Universidad Loyola de América

UNIVERSIDAD LOYOLA DE AMÉRICA
SISTEMA INCORPORADO-UNAM CLAVE 8911

U L A

“CONTRIBUCIONES AL MODELADO DE SELECCIÓN DE ATRIBUTOS APLICANDO OPTIMIZACIÓN”

T E S I S

Q U E COMO R E Q U I S I T O
PARA OBTENER EL GRADO DE:
LICENCIADA EN ACTUARÍA

P R E S E N T A:
DIANA LISSETE MARTÍNEZ CUENCA

DIRECTOR DE TESIS:
M.C. ISAÍAS GUILLEN MOYA

CUERNAVACA MOR. **OCTUBRE DE 2006**

DEDICATORIAS

A mi padre: José Cruz, por ser un guía en mi vida, por el tiempo que estuvimos juntos y por la esperanza que duro su ausencia, ya que gracias a él termine mi carrera.

A mi madre: Elsa, por el gran ejemplo que siempre me ha dado y porque es una mujer admirable en todos los aspectos.

A mis hermanos: Cyndhy y Ricky, por ponerme retos para ser cada día mejor.

A mis asesores: M.C. Isaías Guillen Moya y M.C.Manuel Mejía Lavalle, por la paciencia que me tuvieron y por todo el apoyo técnico que me brindaron, ya que sin ellos, no habría sido posible esta tesis.

Con cariño:

DIANA

AGRADECIMIENTOS

A mi madre por el ejemplo de dedicación en el empeño de su trabajo y por el apoyo durante toda mi carrera.

A mis profesores: por toda sus enseñansas, base sólida de mi formación profesional.

A mis asesores: M.C. Manuel Mejía Lavalle y M.C. Isaías Guillen Moya, por su valiosa retroalimentación, por sus asesorias y por el tiempo que me brindaron para la elaboración de esta tesis.

Al Instituto de Investigaciones Eléctricas, por el lugar, el equipo y la beca que me otorgó.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

En el presente trabajo se pretenden presentar tres variantes o extensiones de algoritmos de optimización para la selección de atributos como pre-procesamiento a la minería de datos. Estas tres variantes se aplican al problema de la detección de usuarios que consumen electricidad en forma ilícita. Las variantes propuestas buscan reducir el tiempo de procesamiento sin afectar la calidad de la solución.

En particular, el problema del consumo ilícito de energía eléctrica lo enfrentan las compañías de electricidad de todo el mundo. Las cuales tienen una alta incidencia de robos de energía eléctrica por parte de los consumidores que utilizan diversos mecanismos como tomas clandestinas y alteración del funcionamiento de los medidores. Para enfrentar dicha problemática, día a día se perfeccionan procedimientos operativos y dispositivos de ayuda para la detección de ilícitos.

No obstante que se cuenta con avances tecnológicos en el campo de la medición, las estadísticas siguen siendo adversas para la mayoría de las empresas de electricidad. Para tratar de aliviar la problemática de pérdidas de energía eléctrica, en su mayoría por ilícitos, en esta tesis, se realizó minería de datos sobre un subconjunto de datos provenientes del Sistema Comercial (SICOM), de Comisión Federal de Electricidad (CFE). Realizando la evaluación de la utilidad de la minería de datos en detección de anomalías no técnicas en la distribución de energía eléctrica, con el fin de detectar si existen patrones de datos que identifiquen a aquellos usuarios probables de incurrir en usos ilícitos de la energía.

Dado un problema planteado que principalmente tenia un algoritmo de complejidad exponencial, se hacen nuevos algoritmos utilizando linealización (por medio de las series de Taylor), de esta manera, el problema se modificó a tiempo polinomial, esto se hizo a través del método de punto interior, el cual tiene la ventaja de que los problemas grandes no requieren muchas más iteraciones que los problemas pequeños. Por lo que son más rápidos y menos costosos que otros métodos para problemas muy grandes. La solución consiste en eliminar atributos irrelevantes para mejorar los tiempos de procesamiento y la calidad de la solución.

Se hicieron experimentos con una base de datos de facturación eléctrica en México, y se compararon los resultados obtenidos contra los ya establecidos de las técnicas tradicionales. Se encontraron soluciones más eficientes y eficaces que las conocidas hasta ahora para seleccionar atributos, dentro de las cuales están: tiempo de procesamiento, calidad predictiva y reducción de atributos. De esta experimentación resultó la reducción en gran medida del tiempo de procesamiento, ya que las iteraciones del proceso disminuyeron en un 99%, esto sin afectar la selección de atributos. Así mismo, se mejoró el porcentaje de las instancias correctamente clasificadas.

CAPITULO 1

FUNDAMENTOS TEÓRICOS

1.1 ANTECEDENTES DE INVESTIGACIÓN DE OPERACIONES

El inicio de La actividad llamada Investigación de Operaciones (IO), se atribuye a los servicios militares prestados a principios de la Segunda Guerra Mundial. Debido a los esfuerzos bélicos, existía una necesidad urgente de asignar recursos escasos a las distintas operaciones militares y a las actividades dentro de cada operación, en la forma más efectiva.

Debido a esto, las administraciones militares americana e inglesa hicieron un llamado a un gran número de científicos para que aplicaran el método científico a éste y a otros problemas estratégicos y tácticos. De hecho, se les pidió que hicieran investigación sobre operaciones (militares); éstos equipos de científicos fueron los primeros equipos de IO.

Con el desarrollo de métodos efectivos para el uso del nuevo radar, estos equipos contribuyeron al triunfo del combate aéreo inglés. A través de sus investigaciones para mejorar el manejo de las operaciones antisubmarinas y de protección, jugaron también un papel importante en la victoria de la batalla del Atlántico Norte. Esfuerzos similares fueron de gran ayuda en la isla de campaña en el pacífico. Al terminar la guerra, el éxito de la investigación de operaciones en las actividades bélicas generó un gran interés en sus aplicaciones fuera del campo militar. Como la explosión industrial seguía su curso, los problemas causados por el aumento en la complejidad y especialización dentro de las organizaciones pasaron de nuevo a primer plano. Comenzó a ser evidente para un gran número de personas, incluyendo a los consultores industriales que habían trabajado para los equipos de IO durante la guerra, que estos problemas eran básicamente los mismos que los enfrentados por la milicia, pero en un contexto diferente. Cuando comenzó la década de 1950, estos individuos habían introducido el uso de la investigación de operaciones en la industria, los negocios y el gobierno. Desde entonces, esta disciplina se ha desarrollado con rápidez.

Se pueden identificar por lo menos otros dos factores que representaron un papel importante en el desarrollo de la investigación de operaciones durante este período. Uno es el gran progreso que ya se había hecho en el mejoramiento de las técnicas disponibles en esta área.

Después de la guerra, muchos científicos que habían participado en los equipos de IO o que tenían información sobre este trabajo, se encontraban motivados a buscar resultados sustanciales en este campo; de esto resultaron avances importantes.

Un ejemplo sobresaliente es el método simplex para resolver problemas de programación lineal, desarrollado en 1947 por George Dantzing. Muchas de las herramientas características de la investigación de operaciones, como programación lineal, programación dinámica, líneas de espera y teoría de inventarios, fueron desarrolladas casi por completo antes del término de la década de 1950. [Itson], [Inv. Operaciones], [Inv. Operaciones-2].

1.2 ANTECEDENTES DE DESCUBRIMIENTO DEL CONOCIMIENTO EN BASES DE DATOS

En los últimos años se han acumulado enormes cantidades de datos en todas las organizaciones, y esta tendencia continúa a un ritmo acelerado. Esto ha sido posible po

r el amplio uso de los sistemas computarizados, nuevas técnicas de captura de datos, el empleo de códigos de barra, los lectores de caracteres ópticos, las tarjetas magnéticas, etc., y por el avance en la tecnología de almacenamiento y su consiguiente reducción de costos.

El crecimiento explosivo de las bases de datos y el aun mayor de la Internet, urge en la búsqueda de técnicas y herramientas que, en forma automática y eficiente, generen información a partir de los datos almacenados. Este es el objetivo de las técnicas de Minería de Datos (Data Mining) y Descubrimiento del Conocimiento en Bases de Datos (Knowledge Discovery in Databases, o KDD para abreviar).

Las técnicas de análisis estadístico desarrolladas hace tiempo, permiten obtener ciertas informaciones útiles, pero no inducir relaciones cualitativas generales, o leyes previamente desconocidas; para esto se requieren técnicas de análisis inteligente [Frawley, 1991] que todavía no han sido perfectamente establecidas. Por ello, se incrementa de forma continua la diferencia existente entre la cantidad de datos disponibles y el conocimiento extraído de los mismos.

Pero cada vez más investigaciones dentro de la inteligencia artificial están enfocadas a la inducción de conocimiento en bases de datos. Se denomina descubrimiento de conocimiento en bases de datos (KDD) al proceso global de búsqueda de nuevo conocimiento a partir de los datos de una base de datos. Este proceso incluye no sólo el análisis inteligente de los datos con técnicas de minería de datos, sino también los pasos previos, como el filtrado y preprocesado de los datos, y los posteriores como la interpretación y validación del conocimiento extraído.

El término descubrimiento de conocimiento en bases de datos empezó a utilizarse en 1989 para referirse al amplio proceso de búsqueda de conocimiento en bases de datos y para enfatizar la aplicación a "alto nivel" de métodos específicos de minería de datos [Fayyad, 1996].

En general, el descubrimiento es un tipo de inducción de conocimiento, no supervisado [Michalski, 1987], que implica dos procesos:

- Búsqueda de regularidades interesantes entre los datos de partida
- Formulación de leyes que las describan

Entre la literatura dedicada al tema, se pueden encontrar varias definiciones para descubrimiento:

Según [Frawley, 1991] el descubrimiento de conocimiento es la extracción no trivial de información implícita, previamente desconocida y potencialmente útil, a partir de un conjunto de datos. Dado un conjunto de hechos (datos) H, un lenguaje L, y alguna medida de la certidumbre C, se define una regularidad (pattern) como una sentencia S en L que describe relaciones dentro de un subconjunto Hs de H con una certidumbre C, de forma que S es más sencillo que la enumeración de todos los hechos de Hs. Una regularidad que sea interesante y bastante cierta (según criterios definidos por el usuario) se denomina conocimiento. Un sistema de descubrimiento será un programa que toma como entrada el conjunto de hechos y extrae las regularidades existentes. Cuando el conocimiento se extrae partiendo de los datos de una base de datos, se tiene KDD.

En [Brachman y Anand, 1996] se define el proceso de KDD, desde un punto de vista práctico, como "una tarea intensiva en conocimiento que consta de complejas interacciones, prolongadas en el tiempo, entre un humano y una (gran) base de datos, posiblemente con la ayuda de un conjunto heterogéneo de herramientas".

Los principales pasos dentro del proceso interactivo e iterativo del KDD pueden verse en la figura 1.1.

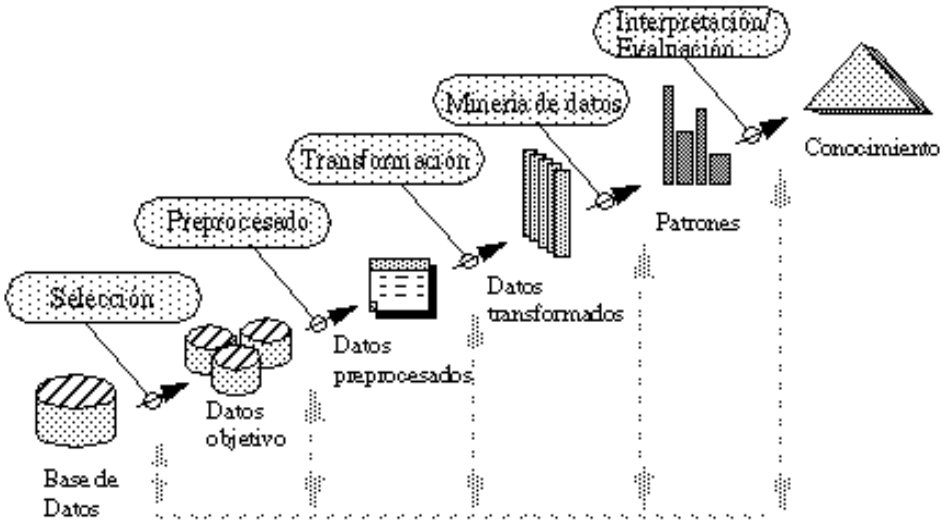

Figura 1.1 Proceso de KDD.

Muchas veces los pasos que constituyen el proceso de KDD no están tan claramente diferenciados como se muestra en la figura 1.1. Las interacciones entre las decisiones tomadas en diferentes pasos, así como los parámetros de los métodos utilizados y la forma de representar el problema suelen ser extremadamente complejos. Pequeños cambios en una parte pueden afectar fuertemente al resto del proceso.

Históricamente, el desarrollo de la estadística ha proporcionado métodos para analizar datos y encontrar correlaciones y dependencias entre ellos. Sin embargo, el análisis de datos ha cambiado recientemente y ha adquirido una mayor importancia, debido principalmente a tres factores [Decker y Focardi, 1995]:

- **Incremento de la potencia de los ordenadores.** Aunque la mayoría de los métodos matemáticos fueron desarrollados durante los años 60 y 70, la potencia de cálculo de los grandes ordenadores de aquella época (equivalente a la de los ordenadores personales de hoy en día), restringía su aplicación a pequeños ejemplos "de juguete", fuera de los cuales los resultados resultaban demasiado pobres. Algo similar ha ocurrido con la capacidad de almacenamiento de los datos y su costo asociado.

- **Incremento del ritmo de adquisición de datos.** El crecimiento de la cantidad de datos almacenados se ve favorecido no sólo por el abaratamiento de los discos y sistemas de almacenamiento masivo, sino también por la automatización de muchos experimentos y técnicas de reducción de datos. Se estima [Frawley, 1991] que la

cantidad de información almacenada en todo el mundo se duplica cada 20 meses; el número y tamaño de las bases de datos probablemente crece más rápidamente. Por ejemplo, se espera que los satélites de observación de la tierra generen, aproximadamente un petabyte (10^{15} bytes) de datos diariamente, por lo que una persona trabajando 24 horas al día, todos los días del año, a un ritmo de procesamiento de una imagen por segundo, necesitaría varios años para mirar las imágenes generadas en sólo un día.

Por último, han surgido nuevos métodos, principalmente de aprendizaje y representación de conocimiento, desarrollados por la comunidad de inteligencia artificial, estadística y física de dinámicas no lineales. Estos métodos complementan a las tradicionales técnicas estadísticas en el sentido de que son capaces de inducir relaciones cualitativas generales, o leyes, previamente desconocidas.

Estos nuevos métodos matemáticos y técnicas de software, para análisis inteligente de datos y búsqueda de regularidades en los mismos, se denominan actualmente técnicas de minería de datos o /data mining/. A su vez, la minería de datos ha permitido el rápido desarrollo de lo que se conoce como descubrimiento de conocimiento en bases de datos. [Gsi].

1.3 OBJETIVO

Realizar variaciones (experimentar) en Clasificación de Instancias y Selección de Atributos, basado en Optimización (Minimización Cóncava [Bradley y Mangasarian, 1998]).

Se busca explorar y encontrar soluciones más eficientes, eficaces y útiles que las conocidas en la actualidad para clasificar instancias y seleccionar atributos.

La tesis aportará resultados experimentales que podrán ayudar a seguir líneas de investigación en un futuro inmediato.

1.4 JUSTIFICACIÓN

El vertiginoso crecimiento de las Bases de Datos actuales obliga a la creación de mecanismos automáticos-computarizados-eficientes capaces de revisar, buscar y encontrar, en estas inmensas montañas de datos, la información relevante que apoye a las grandes empresas modernas a cumplir sus objetivos de negocio. El Descubrimiento Automático de Conocimiento y en particular la Minería de Datos son la respuesta emanada de la Inteligencia Artificial a esta problemática.

De esta manera se requieren algoritmos computacionales eficientes capaces de extraer y presentar la información importante e implícita en las bases de datos. Estos algoritmos, surgidos del área del Aprendizaje de la Inteligencia Artificial, conforman lo que se conoce como Minería de Datos (Data Mining).

En general las Grandes Bases de Datos (Very Large Databases ó VLDB) no se pueden minar (procesar) con los algoritmos hasta ahora desarrollados. Para ello, como beneficio se contará con nuevos algoritmos basados en investigación de operaciones, que superen en alguna medida a los tradicionales, o dar nuevas alternativas como líneas de futuras investigaciones.

1.5 ALCANCES Y LIMITACIONES

1.5.1 ALCANCES

- Re-formular el modelo /**F**eature **S**election via Conca**v**e Minimization and Support Vector Machines (FSV)/ de tal forma que no requiera realizar miles de iteraciones para llegar a la solución. Esto se podría lograr aplicando métodos de punto interior o linealizando la formulación (sustituir la expresión exponencial de la formulación por un equivalente lineal).
- FSV en su formulación actual no considera el aspecto de costo-beneficio en la clasificación de las clases, el cual es un aspecto importantísimo en aplicaciones del mundo real; por lo tanto se trabajará en una formulación de FSV que tome en cuenta este aspecto.
- FSV no experimenta con las distancias en las líneas de división, ya que sólo toma en cuenta distancias de 1 y -1 para la clasificación y selección, por lo tanto se experimentará en la reformulación de FSV para que considere este tópico.

Se probaran nuevas formulaciones en un dominio del sector eléctrico.

1.5.2 LIMITACIONES

- Sólo se experimentará con la herramienta Gams (solver optimizador).
- Los resultados obtenidos no necesariamente serán mejores a los reportados por la literatura especializada (lo que se busca es evaluar las formulaciones para al analizarlas, idear nuevas posibles reformulaciones).
- Sólo se realizaran re-formulaciones sobre el esquema FSV.
- Únicamente se exploraran a detalle las combinaciones aparentemente más interesantes.

1.6 ANTECEDENTES DEL PROBLEMA A RESOLVER

Las compañías de electricidad de todo el mundo tienen una alta incidencia de robos de energía eléctrica por parte de los consumidores que utilizan diversos mecanismos como tomas clandestinas y alteración del funcionamiento de los medidores. El porcentaje de pérdidas debido a estos ilícitos se estima en algunos casos equivalentes al total de las pérdidas debidas a otros factores, que llegan a sumar hasta 30% de la energía que se comercializa.

Para enfrentar dicha problemática, día a día se perfeccionan procedimientos operativos y dispositivos de ayuda para la detección de ilícitos. A la fecha, los desarrollos se han centrado casi exclusivamente en el concepto de la medición para fines de comercialización y lo que de ella se pueda inferir, como detección de pérdidas técnicas de energía e interrupciones en el suministro.

Las pérdidas de energía eléctrica son comunes e inherentes a las compañías de electricidad; se tornan en un problema grave cuando éstas rebasan ciertos límites lógicos.

Es práctica común clasificar las pérdidas de energía eléctrica en técnicas y no técnicas. Las pérdidas técnicas se dan en los elementos y equipos de los circuitos eléctricos, por ejemplo en líneas de transmisión, transformadores y bancos de capacitores. Su origen son los principios que rigen la transformación de la energía. Las pérdidas no técnicas se pueden clasificar en tres tipos: accidentales, administrativas, y fraudulentas.

Tratando de aliviar la problemática de pérdidas de energía eléctrica debidas a ilícitos, las compañías de electricidad han implementado una o más acciones. Las principales se describen a continuación.

- Inspección visual de las instalaciones de medición
- Equipos de detección
- Análisis estadístico de consumos
- Educación de los consumidores

En la mayoría de los casos, los métodos tradicionales para reducir las pérdidas por ilícitos no han dado los resultados esperados. La implantación simultánea de más de un método muchas veces no justifica la inversión.

Desgraciadamente, a pesar de lo grave del problema en muchos países, no hay una tendencia tecnológica fuerte para enfrentarlo. Las tendencias en medición se dirigen hacia la lectura automática, como lo demuestran los medidores electrónicos multifunción y los sistemas de lectura automática, ya que ese sigue siendo el mercado global de importancia.

Aunque no con la fuerza deseada, la tendencia es hacia la implantación de sistemas de vigilancia continua, eficaz y de costo aceptable, mediante la combinación de procedimientos automatizados de procesamiento de información y equipos de submedición de costo aceptable.

Las pérdidas de energía eléctrica debidas a ilícitos es un problema que se dejó adelantar demasiado a los procedimientos y a la tecnología para combatirlo. Es un fenómeno parecido al robo de autos: cada dispositivo (alarma) que sale al mercado para combatirlo es descifrado por los ladrones. La gravedad del problema hace que se empiecen a tomar medidas al respecto. Lo que también es cierto es que la energía que algunos roban, todos la tenemos que pagar de alguna manera. [IIE, 1997], [IURPA, 1997], [Ferrer, 1996], [Hodges, 1996] [Reason, 1996], [Hahn, 1996] y [Richardson, 1994].

1.7 ORGANIZACIÓN DEL DOCUMENTO

En la tesis se muestra que:

- Se ha identificado un problema muy valioso para la investigación, pero sobre todo, para obtener información de grandes bases de datos.
- Se resolverá el problema y se señalarán líneas de futuras investigaciones.
- Se aplicarán métodos de linealización y punto interior para eficientar el modelo FSV en el proceso de optimización.

Así mismo, se tratará de contestar a las siguientes preguntas:

- ¿Qué problema se está planteando?
- ¿Es un buen problema?
- ¿Ha sido resuelto?
- ¿Se está haciendo una contribución adecuada al conocimiento?

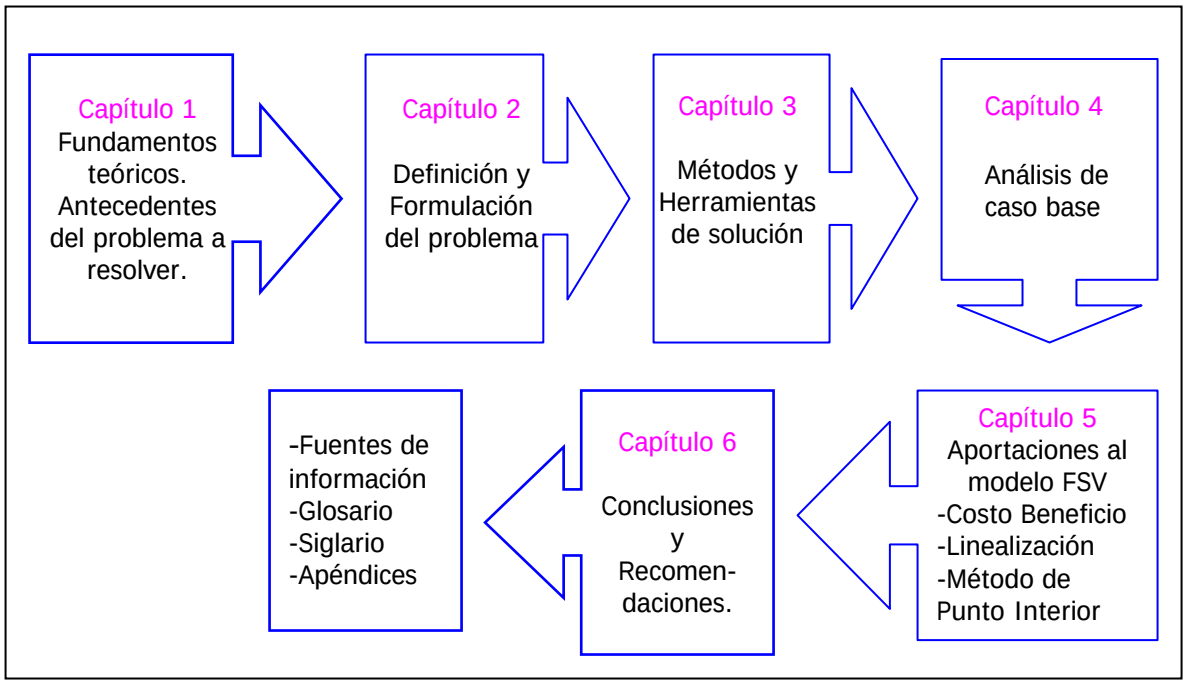

Figura 1.2 Organización de la tesis.

CAPÍTULO 2

DEFINICIÓN Y FORMULACIÓN DEL PROBLEMA

2.1 DEFINICIÓN DEL PROBLEMA

Las compañías de electricidad de todo el mundo sufren en mayor o menor grado, de pérdidas de energía eléctrica. Un gran porcentaje de estas pérdidas se debe, principalmente, a apropiaciones indebidas de energía. Los consumidores, mediante prácticas ilícitas como tomas clandestinas y alteración del funcionamiento de los medidores, llevan a cabo el robo de energía.

Para enfrentar dicha problemática, día a día se perfeccionan procedimientos operativos y dispositivos de ayuda para la detección de ilícitos. No obstante que se cuenta con avances tecnológicos en el campo de la medición, las estadísticas siguen siendo adversas para la mayoría de las empresas de electricidad.

A la fecha (Junio, 2006), los nuevos desarrollos se han orientado a la problemática asociada con la lectura convencional de medidores para fines de comercialización (facturación) y del análisis del historial de consumo se infieren las pérdidas de energía. [Vidrio, Gómez y Castán, 2004].

Para tratar de aliviar la problemática de pérdidas de energía eléctrica, en su mayoría por ilícitos, en la Gerencia de Sistemas Informáticos (GSI) del Instituto de Investigaciones Eléctricas (IIE) se realiza minería de datos sobre un subconjunto de datos provenientes del Sistema Comercial (SICOM) de Comisión Federal de Electricidad (CFE). Realizando la evaluación de la utilidad de la minería de datos en detección de anomalías no técnicas en la distribución de energía eléctrica, con el fin de detectar si existen patrones de datos que identifiquen a aquellos usuarios probables de incurrir en usos ilícitos de la energía.

El trabajo de minería de datos se realiza sobre el subconjunto de datos del Sistema Comercial (SICOM) del proceso de comercialización de energía. El sistema de información de la gerencia de cobranza está formado por los siguientes subsistemas:

SICOM	**Sistema Comercial.**
SINOT	Sistema de Notificación.
SICOSS	Sistema de Control de Solicitud de Servicios.
SIMED	Sistema de Control de Medidores.

Cada uno de estos sistemas contiene información propia para la ejecución del proceso de facturación y cobranza que lleva a cabo la gerencia. La relación entre estos 4 subsistemas es directa y se muestra en la figura 2.1.

El foco de atención de los experimentos que se realizaran (usando algoritmos de optimización y minería), es detectar patrones de comportamiento de aquellos usuarios que cometen uso ilícito de la energía. Así como la utilidad de la optimización y la minería de datos en detección de anomalías no técnicas en la distribución de energía eléctrica, con el fin de detectar si existen patrones de datos que identifiquen a aquellos usuarios probables de incurrir en usos ilícitos de la energía.

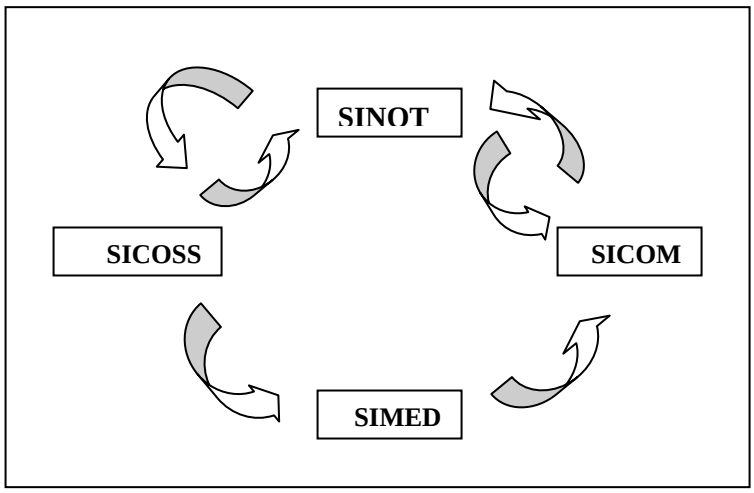

Figura 2.1 Sistema de información de la gerencia de cobranza, CFE.

2.1.1 DESCRIPCIÓN DE LOS DATOS FUENTE

Los datos a minar provienen de un subconjunto de datos del SICOM. Este subconjunto de datos fuente está compuesto básicamente por 13 tablas:

- CATAL1
- CATAL1A
- CATAL2
- CATAL22
- CATAL2A
- CATAL3
- CATAL4
- CATAL4A
- CATAL5
- CATAL6
- CATAL9
- CATALL
- CATALS

De las cuales se detectaron cuatro tablas con campos que sugerían tener relación con los usos ilícitos, el cual es el foco de atención de los experimentos a realizar, las tablas son las siguientes:

- CATAL2A
- CATAL4
- CATAL6
- CATALL

2.1.2 DESCRIPCIÓN DE LA MINA DE DATOS

Después de analizar los datos fuente contenidos en este subconjunto de SICOM se decidió emplear como eje los datos de usos ilícitos de *CATAL4* por ser la que presentaba los datos en forma más completa y con mayor número de casos de usos ilícitos.

Así se extrajo una mina con 35,983 registros con los siguientes atributos:

- RPU
- Año
- Mes
- Tipo-adeudo
- Dígito
- KWH
- Energía
- Cve-facturación
- Total
- Status
- Giro
- Tarifa
- Nombre
- Carga-Instalada
- Carga-Contratada.

De aquí se hicieron algunos atributos "construidos" como por ejemplo:

- ciMcc: Carga instalada menos carga contratada
- kwEci: kwh entre carga instalada
- kwMen: kwh menos energía
- kwMci: kwh menos carga instalada
- toMcl: Total menos carga instalada
- toMen: Total menos energía
- toMkw: Total menos kwh
- toMcc: Total menos carga contratada
- ciEto: carga instalada entre total
- ciEen: carga instalada entre energía.

Y se tomaron 3 atributos Random:

- Aleatorio1
- Aleatorio2
- Aleatorio3

Donde el atributo *Tipo-adeudo* es el que puede tomar el valor "9" que indica un uso ilícito.

Los datos extraídos tienen información de:

- 1,690 usuarios distintos (es decir, diferentes RPU`s).

- Los usuarios tienen la característica en común de ser empresas (grandes usuarios).
- Los registros contienen información que va del año de 1994 al 2000 (aunque en realidad la mayor cantidad de registros están del mes 6 de 1998 al mes 6 del 2000).
- En promedio se tienen 21 registros mensuales por cada usuario, (aunque hay usuarios que sólo tienen 1 registro y otros hasta 33).
- Hay 570 casos registrados como usos ilícitos (del total de 35,983 registros).
- Éstos 570 casos ilícitos representan el 1.5841% del total de los 35,983 casos y se contabilizaron 274 giros diferentes.
- Los kwh consumidos en los 570 casos ilícitos son 33,838,939 que representan el 2.75% del total de 1,231,946,620.

Dado que los datos arrojados por el reportador (localizado en el directorio *sicore2k* del servidor de CFE) contenían algunos caracteres extraños y campos en blanco, se desarrolló un programa para realizar la limpieza sobre los datos, quedando 2770 registros.

Después de limpiar los datos, con esta mina se procedió a realizar los diferentes experimentos de optimización y minería de datos.

2.2 FORMULACIÓN DEL MODELO

Se busca seleccionar atributos de una base de datos, para lo cual se quiere obtener un modelo que prediga cuando un usuario es susceptible de ser ilícito o no. El problema usa una base de datos de facturación eléctrica en México que cuenta con 24 atributos más una clase, 2,200 registros de tipo "1" y 570 registros de tipo "9", es decir, 2770 instancias. Dado que la base de datos es muy grande, se quiere minimizar el promedio de las distancias de los puntos mal clasificados, donde "m" es el número de instancias de "A" y "k" es el número de instancias de "B". De la misma manera se busca eliminar coeficientes "w", que son los 24 atributos (Año, Mes, Dígito, KWH, Energía, Cve-facturación, Total, Status, Tarifa, Carga-Instalada, Carga-Contratada, ciMcc, kwEci, kwMen, kwMci, toMcl, toMen, toMkw, toMcc, ciEto, ciEen, Aleatorio1, Aleatorio2, Aleatorio3), así como reducir el tiempo de procesamiento.

Según [Bradley y Mangasarian, 1998], con FSV se obtiene buen poder de clasificación, en pocas iteraciones y con alta reducción de atributos.

Bradley y Mangasarian proponen un modelo que se usará en esta tesis, que será descrito con mas detalle en la sección 3.1, el cual es:

$$\min_{w,r,y,z,v} imizar \quad (1-\lambda)\left(\frac{e^T y}{m}+\frac{e^T z}{k}\right)+\lambda e^T\left(e-\varepsilon^{-\alpha v}\right)$$

$$\text{sujeto a :} \quad \begin{aligned} &-Aw+e\gamma+e\le y,\\ &Bw-e\gamma+e\le z,\\ &y\ge 0, z\ge 0,\\ &-v\le w\le v. \end{aligned}$$

El modelo matemático comprende principalmente tres conjuntos básicos de elementos que se presentan en las siguientes secciones.

2.2.1 FUNCIÓN OBJETIVO

La función objetivo es la medida de efectividad que permite conocer el nivel de logro de los objetivos y es una función (ecuación) matemática de las variables de decisión. Para este modelo la función objetivo es:

$$\min_{w,r,y,z,v} imizar \quad (1-\lambda)\left(\frac{e^T y}{m}+\frac{e^T z}{k}\right)+\lambda e^T\left(e-\varepsilon^{-\alpha v}\right)$$

Donde, e es un vector de unos en un espacio real de dimensión arbitraria y e^T es la transpuesta de un vector de unos en un espacio real de dimensión arbitraria.

2.2.2 VARIABLES

Las variables de decisión son las incógnitas (o decisiones) que deben determinarse resolviendo el modelo. Para el modelo que se está utilizando en esta tesis las variables son:

w, r, y, z, v y **f**

Donde:

w Son los coeficientes de las variables en la ecuación de la recta.
r Es el término independiente en la ecuación de la recta.
y Representa la magnitud del error de un punto mal ubicado en la tabla A(I,J).
z Representa la magnitud del error de un punto mal ubicado en la tabla B(L,J).
v Modela el vector $|w|$, es el número de las características usadas.
f Es el valor numérico de la función objetivo.

Los parámetros son los valores conocidos que relacionan las variables de decisión con las restricciones y función objetivo. También pueden ser llamados escalares y son:

m, k, λ y **α**

Donde:

m Es el total de puntos de la tabla A(I,J).
k Es el total de puntos de la tabla B(L,J).
λ Es un escalar que toma diferentes valores.
α Es un escalar que toma diferentes valores.

2.2.3 RESTRICCIONES

Las limitantes del problema llamadas restricciones son un conjunto de igualdades o desigualdades que constituyen las barreras y obstáculos para la consecución del objetivo.

Para tener en cuenta las limitaciones tecnológicas, económicas, de tiempo, etc. el modelo debe incluir restricciones (implícitas o explícitas) que restrinjan las variables de decisión a un rango de valores factibles.

$$
\begin{aligned}
&-Aw+e\gamma+e\le y,\\
&Bw-e\gamma+e\le z,\\
&y\ge 0, z\ge 0,\\
&-v\le w\le v.
\end{aligned}
$$

Un modelo siempre debe ser menos complejo que el problema real, es una aproximación abstracta de la realidad con consideraciones y simplificaciones que hacen más manejable el problema y permiten evaluar eficientemente las alternativas de solución.

La solución óptima será aquella que produzca el mejor valor de la función objetivo, sujeta a las restricciones.

Para efectos de esta tesis se hizo este mismo modelo pero adaptado al programa GAMS que se mencionará en la sección 3.2 y que será de mucha ayuda para realizar los objetivos.

CAPÍTULO 3

MÉTODOS Y HERRAMIENTAS DE SOLUCIÓN

3.1 MÉTODO DE SELECCIÓN DE ATRIBUTOS DE BRADLEY Y MANGASARIAN

Paul. S. Bradley y Olvi L. Mangasarian son los pioneros en selección de atributos usando optimización. Esta tesis se basa en un artículo de ellos llamado "Feature Selection via Concave Minimization and Support Vector Machines" (La selección de características vía minimización cóncava y maquinas vectoriales).

En este artículo básicamente se realiza una comparación entre dos métodos de la selección de características que tienen en común encontrar un plano separador que discrimine entre dos conjuntos de puntos en un espacio característico n-dimensional que utilice tan pocas de las n características (dimensiones) como sea posible. En la aproximación de la minimización cóncava ([Mangasarian, 1996] y [Bradley, Mangasarian, Street, 1998]) un plano de separación es generado reduciendo al mínimo una suma ponderada de distancias de puntos clasificados equivocadamente a dos planos paralelos que limitan los sistemas y que determinan la mitad del camino del plano de separación entre ellos. Además, el número de las dimensiones del espacio usado para determinar el plano es minimizado. En el método de la máquina vectorial, además de minimizar la suma ponderada de distancias de puntos clasificados equivocadamente a los planos de limitación, también se maximiza la distancia entre los dos planos de limitación que genera el plano de separación. Los resultados computacionales demuestran que la supresión de las características es una consecuencia indirecta de la aproximación de la máquina vectorial, cuando se utiliza una norma apropiada. Las pruebas numéricas en 6 bases de datos demuestran que los clasificadores entrenados por el método de minimización cóncava y los entrenados por una máquina vectorial tienen comparables 10 -intersecciones dobles- de la validación correcta, es decir, que el programa trabaja sin fallas. Sin embargo, en todas las bases de datos probadas, los clasificadores obtenidos por la minimización cóncava seleccionaron menos características del problema que los entrenados por una máquina vectorial.

FSV: Selección de Atributos Vía Minimización Cóncava

Aquí se describe un procedimiento de la selección de características que ha sido eficaz en medicina y otras aplicaciones. [Mangasarian, 1996] y [Bradley, Mangasarian y Street, 1998].

Dado dos conjuntos de puntos A y B en R^n representado por las matrices A Є R^{mxn} y B Є R^{kxn} respectivamente, se desea excluir entre ellas por un plano de separación:

$$P = \{x \mid x \in R^n, x^T w = \gamma\}, \qquad (1)$$

con la normal w Є R^n y 1-norma, que es la distancia al origen de $\frac{|\gamma|}{\|w\|_\infty}$. Se procura determinar W y γ de modo que el plano de separación P defina dos espacios abiertos $P = \{x \mid x \in R^n, x^T w > \gamma\}$ conteniendo sobre todo puntos de A, y $P = \{x \mid x \in R^n, x^T w < \gamma\}$ conteniéndolos sobre todos los puntos de B. Por lo tanto, sobre la normalización, se desea satisfacer

$$Aw \geq e\gamma + e, Bw \leq e\gamma - e \qquad (2)$$

a la magnitud posible. Las condiciones (2) se pueden satisfacer si y solamente si, los espacios convexos de A y B son disjuntos. Éste no es el caso en muchas aplicaciones del mundo real. Por lo tanto, se procura satisfacer (2) en un mejor sentido minimizando una cierta norma de las violaciones promedio (2), por ejemplo

$$\min_{w,\gamma} f(w,\gamma) = \min_{w,\gamma} \frac{1}{m} \left\| (-Aw + e\gamma + e)_+ \right\|_1 + \frac{1}{k} \left\| (Bw - e\gamma + e)_+ \right\|_1 \qquad (3)$$

Para que el vector x, x_+ denote al vector con los componentes máximos {0, x_i}. Dos razones principales de elegir la 1-norma en (3) son: (1) el problema, (3) es entonces reducible a un programa lineal, (4) con muchas características abstractas importantes que hacen una herramienta de cómputo eficaz [Bennet y Mangasarian, 1992], (2) la 1-norma es menos sensible a las desviaciones tales como ésos que ocurren cuando las distribuciones subyacentes de los datos han pronunciado las colas, por lo tanto (3) tienen un efecto similar al de la regresión robusta [Huber, 1981].

La formulación (3) es equivalente a la formulación de programación linear robusta siguiente propuesta en [Bennet y Mangasarian, 1992] y usada con eficacia para solucionar problemas del mundo real [Mangasarian, Street y Wolberg, 1995]:

$$\begin{aligned} \min_{w,\gamma,y,z} imice \quad & \frac{e^T y}{m} + \frac{e^T z}{k} \\ & -Aw + e\gamma + e \leq y, \\ \text{sujeto a} \quad & Bw - e\gamma + e \leq z, \\ & y \geq 0, z \geq 0. \end{aligned} \qquad (4)$$

El programa lineal (4) o equivalentemente, la formulación (3), define un plano de separación P que satisfaga aproximadamente las condiciones (2) en el sentido siguiente. Cada valor positivo de y_i determina la distancia $\frac{y_i}{\|w\|}$ [Mangasarian, Street y Wolberg, 1995], entre un punto A_i de A que está en el lado incorrecto del plano de limitación $x^T w = \gamma + 1$ para A, que es A_i que se encuentra en el espacio abierto $\{x \mid x^T w < \gamma + 1\}$, y el plano de limitación $x^T w = \gamma + 1$. Similarmente para B y $x^T w = \gamma - 1$. La función objetivo del programa lineal (4) minimiza el promedio de las distancias, ponderada de $\|w\|$`, puntos clasificados equivocadamente a los planos de limitación. El plano de separación P (1) es situado a mitad del camino entre los dos planos y paralelos a ellos.

La selección de características ([Mangasarian, 1996] y [Bradley, Mangasarian, Street, 1998]) se impuso procurando suprimir tantos elementos del vector normal W al plano de separación P que es constante con la obtención de una separación aceptable entre los sistemas A y el B. Logrando esto introduciendo un término adicional con el parámetro $\lambda \in [0,1)$ en el objetivo de (4) mientras que pondera el objetivo original como sigue:

$$\begin{aligned} \min_{\omega,\gamma,y,z} imize \quad & (1-\lambda)\left(\frac{e^T y}{m}+\frac{e^T z}{k}\right)+\lambda e^T |w|_* \\ & -Aw+e\gamma+e\le y, \\ \text{sujeto a} \quad & Bw-e\gamma+e\le z, \\ & y\ge 0, z\ge 0. \end{aligned} \tag{5}$$

Obsérvese que el vector $|w|_* \in R^n$ tiene componentes que son iguales a 1 si los componentes correspondientes de W son distintos a cero y componentes iguales a cero si los componentes correspondientes de W son cero. Recordando que e es un vector de unos y $e^T|w|_*$ es simplemente un conteo de elementos distintos a cero en el vector w. Problema (5) balances del error en la separación de los sistemas A y B, $\left(\frac{e^T y}{m}+\frac{e^T z}{k}\right)$, y el número de elementos distintos a cero de W, $\left(e^T|w|_*\right)$. Además, si un elemento de W es cero, la característica correspondiente se elimina del problema.

Introduciendo la variable V se puede eliminar el valor absoluto del problema (5) que conduce al siguiente equivalente programa paramétrico (Para $\lambda \in [0,1)$):

$$\begin{aligned} \min_{\omega,\gamma,y,z,v} imize \quad & (1-\lambda)\left(\frac{e^T y}{m}+\frac{e^T z}{k}\right)+\lambda e^T |w|_* \\ & -Aw+e\gamma+e\le y, \\ & Bw-e\gamma+e\le z, \\ \text{sujeto a} \quad & y\ge 0, z\ge 0, \\ & -v\le w\le v. \end{aligned} \tag{6}$$

Dado que v aparece ponderado positivamente en la función objetivo y es restringido por $-v\le w\le v$, modela efectivamente el vector $|w|$. Este problema de selección de características será solucionado para un valor de $\lambda \in [0,1)$ para lo cual se obtuvo la clasificación que resultaba del plano de separación (1) medio entre los planos de limitación $x^T w=\gamma\pm 1,$ generaliza lo mejor posible, estimado por un procedimiento de la intersección validada. Esto estará típicamente alcanzado en un espacio de la característica de la dimensionalidad reducida, que es $e^T v_*$ <n (es decir, el número de las características usadas es menor que n).

Debido a la discontinuidad del término de la función $e^T v_*$ aproximamos esto por un exponencial cóncavo en la línea real no negativa [Mangasarian, 1996]. La aproximación del vector del paso v_* por el del exponencial cóncavo:

$$v_* \approx t(v,\alpha)=e-\varepsilon^{-\alpha v}, \quad \alpha>0, \tag{7}$$

conduce al problema reformulado (FSV: Selección de Características Cóncavas):

$$\min_{w,\gamma,y,z,v} imize \quad (1-\lambda)\left(\frac{e^T y}{m}+\frac{e^T z}{k}\right)+\lambda e^T\left(e-\varepsilon^{-\alpha v}\right)$$

$$\text{sujeto a :} \quad \begin{aligned} &-Aw+e\gamma+e\le y,\\ &Bw-e\gamma+e\le z,\\ &y\ge 0, z\ge 0,\\ &-v\le w\le v. \end{aligned} \qquad (8)$$

Puede ser demostrado [Bradley, Mangasarian y Rosen, 1997], que para un valor finito de α (aparece en el exponencial cóncavo) el problema reformulado (8) genera una solución exacta del problema no reformulado (6). Se observa que este problema es la minimización de una función objetivo cóncava sobre un sistema polihedral. El modelo se ve ejemplificado en la figura 3.1. [Bradley y Mangasarian, 1998].

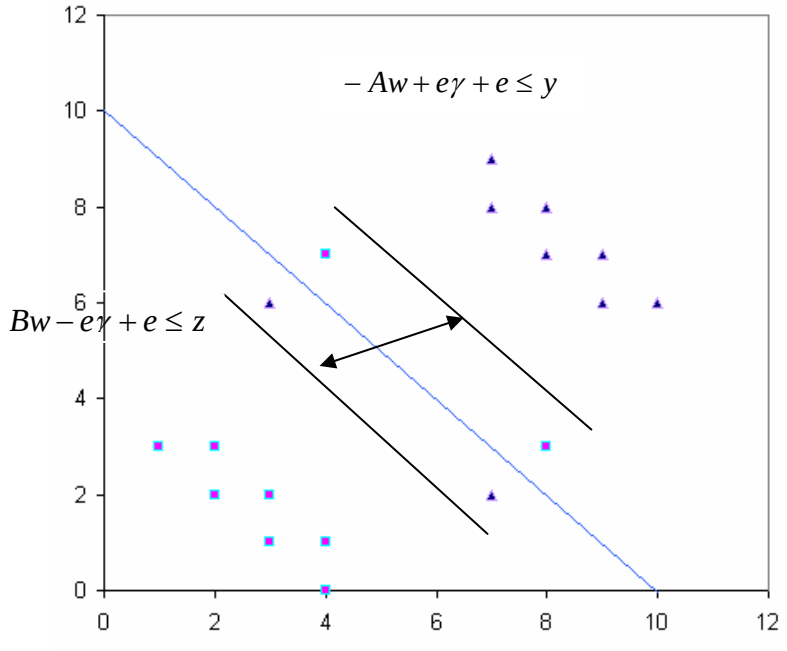

Fig.3.1 Modelo FSV

3.2 HERRAMIENTA GAMS

El programa GAMS [Gams] (General Algebraic Modeling System) es un software desarrollado por A. Brooke, D. Kendrick y A. Meeraus. A diferencia de otros paquetes de software de implementación de algoritmos matemáticos que permiten resolver los problemas de optimización, el programa GAMS presenta la ventaja de plantear un lenguaje de modelización que permite poder escribir en un editor la formulación matemática del problema y posteriormente aplicarle una serie de /solvers/ o programas de resolución.

Este programa fue desarrollado a finales de la década de los 80 en el World Bank por un grupo de economistas, aprovechando la experiencia de su trabajo sobre programas de

desarrollo económico, que requieren en primer lugar una modelización exhaustiva y posteriormente la aplicación de los correspondientes programas de optimización para poder hallar la solución numérica a los modelos propuestos.

Aunque inicialmente el manejo y comprensión de sus estructuras requiere cierto esfuerzo, una vez entendidas se dispone de una herramienta muy versátil capaz de resolver problemas de programación matemática. A pesar de ser una magnifica herramienta, el lector debe ser consciente de las limitaciones impuestas por el estado del arte existente en programación matemática.

Otros lenguajes similares a GAMS son AMPL [Ampl] y AIMMS [Aimms, 1999]. Todos ellos presentan características análogas y, en general, no hay razón alguna para elegir uno u otro. En esta tesis se opta por GAMS dada la familiaridad con este lenguaje.

Entre las características más importantes de GAMS cabe destacar:

- Su capacidad para pasar de resolver problemas de pequeña dimensión (docenas de variables y restricciones) a problemas mayores (miles de variables y restricciones) sin variar el código sustancialmente. El manejo eficiente de sus índices permite escribir de manera compacta restricciones similares mediante una sola restricción.
- Separa el proceso de modelado del proceso de resolución del problema. Así, el usuario de GAMS debe ser capaz de conseguir una formulación consistente del problema, y una vez la expresa en la notación de GAMS, este lenguaje hace uso de alguno de los optimizadores disponibles para obtener su solución. De esta manera, el usuario sólo ha de centrarse en obtener un modelo del problema y puede ignorar el funcionamiento interno del algoritmo que se necesita para resolverlo. La separación de estas dos tareas permite cambiar el modelo para mejorarlo o completarlo cómodamente.
- La forma en que GAMS representa un problema de optimización coincide, prácticamente, con la descripción matemática de ese problema. Por tanto, el código GAMS es sencillo de comprender para aquellos lectores familiarizados con la optimización.
- Además, GAMS proporciona los mecanismos necesarios para resolver problemas de optimización con estructuras similares, como son aquellos que se derivan de las técnicas de descomposición.

El usuario de GAMS debe ser cuidadoso con las reglas "gramaticales" de GAMS. El incumplimiento de una sola de ellas puede provocar muchos errores de compilación.

Entre la bibliografía de este lenguaje de programación cabe destacar el manual de GAMS [Gams], cuyo segundo capítulo ofrece un resumen con las características principales para empezar a programar en este lenguaje, y el articulo [Chattopandhyay, 1999], que proporciona un enfoque ingenieril de GAMS. [Castillo, et al, 2002].

Cabe destacar, que se usó el paquete GAMS con la versión académica, (véase figura 3.2), pero esta versión tiene una serie de limitaciones en cuanto al tamaño de los modelos como:

Máximo número de filas: 300

Máximo número de columnas:	300
Máximo número de elementos distintos de cero:	2000
Máximo número de elementos no lineales:	1000
Máximo número de variables discretas:	50

Por ello se utilizó Neos (que se menciona a continuación), para poder hacer los experimentos necesarios para esta tesis.

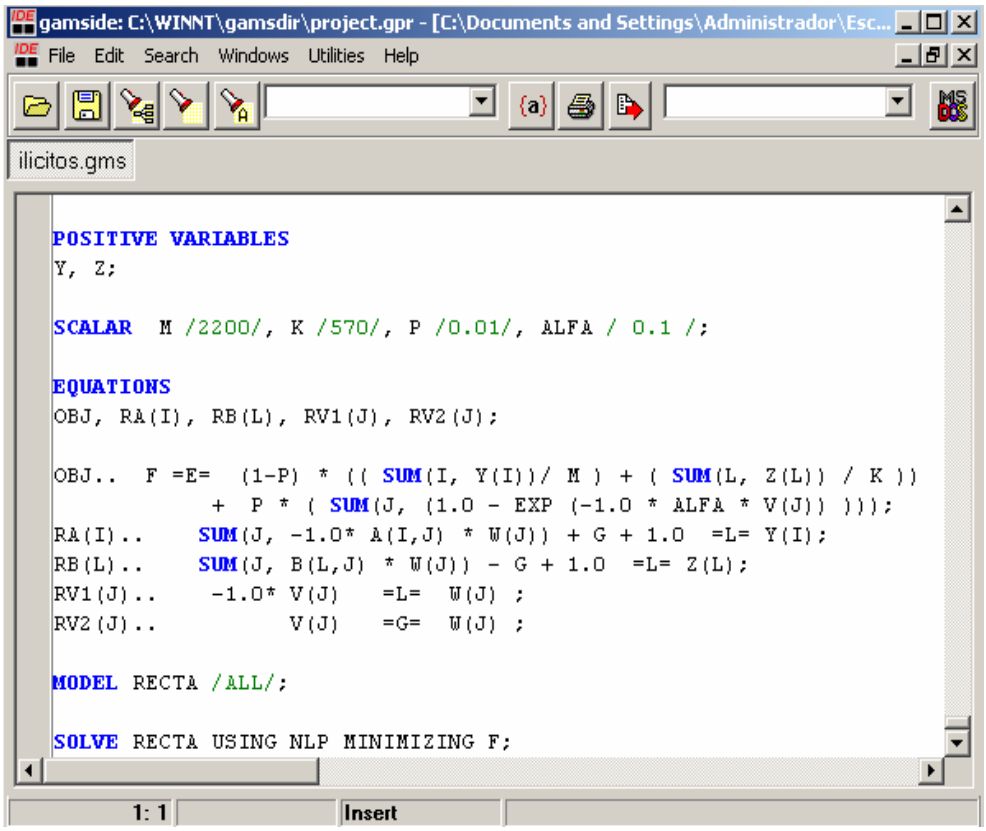

Figura 3.2 El programa GAMS académico, al introducir los datos del modelo.

3.3 NEOS

El servidor NEOS [Neos] permite el acceso general a través de Internet. Es de tipo /Freeware/, es decir, es un programa con derechos de autor que se puede utilizar sin pago alguno. Desarrollado por Argonne National Laboratory y Northwestern University.

En la actualidad NEOS puede resolver problemas en las siguientes áreas: Programación Lineal, Programación No Lineal, Optimización Sin Restricciones, Programación Estocástica, Optimización Lineal en grafos, entre otras. Los problemas de optimización se solucionan automáticamente con información mínima por parte del usuario, los usuarios necesitan solamente una definición del problema de optimización y un modelo matemático, toda la información adicional requerida por el /solver/ de optimización se determina automáticamente.
Para comenzar a usar NEOS, primero, se necesita hacer el problema de optimización, formularlo en uno de los formatos de entrada aceptados por un /solver/. Observar a través

de los /solvers/ en la página del servidor para ver qué formatos están disponibles para su tipo de problema. La mayoría de los /solvers/ cuentan con sitios en la Web que explican la sintaxis de su formato de entrada.

Las compañías proporcionan /solvers/ comerciales libres de ponderación para el acceso a través del servidor de NEOS (Rociada Optimization's XPRESS, Mosek ApS's Mosek, OptiRisk Systems' FortMP, etc.). "Sun Microsys" ha donado siete sitios de trabajo al centro de tecnología de optimización, para el funcionamiento de trabajos de NEOS.

El servidor de NEOS envía archivos de la petición del trabajo a sus estaciones geográficamente distribuidas y ejecuta su /software/. Su /software/ existente lee la información de la petición de los usuarios, y envía una solución del problema de nuevo al servidor de NEOS. El servidor de NEOS envía resultados al usuario, que ha podido utilizar el producto de software sin necesitar descargarlo, instalar, o funcionar en su máquina.

NEOS permite ver una variedad más amplia de /solvers/ que se tienen disponibles en el departamento de informática en la universidad de Wisconsin-Madison, y también ayuda a superar una gran variedad de límites de tiempo y almacenamiento que se aplican a los trabajos. Para resolver problemas más grandes donde se pueden enviar los archivos de GAMS a NEOS y a XPRESS-MP usado para solucionar los problemas.

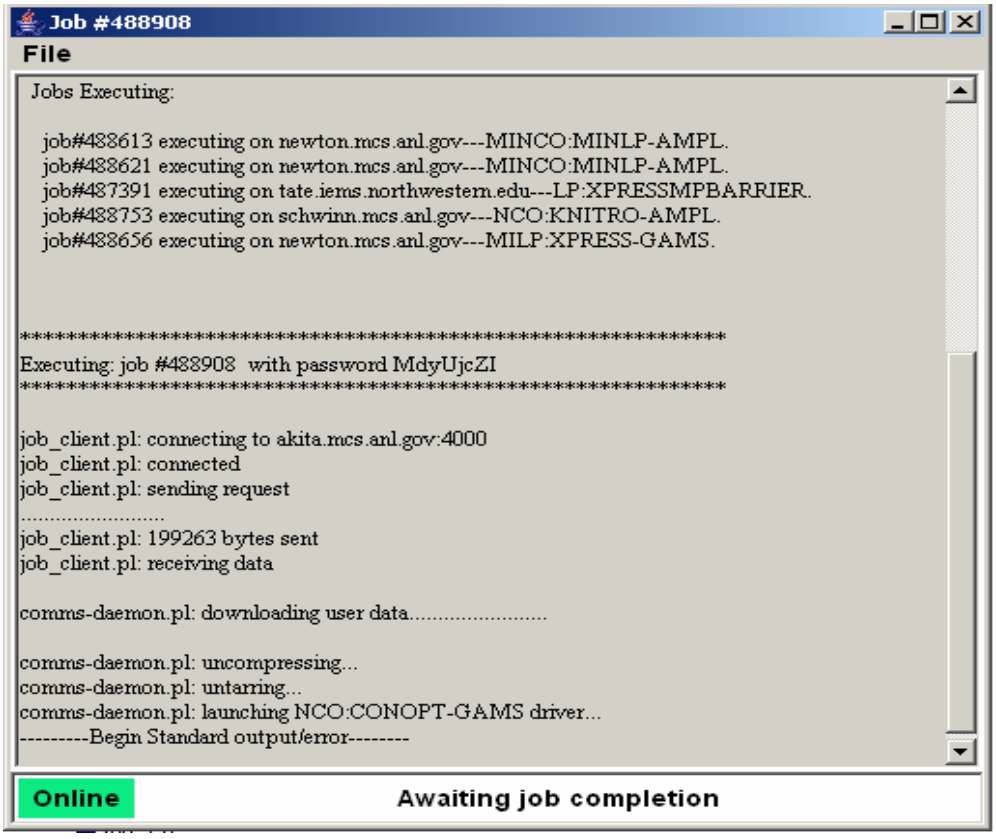

Figura 3.3 Interfaz de NEOS

Los pasos para instalar y ejecutar NEOS son:

- Se necesita tener el entorno Java [Java], en la pc.
- Entrar a la pagina http://www-neos.mcs.anl.gov/neos/
- Descargar InstallSC y Client , que son archivo de MS-DOS
- Darle doble clic a Client
- Aparece Neos en el entorno Java [Java].
- Listo para ser utilizado.

Si se necesita información adicional, se puede consultar: Optimization Software Guide, J. More' and S. Wright, SIAM Publications, 1993.

3.4 WEKA

Weka [Weka] es un software de minería de datos desarrollado en Java[Java]. Su nombre lo toma de un pájaro flightless con una naturaleza inquisitiva, encontrado solamente en las islas de Nueva Zelanda.

Weka es una colección de algoritmos de aprendizaje, para las tareas que minan los datos. Los algoritmos se pueden aplicar directamente a una base de datos o llamar desde su propio código de Java. Weka contiene las herramientas para el proceso previo de los datos, la clasificación, la regresión, la selección, las reglas de la asociación y la visualización. Ésta también se adapta para desarrollar nuevos esquemas de aprendizaje. Weka es software libre publicado bajo licencia al público en general.

Un gran desarrollo en informática es la invención y el uso de métodos de aprendizaje de máquina. Éstos permiten a un programa de computadora analizar automáticamente una gran base de datos y decidir qué información es la más relevante. Esta información obtenida se puede utilizar para hacer automáticamente predicciones o ayudar a tomar las decisiones más rápidas y más exactas.

Este paquete de aprendizaje de máquina está públicamente disponible y presenta una colección de los algoritmos para solucionar problemas que minan de los datos del mundo real. WEKA es un /toolbench/, es decir un conjunto de herramientas para aprendizaje de maquina y minería de datos.

Cualquier algoritmo de aprendizaje en WEKA se deriva de la clase abstracta del clasificador. Para un clasificador básico solo es necesario: una rutina que genera un modelo del clasificador de una base de datos del entrenamiento y de otra rutina que evalúe el modelo generado en una base de datos, (clasificación de instancias).

El paquete de weka.filters se refiere a las clases que transforman bases de datos eliminando o agregando atributos. Este paquete ofrece ayuda útil para el proceso previo de los datos, el cual es un paso importante en el aprendizaje de máquina.

Todos los filtros ofrecen las opciones para especificar bases de datos de entrada, y para especificar bases de datos de la salida. Si cualesquiera de estos parámetros no se dan, éste especifica la entrada estándar respectivamente, es decir, si no tienes datos de entrada los

toma por defaul. Otros parámetros son específicos a cada filtro y se pueden descubrir mediante otra vía, como con cualquier otra clase.

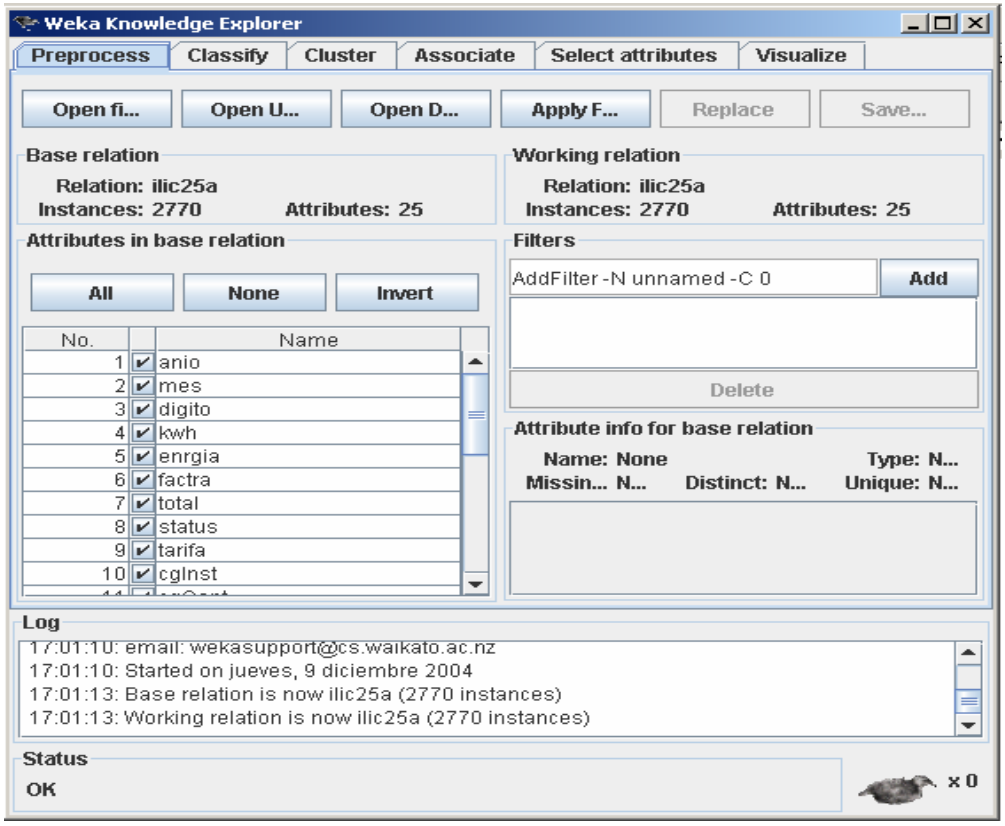

Figura 3.4 Interfaz de WEKA.

CAPÍTULO 4

ANÁLISIS DEL CASO BASE

4.1 CASO BÁSE

Una vez que se tiene el modelo, se espera obtener una solución matemática. Se debe tener en cuenta que las soluciones que se obtienen en este punto del proceso, son matemáticas y se deben interpretar para el mundo real. Además, para la solución del modelo, se debe realizar análisis de sensibilidad, es decir, ver como se comporta el modelo a cambios en las especificaciones y parámetros del modelo. Esto se hace, debido a que los parámetros no necesariamente son precisos y las restricciones pueden estar equivocadas.

El caso base cuenta con 18 puntos, 9 en la tabla A y 9 en la tabla B, los cuales se consideran ilicitos.

4.2 RESULTADOS

Para analizar la solución se debe abrir el archivo completo LST (CLASIFICACION.LST), que en este caso se muestra en el listado 4.1:

```
 1  GAMS Rev 135  Microsoft Window                06/17/04 09:58:54 Page 1
 2  G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
 3  C o m p i l a t i o n
 4
 5
 6
 7   SE TRATA DE RESOLVER UN PROBLEMA DE PROGRAMACIÓN LINEAL,
 8   PARA LA CLASIFICACIÓN DE ATRIBUTOS POR MEDIO DE  MINIMIZACIÓN
 9   CÓNCAVA (BRADLEY & MANGASARIAN).
10   EL PROBLEMA ES:
11                          MIN F(W,R,Y,Z) = ((e**TY/M) + (e**TZ/K))
12                          SUJETO A:          -AW+eR+e<=Y,
13                                              BW-eR+e<=Z,
14                                              Y>=0, Z>=0
15  DONDE:
16
17  W       SON LOS COEFICIENTES DE LAS VARIABLES EN LA ECUACIÓN
18          DE LA RECTA (PLANO O HIPERPLANO).
19  R       ES EL TERMINO INDEPENDIENTE EN LA ECUACIÓN DE LA RECTA
20  Y       REPRESENTA LA MAGNITUD DEL ERROR DE UN PUNTO MAL
21          UBICADO, DE LA TABLA A(I,J)
22  Z       REPRESENTA LA MAGNITUD DEL ERROR DE UN PUNTO MAL
23          UBICADO, DE LA TABLA B(I,J)
24  e       ES UN VECTOR DE UNOS EN UN ESPACIO REAL DE DIMENSIÓN
25          ARBITRARIA
26  e**T    ES LA TRANSPUESTA DE UN VECTOR DE UNOS EN UN ESPACIO
27          REAL DE DIMENSIÓN ARBITRARIA
28  M       ES EL TOTAL DE PUNTOS DE LA TABLA A(I,J)
29  K       ES EL TOTAL DE PUNTOS DE LA TABLA B(I,J)
30  A       SON LOS PUNTOS DE LA TABLA A(I,J) QUE ESTÁN EN EL ESPACIO
31          REAL m*n (DONDE m = INSTANCIAS EN A Y n = ATRIBUTOS EN A)
32  B       SON LOS PUNTOS DE LA TABLA B(I,J) QUE ESTÁN EN EL ESPACIO
33          REAL k*n (DONDE k = INSTANCIAS EN B Y n = ATRIBUTOS EN B)
34  FOBJ    ES LA FUNCIÓN OBJETIVO
35  RA      ES UNA RESTRICCIÓN CON RESPECTO A LOS PUNTOS DE LA
```

```
36            TABLA A(I,J)
37  RB        ES UNA RESTRICCIÓN CON RESPECTO A LOS PUNTOS DE LA
38            TABLA B(I,J)
39  F         ES EL VALOR NUMÉRICO DE LA FUNCIÓN OBJETIVO
40
41  31
42  32  SET
43  33  I /I1*I9/
44  34  J /J1,J2/;
45  35
46  36  TABLE
47  37  A(I,J)
48  38            J1      J2
49  39  I1        10      6
50  40  I2        9       6
51  41  I3        9       7
52  42  I4        8       7
53  43  I5        8       8
54  44  I6        7       8
55  45  I7        7       9
56  46  I8        7       2
57  47  I9        3       6;
58  48
59  49  TABLE
60  50  B(I,J)
61  51            J1      J2
62  52  I1        1       3
63  53  I2        2       2
64  54  I3        2       3
65  55  I4        3       1
66  56  I5        3       2
67  57  I6        4       0
68  58  I7        4       1
69  59  I8        8       3
70  60  I9        4       7;
71  61
72  62  VARIABLES
73  63  W(J),R,Y(I),Z(I),F;
74  64
75  65  POSITIVE VARIABLES
76  66  Y,Z;
77  67
78  68  SCALAR M/9/,K/9/;
79  69
80  70  EQUATIONS
81  71  FOBJ,RA(I),RB(I);
82  72  FOBJ..      F=E=((SUM(I,Y(I))/M)+(SUM(I,Z(I))/K));
83  73  RA(I)..     SUM(J,-1*A(I,J)*W(J))+R+1=L=Y(I);
84  74  RB(I)..     SUM(J,B(I,J)*W(J))-R+1=L=Z(I);
85  75
86  76   MODEL CLASIFICACION /ALL/;
87  77
88  78   OPTION ITERLIM=1e8;
89  79
90  80   SOLVE CLASIFICACION USING LP MINIMIZING F;
91
```

```
92
93  COMPILATION TIME     =        0.000 SECONDS    0.8 Mb      WIN211-135
 94
 95
============================================================
 96  GAMS Rev 135  Microsoft Window                    06/17/04 09:58:54 Page 2
 97  G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
 98  Equation Listing    SOLVE CLASIFICACION Using LP From line 80
 99
100
101  ---- FOBJ  =E=
102
103  FOBJ..  - 0.111111111111111*Y(I1) - 0.111111111111111*Y(I2)
104
105           - 0.111111111111111*Y(I3) - 0.111111111111111*Y(I4)
106
107           - 0.111111111111111*Y(I5) - 0.111111111111111*Y(I6)
108
109           - 0.111111111111111*Y(I7) - 0.111111111111111*Y(I8)
110
111           - 0.111111111111111*Y(I9) - 0.111111111111111*Z(I1)
112
113          - 0.111111111111111*Z(I2) - 0.111111111111111*Z(I3)
114
115          - 0.111111111111111*Z(I4) - 0.111111111111111*Z(I5)
116
117          - 0.111111111111111*Z(I6) - 0.111111111111111*Z(I7)
118
119           - 0.111111111111111*Z(I8) - 0.111111111111111*Z(I9)+ F =E= 0 ; (LHS 0)
120
121
122  ---- RA  =L=
123
124  RA(I1)..  - 10*W(J1) - 6*W(J2) + R - Y(I1) =L= -1 ; (LHS = 0, INFES = 1 ***)
125
126  RA(I2)..  - 9*W(J1) - 6*W(J2) + R - Y(I2) =L= -1 ; (LHS = 0, INFES = 1 ***)
127
128  RA(I3)..  - 9*W(J1) - 7*W(J2) + R - Y(I3) =L= -1 ; (LHS = 0, INFES = 1 ***)
129
130  RA(I3)..  - 9*W(J1) - 7*W(J2) + R - Y(I3) =L= -1 ; (LHS = 0, INFES = 1 ***)
131
132  REMAINING 6 ENTRIES SKIPPED
133
134
135  ---- RB  =L=
136
137  RB(I1)..  W(J1) + 3*W(J2) - R - Z(I1) =L= -1 ; (LHS = 0, INFES = 1 ***)
138
139  RB(I2)..  2*W(J1) + 2*W(J2) - R - Z(I2) =L= -1 ; (LHS = 0, INFES = 1 ***)
140
141  RB(I3)..  2*W(J1) + 3*W(J2) - R - Z(I3) =L= -1 ; (LHS = 0, INFES = 1 ***)
142
143  REMAINING 6 ENTRIES SKIPPED
144
============================================================
145  GAMS Rev 135  Microsoft Window                    06/17/04 09:58:54 Page 3
```

```
146 General Algebraic Modeling System
147 Column Listing      SOLVE CLASIFICACION Using LP From line 80
148
149
150 ---- W
151
152 W(J1)
153                (.LO, .L, .UP = -INF, 0, +INF)
154       -10      RA(I1)
155        -9      RA(I2)
156        -9      RA(I3)
157        -8      RA(I4)
158        -8      RA(I5)
159        -7      RA(I6)
160        -7      RA(I7)
161        -7      RA(I8)
162        -3      RA(I9)
163         1      RB(I1)
164         2      RB(I2)
165         2      RB(I3)
166         3      RB(I4)
167         3      RB(I5)
168         4      RB(I6)
169         4      RB(I7)
170         8      RB(I8)
171         4      RB(I9)
172
173 W(J2)
174                (.LO, .L, .UP = -INF, 0, +INF)
175        -6      RA(I1)
176        -6      RA(I2)
177        -7      RA(I3)
178        -7      RA(I4)
179        -8      RA(I5)
180        -8      RA(I6)
181        -9      RA(I7)
182        -2      RA(I8)
183        -6      RA(I9)
184         3      RB(I1)
185         2      RB(I2)
186         3      RB(I3)
187         1      RB(I4)
188         2      RB(I5)
189         1      RB(I7)
190         3      RB(I8)
191         7      RB(I9)
192
193
194 ---- R
195
196 R
197                (.LO, .L, .UP = -INF, 0, +INF)
198         1      RA(I1)
199         1      RA(I2)
200         1      RA(I3)
201         1      RA(I4)
```

```
202        1       RA(I5)
203        1       RA(I6)
204        1       RA(I7)
205        1       RA(I8)
206        1       RA(I9)
207       -1       RB(I1)
208       -1       RB(I2)
209       -1       RB(I3)
210       -1       RB(I4)
211       -1       RB(I5)
212       -1       RB(I6)
213       -1       RB(I7)
214       -1       RB(I8)
215       -1       RB(I9)
216
217
218  ---- Y
219
220  Y(I1)
221                (.LO, .L, .UP = 0, 0, +INF)
222     -0.1111    FOBJ
223     -1         RA(I1)
224
225  Y(I2)
226                (.LO, .L, .UP = 0, 0, +INF)
227     -0.1111    FOBJ
228     -1         RA(I2)
229
230  Y(I3)
231                (.LO, .L, .UP = 0, 0, +INF)
232     -0.1111    FOBJ
233     -1         RA(I3)
234
235  REMAINING 6 ENTRIES SKIPPED
236
237  ---- Z
238
239  Z(I1)
240                (.LO, .L, .UP = 0, 0, +INF)
241     -0.1111    FOBJ
242     -1         RB(I1)
243
244  Z(I2)
245                (.LO, .L, .UP = 0, 0, +INF)
246     -0.1111    FOBJ
247     -1         RB(I2)
248
249  Z(I3)
250                (.LO, .L, .UP = 0, 0, +INF)
251     -0.1111    FOBJ
252     -1         RB(I3)
253
254  REMAINING 6 ENTRIES SKIPPED
255
256  ---- F
257
```

```
258  F
259             (.LO, .L, .UP = -INF, 0, +INF)
260       1       FOBJ
261
============================================================
262  GAMS Rev 135  Microsoft Window                  06/17/04 09:58:54 Page 4
263  G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
264  Model Statistics    SOLVE CLASIFICACION Using LP From line 80
265
266
267   MODEL STATISTICS
268
269  BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS       19
270  BLOCKS OF VARIABLES       5     SINGLE VARIABLES       22
271  NON ZERO ELEMENTS        90
272
273
274  GENERATION TIME      =       0.081 SECONDS    1.6 Mb      WIN211-135
275
276
277  EXECUTION TIME       =       0.081 SECONDS    1.6 Mb      WIN211-135
278
============================================================
279  GAMS Rev 135  Microsoft Window                  06/17/04 09:58:54 Page 5
280  G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
281  Solution Report     SOLVE CLASIFICACION Using LP From line 80
282
283
284              S O L V E     S U M M A R Y
285
286     MODEL   CLASIFICACION        OBJECTIVE  F
287     TYPE    LP                   DIRECTION  MINIMIZE
288     SOLVER  CPLEX                FROM LINE  80
289
290  **** SOLVER STATUS     1 NORMAL COMPLETION
291  **** MODEL STATUS      1 OPTIMAL
292  **** OBJECTIVE VALUE         0.5333
293
294  RESOURCE USAGE, LIMIT         0.100     1000.000
295  ITERATION COUNT, LIMIT        8         100000000
296
297  GAMS/Cplex    Jun  2, 2003 WIN.CP.NA 21.1 023.025.041.VIS For Cplex 8.1
298  Cplex 8.1.0, GAMS Link 23
299
300  Optimal solution found.
301  Objective :     0.533333
302
303                       LOWER     LEVEL     UPPER    MARGINAL
304
305  ---- EQU FOBJ           .         .         .           1.000
306
307  ---- EQU RA
308
309         LOWER     LEVEL     UPPER    MARGINAL
310
311  I1      -INF      -1.200    -1.000          .
```

```
312  I2      -INF       -1.000       -1.000      3.123E-17
313  I3      -INF       -1.200       -1.000           .
314  I4      -INF       -1.000       -1.000           .
315  I5      -INF       -1.200       -1.000           .
316  I6      -INF       -1.000       -1.000       -0.044
317  I7      -INF       -1.200       -1.000           .
318  I8      -INF       -1.000       -1.000       -0.111
319  I9      -INF       -1.000       -1.000       -0.111
320
321  ---- EQU RB
322
323       LOWER    LEVEL    UPPER   MARGINAL
324
325  I1      -INF       -1.200       -1.000           .
326  I2      -INF       -1.200       -1.000           .
327  I3      -INF       -1.000       -1.000       -0.044
328  I4      -INF       -1.200       -1.000           .
329  I5      -INF       -1.000      -1.000        .
330  I6      -INF       -1.200      -1.000        .
331  I7      -INF       -1.000      -1.000        .
332  I8      -INF       -1.000      -1.000       -0.111
333  I9      -INF       -1.000      -1.000       -0.111
334
335  ---- VAR W
336
337       LOWER    LEVEL    UPPER   MARGINAL
338  J1      -INF        0.200       +INF           .
339  J2      -INF        0.200       +INF           .
340
341                    LOWER    LEVEL    UPPER   MARGINAL
342
343  ---- VAR R         -INF        2.000       +INF           .
344
345  ---- VAR Y
346
347       LOWER    LEVEL    UPPER   MARGINAL
348
349  I1         .          .          +INF        0.111
350  I2         .          .          +INF        0.111
351  I3         .          .          +INF        0.111
352  I4         .          .          +INF        0.111
353  I5         .          .          +INF        0.111
354  I6         .          .          +INF        0.067
355  I7         .          .          +INF        0.111
356  I8         .        1.200        +INF          .
357  I9         .        1.200        +INF          .
358
359  ---- VAR Z
360
361       LOWER    LEVEL    UPPER   MARGINAL
362
363  I1         .          .          +INF        0.111
364  I2         .          .          +INF        0.111
365  I3         .          .          +INF        0.067
366  I4         .          .          +INF        0.111
367  I5         .          .          +INF        0.111
```

```
368  I6        .           .          +INF        0.111
369  I7        .           .          +INF        0.111
370  I8        .         1.200        +INF         .
371  I9        .         1.200        +INF         .
372
373                    LOWER    LEVEL    UPPER    MARGINAL
374
375  ---- VAR F         -INF      0.533       +INF            .
376
377  **** REPORT SUMMARY :       0  NONOPT
378                              0  INFEASIBLE
379                              0  UNBOUNDED
380
381
382  EXECUTION TIME      =       0.010 SECONDS    0.8 Mb      WIN211-135
383
384  USER: GAMS Development Corporation, Washington, DC   G871201:0000XXXXX
385   Free Demo,  202-342-0180,  sales@gams.com,  www.gams.com   DC9999
386
387   **** FILE SUMMARY
388
389   INPUT        C:\INV. GAMS\CLASIFICACION.GMS
390   OUTPUT       C:\GAMSDIR\CLASIFICACION.LST
```

Listado 4.1 Salida de GAMS del archivo CLASIFICACION.LST

4.3 ANÁLISIS DE RESULTADOS

Se explicará la construcción de un archivo de datos completo, al cual se le da el nombre de CLASIFICACION.LST, la extensión es la que por defecto se usa en GAMS para identificar a los archivos de datos de salida.

Siguiendo el listado 4.1.1 se procede a analizar cada bloque de resultados.

- De la línea 1 a la 3 se tiene el nombre del software que se está usando (GAMS), así como la fecha y la hora en que fue compilado y el número de pagina (1, en este caso).

- De la línea 7 a la 39 se tienen las *líneas de comentario*, las cuales no forman parte del modelo y por lo tanto no van a ser compiladas, pero facilitan la lectura posterior tanto del archivo de datos como de la solución, en este caso se especifica el problema y que significa cada una de las variables.

- De la línea 42 a la 44 se tiene el *bloque de conjuntos* SET, el cual consiste en definir una serie de conjuntos o índices y asignarles valores a estos conjuntos. El símbolo (*) permite definir conjuntos de índices de manera compacta.

- De la línea 46 a la 70 se tiene el *bloque de conjuntos* TABLE, que contiene tablas que permiten definir un vector de dos o más dimensiones. En este caso las matrices de datos A(I,J) y B(I,J) tienen el índice I y el índice J; los vectores iniciales de cada matriz se asignan a cada una de las posiciones (I1.J1, I1.J2, . . . I9.J2).

- De la línea 72 a la 76 se tiene el *bloque de variables*, dentro de este bloque se definen las variables y su clase, que se van a usar en el modelo, en este caso son variables positivas.

- En la línea 78 se tiene SCALAR, en el cuál se declara y se le asigna el valor inicial. Por ejemplo; le asigna valor de 9 a M y a K.

- De la línea 80 a la 84 se tiene el *bloque de ecuaciones*, en este bloque se declaran y definen las ecuaciones que van a usarse en el modelo, empezando por la función objetivo (FOBJ) y después se definen las variables. En el renglón 82 se pone el símbolo =E= para indicar que la restricción es de "igualdad" y en los renglones 83 y 84 se pone =L= para indicar que la restricción es de "menor o igual que".

- De la línea 86 a la 90 se tiene el *bloque de modelo y de solución*, aquí se definen las variables que forman parte del modelo y que tipo de modelo es, así como la dirección de optimización que debe seguir (minimizar). Por ejemplo, el renglón 86 significa que se quiere el análisis de todos los datos introducidos en el archivo. En el renglón 88 se tiene OPTION ITERLIM=1e8 con el cual el usuario GAMS puede incrementar el límite máximo por defecto (10,000 iteraciones) a 100,000,000. Y en el renglón 90 SOLVE se utiliza para resolver el problema definido. En este caso este comando indica a GAMS que resuelva el modelo CLASIFICACION usando un optimizador de Programación Lineal (LP) que minimice el valor de la función objetivo.

Hasta aquí es el archivo de entrada, que en este caso se llama CLASIFICACION, que también está incluido dentro del archivo de salida CLASIFICACION.LST.

De aquí en adelante se interpretan los datos que arroja GAMS en el archivo LST.

- En la línea 93 se tiene el tiempo que tardó en realizar el proceso de compilación, que en este caso fue tan rápido que no se alcanza a registrar y el espacio de almacenamiento que ocupó, fue de 0.8 Mb.

- De la línea 96 a la 98 se tiene el nombre del software que se está usando (GAMS), así como la fecha y la hora, y el número de página (2); aquí dice que va a dar un listado de ecuaciones que resolvió para este modelo de nombre clasificación, también dice que está usando Programación Lineal aplicada a la instrucción declarada en la línea 90, (línea 80 de GAMS).

- De la línea 101 a la 119 se tiene el *listado de la función objetivo*. Por ejemplo; para este caso de la función objetivo, GAMS escribe todas las variables en el primer miembro de la ecuación, por eso aparecen con coeficientes negativos las variables principales del problema. El termino (LHS = 0) significa que el término de la izquierda toma el valor de cero. Al no definir un punto de partida inicial, se toma por defecto el cero.

- De la línea 122 a la 143 se tienen las *ecuaciones de las restricciones.* Por ejemplo; en el renglón 124 se observan dos igualdades después del (;) la primera igualdad es la valuación del termino dependiente (LHS) de la restricción RA(I1) para el punto inicial. La segunda indica que esta restricción es infactible en el punto inicial, aunque no significa que sea infactible en cualquier punto. El símbolo (***) al final de la formulación de la restricción afectada indica que el archivo de salida de las restricciones lineales son infactibles para el punto inicial.

- De la línea 145 a la 147 se tiene el nombre del software que se está usando (GAMS), así como la fecha y la hora y el número de página (3), aquí dice que va a dar un listado de columnas que resolvió para el modelo de nombre "clasificación", también dice que está usando Programación Lineal (LP) aplicada a la instrucción declarada en la línea 90, (línea 80 de GAMS).

- De la línea 150 a la 260 se tiene el *listado de columnas o de variables, e*n él aparecen relacionadas todas las variables y los coeficientes que se incorporan en cada ecuación. Se puede observar que las variables W(J1), W(J2), R y F tienen una cota inferior o valor mínimo (LO o LOWER) de –INF, el punto de partida (L o LEVEL) es cero, y la cota superior o valor máximo (UP o UPPER) es +INF. Para las variables Y y Z se tiene una cota inferior (LO o LOWER) de 0 (esto es porque las variables han sido declaradas como no negativas), el punto de partida (L o LEVEL) es cero, y la cota superior o valor máximo (UP o UPPER) es +INF

  Los valores máximos y mínimos que se asocian a una restricción dependen de su carácter. De este modo, si la restricción es del tipo "menor o igual que" como en este caso, el valor mínimo es su término independiente, y el máximo es +∞ (+INF en el archivo de salida GAMS). Si en el archivo de entrada no se especificó el límite inferior de la variable, se considera que este valor es –INF y si no se especificó el límite superior de la variable, el valor es (+INF).

- De la línea 262 a la 264 se tiene el nombre del software que se está usando (GAMS), así como la fecha y la hora, y el número de pagina (4), aquí dice que va a dar estadísticas del modelo que resolvió, también dice que está usando Programación Lineal (LP) aplicada a la instrucción declarada en la línea 90, (línea 80 de GAMS).

- De la línea 267 a la 271 se tienen las *estadísticas del modelo***,** las cuales señalan:
  3 bloques ecuaciones: FOBJ, RA, RB.
  19 ecuaciones: 1 de FOBJ, 9 de RA y 9 de RB.
  5 bloques de variables: W, R, Y, Z y F.
  22 variables: 2 de W, 1 de R, 9 de Y, 9 de Z y 1 de F.
  90 elementos diferentes de cero: 1 de FOBJ, 9 de RA, 9 de RB, 18 de W(J1), 17 de W(J2), 18 de R, 9 de Y y 9 de Z.

- La línea 274 muestra el tiempo que tarda en el proceso de generar (0.081 segundos) y el espacio de almacenamiento que ocupó, el cual fue de 1.6 Mb.

- La línea 277 muestra el tiempo que tarda en el proceso de ejecutar (0.081 segundos) y el espacio de almacenamiento que ocupó, el cual fue de 1.6 Mb.

- De la línea 279 a la 281 se tiene el nombre del software que se está usando (GAMS), así como la fecha y la hora, y el número de página (5), informa que va a dar un reporte sobre la solución que resolvió para este modelo, también dice que está usando Programación Lineal (LP) aplicada a la instrucción declarada en la línea 90, (línea 80 de GAMS).

- De la línea 284 a la 301 se tiene *el resumen de la solución***,** aquí se pueden distinguir tres partes diferenciadas:

  a) De la línea 286 a la 288 se tiene la referida al proceso de solución; la cual dice que el modelo se llama CLASIFICACION, es de tipo lineal, la dirección es minimizar la función objetivo (F) y el /solver/ que usó para resolverlo es CPLEX.
  b) De la línea 290 a la 292 se tiene la referida al valor de las variables y al comportamiento de las ecuaciones. En está parte se observa que el bloque comienza con cuatro asteriscos (****), lo cual indica que es importante, ya que se ha encontrado una solución óptima con un valor de 0.53333.
  c) En las líneas 294 y 295 se tiene la referida a los recursos utilizados, la cual indica que se han usado 0.100 de los 1,000 recursos y se han realizado 8 iteraciones de las 100, 000, 000 posibles.

4.4 ANÁLISIS DE SENSIBILIDAD

- En la línea 305 se tiene la función objetivo, la cual tiene un valor de 1 en su valor marginal (la columna MARGINAL recoge el valor de los multiplicadores de las restricciones, es decir, las variables duales), lo que quiere decir que al hacer un cambio en las variables primales la función objetivo variará una unidad expresada en centímetros.

- A partir de la línea 307 hasta el 319 se muestran las restricciones RA con respecto a la tabla A(I,J), las cuales tiene una cota inferior (LOWER) de –INF, el valor óptimo (LEVEL) es –1.2 y –1.0, la cota superior (UPPER) es –1. En el valor marginal tienen la siguiente interpretación:

  RAI1, RAI3, RAI5, RAI7. Tienen capacidad no utilizada positiva, es decir, (1.200 - 1.000 = 0.200, el signo no se toma en cuenta), por lo tanto el valor interno (precio sombra) de estas distancias es cero.
  RAI6, RAI8, RAI9. Se utilizan todos los recursos (1.000 –1.000 = 0), por lo tanto el valor interno (precio sombra) de esta distancia es abundante y tienen valores de 0.044, 0.111, 0.111 respectivamente.
  RAI2, RAI4. Aquí se utilizan todos los recursos, por lo tanto el valor interno (precio sombra) de esta distancia es abundante, aunque en un valor muy pequeño, ya que su costo marginal es de casi cero, por ejemplo la RAI2 tiene un costo marginal de 0.00000000000000003123 y la RAI4 tiene un valor todavía muy pequeño, tanto que GAMS lo toma como cero.

- De la línea 321 a la 333 se muestran las restricciones Rb con respecto a la tabla B(I,J), las cuales tiene una cota inferior (LOWER) de –INF, el valor óptimo (LEVEL)

es –1.2 y –1.0, la cota superior (UPPER) es –1. En el valor marginal tienen la siguiente interpretación:

RBI1, RBI2, RBI4, RBI6. Tienen capacidad no utilizada positiva, es decir, (1.200 - 1.000 = 0.200, el signo no se toma en cuenta), por lo tanto el valor interno (precio sombra) de estas distancias es cero.
RBI3, RBI8, RBI9. Se utilizan todos los recursos (1.000 –1.000 = 0), entonces, el valor interno (precio sombra) de estas distancias es abundante y tienen valores de 0.044, 0.111, 0.111 respectivamente.
RBI5, RBI7. Aquí se utilizan todos los recursos, por lo tanto el valor interno (precio sombra) de estas distancias es abundante, aunque en un valor muy pequeño, ya que su costo marginal es de casi cero, tanto que GAMS lo toma como cero.

- En las líneas 338, 339 y 343 se interpretan las variables principales que son los coeficientes de la recta, según la solución que proporciona GAMS, se tienen las variables WJ1, WJ2 y R, las cuales tiene un valor de 0.2, 0.2 y 2 respectivamente y su rango puede variar de –INFINITO a +INFINITO. Estas variables no tiene valor marginal, ya que son variables básicas en el primal, por lo tanto son variables de holgura en el dual. No se pueden cambiar, ni variar, ya que un pequeño cambio en ellas cambia la función objetivo.

- De la línea 345 a la 357 se tienen las variables de Y, en las que se puede ver una cota inferior de 0 y una cota superior es +INF. Los costos de oportunidad de las variables de holgura no básicas tienen una interpretación que en este problema está directamente relacionada con la interpretación de las variables duales. Por lo que la solución se mantiene como óptima mientras los beneficios de las diferentes distancias estén comprendidos entre las variables:

  YI1, YI2, YI3, YI4, YI5, YI6, YI7. Tienen un valor de cero, y su costo marginal es de 0.111 y de 0.067 para YI6, lo que indica que si se aumenta una unidad más, ésta variaría 0.111 centímetros y 0.067 centímetros para YI6. Aquí también se puede ver que son puntos bien ubicados, ya que su valor es cero.
  YI8, YI9. Tienen un valor de 1.2, están en el punto óptimo y no pueden variar ningún punto ni en "X" ni en "Y", porque se modificaría la función objetivo. También quiere decir que es un punto mal ubicado ya que tienen valor.

- De la línea 359 a la 371 se tienen las variables de Z en las cuales se puede ver una cota inferior de 0 y una cota superior es +INF. Los costes de oportunidad de las variables de holgura no básicas tienen una interpretación que en este problema está directamente relacionada con la interpretación de las variables duales. Por lo que la solución se mantiene como óptima mientras los beneficios de las diferentes distancias estén comprendidos entre las variables:

  ZI1, ZI2, ZI3, ZI4, ZI5, ZI6, ZI7. Tienen un valor de cero, y su costo marginal es de 0.111 y de 0.067 para ZI3, lo que indica que si se aumenta una unidad más, ésta variaría 0.111 centímetros y 0.067 centímetros para ZI3. Aquí también se puede ver que son puntos bien ubicados, ya que su valor es cero.

ZI8, ZI9. Tienen un valor de 1.2, están en el punto óptimo y no pueden variar ningún punto ni en "X" ni en "Y", porque se modificaría la función objetivo. También quiere decir que es un punto mal ubicado ya que tienen valor.

- En la línea 375 se tiene la interpretación de la función objetivo, la cual representa la distancia mínima promedio, de la recta a un punto mal colocado, dicha distancia será de 0.5333 centímetros. Ésta puede variar desde –INFINITO hasta +INFINITO. Tiene un valor óptimo mínimo.

  Siendo la solución W(J1)=0.2, W(J2)=0.2, R=2, Y(I8)=1.2, Y(I9)=1.2, Z(I8)=1.2, Z(I9)=1.2 con un valor de la función objetivo de 0.533 (F=0.533).

  El valor óptimo de una restricción se obtiene al evaluar su término dependiente una vez que se ha resuelto el problema. Al igual que el valor óptimo, el costo marginal se obtiene como resultado de la optimización. Este valor representa la variación que experimenta el valor de la función objetivo frente a un cambio en el término independiente de la restricción (valor de su variable dual).

- De la línea 377 a la 379 se tiene el *informe resumen*. Todas las instrucciones que tengan cuatro asteriscos (****) son muy importantes. En este caso, no hay soluciones no óptimas, ni infactibles, ni no acotadas, es decir, la solución es normal y óptima.

- En la línea 382 se tiene el tiempo de ejecución, el cual muestra el tiempo que tarda en el proceso para ejecutar (0.010 segundos) y el espacio de almacenamiento que ocupó, el cual fue de 0.8 Mb.

- Las líneas 384 y 385 dicen que se está usando el demo libre de GAMS, además, viene la página web.

- De la línea 387 a la 390 se tiene el *resumen de origen y final de los archivos*, es decir, el nombre de los archivos de entrada y salida, así como su localización.
  Por ejemplo, para el archivo de entrada, está guardado en disco C:/ en la carpeta INV. GAMS, con el nombre de CLASIFICACION.GMS.

En la figura 4.1, los triángulos son los puntos de la tabla A(I,J) y los cuados son puntos de la tabla B(I,J). La línea recta (T) se basa en lo siguiente:

$$0.2\ A + 0.2\ B - 2 = 0$$

Para efectos gráficos lo tenemos así:

$$0.2\ X + 0.2\ Y - 2 = 0$$
$$X=2/0.2 \rightarrow X=10$$
$$Y=2/0.2 \rightarrow Y=10$$

La línea recta está minimizando el error de la clasificación.

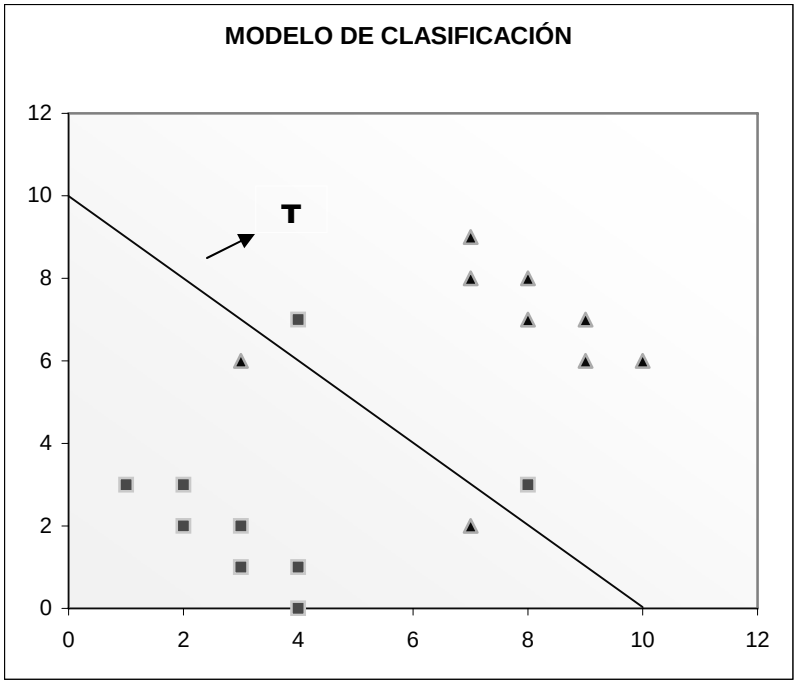

Figura 4.1 Modelo de Clasificación.

4.5 VARIACIÓNES EN ALFA(α) Y LAMBDA(λ)

Se harán pruebas con diferentes alfa's y lambda's con el modelo del caso básico, para saber con cuales se obtienen mejores resultados, basándose en este modelo; y de ahí tomar algunos valores de alfa's y lambda's como base para los demás experimentos. Para posteriormente aplicarlo en el modelo reformulado con todos los datos completos.

Los resultados son los siguientes:

α	λ	F	WJ1	WJ2	WJ3	R
0.1	0.1	0.4878	0.200	0.200	0	2.000
0.1	0.5	0.3059	0.200	0.200	0	2.000
0.1	1.0	0	0	0	0	-1.000
0.2	0.1	0.4878	0.200	0.200	0	2.000
0.2	0.5	0.3059	0.200	0.200	0	2.000
0.2	1.0	0	0	0	0	-1.000
0.3	0.1	0.4916	0.200	0.200	0	2.000
0.3	0.5	0.3249	0.200	0.200	0	2.000
0.3	1.0	0	0	0	0	-1.000
0.4	0.1	0.4954	0.200	0.200	0	2.000
0.4	0.5	0.3436	0.200	0.200	0	2.000
0.4	1.0	0	0	0	0	-1.000
0.5	0.1	0.4990	0.200	0.200	0	2.000
0.5	0.5	0.3618	0.200	0.200	0	2.000
0.5	1.0	0	0	0	0	-1.000
0.6	0.1	0.5026	0.200	0.200	0	2.000
0.6	0.5	0.3797	0.200	0.200	0	2.000
0.6	1.0	0	0	0	0	-1.000
0.7	0.1	0.5061	0.200	0.200	0	2.000
0.7	0.5	0.3973	0.200	0.200	0	2.000
0.7	1.0	0	0	0	0	-1.000

0.8	0.1	0.5096	0.200	0.200	0	2.000
0.8	0.5	0.4145	0.200	0.200	0	2.000
0.8	1.0	0	0	0	0	-1.000
0.9	0.1	0.5129	0.200	0.200	0	2.000
0.9	0.5	0.4314	0.200	0.200	0	2.000
0.9	1.0	0	0	0	0	-1.000
1.0	0.1	0.5163	0.200	0.200	0	2.000
1.0	0.5	0.4479	0.200	0.200	0	2.000
1.0	1.0	0	0	0	0	–1.000
2.0	0.1	0.5459	0.200	0.200	0	2.000
2.0	0.5	0.5963	0.200	0.200	0	2.000
2.0	0.9	0.2000	0	0	0	0
3.0	0.1	0.5702	0.200	0.200	0	2.000
3.0	0.5	0.7083	0.167	0.167	0	1.667
3.0	0.9	0.2000	0	0	0	0
4.0	0.1	0.5901	0.200	0.200	0	2.000
4.0	0.5	0.8014	0.167	0.167	0	1.667
4.0	0.9	0.2000	0	0	0	0
5.0	0.1	0.6064	0.200	0.200	0	2.000
5.0	0.5	0.8802	0.167	0.167	0	1.667
5.0	0.9	0.2000	0	0	0	0.300
6.0	0.1	0.6198	0.200	0.200	0	2.000
6.0	0.5	0.9469	0.167	0.167	0	1.667
6.0	0.9	0.2000	0	0	0	-1.000
7.0	0.1	0.6307	0.200	0.200	0	2.000
7.0	0.5	1.0034	0.167	0.167	0	1.667
7.0	0.9	0.2000	0	0	0	1.000
8.0	0.1	0.6396	0.200	0.200	0	2.000
8.0	0.5	1.0512	0.167	0.167	0	1.667
8.0	0.9	0.2000	0	0	0	1.000
9.0	0.1	0.6469	0.200	0.200	0	2.000
9.0	0.5	1.0917	0.167	0.167	0	1.667
9.0	0.9	0.2000	0	0	0	1.000

Lo que se está buscando es minimizar el valor de la función objetivo el cual se encuentra en la columna tres; para diferentes alfa's y lambda's los valores pequeños son 0 y 0.2 los cuales son infactibles. El valor mínimo factible es de F=0.3059 y el valor máximo factible es de F=1.0917. En la figura 4.2 se muestran las variaciones de la función objetivo.

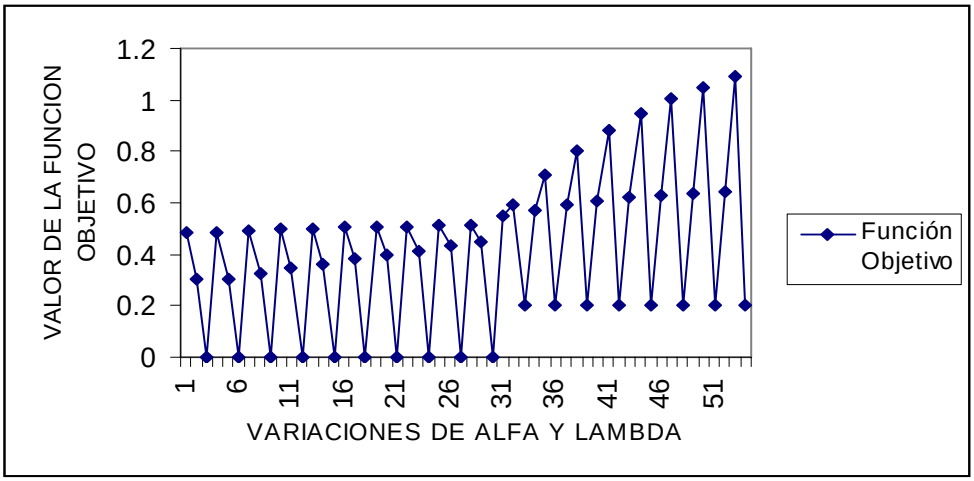

Figura 4.2 Variaciones de la Función Objetivo.

Después de probar con diferentes alfa's y lambda's se optó por utilizar las siguientes:

α=0.1, 1, 5, 20 y 100
λ= 0.01, 0.05, 0.2, 0.05 y 0.9

Porque de este modo se obtienen diferentes valores y los rangos son más variados, así que se podrán tener mejores resultados para ser analizados.

CAPÍTULO 5

APORTACIONES AL MODELO FSV

5.1 EJEMPLIFICACIÓN DEL PROCESO

El proceso que se llevará a cabo en cada uno de los siguientes casos se muestra en la figura 5.1, en donde se tiene una base de ilícitos la cuál es introducida al programa Gams y posteriormente al programa Neos de donde se obtiene la selección de atributos optimizada, de ahí se toman los atributos seleccionados y se introducen al programa Weka, en donde se utilizan los clasificadores j48.J48 y j48Part, los cuáles arrojan el número de árboles y reglas, el tiempo de procesamiento, el porcentaje de instancias correctamente clasificadas y el costo total.

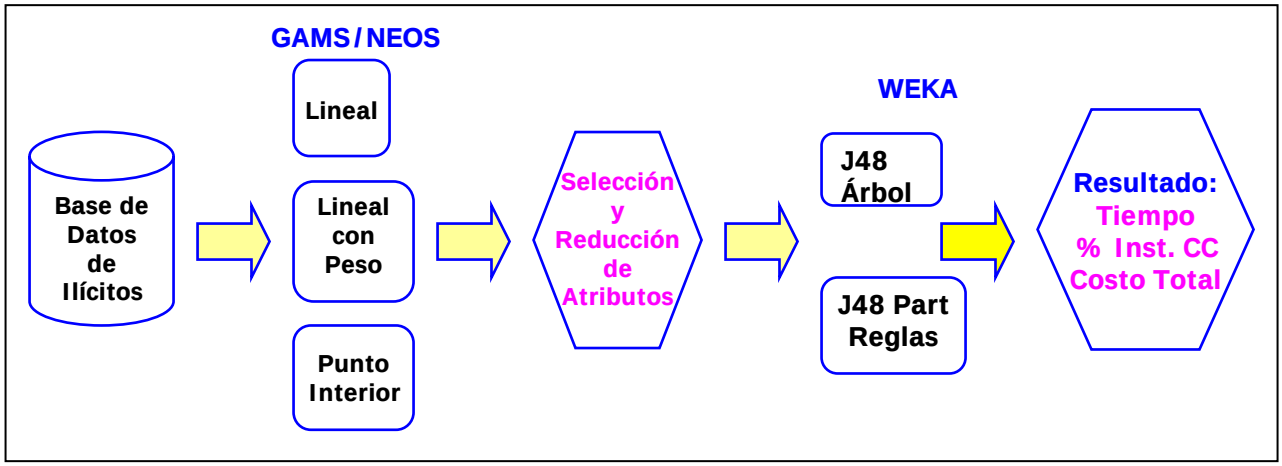

Figura 5.1 Proceso llevado a cabo en cada experimentación.

5.2 CASO 1: COSTO BENEFICIO

El modelo FSV en su formulación actual no considera el aspecto de costo-beneficio en la clasificación de las clases, el cual es un aspecto importantísimo en aplicaciones del mundo real, por lo tanto se trabajará en una formulación de FSV que tome en cuenta este aspecto.

Para tratar de explicarlo, se hará aplicándolo al problema de ilícitos. Como se muestra en la figura 5.1, empezando por la base de datos de ilícitos se elabora un algoritmo, él cual le asigna mas peso a ciertas instancias, este algoritmo es introducido al programa GAMS y posteriormente se manda a NEOS para obtener la selección de atributos, y por ultimo se introducen los atributos seleccionados en el software Weka, ahí se hace minería de datos aplicando costo de sensibilidad. El modelo predice con el software de Weka lo siguiente:

- Si la instancia es ilícito y realmente si lo es gana 97.5
- Si la instancia es ilícito y no lo es pierde 2.5
- Si la instancia no es ilícito y si lo es pierde 97.5
- Si la instancia no es ilícito y realmente no lo es gana 2.5

El costo de que un verificador vaya al lugar del ilícito y compruebe que es cierto o falso lo que el modelo predijo es de 2.5, por eso si comprueba que es cierto el ilícito se gana 100 menos el costo de ir a verificar, se obtiene una ganancia total de 97.5.

5.2.1 RESULTADOS

Después de hacer varios experimentos con /solvers/ como BDMLP, Xpress-MP, SBB, CONOPT, SNOPT, MINOS, MOSEK, MILES y PATH, se observó que para el modelo de esta tesis PATH es más rápido que los demás y lo hace en pocas iteraciones, es por ello que de aquí en adelante se utilizará este /solver/ para hacer los experimentos.

En las siguientes tablas de esta sección la primera columna es PNI:1-PI:2 que significa punto no ilícito (PNI), punto ilícito (PI) y el número 1 y 2 es el peso que se le da a cada uno. Las pruebas dieron los siguientes resultados:

PATH						
PNI:1 - PI:2	**ALFA**	**LAMBDA**	**ATS-GAMS**	**VAL.FUN.OBJ.**	**TIEMPO(SEG)**	**ITERACIONES**
PNI:1 - PI:2	0.1	0.01	17	0.7898	13	1999
PNI:1 - PI:2	0.1	0.05	16	0.7715	11	1844
PNI:1 - PI:2	1	0.01	16	0.8198	16	1852
PNI:1 - PI:1.5	0.1	0.01	17	0.7346	13	1844
PNI:1 - PI:1.5	0.1	0.05	17	0.7185	11	1938
PNI:1 - PI:1.5	1	0.01	16	0.7644	10	1899
PNI:1 - PI:2.5	0.1	0.01	17	0.8417	12	2134
PNI:1 - PI:2.5	0.1	0.05	17	0.8231	11	1939
PNI:1 - PI:2.5	1	0.01	17	0.8736	12	1946
PNI:1 - PI:10	0.1	0.01	18	1.3687	10	2290
PNI:1 - PI:10	0.1	0.05	18	1.3275	10	2256
PNI:1 - PI:10	1	0.01	16	1.3989	21	2227
PNI:1 - PI:50	0.1	0.01	18	1.9186	11	2603
PNI:1 - PI:50	0.1	0.05	18	1.8492	11	2384
PNI:1 - PI:50	1	0.01	18	1.9329	10	2345
PNI:2.5 - PI:97.5	0.1	0.01	18	4.6918	11	2457
PNI:2.5 - PI:97.5	0.1	0.05	18	4.5376	13	2430
PNI:2.5 - PI:97.5	1	0.01	18	4.7395	9	2372
PNI:2 - PI:1	0.1	0.01	15	0.7809	9	978
PNI:2 - PI:1	0.1	0.05	13	0.751	9	960
PNI:2 - PI:1	1	0.01	10	0.7839	19	932
PNI:1.5 - PI:1	0.1	0.01	14	0.7331	9	1264
PNI:1.5 - PI:1	0.1	0.05	13	0.7047	13	1128
PNI:1.5 - PI:1	1	0.01	13	0.7361	7	1101
PNI:2.5 - PI:1	0.1	0.01	14	0.8293	8	986
PNI:2.5 - PI:1	0.1	0.05	16	0.7968	7	847
PNI:2.5 - PI:1	1	0.01	11	0.832	9	840

PNI:10 - PI:1	0.1	0.01	17	1.4246	12	1237
PNI:10 - PI:1	0.1	0.05	15	1.3772	7	1204
PNI:10 - PI:1	1	0.01	14	1.4474	7	1157
PNI:50 - PI:1	0.1	0.01	17	1.8015	6	913
PNI:50 - PI:1	0.1	0.05	16	1.741	12	902
PNI:50 - PI:1	1	0.01	16	1.8287	7	933
PNI:97.5 - PI:2.5	0.1	0.01	17	4.3342	7	1026
PNI:97.5 - PI:2.5	0.1	0.05	16	4.1721	7	1021
PNI:97.5 - PI:2.5	1	0.01	16	4.3614	13	942

Tabla 5.1 Resultados obtenidos por GAMS Y NEOS con el software PATH.

PATH	**L**	**ÁRBOLES**					
PNI:1 – PI:2	**ALFA**	**LAMBDA**	**ATS-GAMS**	**ÁRBOL**	**TIEMPO(SEG.)**	**% INST.CC**	**CT**
PNI:1 - PI:2	0.1	0.01	17	39	2.48	97.3646	54820
PNI:1 - PI:2	0.1	0.05	16	33	2.03	97.509	54650
PNI:1 - PI:2	1	0.01	16	39	1.92	97.4007	55015
PNI:1 - PI:1.5	0.1	0.01	17	39	1.98	97.2924	54430
PNI:1 - PI:1.5	0.1	0.05	17	39	1.96	97.3646	54630
PNI:1 - PI:1.5	1	0.01	16	39	1.88	97.4007	54825
PNI:1 - PI:2.5	0.1	0.01	17	33	1.97	97.4368	54640
PNI:1 - PI:2.5	0.1	0.05	17	39	2.09	97.3285	54815
PNI:1 - PI:2.5	1	0.01	17	39	1.99	97.3646	54820
PNI:1 - PI:10	0.1	0.01	18	39	2.18	97.2924	54240
PNI:1 - PI:10	0.1	0.05	18	39	2.18	97.3285	54435
PNI:1 - PI:10	1	0.01	16	21	1.86	95.5596	48490
PNI:1 - PI:50	0.1	0.01	18	39	2.1	97.4007	54825
PNI:1 - PI:50	0.1	0.05	18	39	2.11	97.3285	54435
PNI:1 - PI:50	1	0.01	18	39	2.65	97.3285	54435
PNI:2.5 - PI:97.5	0.1	0.01	18	39	5.04	97.3646	54440
PNI:2.5 - PI:97.5	0.1	0.05	18	39	2.91	97.2924	54240
PNI:2.5 - PI:97.5	1	0.01	18	39	2.11	97.3285	54435
PNI:2 - PI:1	0.1	0.01	15	39	1.83	97.3646	54630
PNI:2 - PI:1	0.1	0.05	13	46	1.54	96.787	52650
PNI:2 - PI:1	1	0.01	10	33	1.28	90.5415	26135
PNI:1.5 - PI:1	0.1	0.01	14	47	2.2	97.0036	54390
PNI:1.5 - PI:1	0.1	0.05	13	21	1.57	95.5596	48300
PNI:1.5 - PI:1	1	0.01	13	21	1.57	95.5596	48110

PNI:2.5 - PI:1	0.1	0.01	14	39	1.97	97.4007	54445
PNI:2.5 – PI:1	0.1	0.05	16	39	1.83	97.4729	54835
PNI:2.5 – PI:1	1	0.01	11	83	1.41	90.6498	25580
PNI:10 – PI:1	0.1	0.01	17	39	1.92	97.4368	54830
PNI:10 – PI:1	0.1	0.05	15	41	1.74	97.0397	54395
PNI:10 – PI:1	1	0.01	14	53	1.62	96.6787	53585
PNI:50 – PI:1	0.1	0.01	17	39	1.89	97.4729	55025
PNI:50 – PI:1	0.1	0.05	16	39	1.84	97.509	55030
PNI:50 – PI:1	1	0.01	16	39	1.84	97.509	55030
PNI:97.5 - PI:2.5	0.1	0.01	17	39	2.05	97.509	55030
PNI:97.5 - PI:2.5	0.1	0.05	16	39	1.8	97.509	55030
PNI:97.5 - PI:2.5	1	0.01	16	39	1.88	97.509	55030

Tabla 5.2 Resultados según Weka con el clasificador j48.J48, en forma de árbol.

L PATH		**REGLAS**					
PNI:1-PI:2	**ALFA**	**LAMBDA**	**ATS-GAMS**	**REGLAS**	**TIEMPO(SEG.)**	**% INST.CC**	**CT**
PNI:1 - PI:2	0.1	0.01	17	21	3.56	97.1119	55165
PNI:1 - PI:2	0.1	0.05	16	16	2.96	97.1841	54225
PNI:1 - PI:2	1	0.01	16	19	4.1	97.0758	55160
PNI:1 - PI:1.5	0.1	0.01	17	13	3.52	97.0397	54585
PNI:1 - PI:1.5	0.1	0.05	17	20	3.33	97.2202	54420
PNI:1 - PI:1.5	1	0.01	16	20	3.19	97.1841	54605
PNI:1 - PI:2.5	0.1	0.01	17	21	3.75	97.2563	54805
PNI:1 - PI:2.5	0.1	0.05	17	21	3.52	97.2202	55370
PNI:1 - PI:2.5	1	0.01	17	21	3.48	97.1119	55165
PNI:1 - PI:10	0.1	0.01	18	20	3.85	97.148	54030
PNI:1 - PI:10	0.1	0.05	18	23	3.74	97.3285	54815
PNI:1 - PI:10	1	0.01	16	16	3.22	95.343	44850
PNI:1 - PI:50	0.1	0.01	18	16	3.93	97.2563	54425
PNI:1 - PI:50	0.1	0.05	18	23	3.72	97.3285	54815
PNI:1 - PI:50	1	0.01	18	23	3.89	97.3285	54815
PNI:2.5 – PI:97.5	0.1	0.01	18	19	3.88	97.1119	53835
PNI:2.5 – PI:97.5	0.1	0.05	18	20	3.93	97.148	54030
PNI:2.5 – PI:97.5	1	0.01	18	23	3.94	97.3285	54815
PNI:2 - PI:1	0.1	0.01	15	20	3.11	97.2924	54810
PNI:2 - PI:1	0.1	0.05	13	20	2.65	96.6787	53585
PNI:2 - PI:1	1	0.01	10	22	3.16	90.5415	24425

PNI:1.5 - PI:1	0.1	0.01	14	24	3.25	96.3899	53355
PNI:1.5 - PI:1	0.1	0.05	13	18	2.71	95.1986	49390
PNI:1.5 - PI:1	1	0.01	13	13	2.78	95.8123	47385
PNI:2.5 - PI:1	0.1	0.01	14	16	3.19	97.0758	53450
PNI:2.5 - PI:1	0.1	0.05	16	19	3.04	96.3899	51265
PNI:2.5 - PI:1	1	0.01	11	14	2.87	90.1083	26265
PNI:10 – PI:1	0.1	0.01	17	17	3.62	97.2202	54990
PNI:10 – PI:1	0.1	0.05	15	19	3.16	96.7148	53780
PNI:10 – PI:1	1	0.01	14	26	2.89	96.5343	52805
PNI:50 – PI:1	0.1	0.01	17	12	3.47	97.2202	55180
PNI:50 – PI:1	0.1	0.05	16	12	3.42	97.0758	54590
PNI:50 – PI:1	1	0.01	16	12	3.52	97.0758	54590
PNI:97.5 - PI:2.5	0.1	0.01	17	17	3.62	97.1119	54215
PNI:97.5 - PI:2.5	0.1	0.05	16	12	3.4	97.0758	54590
PNI:97.5 - PI:2.5	1	0.01	16	13	3.47	97.0758	54590

Tabla 5.3 Resultados según Weka con el clasificador j48.PART, en forma de regla.

5.2.2 ANÁLISIS DE RESULTADOS

Para realizar los experimentos se tomaron:

- α=0.1 con λ=0.01
- α=0.1 con λ=0.05
- α=1 con λ=0.01,

Estos fueron los mejores resultados obtenidos de una serie de experimentos para escoger las mejores combinaciones de alfa y lambda, de ahí se hicieron pruebas, las cuales arrojaron lo siguiente:

En la tabla 5.1, se observa que las pruebas dieron diferentes resultados, los cuales después de analizarlos se obtiene que los mejores son:

- PNI:2-PI:1 con α=1 y λ=0.01
- PNI:2.5-PI:1 con α=0.1 y λ=0.05

Como se puede observar son diferentes datos los que se consideran los mejores ya que en PNI:2-PI:1 con α=1 y λ=0.01, se obtienen menos atributos que en los demás y el valor de la función objetivo es muy satisfactorio, el tiempo no es tan bueno sin embargo las iteraciones son pocas, que es lo que se desea minimizar.

En la figura 5.2, se observa que el número máximo de atributos es de 18 y que existen varias instancias con ese mismo número de atributos, sin embargo el número mínimo de atributos es 10 y lo tiene PNI:2-PI:1 con α=1 y λ=0.01. En la misma figura también se puede ver que

el valor máximo de la función objetivo es de 4.7395 con PNI:2.5-PI:97.5, α=1 y λ=0.01, y el valor mínimo es de 0.7047 con PNI:1.5-PI:1, α=0.1 y λ=0.05. Por último se puede observar que el tiempo máximo que tarda el programa en compilar los datos es de 21 segundos con PNI:1-PI:10, α=1 y λ=0.01 y el tiempo mínimo es de 6 segundos con PNI:50-PI:1, α=0.1 y λ=0.01.

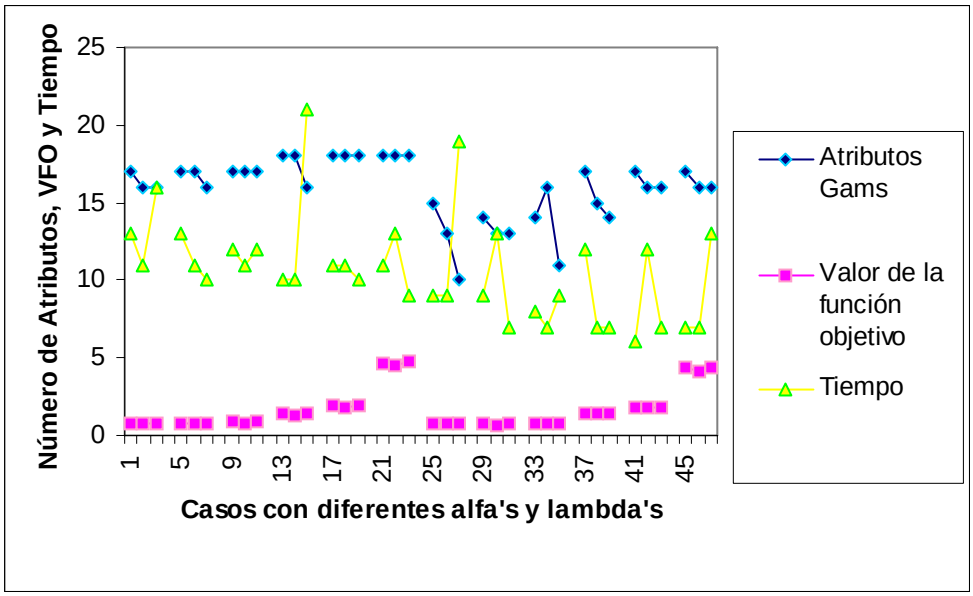

Figura 5.2 Número de atributos, Valor de la Función objetivo y Tiempo, según Gams y Neos.

En cuanto al número de iteraciones se refiere el valor máximo es de 2603 iteraciones con PNI:1-PI:50, α=0.1 y λ=0.01 y el valor mínimo es de 840 iteraciones con PNI:2.5-PI:1, α=1 y λ=0.01, para visualizarlo mejor obsérvese la figura 5.3, en donde se pueden ver las iteraciones para todos los casos con diferentes alfa's y lambda's.

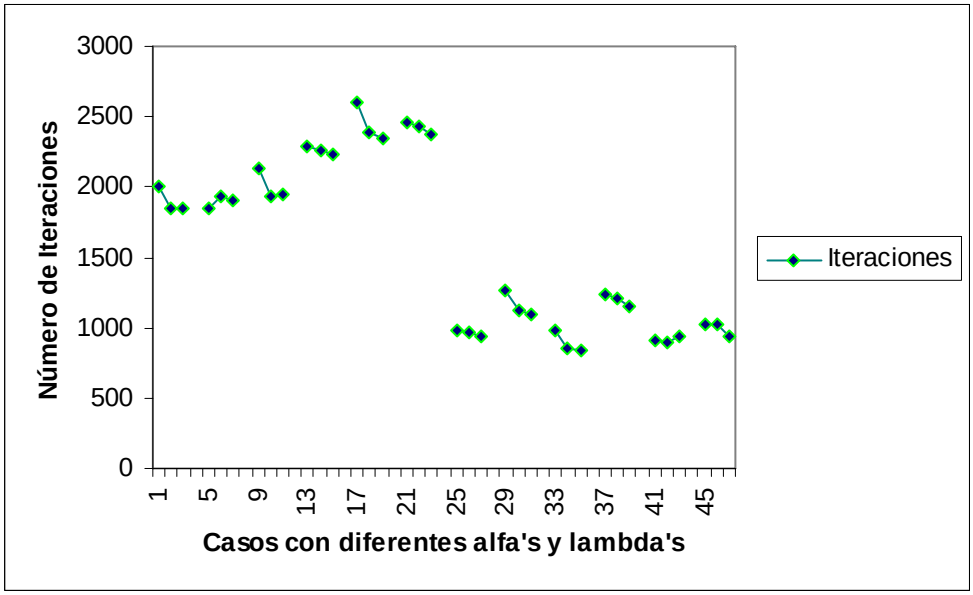

Figura 5.3 Número de Iteraciones, según Gams y Neos.

En la figura 5.4 se observa el tamaño de árboles y reglas, en donde se puede apreciar que el valor máximo de árboles es de 83, un poco disparado para los demás valores, este valor se obtuvo con PNI:2.5-PI:1, α=1 y λ=0.01. El tamaño de árbol mínimo es de 21 y son tres instancias las cuales obtienen el mismo resultado. Sin embargo según Weka para el clasificador en forma de reglas el tamaño máximo es de 20 con PNI:10-PI:1, α=1 y λ=0.01 y el tamaño mínimo es de 12 y son cuatro instancias las que obtiene este mismo resultado, como puede apreciarse en la figura.

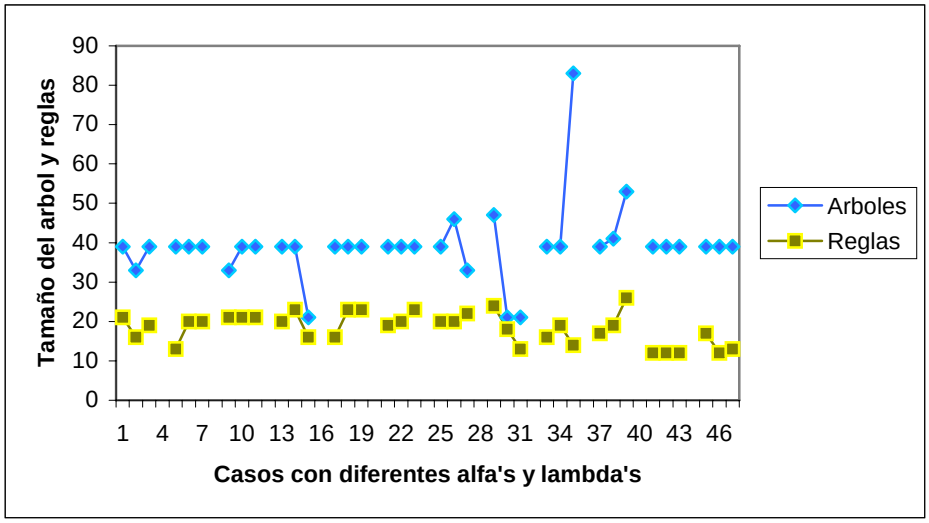

Figura 5.4 Tamaño de árboles y reglas, según Weka.

El tiempo máximo que tarda el programa Weka con el clasificador j48.J48 en forma de árboles es de 5.04 segundos con PNI:2.5-PI:97.5, α=0.1 y λ=0.01 y el tiempo mínimo que tarda es de 1.41 segundos con PNI:2.5-PI:1, α=1 y λ=0.01. Para el clasificador j48.Part en forma de reglas el tiempo máximo es de 4 segundos con PNI:1-PI:2, α=1 y λ=0.01 y el valor mínimo es de 2.65 segundos con PNI:2-PI:1, α=1 y λ=0.01, como se puede ver en la figura 5.5.

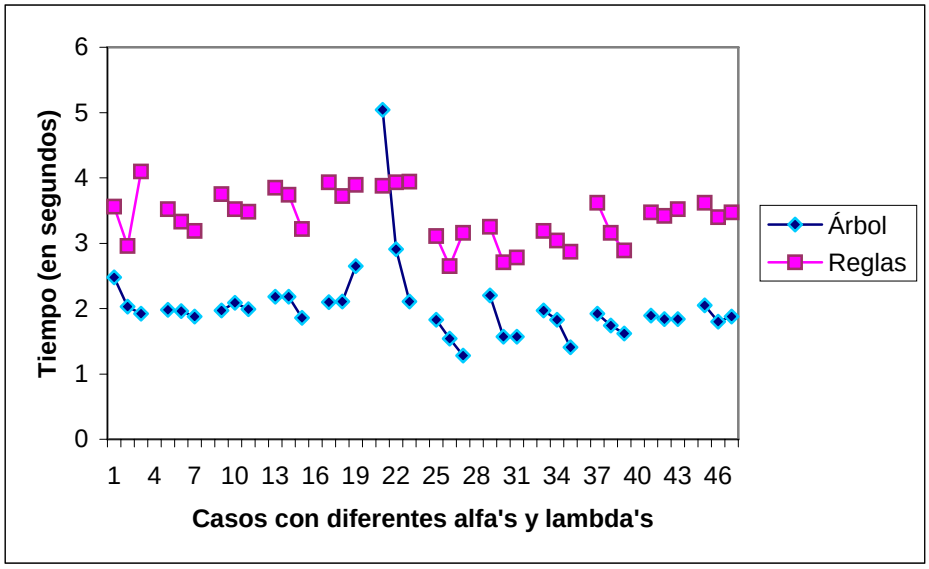

Figura 5.5 Tiempo en segundos, según Weka.

Dando como mejor instancia correctamente clasificada un porcentaje de 97.509, que es altamente favorable y corresponde a cinco instancias por el clasificador en forma de árbol, seguido de 97.3285 con cuatro instancias con el clasificador en forma de reglas. Y el porcentaje mínimo de instancias correctamente clasificadas es de 90.5415 con PNI:2-PI:2, α=1 y λ=0.01 para árboles y de 90.1083 con PNI:2.5-PI:1, α=1 y λ=0.01 para reglas, como se puede observar en la figura 5.6.

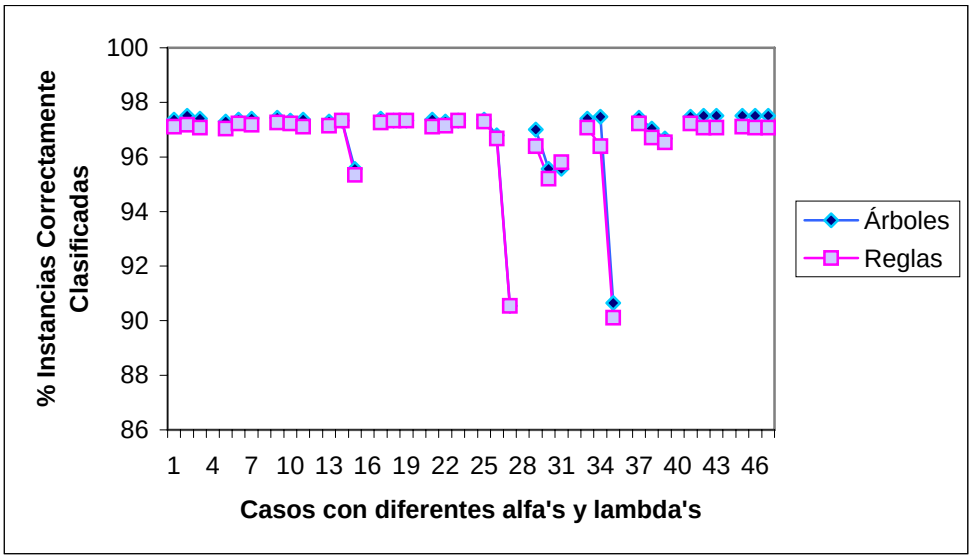

Figura 5.6 Porcentaje de Instancias Correctamente Clasificadas, según Weka.

En la figura 5.7 se observa que el costo total más alto es de 55030 y lo tienen las ultimas cinco instancias de árboles, mientras que el costo total más bajo es de 26135 con PNI:2-PI:1, α=1 y λ=0.01. En cuanto a reglas se refiere el costo total más alto es de 55180 con PNI:50-PI:1, α=0.1 y λ=0.01, y el costo total mas bajo es de 24425 con PNI:2-PI:1, α=1 y λ=0.01.

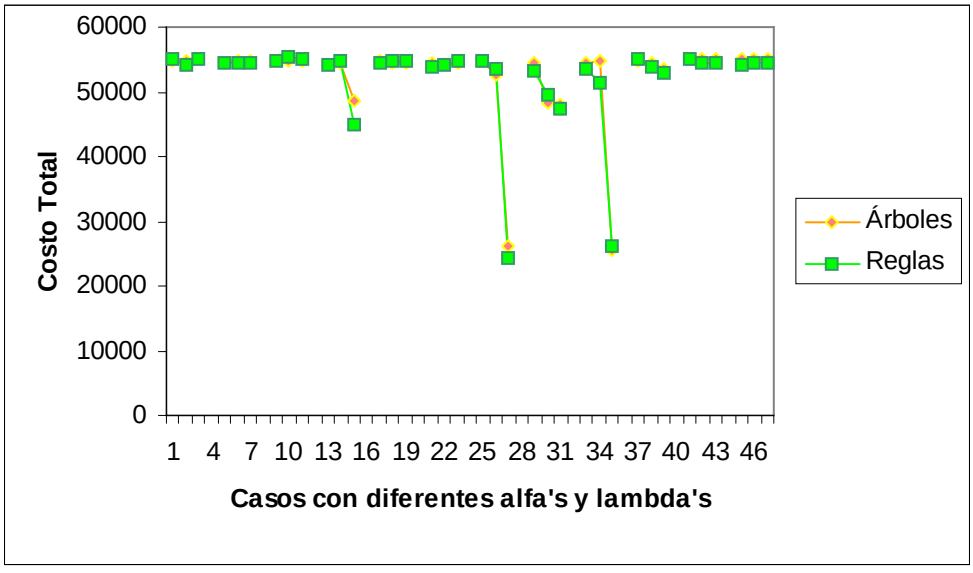

Figura 5.7 Costo Total, según Weka.

Después de un difícil análisis se opto por escoger a peso PNI:1-PI:2.5, α=0.1, λ=0.05 en el cual se seleccionaron 17 atributos, 21 reglas, 3.52 segundos, 97.2202 porcentaje de instancias correctamente clasificadas, 55370 de costo total, él cual fue él más alto obtenido en esta sección.

5.3 CASO 2: LINEALIZACIÓN

Los Problemas de Programación Lineal (PPL) se restringen a problemas cuya función objetivo es lineal y también lo son sus restricciones. Cuando la función objetivo o alguna de las restricciones no es lineal, se tiene un Problema de Programación No Lineal (PPNL). Sin embargo, para ciertos casos, es posible transformar un PPNL en un PPL equivalente. Para el problema que se está usando en esta tesis, se hace la transformación por medio de las series de Taylor, aunque hay diferentes y varios métodos para linealizar.

Se quiere linealizar el modelo para tratar de reducir el tiempo sin afectar el resultado, al mismo tiempo reducir atributos, ya que aplicando linealización se obtiene un resultado más satisfactorio con un número menor de iteraciones.

5.3.1 TEOREMA DE TAYLOR

Se sabe que la recta tangente R(x), (la mejor aproximación lineal a la figura de f en las cercanías del punto de tangencia (x_o, $f(x_o)$)), es aquella recta que pasa por el mencionado punto y tiene la misma pendiente que la curva f(x) en ese punto (primera derivada en el punto), lo que hace que la recta tangente y la curva sean prácticamente indistinguibles en las cercanías del punto de tangencia. En la figura 5.8 se puede observar que la curva toca "suavemente" a la recta en este entorno, de tal manera que de todas las rectas que pasan por el punto, es esta recta la que más se parece a la curva cerca del punto.

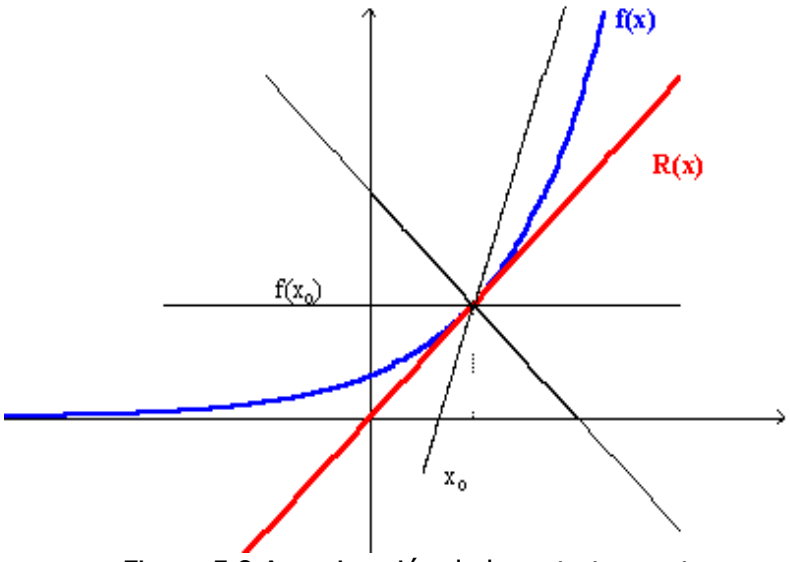

Figura 5.8 Aproximación de la recta tangente.

Cerca del punto de tangencia, la curva se comporta casi linealmente, como se puede apreciar en la figura 5.9, en donde se hacen acercamientos a la figura anterior.

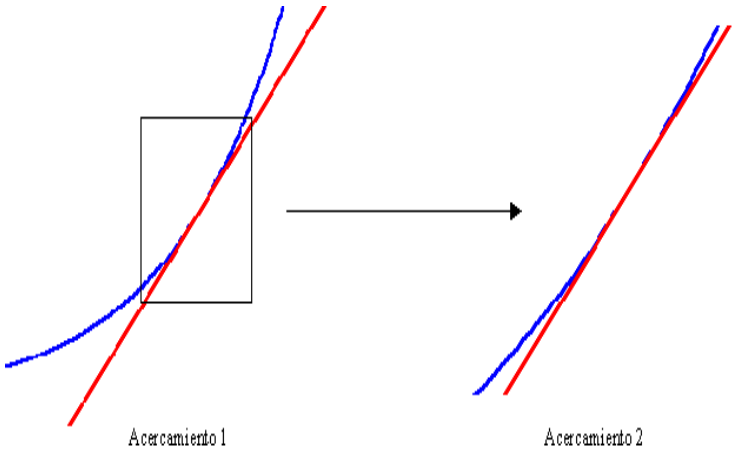

Figura 5.9 Acercamiento de la figura 5.8.

Si x se encuentra alejada de x_o, la recta tangente ya no funciona como aproximador. La recta tangente es un polinomio de grado uno, el más sencillo tipo de función que se puede encontrar, por lo que se puede tratar de ver si es posible encontrar un polinomio de grado dos que sirva para aproximar la función en un rango más grande que la recta tangente.

En lugar de aproximarse con una recta se trata de hacerlo con una parábola, es decir, se tratará de encontrar de todas las parábolas que pasan por $(x_o, f(x_o))$, la que mejor se aproxima a la curva, es decir, es la que tratara de encontrar la parábola tangente P(x), como se muestra en la figura 5.10.

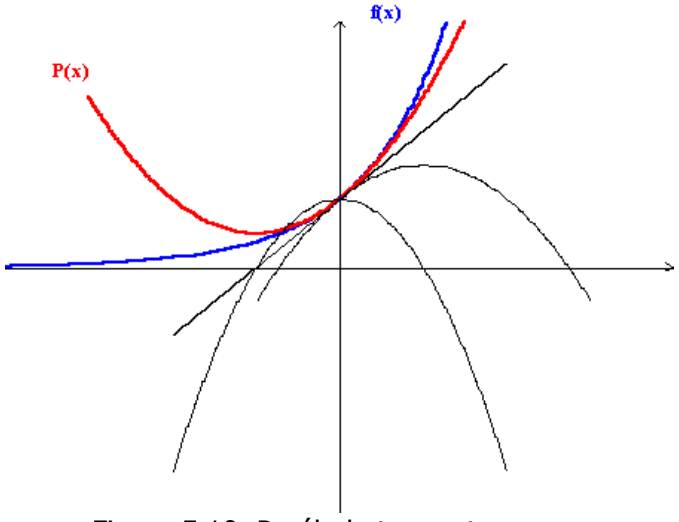

Figura 5.10. Parábola tangente

A esta parábola $\mathbf{P(x) = a + b(x- x_o) + c(x- x_o)^2}$ se le debe pedir que pase por el punto, que tenga la misma inclinación (primera derivada) y la misma concavidad que la parábola (segunda derivada), es decir se le debe pedir:

a) $P(x_o) = f(x_o)$
b) $P'(x_o) = f'(x_o)$
c) $P''(x_o) = f''(x_o)$

Si f no es un polinomio, obviamente no podrá representarse de la misma manera, sin embargo en vista de que para la recta tangente que es un polinomio de grado 1, se cumple que para x cercano a x_o:

$$f(x) \cong f(x_o) + f'(x_o)(x - x_o)$$

y gráficamente se observa que para x cercano a x_o, la función es muy parecida a su "parábola tangente", es decir:

$$f(x) \cong f(x_o) + f'(x_o)(x - x_o) + \frac{f^{(2)}(x_o)}{2!}(x - x_o)^2$$

surge de manera natural preguntarse si para valores cercanos a x_o, se cumplirá:

$$f(x) \cong f(x_o) + f'(x_o)(x - x_o) + \frac{f^{(2)}(x_o)}{2!}(x - x_o)^2 + \ldots + \frac{f^{(n)}(x_o)}{n!}(x - x_o)^n$$

y se puede intentar verlo en algunos casos particulares. Al polinomio:

$$P_n(x) = f(x_o) + f'(x_o)(x - x_o) + \frac{f^{(2)}(x_o)}{2!}(x - x_o)^2 + \ldots + \frac{f^{(n)}(x_o)}{n!}(x - x_o)^n$$

se le llama ***polinomio de Taylor*** de grado n para f, en el punto x_o.

En estos términos, la recta tangente y la parábola tangente, vienen siendo los polinomios de Taylor para f de grados 1 y 2 respectivamente.

En la tabla 5.4 se hace una comparación de la función exponencial $f(x) = e^x$ (última columna) con los polinomios de Taylor correspondientes de grados 1 hasta 4. La segunda columna corresponde a la recta tangente y la tercera columna a la parábola tangente.

x	1+x	$1 + x + \frac{x^2}{2}$	$1 + x + \frac{x^2}{2} + \frac{x^3}{6}$	$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}$	e^x
1	2	2.5	2.666666	2.7083333	2.718281828
0.5	1.5	1.625	1.645833	1.6484375	1.6487212707
0.3	1.3	1.345	1.3495	1.3498375	1.34985880757
0.1	1.1	1.105	1.10516667	1.10517083	1.10517091807
0.01	1.01	1.01005	1.01005017	1.01005017	1.010050167
0.001	1.001	1.0010005	1.00100050000	1.00100050017	1.00100050016

Tabla 5.4 Comparación de la función e^x con los polinomios de Taylor.

Si se analiza con detenimiento la información proporcionada en la tabla 5.4, se verá lo siguiente:

- En cada columna, se muestra que la aproximación del correspondiente polinomio de Taylor es mejor cuanto más cercano se encuentre x a 0.
- En cada renglón, se muestra que para cada valor fijo de x, no importa si está cerca o no de 0, la aproximación va mejorando conforme aumenta el grado del polinomio de Taylor.

Una representación de esta situación se muestra en la figura 5.11 para los polinomios de Taylor de grado 1 (línea roja), 2 (parábola rosa), y 3 (parábola cúbica negra).

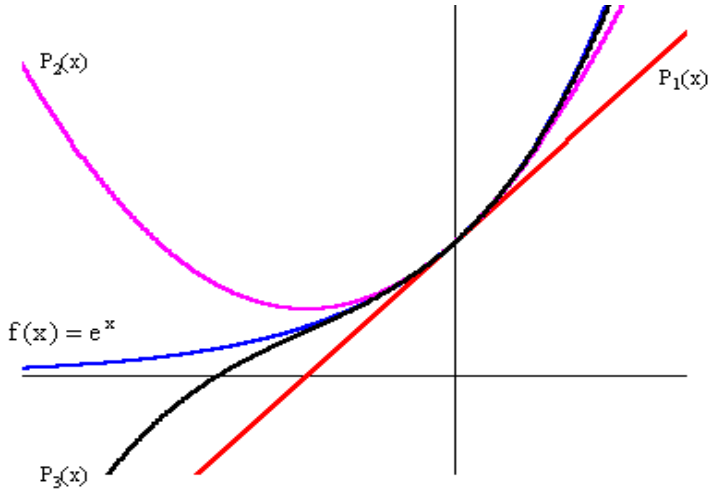

Figura 5.11 Polinomios de Taylor.

El Teorema de Taylor dice que bajo ciertas condiciones, una función puede expresarse como un polinomio de Taylor más un cierto error, es decir:

$$f(x) = P_n(x) + E_n$$

La estimación del error puede hacerse independientemente del cálculo de la aproximación, es decir, antes de calcular ésta se puede preguntar por el grado del polinomio de Taylor que dé la precisión deseada, [Mat].

Para la función con la que se está trabajando en esta tesis y que se quiere linealizar se tiene $e^{-\alpha v}$, donde α toma valores arbitrarios y v es obtenida al correr el programa.

Utilizando el Teorema de Taylor se tiene:

$$f(x) = f(0) + f'(0)(x-0) + \frac{f''(0)}{2!}(x-0)^2 + \frac{f'''(0)}{3!}(x-0)^3 + E_3$$

Para $f(x) = e^x$ en x_o=0

Como la función exponencial y todas sus derivadas son iguales, $f^{(n)}(0)$=1, la fórmula queda:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + E_3$$

Evaluando en x=-0.5, se tiene:

$$e^{-0.5} = 1 + (-0.5) + \frac{(-0.5)^2}{2!} + \frac{(-0.5)^3}{3!} + E^3$$

$$e^{-0.5} = 1 - 0.5 + 0.125 - 0.2083333 + E_3$$

$$e^{-0.5} = .604166667 + E_3$$

Obtenida la linealización se procede a teclear el problema del caso básico al Gams y se compara el resultado obtenido del problema no lineal contra el linealizado. Los datos de entrada del problema lineal se encuentran en el listado 5.1:

```
$ONTEXT

SE TRATA DE RESOLVER UN PROBLEMA DE PROGRAMACIÓN NO LINEAL, PARA LA CLASIFICACIÓN DE
ATRIBUTOS POR MEDIO DE MINIMIZACIÓN CONCAVA (MANGASARIAN & BRADLEY).
EL PROBLEMA ES:
MIN F(W,R,Y,Z,V)=(1-LAMBDA)((e**TY/M)+(e**TZ/K))+(LAMBDA*e**T)(e-E**-  (ALFA*V))
          SUJETO A:      -AW+eR+e<=Y,
                         BW-eR+e<=Z,
                         Y>=0, Z>=0
                         -V<=W<=V
SE LINEALIZO LA FUNCIÓN OBJETIVO POR MEDIO DE LAS SERIES DE TAYLOR, POR LO TANTO EL PROBLEMA SE
CONVIERTE EN UN PROBLEMA DE PROGRAMACIÓN LINEAL (PPL).

$OFFTEXT

SET
I /I1*I9/
J /J1,J2,J3/;

TABLE
A(I,J)
       J1     J2     J3
I1      10     6      8
I2      9      6      8
I3      9      7      8
I4      8      7      8
I5      8      8      8
I6      7      8      8
I7      7      9      8
I8      7      2      8
I9      3      6      8;

TABLE
B(I,J)
       J1     J2     J3
I1      1      3      8
I2      2      2      8
I3      2      3      8
I4      3      1      8
I5      3      2      8
I6      4      0      8
I7      4      1      8
I8      8      3      8
I9      4      7      8;

VARIABLES
W(J), R, Y(I), Z(I), V(J), F;

POSITIVE VARIABLES
Y, Z;

SCALAR M/9/, K/9/, LAMBDA/0.5/, ALFA/0.95/;

EQUATIONS
FOBJ, RA(I), RB(I), RC(J), RD(J);
```

```
FOBJ..    F=E=(1-LAMBDA)*((SUM(I,Y(I))/M)+(SUM(I,Z(I))/K))+LAMBDA*(SUM(J,(1-(EXP(-1*ALFA*V(J))))));
RA(I)..   SUM(J,-1*A(I,J)*W(J))+R+1=L=Y(I);
RB(I)..   SUM(J,B(I,J)*W(J))-R+1=L=Z(I);
RC(J)..   -1*V(J)=L=W(J);
RD(J)..   V(J)=G=W(J);

MODEL LINEALIZADO /ALL/;

SOLVE LINEALIZADO USING LP MINIMIZING F;
```

Listado 5.1 Datos de entrada en Gams, del problema lineal.

Un dato importante de la solución es que el problema linealizado da un valor en la función objetivo de 0.4397 contra 0.4397 que es el valor arrojado del problema de PNL.

5.3.2 RESULTADOS

Los siguientes resultados son del problema de uso ilícito de energía eléctrica, que es el problema que se está manejando en esta tesis; por cuestión de espacio, los resultados de los programas GAMS y NEOS están en el anexo B.

PATH					
ALFA	**LAMBDA**	**ATS-GAMS**	**VAL.FUN.OBJ.**	**TIEMPO(SEG)**	**ITERACIONES**
0.1	0.01	17	0.6719	8	1991
0.1	0.05	16	0.6555	21	2019
0.1	0.2	13	0.5599	7	1461
0.1	0.5	12	0.3624	7	1306
0.1	0.9	6	0.0992	7	1554
1	0.01	15	0.6878	8	1760
1	0.05	10	0.6736	7	1348
1	0.2	10	0.6199	7	1298
1	0.5	7	0.5124	6	1606
1	0.9	5	0.1733	9	2671
5	0.01	11	0.7013	7	1384
5	0.05	5	0.8626	7	2662
5	0.2	6	0.8866	6	1701
5	0.5	5	0.8626	8	2662
5	0.9	5	0.178	8	2696
20	0.01	10	0.7513	7	1296
20	0.05	8	0.9903	8	1739
20	0.2	5	1.3801	8	2662
20	0.5	5	0.8753	9	2680
20	0.9	4	0.1809	8	2737
100	0.01	7	1.018	6	1613
100	0.05	5	1.6394	9	2671
100	0.2	5	1.4061	8	2698

100	0.5	5	0.9015	9	2688
100	0.9	2	0.1828	8	2688

Tabla 5.5 Resultados obtenidos con Gams y Neos, con el software PATH.

PATH	ARBOLES					
ALFA	LAMBDA	ATS-GAMS	ARBOL	TIEMPO(SEG.)	% INST.CC	COSTO TOTAL
0.1	0.01	17	39	2.62	97.4729	55405
0.1	0.05	16	39	1.96	97.509	55410
0.1	0.2	13	21	1.62	95.704	49270
0.1	0.5	12	21	1.54	95.5235	48105
0.1	0.9	6	3	0.57	90.1444	27790
1	0.01	15	41	1.82	97.1841	54415
1	0.05	10	3	1.15	90	28150
1	0.2	10	9	1.17	90.4693	27075
1	0.5	7	3	0.68	90.0722	27780
1	0.9	5	1	0.65	79.3863	-50080
5	0.01	11	31	1.36	95.4874	47530
5	0.05	5	1	0.61	79.3863	-50080
5	0.2	6	3	0.57	90.1444	27790
5	0.5	5	1	0.61	79.3863	-50080
5	0.9	5	1	0.6	79.3863	-50080
20	0.01	10	9	1.18	90.4693	28405
20	0.05	8	9	0.98	90.4332	28210
20	0.2	5	1	0.65	79.3863	-50080
20	0.5	5	1	0.63	79.3863	-50080
20	0.9	4	1	0.56	79.4224	-50075
100	0.01	7	9	1.23	90.6137	27855
100	0.05	5	1	0.63	79.3863	-50080
100	0.2	5	1	0.66	79.3863	-50080
100	0.5	5	1	0.74	79.4224	-50075
100	0.9	2	1	0.27	79.4224	-50075

Tabla 5.6 Resultados obtenidos con Weka con j48.J48 en forma de árbol.

PATH	REGLAS					
ALFA	LAMBDA	ATS-GAMS	REGLAS	TIEMPO(SEG.)	% INST.CC	COSTO TOTAL
0.1	0.01	17	22	3.64	97.0758	53640
0.1	0.05	16	22	3.5	97.0397	53635
0.1	0.2	13	16	2.78	95.1264	44440
0.1	0.5	12	17	2.65	95.3069	47885
0.1	0.9	6	2	0.63	90.1444	27790
1	0.01	15	17	3.01	97.0397	54775
1	0.05	10	8	2.69	90.1805	27795

1	0.2	10	10	2.37	89.9278	25480
1	0.5	7	2	0.74	90.0722	27780
1	0.9	5	1	0.65	79.4224	50075
5	0.01	11	14	2.03	94.9458	45365
5	0.05	5	1	0.61	79.4224	-50075
5	0.2	6	2	0.65	90.1444	27790
5	0.5	5	1	0.62	79.4224	-50075
5	0.9	5	1	0.62	79.4224	-50075
20	0.01	10	11	2.99	90.1083	27405
20	0.05	8	15	2.43	90.2888	28380
20	0.2	5	1	0.67	79.4224	-50075
20	0.5	5	1	0.62	79.4224	-50075
20	0.9	4	1	0.57	79.4224	-50075
100	0.01	7	13	2.03	90.2527	27805
100	0.05	5	1	0.62	79.4224	-50075
100	0.2	5	1	0.6	79.4224	-50075
100	0.5	5	4	0.96	97.4224	-50075
100	0.9	2	2	0.38	79.4224	-50075

Tabla 5.7 Resultados obtenidos con Weka con J48 Part en forma de regla.

5.3.3 ANÁLISIS DE RESULTADOS

Después de hacer las pruebas necesarias se observa que efectivamente es más rápido y a veces se obtienen mejores resultados linealizando, dependiendo de los valores de alfa y lambda. Se hicieron las siguientes figuras para visualizarlo mejor.

Como se puede observar en la figura 5.12, el método lineal obtiene menos atributos que el no lineal y uno de los objetivos es reducir el número de atributos, así que para esta característica es mejor el método lineal. Además está dividido por cinco grupos, el primer grupo corresponde a alfa=0.1, el segundo a alfa=1, el tercero a alfa=5, el cuarto a alfa=20 y el último a alfa=100, cada uno con diferentes lambdas.

Se puede ver en la figura 5.13, la función objetivo al principio es muy semejante en los dos métodos, en medio como es el caso de $\alpha=5$ con los diferentes valores de λ dista un poco, solo hay diferencia con un solo valor de la función objetivo que se dispara y es por el método no lineal.

En la figura 5.14 se nota claramente que para todos los valores o puntos, el tiempo es menor por el método lineal además de que se ve la diferencia entre los dos métodos. El tiempo es algo que se desea reducir y por este método se cumple con el objetivo propuesto anteriormente.

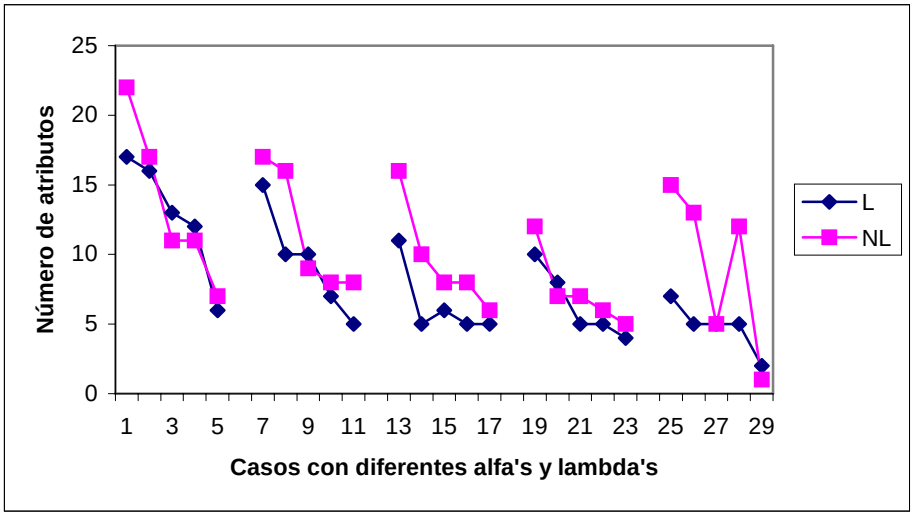

Figura 5.12 Comparación de atributos, según Gams y Neos.

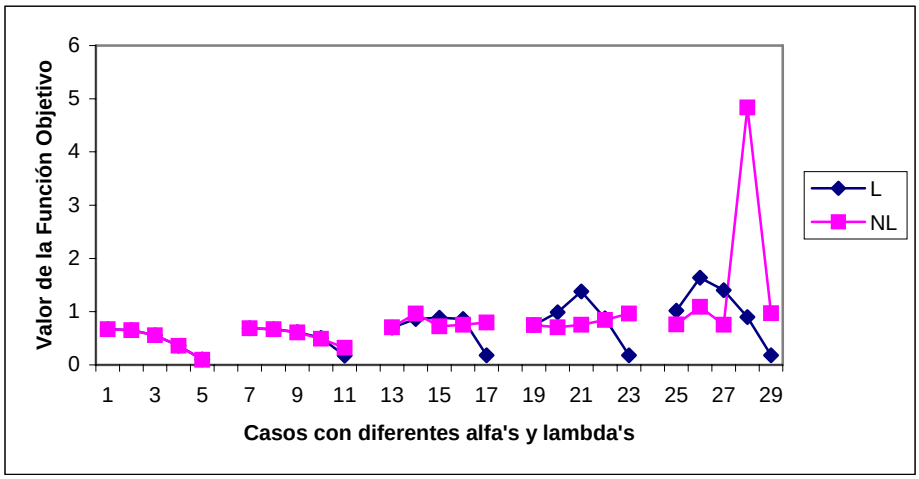

Figura 5.13 Comparación del valor de la función objetivo, según Gams y Neos.

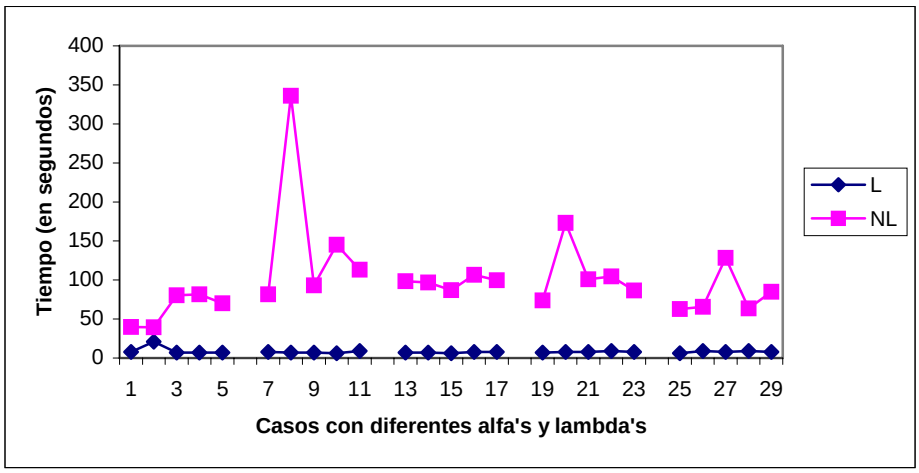

Figura 5.14 Comparación del tiempo (en segundos), según Gams y Neos.

En cuanto a las iteraciones, son mayores por el método lineal, sin embargo esto no repercute a la hora del tiempo en que tarda el programa en iterar, como se pudo apreciar anteriormente ya que el tiempo es menor a pesar de que el método no lineal tiene mucho menos iteraciones, como se puede observar en la figura 5.15.

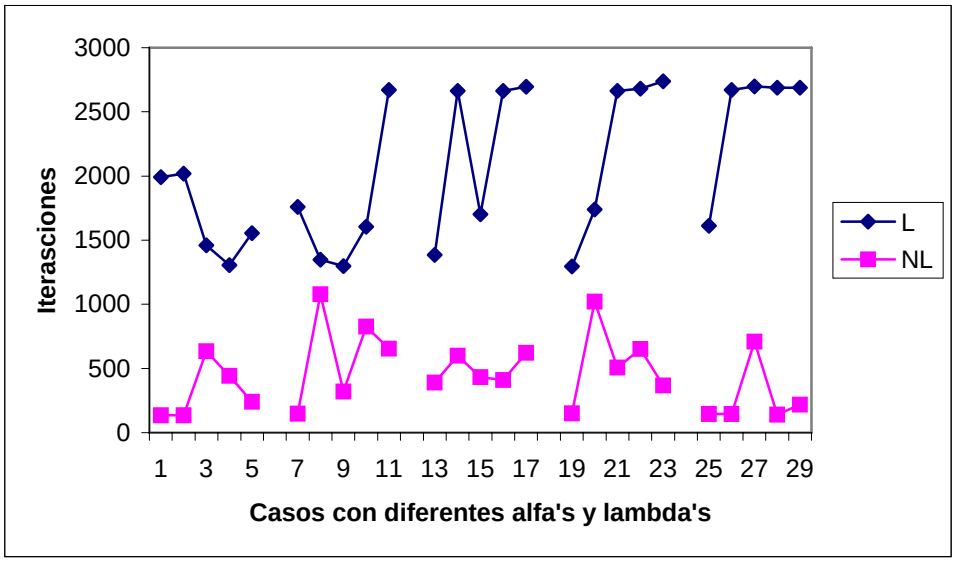

Figura 5.15 Comparación de las iteraciones, según Gams y Neos.

Ahora, se analizaran los resultados (tablas 5.6 y 5.7) obtenidos por Weka con j48.J48 y j48.PART respectivamente, mediante el método lineal y se hará la comparación con el método no lineal.

En la figura 5.16 se observa que el árbol más grande (de 41 árboles), corresponde al método lineal con $\alpha=1$ y $\lambda=0.01$, y el árbol más pequeño también corresponde al mismo método con solo 1 árbol y/o 1 regla. Mientras que en método no lineal el más grande tiene 39 árboles y son cinco instancias las que tienen este mismo número y el árbol más pequeño es de 3, siendo varias instancias las que lo contienen, obsérvese en la figura 5.16.

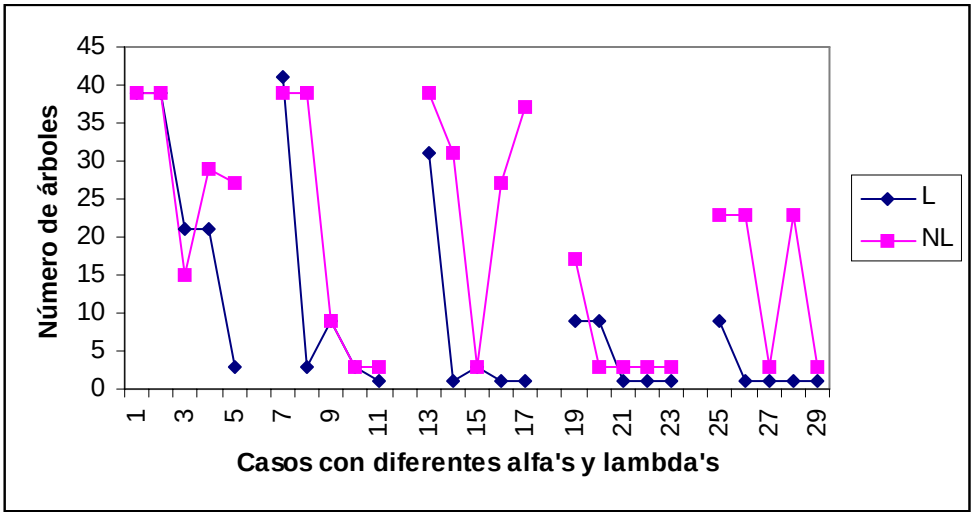

Figura 5.16 Comparación de árboles según Weka, mediante el método lineal y no lineal.

Como se puede observar hay mucha variación entre los resultados obtenidos por los diferentes árboles y reglas, ya que en reglas por el método lineal el máximo número es de 22 con α=0.1-λ=0.01 y α=0.1-λ=0.05, y el mínimo es de 1 regla y son varias instancias las que contienen este mismo número. Para el método no lineal el máximo número de reglas son las mismas que para el método lineal (22 reglas), y el mínimo número de reglas es de 2, como puede observarse en la figura 5.17.

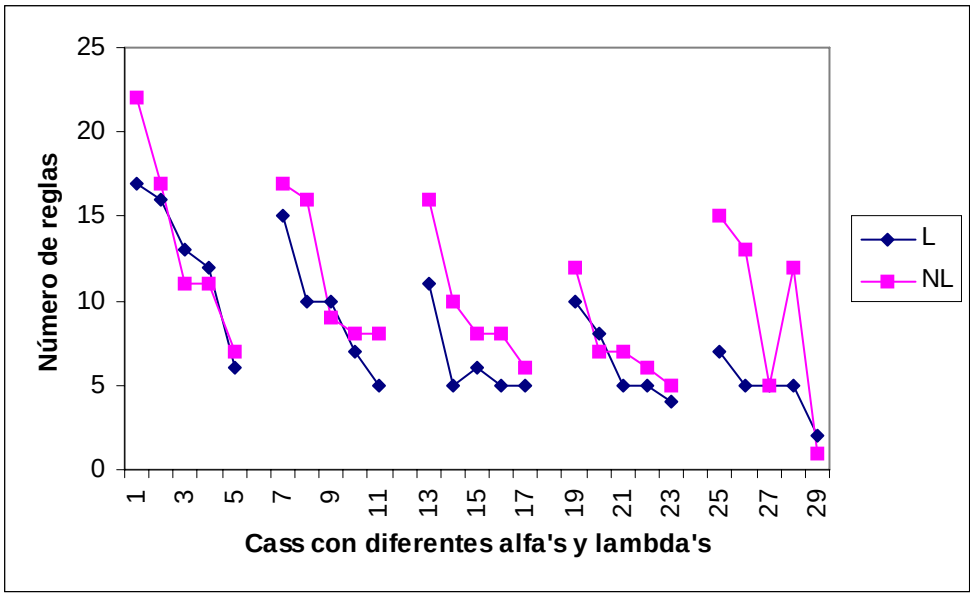

Figura 5.17 Comparación de reglas según Weka, mediante el método lineal y no lineal.

Se observa que en la figura 5.18 que el tiempo más largo por el método lineal para árboles es de 2.62 segundos y el tiempo más pequeño por el mismo método es de 0.27 segundos. Mientras que por el método no lineal el tiempo máximo es de 3.07 segundos con α=0.1-λ=0.01 y el tiempo mínimo es de 0.03 segundos con α=100-λ=0.9.

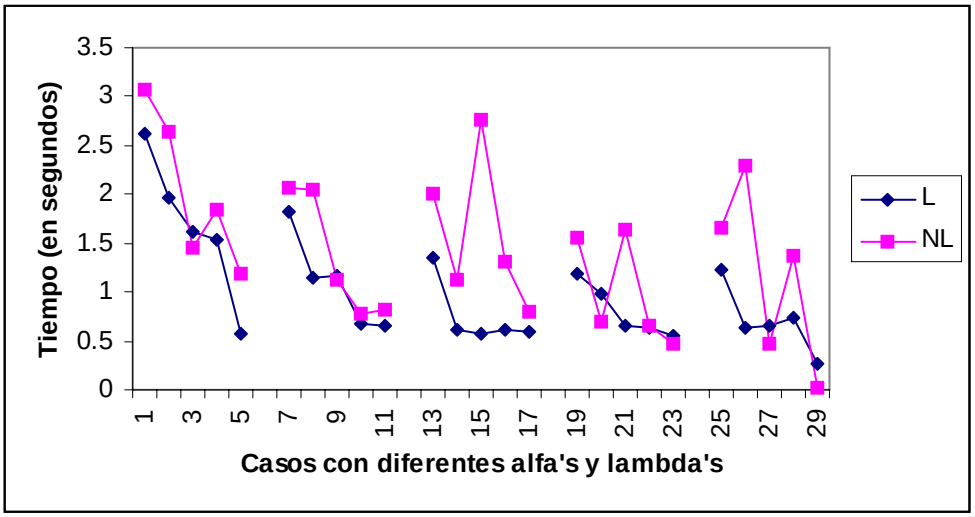

Figura 5.18 Comparación del tiempo en segundos en forma de árboles según Weka, mediante el método lineal y no lineal.

Para el método lineal en forma de reglas se obtiene un tiempo máximo de 3.64 segundos con α=0.1-λ=0.01 y un tiempo mínimo de 0.38 segundos con α=100-λ=0.9. Y para el método no lineal en forma de reglas el tiempo más largo es de 5.47 segundos con α=100-λ=0.01, mientras que el más pequeño es de 0.04 segundos con α=100-λ=0.9, como se observa en la figura 5.19.

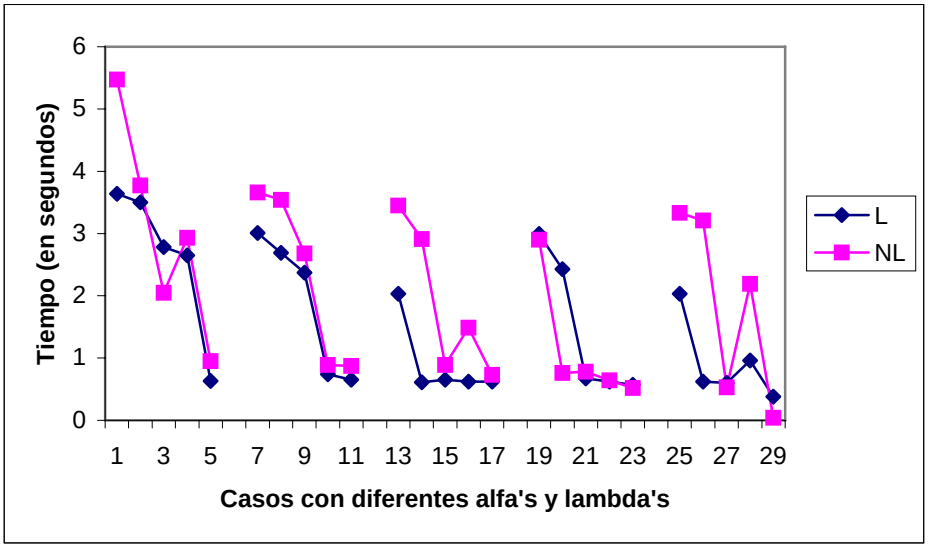

Figura 5.19 Comparación del tiempo en segundos en forma de reglas según Weka, mediante el método lineal y no lineal.

Haciendo una comparación del porcentaje de instancias correctamente clasificadas en árboles y reglas, se observa que hay mucha similitud como puede verse en las figuras 5.20 y 5.21. En la figura 5.20 en forma de árboles por el método lineal el porcentaje más alto es de 97.4729 con α=0.1-λ=0.01 y el más pequeño es de 79.3863% con ocho instancias que contiene la misma cantidad. En cambio para el método no lineal el porcentaje más alto es de 97.509 con α=0.1-λ=0.05 y α=1-λ=0.01, y el más pequeño es de 90.0722 con α=1-λ=0.5 y α=1-λ=0.09.

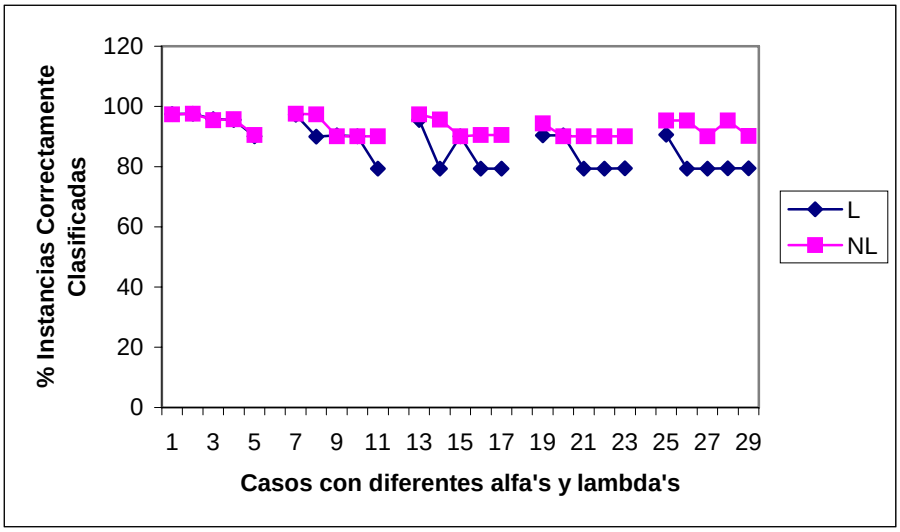

Figura 5.20 Comparación del porcentaje de instancias correctamente clasificadas en forma de árboles según Weka, mediante el método lineal y no lineal.

En la figura 5.21 se puede ver que para el método lineal el porcentaje más grande es de 97.4224 con α=100-λ=0.05 y el más pequeño es de 79.4224% con varias instancias. Y para el método no lineal 97.2924 es el porcentaje de instancias correctamente clasificadas más alto, con α=5-λ=0.01 y 90.0722 con α=1-λ=0.9 es el porcentaje que contiene menos instancias correctamente clasificadas.

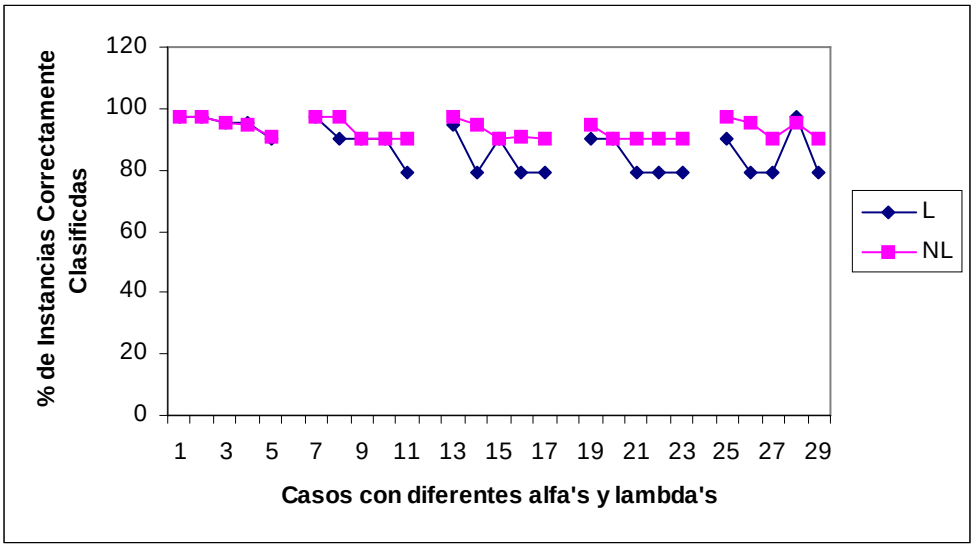

Figura 5.21 Comparación del porcentaje de instancias correctamente clasificadas en forma de reglas según Weka, mediante el método lineal y no lineal.

Al comparar los costos totales de los dos métodos (lineal y no lineal) en forma de árboles, se observa que el método lineal obtuvo perdidas en ocho instancias con –50080 y el costo total más alto es de 55410 con α=0.1-λ=0.05, la variación que hay entre estos dos valores depende de alfa y de lambda, ya que para el método no lineal el costo total más alto es de 55220 con α=0.1-λ=0.05 y el más bajo es de 25380 y son tres instancias las que contienen este valor, como se observa en la figura 5.22.

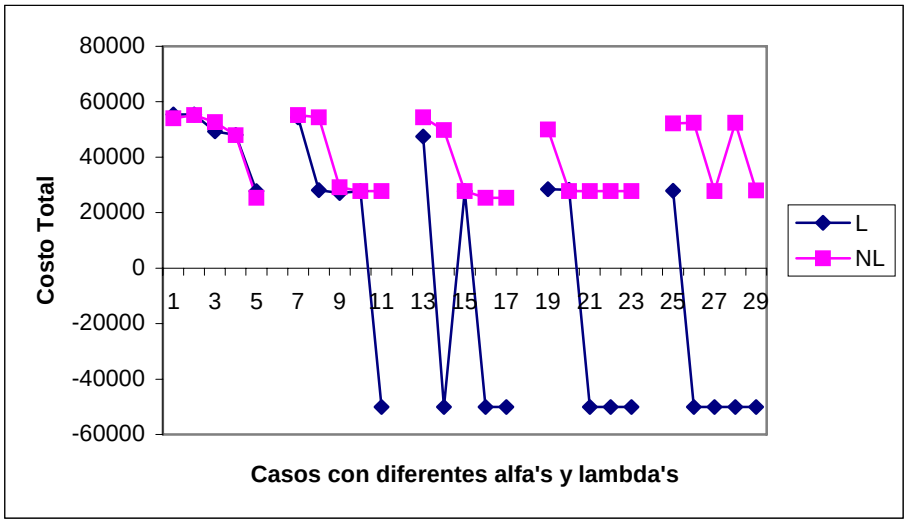

Figura 5.22 Comparación del costo total en forma de árboles según Weka, mediante los métodos lineal y no lineal.

En la figura 5.23 se observa el costo total en forma de reglas y se nota claramente que el método lineal obtiene los costos más bajos como es el caso de –50075, que lo tienen varias instancias y el más alto es de 54775 con α=1-λ=0.01; mientras que por el método no lineal obtiene un costo total máximo de 54405 con α=100-λ=0.01 y un costo mínimo de 25385 con α=5-λ=0.05.

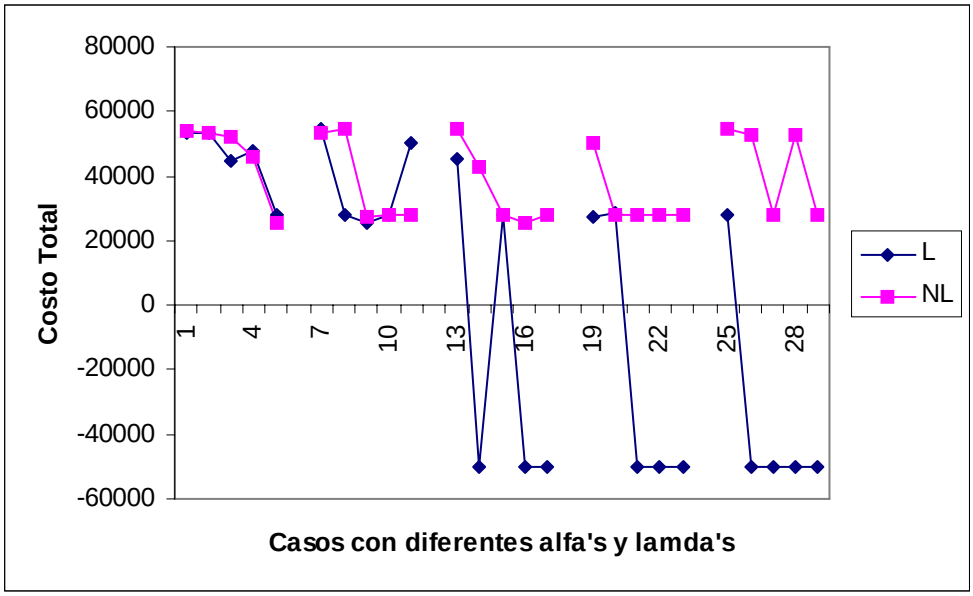

Figura 5.23 Comparación del costo total en forma de reglas según Weka, mediante los métodos lineal y no lineal.

5.4 CASO 3: ALGORITMO DE PUNTO INTERIOR DE KARMARKAR

En 1984 Karmarkar[1] desarrolló un nuevo algoritmo de tiempo polinomial que corta a través del interior del espacio de la solución. La efectividad del algoritmo parece estar en la solución de problemas extremadamente grandes de programación lineal.

El método símplex obtiene la solución óptima moviéndose por los bordes del espacio de la solución de un punto extremo al siguiente; en la práctica el método símplex ha servido para resolver problemas muy grandes, aunque el número de iteraciones necesarias para llegar a la solución óptima puede crecer exponencialmente, dependiendo del problema.

El algoritmo de Karmarkar comparte alguna de sus características. Por ejemplo, es un algoritmo iterativo que comienza por identificar una solución de prueba factible. Después continúa este proceso hasta llegar a una solución prueba (solución inicial arbitraria), que es (en esencia) óptima. Para el algoritmo de Karmarkar, las soluciones prueba son puntos interiores, es decir, puntos dentro de la frontera de la región factible. Ésta es una razón por la que se hace referencia al algoritmo de Karmarkar y sus variantes como algoritmos de Punto Interior.

[1] Narendra Karmarkar un joven matemático hindú de 28 años, egresado del Instituto de Tecnología de Bombay, del Instituto de Tecnología de California y de la Universidad de California en Berkeley, investigador de los laboratorios AT&T Bell; publicó el artículo “A New Polinomial-Time Algorithm for Linear Programming”.

Una manera de comparar los algoritmos de punto interior con el método símplex es examinar sus propiedades teóricas en cuanto a complejidad computacional. Karmarkar demostró que la versión original de su algoritmo es un algoritmo de tiempo polinomial, es decir, el tiempo que se requiere para resolver cualquier problema de programación lineal, se puede acotar por arriba por una función polinomial del tamaño del problema.

Se ha tratado de demostrar que el método símplex no posee esta propiedad y que es un algoritmo de tiempo exponencial. La diferencia en el desempeño en el peor de los casos es considerable, ya que los dos factores básicos que determinan el desempeño de un algoritmo para un problema real son el tiempo promedio de computadora por iteración y el número de iteraciones. Los algoritmos de punto interior son mucho más complicados que el método símplex. Se requiere mucho más cálculo en cada iteración para encontrar la solución prueba, por lo tanto, el tiempo de computadora por iteración para un algoritmo de punto interior es muchas veces mayor que para el método símplex; pero el número de iteraciones puede ser menor, dependiendo del tamaño del problema.

Una ventaja de los algoritmos de punto interior es que los problemas grandes no requieren muchas más iteraciones que los problemas pequeños. Por ejemplo un problema con 10,000 restricciones funcionales tal vez requiera menos de 100 iteraciones. Por el contrario, el método símplex puede requerir 20,000 iteraciones por lo que puede no terminar en un tiempo razonable. Por lo tanto, es muy probable que los algoritmos de punto interior sean más rápidos que el método símplex para problemas muy grandes.

La razón de esta gran diferencia en el número de iteraciones en problemas muy grandes es la diferencia en las trayectorias seguidas. En cada iteración el método símplex se mueve de la solución factible en un vértice actual a una solución factible en un vértice adyacente por una arista de la frontera de la región factible. Los problemas grandes tienen propiedades astronómicas de soluciones factibles en vértices.

La trayectoria de la solución factible en un vértice inicial hasta una solución óptima puede dar muchas vueltas por la frontera, tomando numerosos pasos a cada solución factible en un vértice adyacente siguiente. En contraste, un algoritmo de punto interior se salta todo esto caminando por el interior de la región factible hacia la solución óptima. Al agregar restricciones funcionales se agregan las aristas a la frontera de la región factible, pero esto tiene muy poco efecto sobre el número de soluciones prueba que se necesitan en la trayectoria a través del interior.

Los enfoques de punto interior tienen mucho mayor potencial que el método símplex para aprovechar el procesamiento en paralelo. Una desventaja grave del enfoque de punto interior es su limitada capacidad para realizar un análisis de sensibilidad. La idea principal de Karmarkar es empezar desde un punto interior representado por el centro de la símplex y después avanzar en dirección del gradiente proyectado para determinar un nuevo punto de la solución. El nuevo punto debe ser estrictamente un punto interior, significando que todas sus coordenadas deben ser positivas.

La idea puede resumirse así:

- Obtener del interior de la región factible, una solución factible que lleve a la solución óptima.

- Moverse en la dirección que mejore el valor de la función objetivo lo más rápido posible.
- Transformar la región factible para colocar la solución prueba actual cerca del centro, permitiendo así una mejora grande cuando se aplique el concepto anterior.

Ejemplo para el algoritmo de punto interior:

Maximizar $Z = x_1 + 2x_2$

Sujeto a: $x_1 + x_2 \leq 8$

$x_1 \geq 0,\ x_2 \geq 0$

En la figura 5.24 se observa la gráfica de este problema, en donde se puede observar que la solución óptima es (0,8) = (x_1, x_2) con Z=16.

El algoritmo comienza con una solución prueba inicial que (al igual que todas las soluciones prueba subsecuentes), se encuentra en el interior de la región factible. La solución no debe estar en ninguna de las tres rectas (x_1=0, x_2=0, x_1+x_2=8) que forman la frontera de esta región, porque esto llevaría a una operación matemática no definida de división entre cero en algún punto del algoritmo.

Se elige (x_1, x_2) = (2,2) de manera arbitraria como la solución prueba inicial (SP_o). Se observa en la figura 4.4 que la dirección del movimiento desde el punto (2,2) que aumenta el valor de Z a la mayor tasa posible es perpendicular a la recta de la función objetivo, $Z=16=x_1+2x_2$. Esta dirección se indica con flechas de (2,2) a (3,4) -> (SP_1). Sumando vectores se tiene:

$$(2,2) + (1,2) = (3,4)$$

donde el vector (1,2) es el gradiente de la función objetivo. Los componentes de (1,2) son justo los coeficientes de la función objetivo. El gradiente (1,2) define la dirección ideal para moverse.

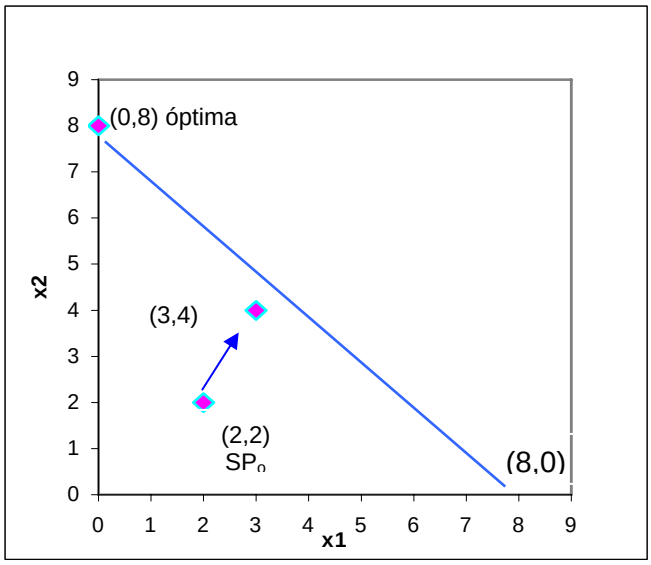

Figura 5.24 Ejemplo del algoritmo de punto interior.

El algoritmo opera sobre los problemas de programación lineal una vez que se han escrito en la forma aumentada. Sea x_3 la variable de holgura de la restricción funcional del ejemplo, esta forma es:

Maximizar $Z = x_1 + 2x_2$,
sujeta a $x_1 + x_2 + x_3 = 8$
y $x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0.$

En notación matricial, la forma aumentada se puede escribir en general como:

Maximizar $Z = c^T x$,
sujeta a $Ax = b$
y $x \geq 0$,

donde,

$$c=\begin{bmatrix}1\\2\\0\end{bmatrix},\quad x=\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix},\quad A=\begin{bmatrix}1 & 1 & 1\end{bmatrix},\quad b=\begin{bmatrix}8\end{bmatrix},\quad x \geq 0=\begin{bmatrix}0\\0\\0\end{bmatrix}$$

Se nota que $c^T = \begin{bmatrix}1 & 2 & 0\end{bmatrix}$ es ahora el gradiente de la función objetivo.

En la figura 5.25 se muestra la gráfica de la forma aumentada del ejemplo. Ahora la región factible consiste en el triángulo con vértices (8,0,0), (0,8,0) y (0,0,8). Los puntos en el interior de esta región factible son aquellos en donde $x_1 > 0$, $x_2 > 0$ y $x_3 > 0$. Cada una de estas tres condiciones $x_j > 0$ tiene el efecto de forzar a (x_1, x_2) a alejarse de una de las tres líneas que forman la frontera de la región factible.

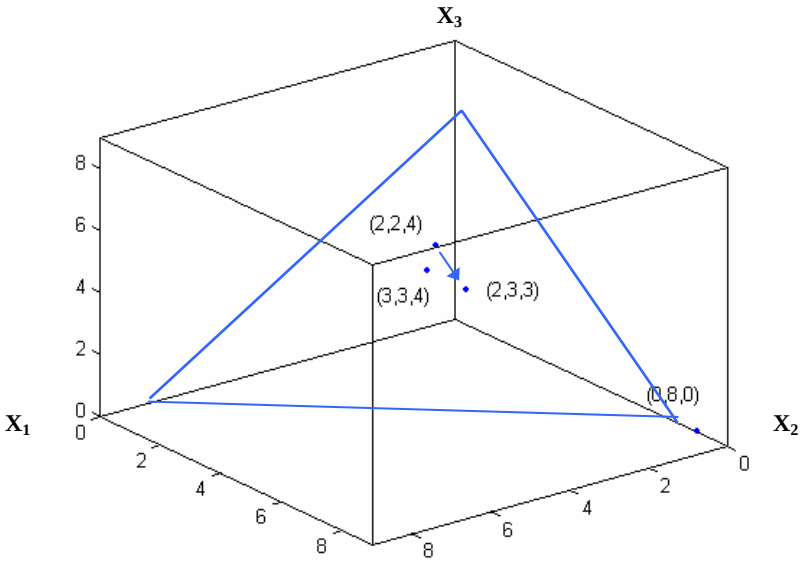

Figura 5.25 Ejemplo de la forma aumentada para el algoritmo de punto interior.

En la forma aumentada, la solución prueba inicial (SP_o) para el ejemplo es $(x_1, x_2, x_3) = (2,2,4)$. Si se agrega el gradiente (1,2,0) se obtiene

(2,2,4) + (1,2,0) = (3,4,4)

Sin embargo, el algoritmo no se puede mover de (2,2,4) hacia (3,4,4) porque (3,4,4) es no factible ya que se viola la restricción de que $x_1+x_2+x_3=8$ porque seria 3+4+4=11 y no 8. Sí $x_1=3$ y $x_2=4$, entonces $x_3=8-x_1-x_2$ -> $x_3=8-3-4$ -> $x_3=1$ si es factible. El punto (3,4,4) se encuentra en la cara de enfrente al ver el triángulo factible. Para que siga siendo factible el algoritmo visualiza el punto (3,4,4) al triángulo factible con el trozo de una recta perpendicular a este triángulo. Un vector de (0,0,0) a (1,1,1) es perpendicular a este triángulo, por lo que la recta perpendicular que pasa por (3,4,4) está dada por la ecuación

$(x_1,x_2,x_3) = (3,4,4) – \theta(1,1,1)$,

donde θ es un escalar. Como el triángulo satisface la ecuación $x_1+x_2+x_3 = 8$, esta recta perpendicular intercepta al triangulo en (2,3,3). Como

Si θ = 1 $\qquad\qquad$ (3,4,4) – (1,1,1) = (2,3,3)

(2,3,3) - (2,2,4) = (0,1,-1)

el gradiente proyectado de la función objetivo es (0,1,-1). Es este gradiente proyectado al que se define la dirección del movimiento para el algoritmo.
Continuando con el problema, se dispone de una fórmula para él cálculo directo del gradiente proyectado. Sea P la matriz de proyección

$P = I – A^T(AA^T)^{-1}A$,

donde I es la matriz identidad,

El gradiente proyectado en forma de columna es $c_p = Pc$

Así, por ejemplo,

P = I - A^T (A $A^T)^{-1}$ A

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} (3)^{-1} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ \frac{-1}{3} & \frac{2}{3} & \frac{-1}{3} \\ \frac{-1}{3} & \frac{-1}{3} & \frac{2}{3} \end{bmatrix},$$

$$c_p = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ \frac{-1}{3} & \frac{2}{3} & \frac{-1}{3} \\ \frac{-1}{3} & \frac{-1}{3} & \frac{2}{3} \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}.$$

El movimiento de (2,2,4) en la dirección del gradiente proyectado (0,1,-1) implica un incremento de α a partir de cero en la fórmula

$$x = SP_o + 4\alpha c_p \quad x = \begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix} + 4\alpha c_p = \begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix} + 4\alpha \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix},$$

donde el coeficiente 4 se usa sólo para dar una cota superior de 1 a α a fin de mantener la factibilidad (toda $x_j \geq 0$). Se observa que al aumentar α a α=1 el resultado es que x_3 disminuye a $x_3=4+4(1)(-1)=0$, donde α>1 lleva a $x_3<0$. Así, α mide la fracción usada de la distancia que se puede mover antes de dejar la región factible.

El incremento en Z es proporcional al de α, un valor cercano a 1 es bastante bueno para dar un paso relativamente grande hacia la optimalidad en la iteración actual. Sin embargo, el problema de un valor muy cercano a 1 es que la siguiente solución prueba puede quedar amontonada con una frontera de restricción, haciendo difícil realizar mejoras grandes en las iteraciones subsecuentes.

Karmarkar estableció para su algoritmo que un valor de α=0.25 debe ser "confiable". En la práctica, a veces se usan valores mucho mayores (como α=0.9); para desarrollar este ejemplo se elige α=0.5.

Sólo falta un paso para completar la descripción del algoritmo, así como un esquema especial para transformar la región factible de manera que la solución prueba actual se acerque al centro. Otro beneficio importante del esquema de centrado es el cambio constante de la dirección del gradiente proyectado para que apunte más de cerca hacia una solución óptima conforme el algoritmo converge a esta solución.

Esta idea básica del esquema de centrado es directa, sencillamente se cambia la escala para cada variable de manera que la solución prueba quede equidistante de las fronteras de restricción en el nuevo sistema de coordenadas. En el ejemplo se tienen tres fronteras de restricción, cada una corresponde a un valor de cero para una de las tres variables del problema en la forma aumentada, es decir, $x_1=0$, $x_2=0$ y $x_3=0$.

En la figura 5.26 se observa que estas tres restricciones cortan el plano Ax=b ($x_1+x_2+x_3=8$) para formar la frontera de la región factible. La solución prueba inicial es (x_1,x_2,x_3)=(2,2,4), y se encuentra dos unidades alejada de las restricciones $x_1=0$ y $x_2=0$ y alejada cuatro unidades de $x_3=0$, si se usan las unidades de las variables respectivas. Sin embargo, cualesquiera que sean estas unidades, son bastante arbitrarias y se pueden cambiar como se desee sin variar el problema. Por lo tanto, se da la siguiente escala a las variables:

$$\tilde{x}_1 = \frac{x_1}{2}, \quad \tilde{x}_2 = \frac{x_2}{2}, \quad \tilde{x}_3 = \frac{x_3}{4}$$

Para hacer que la solución prueba actual (x_1,x_2,x_3)=(2,2,4) se transforme en

$$\left(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3 \right) = (1, 1, 1)$$

Substituyendo $2\tilde{x}_1$ por x_1, $2\tilde{x}_2$ por x_2 y $4\tilde{x}_3$ por x_3), el problema se convierte en:

Maximizar $Z = 2\tilde{x}_1 + 4\tilde{x}_2$,

sujeta a $2\tilde{x}_1 + 2\tilde{x}_2 + 4\tilde{x}_3 = 8$

y $\tilde{x}_1 \geq 0, \tilde{x}_2 \geq 0, \tilde{x}_3 \geq 0$

como se muestra en la figura 5.26.

Se observa que la solución prueba (1,1,1) en la figura 5.26 equidista de las fronteras de restricción $\tilde{x}_1 = 0, \tilde{x}_2 = 0$ y $\tilde{x}_3 = 0$. En cada iteración subsiguiente se da una nueva escala al problema para que la solución prueba sea siempre (1,1,1) en las coordenadas actuales.

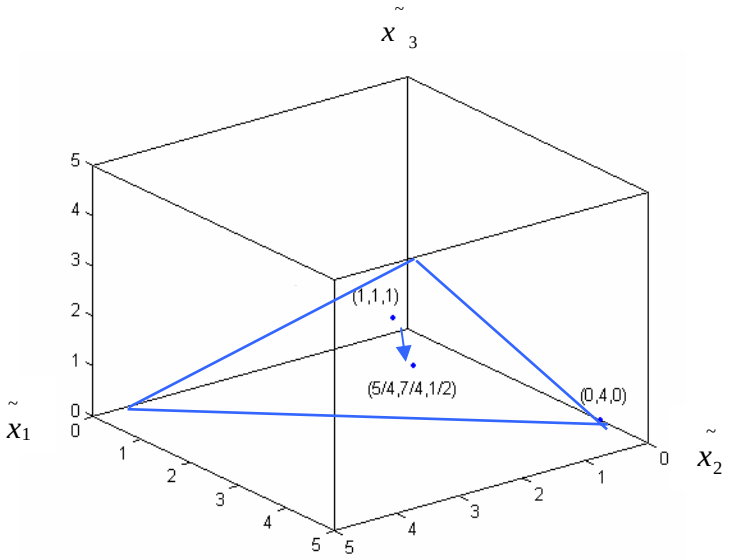

Figura 5.26 Ejemplo con la nueva escala para la iteración I.

El ejemplo presentado es muy pequeño, por lo tanto, el algoritmo requiere un número relativamente grande de cálculos y después de muchas iteraciones obtiene solo una aproximación de la solución óptima. En cambio, el método símplex requiere solo una iteración rápida. Esto no es razón para menospreciar la eficiencia del algoritmo de punto interior, ya que este algoritmo está diseñado para manejar problemas grandes que tienen miles de iteraciones. En este tipo de problemas al obtener una solución en el interior de la región factible, el algoritmo de punto interior tiende a requerir un número mucho menor de iteraciones.

5.4.1 RESULTADOS

Después poner en practica el método de punto interior, se obtuvieron los siguientes resultados; para obtenerlos no fue necesario modificar el modelo hecho en Gams, ya que el problema estaba linealizado, sólo se cambio el tipo de problema, es decir, en lugar de poner la instrucción LP, se cambio por MIP.

	MOSEK LINEAL PUNTO INTERIOR				
ALFA	**LAMBDA**	**ATS-GAMS**	**VAL.FUN.OBJ.**	**TIEMPO(SEG)**	**ITERACIONES**
0.1	0.01	18	0.672	4	3
0.1	0.05	16	0.6557	4	0
0.1	0.2	11	0.56	4	4
0.1	0.5	11	0.3625	3	0
0.1	0.9	7	0.0992	5	0
1	0.01	15	0.688	5	0
1	0.05	9	0.6737	4	0
1	0.2	9	0.62	10	0
1	0.5	7	0.5125	5	0
1	0.9	5	0.1733	5	0
5	0.01	11	0.7014	5	0
5	0.05	9	0.7404	5	0
5	0.2	6	0.8867	4	0
5	0.5	5	0.8627	6	0
5	0.9	5	0.178	4	0
20	0.01	10	0.7514	4	0
20	0.05	7	0.9904	4	0
20	0.2	5	1.3803	5	0
20	0.5	5	0.8755	4	0
20	0.9	4	0.1809	4	0
100	0.01	7	1.0181	5	0
100	0.05	5	1.6396	4	0
100	0.2	5	1.4062	5	0
100	0.5	5	0.9016	5	0
100	0.9	2	0.1828	4	0

Tabla 5.8 Resultados arrojados por el Método de Punto Interior, con Gams y Neos.

MOSEK LINEAL PUNTO INTERIOR						ÁRBOLES
ALFA	LAMBDA	ATS-GAMS	ARBOL	TIEMPO(SEG.)	% INST.CC	COSTO TOTAL
0.1	0.01	18	39	2.93	97.4368	55210
0.1	0.05	16	39	1.89	97.509	55410
0.1	0.2	11	15	1.36	95.3791	52645
0.1	0.5	11	29	1.38	95.704	47940
0.1	0.9	7	27	0.67	90.6498	25390
1	0.01	15	39	1.88	97.4007	55015
1	0.05	9	15	1.08	95.343	52450
1	0.2	9	9	1.02	90.1444	29120
1	0.5	7	27	0.73	90.5776	25380
1	0.9	5	1	0.61	79.3863	-50080
5	0.01	11	23	1.66	95.2347	52625
5	0.05	9	9	1.05	90.1444	29120
5	0.2	6	3	0.54	90.1444	27790
5	0.5	5	1	1.36	79.3863	-50080
5	0.9	5	1	0.63	79.3863	-50080
20	0.01	10	9	1.17	90.5054	28410
20	0.05	7	27	0.73	90.5776	25380
20	0.2	5	1	0.61	79.3863	-50080
20	0.5	5	1	0.61	79.3863	-50080
20	0.9	4	1	0.61	79.4585	-49880
100	0.01	7	27	0.74	90.5776	25380
100	0.05	5	1	0.66	79.3863	-50080
100	0.2	5	1	0.84	79.3863	-50080
100	0.5	5	1	0.77	79.4224	-50075
100	0.9	2	1	0.31	79.4224	-50075

Tabla 5.9 Resultados arrojados por el Método de Punto Interior con Weka y con el clasificador j48.J48, en forma de árbol.

MOSEK LINEAL PUNTO INTERIOR					REGLAS	
ALFA	LAMBDA	ATS-GAMS	REGLAS	TIEMPO(SEG.)	% INST. CC	COS.TOT
0.1	0.01	18	21	5.34	97.148	54600
0.1	0.05	16	22	3.5	97.0397	53635
0.1	0.2	11	8	2.02	95.4513	52085
0.1	0.5	11	14	2.23	95.0181	46135
0.1	0.9	7	10	0.94	90.5776	25190
1	0.01	15	19	3.16	97.0758	55160
1	0.05	9	8	1.61	95.5596	52290
1	0.2	9	11	2.65	90.361	27440
1	0.5	7	7	0.94	90.6859	25585

1	0.9	5	1	0.63	79.4224	-50075
5	0.01	11	9	1.97	95.6679	52875
5	0.05	9	11	2.89	90.361	27440
5	0.2	6	2	0.63	90.1444	27790
5	0.5	5	1	0.67	79.4224	-50075
5	0.9	5	1	0.64	79.4224	-50075
20	0.01	10	11	2.94	90.361	27060
20	0.05	7	7	0.95	90.6859	25585
20	0.2	5	1	0.62	79.4224	-50075
20	0.5	5	1	0.67	79.4224	-50075
20	0.9	4	6	0.9	79.4585	-49880
100	0.01	7	7	0.89	90.6859	25585
100	0.05	5	1	0.6	79.4224	-50075
100	0.2	5	1	0.8	79.4224	-50075
100	0.5	5	4	1.11	79.4224	-50075
100	0.9	2	2	0.38	79.4224	-50075

Tabla 5.10 Resultados arrojados por el Método de Punto Interior con Weka y con el clasificador PART, en forma de reglas.

5.4.2 ANÁLISIS DE RESULTADOS

En la tabla 5.8 se mejoro mucho el tiempo, ya que en la tabla 5.5 era de 8.12 segundos en promedio y se mejoro a 4.68 segundos. En las iteraciones se vio un cambio muy notorio ya que en promedio por el método lineal eran de 2063.64 y por el Método de Punto Interior son en promedio 0.28 iteraciones. La función objetivo que es la que se quiere minimizar no se ve afectada y eso va a depender de los valores que tome alfa y lambda ya que si se comparan las dos tablas se observa que los resultados de la tabla 5.5 contra los de la tabla 5.8, en la columna del valor de la función objetivo no cambia mucho, cambia por centésimas, pero al redondearlo sigue siendo el mismo valor para los dos tablas que en promedio tienen un valor de 0.71.

Por el método de Punto Interior el valor máximo de la función objetivo es de 1.6396 con un $\alpha=100$ y $\lambda=0.05$ y el valor mínimo es de 0.1733 con $\alpha=1$ y $\lambda=0.9$. El tiempo máximo es de 10 segundos con $\alpha=1$ y $\lambda=0.2$ y el tiempo mínimo es de 4 segundos, tomando este valor varias instancias, como se observa en la figura 5.27, donde también se puede ver que el número máximo de iteraciones es de 4 con $\alpha=0.1$ y $\lambda=0.02$, mientras que el mínimo es de cero iteraciones. En cuanto atributos se refiere, en donde se seleccionaron más fue en $\alpha=0.1$ y $\lambda=0.01$ con 18 atributos, en tanto que en $\alpha=100$ y $\lambda=0.9$ tan sólo se seleccionaron 2 atributos.

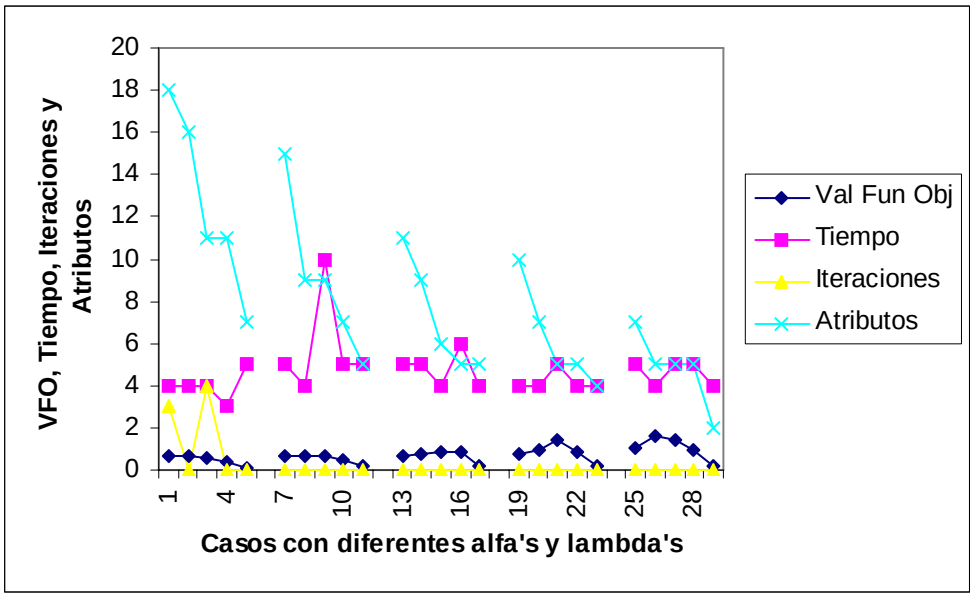

Figura 5.27 Valor de la Función Objetivo, Tiempo en Segundos, Iteraciones y Atributos, según Gams y Neos.

En la figura 5.28 se hace la comparación entre árboles y reglas, y se puede observar a simple vista que el número de árboles es mayor que el número de reglas, ya que el número máximo de árboles es de 39 siendo tres instancias las que contienen el mismo valor, las cuales son: α=0.1-λ=0.01, α=0.1-λ=0.05 y α=1-λ=0.01; el valor mínimo de árboles es de 1 como se puede observar en la figura. En cuanto a reglas se refiere el valor máximo es de 21 con α=0.1 y λ=0.01, teniendo varias instancias el tamaño mínimo de reglas que es 1.

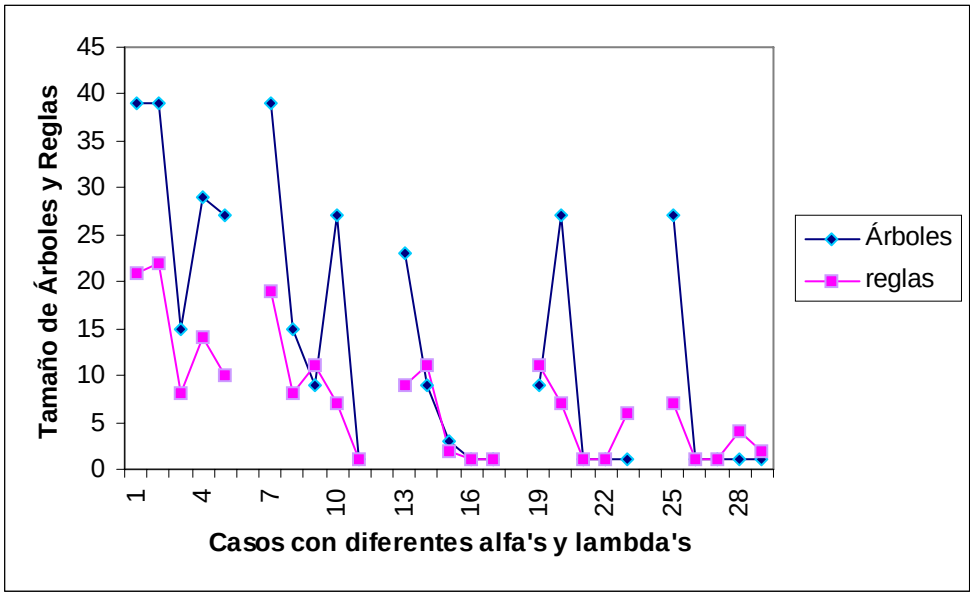

Figura 5.28 Tamaño de árboles y reglas, según Weka.

El tiempo máximo que tardo el programa Weka con el software j48.J48 en forma de árboles de 2.93 segundos con α=0.1 y λ=0.01 y el mínimo es 0.31 segundos con α=100 y λ=0.09. Mientras que para reglas el máximo es de 5.34 segundos con α=0.1 y λ=0.01 y el tiempo mínimo es de 0.38 segundos con α=100 y λ=0.09, obsérvese la figura 5.29 en donde se muestra el tiempo.

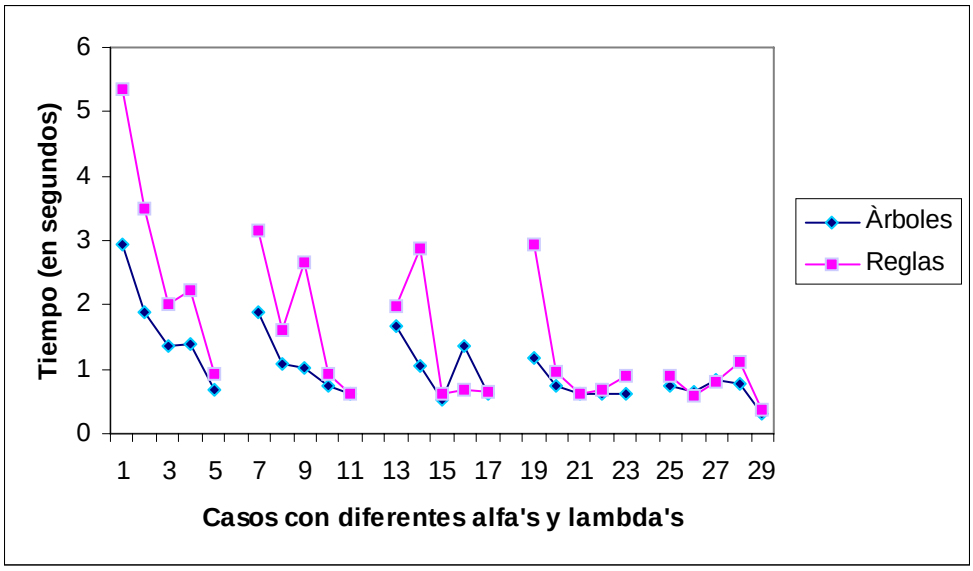

Figura 5.29 Tiempo en segundos de árboles y reglas según Weka.

Para analizar el porcentaje de instancias correctamente clasificadas, se observa la figura 5.30 en donde se puede observar que el máximo porcentaje corresponde a árboles y es de 97.509 con α=0.1 y λ=0.05, seguido de 97.148 que es el máximo valor para reglas. Existen varios valores mínimos con 79.3863% de instancias correctamente clasificadas, en forma de árboles y 79.4224% en forma de reglas.

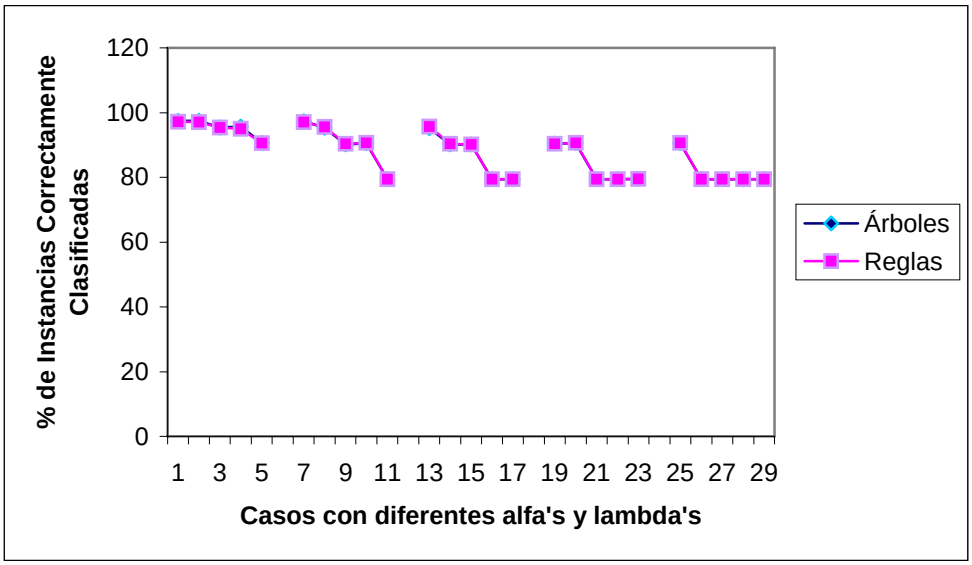

Figura 5.30 Porcentaje de Instancias Correctamente Clasificadas de árboles y reglas, según Weka.

En la figura 5.31 se observa que los costos totales más altos son 55410 para árboles con α=0.1 y λ=0.05, y 54600 con α=0.1 y λ=0.01 para reglas. Los costos mínimos son –50080 para árboles y –50075 para reglas, como se puede observar hay mucha diferencia entre los costos máximos y los mínimos, eso va a depender de las alfa's y lambda's que se escojan para la selección de atributos. También se observa que los costos de árboles y reglas son casi los mismos.

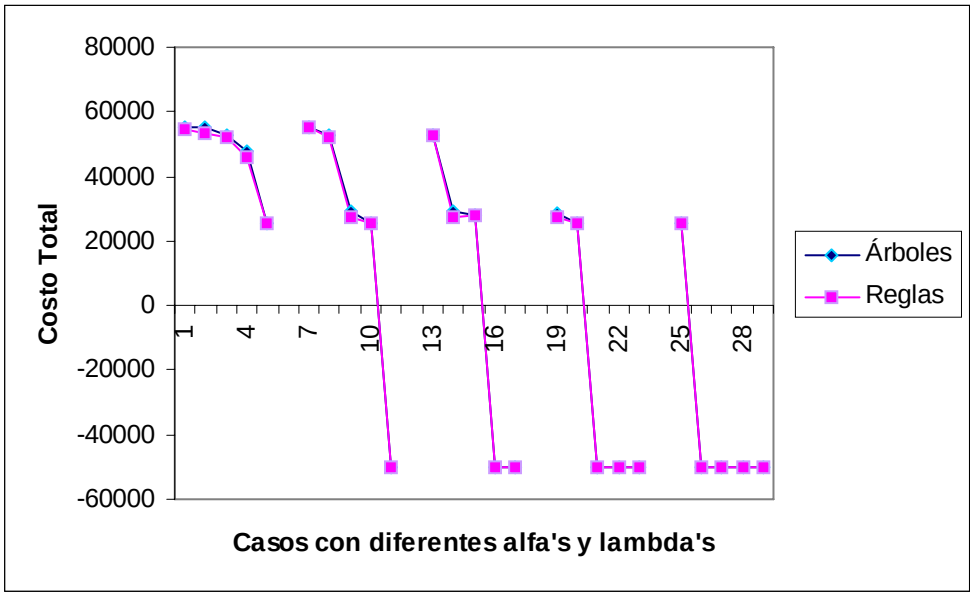

Figura 5.31 Costo Total de árboles y reglas según Weka.

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Después de realizar los experimentos necesarios en Clasificación de Instancias y Selección de Atributos, basados en Optimización (minimización cóncava FSV). Se analizaron y encontraron soluciones que en algunos casos resultaron más eficientes y útiles que las conocidas en la actualidad para clasificar instancias y seleccionar atributos.

Se obtuvieron resultados interesantes en lo propuesto por esta tesis, ya que al linealizar el modelo y hacer experimentos por el Método de Punto Interior, se redujo el tiempo de procesamiento hasta en un 99% y se minimizaron a cierta medida los atributos, que es lo que se pretendía primordialmente.

Actualizando el modelo FSV para sus nuevas aplicaciones y poder usarse para futuras investigaciones el modelo quedo de la siguiente manera:

$$\min_{w,\gamma,y,z,v} \; (1-\lambda)\left(\frac{e^T y}{m}+\frac{e^T z}{k}\right)+\lambda e^T\left(e-\left(1+\left(-\alpha v\right)\right)\right)$$

$$\text{sujeto a :} \quad \begin{aligned} -Aw+e\gamma+e &\le y,\\ Bw-e\gamma+e &\le z,\\ y\ge 0, z &\ge 0,\\ -v\le w &\le v. \end{aligned}$$

La tesis aportó resultados experimentales que ayudan a seguir líneas de investigación en un futuro inmediato. El modelo utilizado puede servir a otros problemas en diferentes campos y ramas de la investigación, pero sobre todo para la selección de atributos.

Los resultados por cada extensión son:

- En Linealización: Se redujo el tiempo de procesamiento y a la vez el número de atributos, la función objetivo no se ve afectada, asi que se considera como opción para sustituir el Método No Lineal.

- En Linealización con Peso: Se cumple el objetivo que es clasificar las instancias de acuerdo a un peso o valor, es decir darle más importancia a ciertas instancias que a otras.

- En Método de Punto Interior: Se redujo el tiempo en un 99%, comparado con el Método No Lineal. Tambien hubó gran reducción de atributos, y la función objetivo no se ve afectada, así que este método es una buena opción, siempre y cuando el problema sea de gran tamaño.

Se encontraron soluciones más eficientes y eficaces con cierto parámetros, que las conocidas hasta ahora para clasificar y seleccionar atributos, dentro de las cuales están: tiempo de procesamiento, calidad predictiva y reducción de atributos. De esta experimentación resultó la reducción en gran medida del tiempo de procesamiento, ya que las iteraciones del proceso disminuyeron en un 99.9%, esto sin afectar la selección de atributos. Así mismo, se mejoro el porcentaje de las instancias correctamente clasificadas.

Además, el problema con el que se trabajo en esta tesis es de 2 770 restricciones y cabe mencionar que es un problema real con 3.5 veces más grande que el problema que se usó para FSV.

6.2 RECOMENDACIONES

Utilizar el modelo linealizado por el Método de Punto Interior, ya que así se reduce el tiempo de procesamiento. Probar los diferentes α´s y λ`s dependiendo del modelo, ya que los resultados pueden variar, dependiendo del problema.

La participación de los expertos del dominio resulta crucial para la definición de las variables importantes relacionadas con el uso ilícito, la definición del tipo de usuarios y áreas a procesar, así como en la interpretación y validación de resultados.

6.3 APORTACIONES

Las aportaciones al modelo FSV son:

- Se linealizó el modelo FSV
- Se redujo el tiempo de procesamiento
- Se hicieron nuevos algoritmos
- Linealizado el problema se introdujo al software Gams por medio de Programación Lineal
- Ya linealizado el problema, se hizo por el Método de Punto Interior.

6.4 TRABAJO FUTURO

- El modelo FSV puede ser mejorado, dependiendo del modelo que se quiera minimizar, se deben encontrar α`s y λ`s óptimos, para cada problema.
- Se pueden probar otros paquetes de optimización, así como de clasificación.
- Se podrían hacer experimentaciones, probando otros métodos.
- Realizar más experimentos con otras bases datos de diferentes magnitudes, para ver la efectividad del modelo.
- Aplicar "Sampling", reducir instancias en lugar de atributos.

FUENTES DE INFORMACIÓN

[Aimms, 1999] Bisschop, J., and Roelofs, M., AIMMS, The User´s Guide, Paragon Decision Technology, 1999. http://www.aimms.com

[Ampl] http://www.ampl.com

[Bazaraa, 1943] Mokhtar S. Bazaraa "Nonlinear Programming Theory and algorithms". C.M. Shetty. School of industrial and Sistems Engineering. Georgia Institute of technology. 1943.

[Bennet y Mangasarian, 1992] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. Optimization Methods and Software, 1992.

[Brachman y Anand, 1996] R. J. Brachman y T. Anand. "The Process of Knowledge Discovery in Databases". Eds., AAAI Press, Menlo Park, California, 1996.

[Bradley, Mangasarian y Rosen, 1997] P. S. Bradley, O. L. Mangasarian, and J. B. Rosen. "Parsimonious least norm approximation". Technical Report 97-03, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, March 1997. Computational Optimization and Applications, to appear. ftp://ftp.cs.wisc.edu/mathprog/tech-reports/97-03.ps.Z.

[Bradley, Mangasarian y Street, 1998] P.S. Bradley, O. L. Mangasarian, and W. N. Street. Feature selection via mathematical programming. INFORMS Journal on Computing, 1998. To appear. ftp://ftp.cs.wisc.edu/mathprog/tech-reports/95-21.ps.Z.

[Bradley-Mangasarian, 1998] P. S. Bradley y O. L. Mangasarian. Feature Selection via Concave Minimization and Support Vector Machines. Febrero 1998. http://citeseer.ist.psu.edu/bradley98feature.html

[Burden, Richard, Douglas, 1985] Burden, Richard L., J. Douglas Faires, "Análisis Numérico". Grupo Editorial Iberoamérica, Primera Edición, 1985.

[Castillo, et al, 2002] Enrique Castillo, Antonio J. Conejo, Pablo Pedregal, Ricardo Garcia y Natalia Alguacil. "Formulación y Resolución de modelos de Programación Matematica en Ingenieria y Ciencia". Febrero 2002.

[Chattopandhyay, 1999] Chattopadhyay, D., "Aplication of General Algebraic Modeling System to Power System Optimization," IEEE Trans. Power System 14(1), Febrero. 1999.

[Decker y Focardi, 1995] K. M. Decker y S. Focardi. "Technology Overview: A Report on Data Mining". En CSCS TR-95-02 Technical Report, Swiss Scientific Computing Center, Mayo 1995. (En URL ftp://ftp.cscs.ch/pub/CSCS/techreports/1995/CSCS-TR-95-02.ps.Z).

[Fayyad, 1996] U. M. Fayyad, G. Piatetsky-Shapiro y P. Smyth. "From Data Mining to Knowledge Discovery". Eds., AAAI Press, Menlo Park, California, 1996.

[Ferrer, 1996] Ferrer, J., "Investigators use AMR to detect millions in energy theft", AMRA News, vol. 9, núm. 12, diciembre de 1996.

[Frawley, 1991] W. J. Frawley, G. Piatetski-Shapiro y C. J. Matheus. "Knowledge Discovery in Databases: An Overview". AAAI-MIT Press, Menlo Park, California, 1991.

[Gams] http://www.gams.com

[Gsi] www.gsi.dit.upm.es/~anto/**tesis**/html/stateart.html

[Hahn, 1996] Hahn, A. E., "Electronic submeters", Electrical Contractor, reprinted by E-MON, junio de 1996.

[Hiller, Lieberman, 2004]Frederick S. Hiller y Gerald J. Lieberman. "Investigación de operaciones". Séptima edición. Mc Graw Hill, Mayo 2004.

[Hodges, 1996] Hodges, S., "Increase AMR benefits with outage and power-quality reporting", Utility Automation, vol. 1, núm. 5, septiembre/octubre de 1996.

[Hornbeck, Robert, 1975] Hornbeck, Robert W., "Numerical Methods" Quantum Publishers, Inc. 1975.

[Huber, 1981] P. J. Huber. Robust Statistics. John Wiley, New Cork, 1981.

[IIE] www.iie.org.mx/publica/bolja97/tenja97.htm

[Inv.Operaciones]http://www.investigacion-operaciones.com/Presentacion_modelos/INTRODUCCION%20INV.%20OPER.ppt

[Inv. Operaciones-2]http://www.investigacion-operaciones.com/operaciones.htm

[Itson] http://www.itson.mx/dii/elagarda/apagina2001/PM/uno.html#introduccion

[IURPA] IURPA, Energy theft related news group posts, julio de 1997.
[Java] http://java.sun.com/j2se/

[Mangasarian, 1996] O. L. Mangasarian. Machine learning via polyhedral concave minimization. In H. Fischer, B. Riedmueller, and S. Schaeffler, editors, Applied Mathematics and Parallel Computing – Festschrift for Klaus Ritter. Physica-Verlag A Springer-Verlag Company, Heidelberg, 1996. ftp://ftp.cs.wisc.edu/mathprog/tech-reports/95-20.ps.Z.

[Mangasarian, Street y Wolberg, 1995] O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Operations Research, Julio-Agosto 1995.

[Mat] http://www.mat.uson.mx/eduardo/calculo2/soltaylor/soltaylorHTML/taylor.htm

[Michalski, 1987] R. S. Michalski. "Concept Learning". Encyclopedia of Artificial Intelligence, Stuart C. Shapiro, Ed. John Wiley & Sons, 1987.

[Neos, 2003] NEOS (Network-Enabled Optimization System):http://www-fp.mcs.anl.gov/otc/Guide/CaseStudies/ , 2003.

[Neos] www-**neos**.mcs.anl.gov/**neos**/

[Reason, 1996] Reason, J., "Para combatir las pérdidas, averigua a dónde va cada Kwh", Energía Eléctrica, junio de 1996.

[Richardson, 1994] Richardson, W., Energy division and theft of service in a changing environment, 7th Annual AMRA Symposium, septiembre de 1994.

[Taha, 1988] Hamdy A. Taha "Investigación de operaciones una introducción". Sexta edición Prentice Hall, 1988.

[Vanderplaats, 1984] Garret N. Vanderplaats. "Numerical Optimization Techniques for Engineering Design With Aplications". Mc Graw-Hill Book Company, 1984.

[Vidrio, Gómez y Castán, 2004] Gilberto Vidrio L., J. Martín Gómez L. y Roberto Castán L. "Sistema de Medición y Detección de Pérdidas de Energía Eléctrica (SIM-IV)". Boletín IIE. Julio-Septiembre de 2004.

[Waikato] http://www.cs.waikato.ac.nz/~ml/weka/

[Weka] http://www.weka.fr/home.php
www.uv.es/~sala/**gams**/14.PDF

MEDIOS ELECTRÓNICOS COMPLEMENTARIOS

http://delta.cs.cinvestav.mx/~mcintosh/comun/an1/node3.html
http://200.13.98.241/~antonio/cursos/control/notas/dynamics/lineal.pdf
http://200.1398.241~antonio/cursos/control/notas/dynamies/lineal.pdf
http://delta.cs.cinvestav.mx/~mcintosh/comun/an1/node2.html
http://http://members.tripod.com/hernangabriel/investigacionoperativa.html
http://www.chartwellyorke.com/derive.html
http://www.chartwellyorke.com/derive.html
http://www.economics.ltsn.ac.uk/software/maths.htm#GAMS
http://www.gams.com/docs/contributed/modelado_en_gams.pdf
http://www.ifors.org/ Federación Internacional de Sociedades de Investigación de Operaciones.
http://www.informs.org/ Sociedad Americana de Investigación Operativa.
http://www.investigacion-operaciones.com/
http://www.mat.ucm.es/deptos/maq/docencia/calcnum/trans-caln-foils.pdf
http://www.mat.uson.mx/eduardo/calculo2/soltaylor/soltaylorHTML/taylor.htm
http://www.orie.cornell.edu/ Departamento de Investigación Operativa de la Universidad de Cornell en Nueva York.
http://www.uv.es/~sala/CUADERN.pdf
http://www.uv.es/~sala/gams/14.PDF--->MANUAL GAMS
http://www.uv.es/~sala/gams/ascplex.pdf
http://www.uv.es/~sala/gams/Lineal01.pdf
http://www.worms.ms.unimelb.edu.au/ Información genérica sobre Investigación Operativa.
http://www-neos.mcs.anl.gov/neos/
http://www-unix.mcs.anl.gov/otc/Guide/faq/nonlinear-programming-faq.html
www.cnice.mecd.es/Descartes/Analisis/
www.itson.mx/dii/elagarda/apagina2001/PM/uno.html
http://www.uv.es/~sala/gams/entera.PDF
www.mat.uson.mx/eduardo/calculo2/ soltaylor/soltaylorHTML/taylor.htm - 45k
www.matematicas.unal.edu.co/~hmora/kar_m_c.pdf

GLOSARIO

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

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

Sistemas de ecuaciones. Cuando no existe una función objetivo como tal. Únicamente interesa encontrar una solución factible a un problema con un conjunto de restricciones.

Solución factible. Un punto X=(X1,X2,...Xn) que satisface todas las restricciones se denomina solución factible. El conjunto de todas esas soluciones es la *región de factibilidad*.

Solución óptima. Un punto factible $\bar{x}$ tal que f(x) ≥ f($\bar{x}$) para cualquier otro punto factible x se denomina una solución óptima del problema.

Variable de holgura. Para las restricciones del tipo ≤ (en este caso), el lado derecho por lo común representa el límite sobre la disponibilidad de un recurso y el lado izquierdo representa el empleo que hacen de ese recurso limitado las diferentes actividades (variables) del modelo. Una holgura representa la cantidad en la cuál la cantidad disponible del recurso excede al empleo que le dan las actividades.

Variable de superávit. Las restricciones del tipo ≥ por lo común determina requerimientos mínimos de especificaciones. En este caso, un superávit representa el exceso mínimo del lado izquierdo, sobre el requerimiento mínimo.

Variable no restringida. En algunos problemas la naturaleza de las variables requiere que asuman valores no negativos. Hay situaciones en las cuales una variable puede asumir cualquier valor real. El hecho de que X sea no restringida permite que la variable no desempeñe los papeles tanto de una holgura como de un superávit, según se desee.

Variables. Representan las decisiones que se pueden tomar para afectar el valor de la función objetivo.

SIGLARIO

MD: Mineria de Datos
IO: Investigación de Operaciones
FSV: Feature Selection Via Concave Minimization
KDD: Knowledge Discovery in Databases (Descubrimiento del Conocimiento en Bases de Datos).
MIP: Método de Punto Interior
PL: Programación Lineal
PMIP: Programación del Método de Punto Interior
PNL: Programación No Lineal
PPL: Problema de Programación Lineal
PPNL: Problema de Programación No Lineal

APÉNDICES

APÉNDICE A: Formulación del Problema No Lineal

Se muestra la entrada de los datos del Problema No Lineal (PNL) al software GAMS y la salida de los mismos por medio del programa NEOS.

```
$ONTEXT

 RESOLVER UN PROBLEMA DE PROGRAMACIÓN NO LINEAL PARA LA :
   -CLASIFICACIÓN DE ILICITOS (USANDO INDICES)
   -SELECCIÓN DE ATRIBUTOS

 EL PROBLEMA ES:
 MINIMIZAR F(W,R,Y,Z,V)=(1-LAMBDA)((e**TY/M)+(e**TZ/K))+(LAMBDA*e**T)(e-E**-(ALFA*V))
 SUJETO A:        -AW+eR+e<=Y,
                   BW-eR+e<=Z,
                   Y>=0, Z>=0
                  -V<=W<=V

 EL PROBLEMA USA DATOS DE UNA BASE DE DATOS DE ILICITOS LA CUAL CONTIENE 24 ATRIBUTOS,
2,200 REGISTROS DE TIPO 1 Y 570 REGISTROS DE TIPO 9. CON:
LAMBDA=0.01
ALFA  =0.1

$OFFTEXT

SET
I  / 1*2200/
J  / 1*24/
L  / 1*570/;

TABLE
A(I,J); (Aquí se introduce una matriz de datos de 2200 registros x 24 atributos)

TABLE
B(L,J); (Aquí se introduce una matriz de datos de 570 registros x 24 atributos)

VARIABLES
W(J), R, Y(I), Z(L), V(J), F;

POSITIVE VARIABLES
Y, Z;

SCALAR  M /2200/, K /570/, LAMBDA /0.01/, ALFA /0.1/;

EQUATIONS
FOBJ, RA(I), RB(L), RV1(J), RV2(J);

FOBJ..     F=E=  (1-LAMBDA) * (( SUM(I, Y(I))/ M ) + ( SUM(L, Z(L)) / K ))
                    +  LAMBDA * ( SUM(J, (1 - EXP (-1 * ALFA * V(J)) )));
RA(I)..    SUM(J, -1 * A(I,J) * W(J)) + R + 1  =L= Y(I);
RB(L)..    SUM(J, B(L,J) * W(J)) - R + 1.0  =L= Z(L);
RV1(J)..    -1.0* V(J)   =L=  W(J) ;
RV2(J)..          V(J)   =G=  W(J) ;

MODEL SELECCION /ALL/;

SOLVE SELECCION USING NLP MINIMIZING F;

*****************************************************************************
                  NEOS Server Version 4.0
   Job#  : 462377
   Solver: CONOPT
```

```
   Start  : 10/18/2004 16:38:53
   End    : 10/18/2004 16:39:36
   Host   : akita.mcs.anl.gov

   Announcements:

   The latest version of the NEOS server can be found at:
   http://www-neos.mcs.anl.gov/
   Users can consult the NEOS Guide at:
   http://www.mcs.anl.gov/otc/Guide/
   which has extensive information on optimization.

   Disclaimer:

   This information is provided without any express or
   implied warranty. In particular, there is no warranty
   of any kind concerning the fitness of this
   information  for any particular purpose.

*******************************************************************************
You chose to use GAMS/CONOPT as your nonlinearly constrained optimization solver

%% GAMS OUTPUT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GAMS Rev 139  Intel /Linux                    10/18/04 16:38:55 Page 1
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
C o m p i l a t i o n

COMPILATION TIME     =  0.093 SECONDS    4.8 Mb  LNX214-139 Sep 01, 2004

GAMS Rev 139  Intel /Linux                    10/18/04 16:38:55 Page 2
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
Model Statistics    SOLVE SELECCION Using NLP From line 2832

MODEL STATISTICS

BLOCKS OF EQUATIONS       5      SINGLE EQUATIONS      2819
BLOCKS OF VARIABLES       6      SINGLE VARIABLES      2820
NON ZERO ELEMENTS     71152      NON LINEAR N-Z          24
DERIVATIVE POOL          30      CONSTANT POOL           16
CODE LENGTH             222


GENERATION TIME      =    0.274 SECONDS    7.9 Mb  LNX214-139 Sep 01, 2004

EXECUTION TIME       =    0.276 SECONDS    7.9 Mb  LNX214-139 Sep 01, 2004
GAMS Rev 139  Intel /Linux                    10/18/04 16:38:55 Page 3

G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
Solution Report     SOLVE SELECCION Using NLP From line 2832


          S O L V E     S U M M A R Y

     MODEL   SELECCION         OBJECTIVE  F
     TYPE    NLP               DIRECTION  MINIMIZE
     SOLVER  CONOPT            FROM LINE  2832

**** SOLVER STATUS             1 NORMAL COMPLETION
**** MODEL STATUS              2 LOCALLY OPTIMAL
**** OBJECTIVE VALUE           0.6719

 RESOURCE USAGE, LIMIT         39.750      1000.000
 ITERATION COUNT, LIMIT        135         10000
 EVALUATION ERRORS             0           0

    C O N O P T 3   Intel /Linux version 3.14D-015-051
    Copyright (C)   ARKI Consulting and Development A/S
```

Bagsvaerdvej 246 A
DK-2880 Bagsvaerd, Denmark

Cannot find option file
"/nfs/mcs-homes64/neosotc/.comms/jobs/neos.mcs.anl.gov:3333/462377/conopt.opt"
Using default options.

The model has 2820 variables and 2819 constraints
with 71152 Jacobian elements, 24 of which are nonlinear.
The Hessian of the Lagrangian has 24 elements on the diagonal,
0 elements below the diagonal, and 24 nonlinear variables.

** Warning ** The variance of the derivatives in the initial point is large (= 6.7). A better initial point, a better scaling, or better bounds on the variables will probably help the optimization.

** Optimal solution. There are no superbasic variables.

CONOPT time Total 39.870 seconds
of which: Function evaluations 0.007 = 0.0%
1st Derivative evaluations 0.003 = 0.0%

Work length = 32.76 Mbytes
Estimate = 32.76 Mbytes
Max used = 17.34 Mbytes

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU FOBJ	0.240	0.240	0.240	1.000

---- EQU RA

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	-3.886	-1.000	.
:				
:				
2200	-INF	-1.000	-1.000	-4.500E-4

---- EQU RB

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	-2.814	-1.000	.
:				
:				
570	-INF	-1.000	-1.000	-0.002

---- EQU RV1

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	.	.	-8.435E-4
:				
:				
24	-INF	.	.	-0.001

---- EQU RV2

	LOWER	LEVEL	UPPER	MARGINAL
1	.	3.404	+INF	.
:				
:				
24	.	3.1348E-4	+INF	.

---- VAR W

```
     LOWER     LEVEL       UPPER    MARGINAL

1     -INF      -1.702       +INF      .
2     -INF       0.003       +INF      .
3     -INF       0.003       +INF      .
4     -INF       2.5989E-5   +INF      .
5     -INF       2.8486E-4   +INF      .
6     -INF      -0.669       +INF      .
7     -INF         .         +INF      .
8     -INF       0.042       +INF      .
9     -INF       0.033       +INF      .
10    -INF      -4.07E-20    +INF      .
11    -INF       3.5337E-4   +INF      .
12    -INF       2.170E-19   +INF      .
13    -INF       0.130       +INF      .
14    -INF      -4.490E-5    +INF      .
15    -INF     6.661E-16     +INF      .
16    -INF        .          +INF      .
17    -INF    -1.73E-18      +INF      .
18    -INF    -2.844E-5      +INF      .
19    -INF    -2.074E-4      +INF      .
20    -INF    -0.009         +INF      .
21    -INF     0.542         +INF      .
22    -INF     0.008         +INF      .
23    -INF    -0.016         +INF      .
24    -INF   -1.567E-4        +INF      .

                 LOWER     LEVEL       UPPER    MARGINAL

---- VAR R          -INF    -3400.052     +INF       .

---- VAR Y

       LOWER     LEVEL     UPPER    MARGINAL

1         .          .        +INF     4.5000E-4
:
:
2200      .        1.005      +INF       .

---- VAR Z

      LOWER     LEVEL     UPPER    MARGINAL

1         .          .        +INF      0.002
:
:
570       .         0.155     +INF       .

---- VAR V

     LOWER     LEVEL       UPPER    MARGINAL

1     -INF       1.702        +INF        .
:
:


.
24    -INF     1.5674E-4     +INF        .

                 LOWER     LEVEL     UPPER    MARGINAL

---- VAR F           -INF      0.672     +INF       .


**** REPORT SUMMARY : 0  NONOPT
                      0 INFEASIBLE
```

```
                            0 UNBOUNDED
                            0 ERRORS

EXECUTION TIME = 0.062 SECONDS    3.7 Mb  LNX214-139 Sep 01, 2004

USER: MCS Department                          G030617:0959CS-LNX
      Argonne National Labs                         DC2747

**** FILE SUMMARY

Input      MODEL.gms
Output     solve.out

%% GAMS LOGFILE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GAMS Rev 139  Copyright (C) 1987-2004 GAMS Development. All rights reserved
Licensee: MCS Department                      G030617:0959CS-LNX
          Argonne National Labs                       DC2747
--- Starting compilation
--- MODEL.gms(2832) 5 Mb
--- Starting execution
--- Generating model SELECCION
--- MODEL.gms(2832) 10 Mb
---    2819 rows, 2820 columns, and 71152 non-zeroes.
--- MODEL.gms(2832) 8 Mb
--- Executing CONOPT
C O N O P T 3 Sep  6, 2004 LNX.CO.CO 21.4 015.051.041.LXI Library 314D

    C O N O P T 3   Intel /Linux version 3.14D-015-051
    Copyright (C)   ARKI Consulting and Development A/S
                    Bagsvaerdvej 246 A
                    DK-2880 Bagsvaerd, Denmark
```

APENDICE B: Formulación del Problema Lineal

Se muestra la entrada de los datos del Problema Lineal (PL) al software GAMS y la salida de los mismos por medio del programa NEOS.

```
$ONTEXT

RESOLVER UN PROBLEMA DE PROGRAMACIÓN  LINEAL PARA LA :

  -CLASIFICACIÓN DE ILICITOS (USANDO INDICES)
  -SELECCIÓN DE ATRIBUTOS

 EL PROBLEMA ES:

 MINIMIZAR F(W,R,Y,Z,V)=(1-LAMBDA) ((e**TY/M) + (e**TZ/K)) + (LAMBDA*e**T) (e-E**-(ALFA*V))
   SUJETO A    -AW+eR+e<=Y,
                BW-eR+e<=Z,
                Y>=0, Z>=0
                -V<=W<=V

EL PROBLEMA USA DATOS DE UNA BASE DE DATOS DE  ILÍCITOS LA CUAL CONTIENE 24 ATRIBUTOS, 2,200 REGISTROS DE
TIPO 1 Y 570 REGISTROS DE TIPO 9. CON:
LAMBDA=0.05
ALFA=0.1

$OFFTEXT

SET
 I  / 1*2200/
 J  / 1*24/
 L  / 1*570/;

TABLE
A(I,J); (Aquí se introduce una matriz de datos de 2200 instancias x 24 atributos)

TABLE
B(L,J); (Aquí se introduce una matriz de datos de 2200 instancias x 24 atributos)

VARIABLES
W(J), R, Y(I), Z(L), V(J), F;

POSITIVE VARIABLES
Y, Z;

SCALAR  M /2200/, K /570/, LAMBDA /0.05/, ALFA /0.1/;

EQUATIONS
FOBJ, RA(I), RB(L), RV1(J), RV2(J);

FOBJ..     F=E=  (1-LAMBDA) * (( SUM(I, Y(I))/ M ) + ( SUM(L, Z(L)) / K ))
                    +  LAMBDA * ( SUM(J, (1 -(1 + (-1 * ALFA * V(J)) ))));
RA(I)..    SUM(J, -1 * A(I,J) * W(J)) + R + 1  =L= Y(I);
RB(L)..    SUM(J, B(L,J) * W(J)) - R + 1.0  =L= Z(L);
RV1(J)..    -1.0* V(J)   =L=  W(J) ;
RV2(J)..          V(J)   =G=  W(J) ;

MODEL SELECCION /ALL/;

SOLVE SELECCION USING LP MINIMIZING F;

**************************************************************************
                  NEOS Server Version 4.0

   Job#   : 463506
   Solver : PATH (GAMS input)
```

```
   Start   : 10/21/2004 12:50:49
   End     : 10/21/2004 12:50:58
   Host    : newton.mcs.anl.gov

   Announcements:

   The latest version of the NEOS server can be found at:
   http://www-neos.mcs.anl.gov/
   Users can consult the NEOS Guide at:
   http://www.mcs.anl.gov/otc/Guide/
   which has extensive information on optimization.

   Disclaimer:

   This information is provided without any express or
   implied warranty. In particular, there is no warranty
   of any kind concerning the fitness of this
   information  for any particular purpose.

*******************************************************************************
You chose to use GAMS/PATH as your complementarity problem solver

%% GAMS OUTPUT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GAMS Rev 139  Intel /Linux                      1004 12:50:53 Page 1
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
C o m p i l a t i o n

COMPILATION TIME     = 0.069 SECONDS    4.8 Mb  LNX214-139 Sep 01, 2004

GAMS Rev 139  Intel /Linux                      1004 12:50:53 Page 2
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
Model Statistics    SOLVE SELECCION Using LP From line 2833

MODEL STATISTICS

BLOCKS OF EQUATIONS       5     SINGLE EQUATIONS     2819
BLOCKS OF VARIABLES       6     SINGLE VARIABLES     2820
NON ZERO ELEMENTS     71152

GENERATION TIME = 0.193 SECONDS    7.9 Mb  LNX214-139 Sep 01, 2004

EXECUTION TIME   = 0.194 SECONDS    7.9 Mb  LNX214-139 Sep 01, 2004

GAMS Rev 139  Intel /Linux                      1004 12:50:53 Page 3
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
Solution Report     SOLVE SELECCION Using LP From line 2833

          S O L V E     S U M M A R Y

    MODEL   SELECCION         OBJECTIVE  F
    TYPE    LP                DIRECTION  MINIMIZE
    SOLVER  XPRESS            FROM LINE  2833

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE   0.6555

 RESOURCE USAGE, LIMIT      4.310      1000.000
 ITERATION COUNT, LIMIT     2019       10000

Xpress-MP     Sep  6, 2004 LNX.XP.XP 21.4 026.028.041.LXI Xpress lib 15.10
Xpress-MP licensed by Dash to GAMS Development Corp. for GAMS

optimal LP solution found: objective value 0.65547726413

                  LOWER     LEVEL     UPPER    MARGINAL
```

---- EQU FOBJ . . . 1.000

---- EQU RA

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	-2.119	-1.000	.
:				
:				
2200	-INF	-1.000	-1.000	-4.318E-4

---- EQU RB

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	-1.663	-1.000	.
:				
:				
570	-INF	-1.000	-1.000	-0.002

---- EQU RV1

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	.	.	-0.005
:				
:				
24	-INF	.	.	-0.005

---- EQU RV2

	LOWER	LEVEL	UPPER	MARGINAL
1	.	1.282	+INF	.
:				
:				
24	.	7.8950E-5	+INF	.

---- VAR W

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	-0.641	+INF	.
2	-INF	0.002	+INF	.
3	-INF	0.003	+INF	.
4	-INF	1.9668E-5	+INF	.
5	-INF	2.1239E-4	+INF	.
6	-INF	-0.462	+INF	.
7	-INF	.	+INF	.
8	-INF	0.010	+INF	.
9	-INF	0.008	+INF	.
10	-INF	.	+INF	.
11	-INF	8.9140E-5	+INF	.
12	-INF	.	+INF	.
13	-INF	0.069	+INF	.
14	-INF	-1.013E-4	+INF	.
15	-INF	.	+INF	.
16	-INF	.	+INF	.
17	-INF	.	+INF	.
18	-INF	-2.039E-5	+INF	.
19	-INF	-8.965E-5	+INF	.
20	-INF	.	+INF	.
21	-INF	0.196	+INF	.
22	-INF	.	+INF	.
23	-INF	-0.011	+INF	.
24	-INF	-3.948E-5	+INF	.

```
                 LOWER     LEVEL     UPPER    MARGINAL

---- VAR R             -INF  -1281.444     +INF       .

---- VAR Y

       LOWER     LEVEL     UPPER    MARGINAL

1          .          .       +INF       4.3182E-4
:
:
2200       .        0.440   +INF       .

---- VAR Z

      LOWER     LEVEL     UPPER    MARGINAL

1       .        .       +INF      0.002
:
:
570     .     1.281    +INF       .

---- VAR V

     LOWER     LEVEL     UPPER    MARGINAL

1    -INF       0.641       +INF        .
:
:
24   -INF    3.9475E-5    +INF        .

                 LOWER     LEVEL     UPPER    MARGINAL

---- VAR F         -INF       0.655       +INF        .

**** REPORT SUMMARY :  0  NONOPT
                       0 INFEASIBLE
                       0 UNBOUNDED

EXECUTION TIME  = 0.046 SECONDS    3.7 Mb  LNX214-139 Sep 01, 2004

USER: MCS Department                          G030617:0959CS-LNX
Argonne National Labs                           DC2747

**** FILE SUMMARY

Input./MODEL
Output./solve.out

%% GAMS LOGFILE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GAMS Rev 139  Copyright (C) 1987-2004 GAMS Development. All rights reserved
Licensee: MCS Department                                G030617:0959CS-LNX
         Argonne National Labs                                DC2747
--- Starting compilation
--- MODEL(2833) 5 Mb
--- Starting execution
--- Generating model SELECCION
--- MODEL(2833) 8 Mb
---    2819 rows, 2820 columns, and 71152 non-zeroes.
--- Executing XPRESS

Xpress-MP     Sep  6, 2004 LNX.XP.XP 21.4 026.028.041.LXI Xpress lib 15.10
Xpress-MP licensed by Dash to GAMS Development Corp. for GAMS

Reading data . . . done.
```

```
Reading Problem gmsxp_xx
Problem Statistics
        2819 (0 spare) rows
        2820 (0 spare) structural columns
       71152 (0 spare) non-zero elements
Global Statistics
           0 entities        0 sets        0 set members
Presolved problem has:2818 rows    2818 cols   68356 non-zeros
```

APÉNDICE C: Formulación del Problema Lineal con Peso

Se muestra la entrada de los datos del Problema Lineal (PL), (pero con la asignación de un peso), al software GAMS y la salida de los mismos por medio del programa NEOS.

```
$ONTEXT

 RESOLVER UN PROBLEMA DE PROGRAMACIÓN LINEAL PARA LA :
  -CLASIFICACIÓN DE ILICITOS (USANDO INDICES)
  -SELECCIÓN DE ATRIBUTOS

 EL PROBLEMA ES:
 MINIMIZAR F(W,R,Y,Z,V)=(1-LAMBDA)((e**TY/M)+(e**TZ/K))+(LAMBDA*e**T)(e-(1+(-ALFA*V))
 SUJETO A:     -AW+eR+e<=Y,
                BW-eR+e<=Z,
                Y>=0, Z>=0
               -V<=W<=V

 EL PROBLEMA USA DATOS DE UNA BASE DE DATOS DE ILICITOS LA CUAL CONTIENE 24 ATRIBUTOS,
 2,200 REGISTROS DE TIPO 1 Y 570 REGISTROS DE TIPO 9. CON:
 LAMBDA=0.05
 ALFA  =0.1

$OFFTEXT

SET
  I  / 1*2200/
  J  / 1*24/
  L  / 1*570/;

TABLE
A(I,J); (Aquí se introduce una matriz de datos de 2200 instancias x 24 atributos)

TABLE
B(L,J); (Aquí se introduce una matriz de datos de 570 instancias x 24 atributos)

VARIABLES
W(J), R, Y(I), Z(L), V(J), F;

POSITIVE VARIABLES
Y, Z;

SCALAR  M /2200/, K /570/, LAMBDA /0.05/, ALFA /0.1/, PNI/1/, PI/2/;

EQUATIONS
FOBJ, RA(I), RB(L), RV1(J), RV2(J);

FOBJ..     F=E=  (1-LAMBDA) * ((PNI*( SUM(I, Y(I))/ M )) + (PI*( SUM(L, Z(L)) / K )))
                +  LAMBDA * ( SUM(J, (1 -(1 + (-1 * ALFA * V(J)) ))));
RA(I)..    SUM(J, -1 * A(I,J) * W(J)) + R + 1  =L= Y(I);
RB(L)..    SUM(J, B(L,J) * W(J)) - R + 1.0  =L= Z(L);
RV1(J)..    -1.0* V(J)   =L=  W(J) ;
RV2(J)..     V(J)   =G=  W(J) ;

MODEL SELECCION /ALL/;

SOLVE SELECCION USING LP MINIMIZING F;

*****************************************************************************
                  NEOS Server Version 4.0

   Job#   : 466542
   Solver : PATH (GAMS input)
   Start  : 10/29/2004 12:35:17
   End    : 10/29/2004 12:35:28
```

```
   Host   : newton.mcs.anl.gov

   Announcements:

   The latest version of the NEOS server can be found at:
   http://www-neos.mcs.anl.gov/
   Users can consult the NEOS Guide at:
   http://www.mcs.anl.gov/otc/Guide/
   which has extensive information on optimization.

   Disclaimer:

   This information is provided without any express or
   implied warranty. In particular, there is no warranty
   of any kind concerning the fitness of this
   information  for any particular purpose.

********************************************************************************
You chose to use GAMS/PATH as your complementarity problem solver

%% GAMS OUTPUT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GAMS Rev 139  Intel /Linux                    1004 12:35:21 Page 1
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
C o m p i l a t i o n

COMPILATION TIME     = 0.090 SECONDS    4.8 Mb  LNX214-139 Sep 01, 2004

GAMS Rev 139  Intel /Linux                    1004 12:35:21 Page 2
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
Model Statistics    SOLVE SELECCION Using LP From line 2833

MODEL STATISTICS

BLOCKS OF EQUATIONS       5     SINGLE EQUATIONS     2819
BLOCKS OF VARIABLES       6     SINGLE VARIABLES     2820
NON ZERO ELEMENTS      71152

GENERATION TIME      = 0.293 SECONDS    7.9 Mb  LNX214-139 Sep 01, 2004

EXECUTION TIME       = 0.293 SECONDS    7.9 Mb  LNX214-139 Sep 01, 2004

GAMS Rev 139  Intel /Linux                    1004 12:35:21 Page 3
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
Solution Report     SOLVE SELECCION Using LP From line 2833
           S O L V E     S U M M A R Y

    MODEL   SELECCION         OBJECTIVE  F
    TYPE    LP                DIRECTION  MINIMIZE
    SOLVER  XPRESS            FROM LINE  2833

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE   0.7715

 RESOURCE USAGE, LIMIT         3.850      1000.000
 ITERATION COUNT, LIMIT        1844       10000

Xpress-MP     Sep  6, 2004 LNX.XP.XP 21.4 026.028.041.LXI Xpress lib 15.10
Xpress-MP licensed by Dash to GAMS Development Corp. for GAMS

optimal LP solution found: objective value 0.77146693329

                 LOWER     LEVEL     UPPER    MARGINAL

---- EQU FOBJ        .           .           .         1.000
```

---- EQU RA

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	-3.384	-1.000	.
:				
:				
2200	-INF	-1.000	-1.000	-4.318E-4

---- EQU RB

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	-3.010	-1.000	.
:				
:				
570	-INF	-1.015	-1.000	.

---- EQU RV1

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	.	.	-0.005
:				
:				
24	-INF	.	.	-0.005

---- EQU RV2

	LOWER	LEVEL	UPPER	MARGINAL
1	.	3.905	+INF	.
:				
:				
24	.	1.0446E-4	+INF	.

---- VAR W

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	-1.953	+INF	.
2	-INF	0.002	+INF	.
3	-INF	.	+INF	.
4	-INF	2.9530E-5	+INF	.
5	-INF	1.6821E-4	+INF	.
6	-INF	-0.678	+INF	.
7	-INF	.	+INF	.
8	-INF	0.055	+INF	.
9	-INF	0.086	+INF	.
10	-INF	.	+INF	.
11	-INF	.	+INF	.
12	-INF	-9.356E-5	+INF	.
13	-INF	0.093	+INF	.
14	-INF	.	+INF	.
15	-INF	-3.207E-5	+INF	.
16	-INF	-8.597E-5	+INF	.
17	-INF	.	+INF	.
18	-INF	.	+INF	.
19	-INF	-1.690E-4	+INF	.
20	-INF	-0.006	+INF	.
21	-INF	0.447	+INF	.
22	-INF	.	+INF	.
23	-INF	-0.004	+INF	.
24	-INF	-5.223E-5	+INF	.

	LOWER	LEVEL	UPPER	MARGINAL

```
---- VAR R          -INF     -3897.701     +INF         .

---- VAR Y

       LOWER     LEVEL     UPPER    MARGINAL

1          .          .        +INF     4.3182E-4
:
:
2200       .        1.339      +INF       .

---- VAR Z

      LOWER     LEVEL     UPPER    MARGINAL

1         .          .        +INF       0.003
:
:
570       .          .        +INF       0.003

---- VAR V

     LOWER     LEVEL     UPPER    MARGINAL

1      -INF       1.953       +INF         .
:
:
24     -INF    5.2228E-5   +INF         .

                  LOWER     LEVEL     UPPER    MARGINAL

---- VAR F          -INF       0.771      +INF          .


**** REPORT SUMMARY : 0  NONOPT
                       0 INFEASIBLE
                       0 UNBOUNDED

EXECUTION TIME       = 0.067 SECONDS     3.7 Mb  LNX214-139 Sep 01, 2004

USER: MCS Department                          G030617:0959CS-LNX
      Argonne National Labs                         DC2747

**** FILE SUMMARY

Input./MODEL
Output./solve.out
```

APÉNDICE D: Formulación del Problema del Método de Punto Interior

Se muestra la entrada de los datos del Problema del Método de Punto Interior (MIP) al software GAMS y la salida de los mismos por medio del programa NEOS.

```
$ONTEXT

 RESOLVER UN PROBLEMA DE PROGRAMACIÓN NO LINEAL PARA LA :
 -CLASIFICACIÓN DE ILICITOS (USANDO INDICES)
 -SELECCIÓN DE ATRIBUTOS

 EL PROBLEMA ES:
MINIMIZAR F(W,R,Y,Z,V) = (1-LAMBDA) ((e**TY/M) + (e**TZ/K)) + (LAMBDA *e**T) (e-E**-(ALFA*V))
 SUJETO A:      -AW+eR+e<=Y,
                 BW-eR+e<=Z,
                 Y>=0, Z>=0
                -V<=W<=V

EL PROBLEMA USA DATOS DE UNA BASE DE DATOS DE ILICITOS LA CUAL CONTIENE 24 ATRIBUTOS, 2,200 REGISTROS DE
TIPO 1 Y 570 REGISTROS DE TIPO 9. CON:
LAMBDA=0.01
ALFA=0.1

$OFFTEXT

SET
I  / 1*2200/
J  / 1*24/
L  / 1*570/;

TABLE
A(I,J); (Aquí se introduce una matriz de datos de 2200 instancias x 24 atributos)

TABLE
B(L,J); (Aquí se introduce una matriz de datos de 570 instancias x 24 atributos)

VARIABLES
W(J), R, Y(I), Z(L), V(J), F;

POSITIVE VARIABLES
Y, Z;

SCALAR  M /2200/, K /570/, LAMBDA /0.01/, ALFA /0.1/;

EQUATIONS
FOBJ, RA(I), RB(L), RV1(J), RV2(J);

FOBJ..     F=E=  (1-LAMBDA) * (( SUM(I, Y(I))/ M ) + ( SUM(L, Z(L)) / K ))
                 +  LAMBDA * ( SUM(J, (1 -(1 + (-1 * ALFA * V(J)) ))));
RA(I)..    SUM(J, -1 * A(I,J) * W(J)) + R + 1  =L= Y(I);
RB(L)..    SUM(J, B(L,J) * W(J)) - R + 1.0  =L= Z(L);
RV1(J)..    -1.0* V(J)   =L=  W(J) ;
RV2(J)..     V(J)   =G=  W(J) ;

MODEL SELECCION /ALL/;

SOLVE SELECCION USING MIP MINIMIZING F;

*****************************************************************************
                  NEOS Server Version 4.0

   Job#   : 477594
   Solver : MOSEK
   Start  : 11/19/2004 11:50:40
```

```
   End    : 11/19/2004 11:50:44
   Host   : akita.mcs.anl.gov

   Announcements:

   The latest version of the NEOS server can be found at:
   http://www-neos.mcs.anl.gov/
   Users can consult the NEOS Guide at:
   http://www.mcs.anl.gov/otc/Guide/
   which has extensive information on optimization.

   Disclaimer:

   This information is provided without any express or
   implied warranty. In particular, there is no warranty
   of any kind concerning the fitness of this
   information  for any particular purpose.

*******************************************************************************
You chose to use GAMS/MOSEK_LP as your linear program solver

%% GAMS OUTPUT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GAMS Rev 139  Intel /Linux                   11/19/04 11:50:42 Page 1
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
C o m p i l a t i o n

COMPILATION TIME     =0.092 SECONDS    4.8 Mb  LNX214-139 Sep 01, 2004

GAMS Rev 139  Intel /Linux                   11/19/04 11:50:42 Page 2
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
Model Statistics    SOLVE SELECCION Using MIP From line 2832

MODEL STATISTICS

BLOCKS OF EQUATIONS       5     SINGLE EQUATIONS     2819
BLOCKS OF VARIABLES       6     SINGLE VARIABLES     2820
NON ZERO ELEMENTS     71152

GENERATION TIME      = 0.316 SECONDS    7.9 Mb  LNX214-139 Sep 01, 2004

EXECUTION TIME       = 0.316 SECONDS    7.9 Mb  LNX214-139 Sep 01, 2004

GAMS Rev 139  Intel /Linux                   11/19/04 11:50:42 Page 3
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
Solution Report     SOLVE SELECCION Using MIP From line 2832

          S O L V E     S U M M A R Y

    MODEL   SELECCION         OBJECTIVE  F
    TYPE    MIP               DIRECTION  MINIMIZE
    SOLVER  MOSEK             FROM LINE  2832

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE   0.6720

 RESOURCE USAGE, LIMIT         0.850      1000.000
 ITERATION COUNT, LIMIT        3          10000

 MOSEK Link    Sep  6, 2004 LNX.MK.MK 21.4 007.028.041.LXI DLL/SO 3.0.1.28

 M O S E K       version 3.0.1.28(RC) (Build date: Jan 12 2004 09:23:28)
 Copyright (C)   MOSEK ApS, Fruebjergvej 3, Box 16
 DK-2100 Copenhagen, Denmark
 http://www.mosek.com
```

MOSEK Warning 50 - Could not open the parameter file '/nfs/mcs-homes64/neos
otc/.comms/jobs/neos.mcs.anl.gov:3333/477594/mosek.opt'.

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU FOBJ	.	.	.	1.000

---- EQU RA

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	-3.878	-1.000	.
:				
:				
2200	-INF	-1.000	-1.000	-4.500E-4

---- EQU RB

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	-2.807	-1.000	.
:				
:				
570	-INF	-1.000	-1.000	-0.002

---- EQU RV1

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	.	.	-0.001
:				
:				
24	-INF	.	.	-0.001

---- EQU RV2

	LOWER	LEVEL	UPPER	MARGINAL
1	.	3.394	+INF	.
:				
:				
24	.	3.4911E-4	+INF	.

---- VAR W

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	-1.697	+INF	.
2	-INF	0.003	+INF	.
3	-INF	0.003	+INF	.
4	-INF	2.5988E-5	+INF	.
5	-INF	2.8482E-4	+INF	.
6	-INF	-0.668	+INF	.
7	-INF	.	+INF	.
8	-INF	0.042	+INF	.
9	-INF	0.033	+INF	.
10	-INF	.	+INF	.
11	-INF	3.4943E-4	+INF	.
12	-INF	.	+INF	.
13	-INF	0.132	+INF	.
14	-INF	-4.352E-5	+INF	.
15	-INF	.	+INF	.
16	-INF	.	+INF	.
17	-INF	.	+INF	.
18	-INF	-2.843E-5	+INF	.
19	-INF	-2.088E-4	+INF	.
20	-INF	-0.008	+INF	.
21	-INF	0.542	+INF	.
22	-INF	0.010	+INF	.

```
23   -INF     -0.017       +INF       .
24   -INF  -1.746E-4      +INF       .

                  LOWER     LEVEL       UPPER    MARGINAL

---- VAR R         -INF     -3390.994     +INF         .

---- VAR Y

      LOWER     LEVEL     UPPER    MARGINAL
1         .         .        +INF     4.5000E-4
:
:
2200      .        1.006     +INF       .

---- VAR Z

     LOWER     LEVEL     UPPER    MARGINAL

1         .         .        +INF       0.002
:
:
570       .       0.160       +INF       .

---- VAR V

    LOWER     LEVEL     UPPER    MARGINAL

1     -INF      1.697      +INF          .
:
:
24    -INF   1.7455E-4   +INF          .

                  LOWER     LEVEL     UPPER    MARGINAL

---- VAR F          -INF       0.672      +INF          .

**** REPORT SUMMARY: 0  NONOPT
                     0 INFEASIBLE
                     0 UNBOUNDED

EXECUTION TIME       = 0.063 SECONDS     3.7 Mb  LNX214-139 Sep 01, 2004

USER: MCS Department                          G030617:0959CS-LNX
      Argonne National Labs                         DC2747

**** FILE SUMMARY

Input      MODEL.gms
Output     solve.out
```