] = new dVECTOR(nn, "soluciones");
242
243         x[k] = (*bloque[k]).SolucBloque(*cprbit[k]);
244
245         // calcula C_Barra de cada bloque
246
247         Zt = (*bloque[k]).Z();
248
249         CBarEs = Zt + (*Pi)[m + k];
250
251         // prueba optimalidad con C_Barra
252
253         if (CBarEs > 0)
254         {
255             // la columna asociada a C_Barra es candidato
256             // a entrar a la base
257
258             if (CBarEs > CMax)
259             {
260                 // .. entrara el maximo de los candidatos C_Barra
261
262                 CMax = CBarEs;
263                 max = k;
264             }
265         }
266     }
267
268     // si el maximo de los C_Barra es menor o igual a cero,
269     // estamos en el optimo
270
271     optimo = (CMax < EPS);
272
273     if (!optimo)
274     {
275         // NO estamos en el optimo, i. e.
276         // hay C_Barra que puede entrar a la base
277
278         // guarda la sol. que entra (asociada al bloque max)
279
280         nn = ProgMaestro.N();
281         (*bloque[max]).GuardSolBlo(nn, x[max]);
282
283         // calcula columna entrante S_Estrella
284
285         DetSEstr(M, m, max, bloque, NDif0, a, x[max], SEstr, cEstr,
286                 InDif0, ColAnt);
287
288         // agrega la columna S_Estrella al programa maestro ...
289
290         ProgMaestro.AumentaVar(NDif0, InDif0, SEstr, cEstr);
291
292         // .. y resuelve el nuevo programa
293
294         ProgMaestro.SolucionPrin();
295
296         // actualiza pi
297
298         Pi = ProgMaestro.DamePi();
299
300     }
```

Pag: 6

```

301         ) /* end NO optimo */
302
303     ) while (!optimo);
304
305     // devuelve FALSE si no hubo ningun problema y se encontro el
306     // optimo
307
308     return (FALSE);
309 } /* end descomp II */
310
311
312 int descomp(WORD m, WORD n, WORD m1, WORD m2, WORD m3,
313            pBLOQUE *bloque, WORD NumTit, wVECTOR &ColAnt, dMAT &a,
314            PdVECTOR *ct, dVECTOR &b, PdVECTOR *xopt)
315 {
316     // ESTA FUNCION IMPLANTA EL ALGORITMO DE DESCOMPOSICION
317     // DE DANTZIG-WOLFE
318
319     WORD NDif0, M, mn, nn;
320     WORD i, k, NVar, N;
321     dVECTOR B(m + NumTit, "B");
322     dVECTOR St(m, "st");
323     wVECTOR InDif0(n, "InDif0");
324     dMAT as((NumTit + m1 + m2), "as");
325     int desc;
326     double z;
327
328     // construye la matriz de restricciones del programa maestro
329     // fase I
330
331     for (k = 0; k < NumTit; k++)
332     {
333         // aplica la transformacion St = (at*xt) con la solucion inicial
334         // de cada bloque
335
336         nn = (*bloque[k]).N();
337
338         Transforma(m, nn, ColAnt, k, a, (*bloque[k]).SolucNum(0), St);
339
340         // guarda la columna St obtenida, en la matriz del programa
341         // maestro (as)
342
343         for (i = 0; i < m; i++)
344         {
345             if (fabs(St[i]) > EPS)
346                 as.guarda(k, i, St[i]);
347         }
348
349         // guarda 1 en la restriccion del titulo k
350
351         as.guarda(k, (m + k), 1);
352
353         // la solucion inicial del bloque k esta
354         // asociada con la columna k de la matriz as
355
356         (*bloque[k]).GuardaAsoc(0, k);
357     }
358
359     // agrega variables de holgura para fase I de descomposicion
360

```

Pag: 7

```

361 // hay m1 restricciones menor o igual
362
363 for (i = 0; i < m1; i++)
364     as.guarda((NumTit + i), i, 1);
365
366 // hay m2 restricciones mayor o igual
367
368 for (i = m1; i < (m1+m2); i++)
369     as.guarda((NumTit + i), i, -1);
370
371 // construye vector de lados derechos (B) del Programa Maestro
372
373 for(i = 0; i < m; i++)
374     B[i] = b[i];
375 for (i = m; i < (m + NumTit); i++)
376     B[i] = 1;
377
378 // la base del programa maestro debe tener m+t vectores
379
380 M = m + NumTit;
381 N = NumTit + m1 + m2;
382
383 dVECTOR xub(2*M + N, "xub0");
384 dVECTOR xlb(2*M + N, "xlb");
385 dVECTOR c(2*M + N, "c");
386 Solucion *apsol = NULL;
387 PwVECTOR base;
388
389 // inicializa cotas superiores del progama maestro
390
391 for (i = 0; i < N; i++)
392     xub[i] = INFINITO;
393
394 // define el programa maestro como programa lineal.
395 // al definirlo se determina si tiene solucion factible inicial,
396 // sin necesidad de agregar variables artificiales
397
398 PROGEQRED ProgMaestro(M, N, 0, 0, M, as, B, c, xub, xlb,
399     "ProgMaestro");
400
401 if (!ProgMaestro.HaySol())
402 {
403     // no hay solucion factible inicial. Debe hacerse procedimiento
404     // de fase I, agregando variables artificiales
405
406     NVar = NumTit + m1 + m2;
407
408     N = NVar + m2 + m3;
409
410     // asignamos memoria para los coeficientes de costo que tendran
411     // los bloques en la fase I
412
413     PdVECTOR *cct = NULL;
414
415     cct = new PdVECTOR(NumTit);
416
417     if(cct == NULL)
418     {
419         // no hay memoria para cct
420

```

Pag: 8

```
421         return(1700);
422     }
423
424     // agregamos variables artificiales al programa maestro
425     NDif0 = 1;
426     for (i = 0; i < (m2 + m3); i++)
427     {
428         // (m2+m3) variables artificiales
429
430
431         InDif0[0] = m1 + i;
432
433         // cuyas columnas son vectores E(m1+i)
434
435         St[m1 + i] = 1;
436
437         // y con coef de costo 1
438
439         ProgMaestro.AumentaVar(NDif0, InDif0, St, 1);
440     }
441
442     // inicializa coeficientes de costos por cada bloque
443
444     for (i = 0; i < NumTit; i++)
445     {
446         mm = (*bloque[i]).M();
447         nn = (*bloque[i]).N();
448
449         cct[i] = new dVECTOR((2 * mm) + nn, "cct");
450
451         // cambia coeficientes costos de cada bloque, por ceros
452
453         (*bloque[i]).CambiaC(*cct[i]);
454     }
455
456     // FASE I del Algoritmo de Descomposicion Dantzig-Wolfe
457     // intenta minimizar la suma de las variables artificiales
458
459     desc = descompII(N, m, bloque, ProgMaestro, NumTit, ColAnt, a);
460
461     // determina el valor de la funcion objetivo FASE I
462     // para decidir si hay sol inicial
463
464     z = ProgMaestro.Z();
465
466     // si el valor de la funcion objetivo es mayor que cero
467     // no hay solucion inicial
468
469     if (fabs(z) > EPS)
470     {
471         mensadv("El valor optimo de la FASE I es mayor que cero.");
472         return (1702);
473     }
474 } /* end si no hubo solucion inicial del prog maestro */
475
476 // regresamos los coeficientes de costos originales de cada bloque
477
478 for (k = 0; k < NumTit; k++)
479     (*bloque[k]).CambiaC(*ct[k]);
480
```

Pag: 9

```

481 // determinamos c_gorro asociado a cada columna basica del
482 // programa maestro
483
484 N = ProgMaestro.N();
485 base = ProgMaestro.DameBase();
486
487 for (k = 0; k < NumTit; k++)
488 {
489     nn = (*bloque[k]).N();
490
491     // barre las soluciones de cada bloque
492
493     apsol = (*bloque[k]).BuscaSol(0);
494     while (apsol != NULL)
495     {
496         for (i = 0; i < N; i++)
497         {
498             // calcula c_gorro de la solucion si su columna asociada
499             // es basica
500
501             if ((*base)[i] == apsol->asociada)
502                 c[apsol->asociada] = CGorro(k, nn, bloque, apsol->valores);
503         }
504         apsol = apsol->siguiente;
505     }
506 }
507
508 // la funcion objetivo del Programa Maestro es cgorro
509
510 ProgMaestro.CambiaC(c);
511
512 // como cambio la funcion objetivo del programa maestro,
513 // se debe recalcular el vector de costos asociado a la base
514 // del programa maestro (CB)
515
516 ProgMaestro.CalculaCb(c);
517
518 // lo mismo para el vector pi
519
520 ProgMaestro.CalculaPi();
521
522 // podemos inicial el ciclo principal del algoritmo de
523 // descomposicion, para saber si la solucion actual es optima,
524 // y si no, encontrar el optimo.
525
526 // FASE II del algoritmo de descomposicion Dantzig-Wolfe
527
528 desc = descompII(M, m, bloque, ProgMaestro, NumTit, ColAnt, a);
529
530 // encuentra solucion optima del programa original,
531 // con el teorema de puntos extremos
532
533 ProgMaestro.SolOpt(NumTit, M, bloque, xopt);
534
535 return (desc);
536 } /* end Descomp */
537

```

Referencias Bibliográficas

- [1] R. Ackoff y M. Sasieni. *Fundamentos de Investigación de Operaciones*. Limusa, México, 1979.
- [2] M. Bazaraa y J. Jarvis. *Programación Lineal y Flujo en Redes*. Limusa, México, 1995.
- [3] D. Burras. *The U.S. Payments System: Costs, Pricing, Competition and Risk*. Monograph Series in Finance and Economics, U.S.A., 1984.
- [4] G. Dantzig. *Linear Programming and Extensions*. Princeton University Press, New Jersey, U.S.A., 1963.
- [5] F. Gil. *The Role of the Central Bank in the Payments System: The Case of Mexico*. In Conference on Risk Reduction in Payments, Clearance and Settlement Systems, New York, U.S.A., January 1996.
- [6] F. Hillier y G. Lieberman. *Introducción a la Investigación de Operaciones*. McGraw-Hill, México, 1991.
- [7] B. Kernighan and D. Ritchie. *The C Programming Language*. Prentice-Hall, Englewood Cliffs, U.S.A., 1988.
- [8] J. Prawda. *Métodos y Modelos de Investigación de Operaciones*, vol. 1. Limusa, México, 1979.
- [9] H. Schildt. *Turbo C/C++ 3.1. Manual de Referencia*. McGraw-Hill, España, 1994.