Estadística Española • Volumen 54, número 179 / 2012, pp. 329-355

Desarrollo de una aplicación de código abierto para el control de la confidencialidad estadística en la difusión de datos de la AEAT

Diego Porras Escalada
Servicio de Estudios Tributarios y Estadísticas, AEAT

Félix Saz Marco
Servicio de Estudios Tributarios y Estadísticas, AEAT

David Pérez Fernández
Servicio de Estudios Tributarios y Estadísticas, AEAT

Resumen

La supresión de celdas es una técnica de protección de la confidencialidad estadística ampliamente conocida y aplicada a las publicaciones tabulares de datos estadísticos.

El problema de encontrar la supresión de celdas óptima que, garantizando la confidencialidad de los datos, oculta la menor cantidad de información es conocido como problema de supresión de celdas (*CSP: Cell Suppression Problem*).

En este artículo, se describe una aplicación que resuelve el problema CSP asociado a la publicación de estadísticas tabulares basado en estándares abiertos y empleando un optimizador lineal de código abierto (GLPK). Dicha aplicación está puesta a disposición del público a través del CTT (Centro de Transferencia de Tecnología de la Administración Pública[1])

En el artículo se muestran resultados computacionales obtenidos por la aplicación, sobre datos reales de las publicaciones de estadísticas realizadas en el Servicio de Estudios Tributarios y Estadísticas (SETE). También se incluyen los resultados de la batería de test conocida como librería CSPlib.

Por otra parte, se muestra el esquema de la integración de la aplicación de protección de la confidencialidad estadística con el sistema de difusión de datos empleado por la Agencia Tributaria. Dicho sistema se basa en una arquitectura OLAP (*OnLine Analytical Processing*).

[1] JCsp: http://administracionelectronica.gob.es/es/ctt/jcsp

Finalmente, se plantean futuras mejoras en la aplicación; tanto en el algoritmo de cálculo, optimizadores lineales así como en la integración con el sistema de difusión estadística de la AEAT.

Palabras clave: control de la confidencialidad estadística, confidencialidad estadística, protección de datos tabulares, supresión de celdas, OLAP en la difusión de datos estadísticos, método Branch-and-Cut, programación entera.

Clasificación AMS: 62P99, 65K05

Development of an open source application for statistical disclosure control on the Spanish Tax Agency

Abstract

The cell suppression is a well known statistical disclosure control technique for tabular data protection.

The cell suppression problem (CSP) is a problem consisting on find the optimal suppression that minimizes the lost of information guaranteeing data confidentiality.

This paper describes an application that solves the CSP problem related to a given statistical tabular data publication. It is based on open source standards and uses an open linear optimizer called GLPK. This application has been shared on Spanish CTT[2].

The current paper shows the application computational results for the information published by SETE (Spanish Tax Agency, statistical information dissemination department). It also contains results running the known CSPLib test library.

Spanish Tax Agency statistical information diffusion system is based on the OLAP (OnLine Analytical Processing) paradigm. The Statistical disclosure control system, using the cell suppression technique, has been integrated with the diffusion system.

Finally, some improvements about statistical disclosure calculation algorithms, linear optimizer and OLAP system integration are proposed.

*Key Words***:** Statistical disclosure control, confidentiality control, tabular data protection, cell suppression, OLAP, branch-and-cut method, statistical data dissemination, linear programming.

AMS clasification: 62P99, 65K05

[2] JCsp: http://administracionelectronica.gob.es/es/ctt/jcsp

1. Introducción

1.1 Información estadística y confidencialidad estadística

La información estadística cumple una función esencial para el conocimiento de la realidad económica y social. Por esta razón, entre las competencias de la Agencia Estatal de Administración Tributaria se encuentra la tarea de difusión de la información estadística de origen tributario.

La Agencia Tributaria debe velar porque la información estadística sea suministrada de forma adecuada, según criterios de máxima accesibilidad, calidad y plena transparencia. Por otra parte, la Agencia Tributaria, en su labor recaudatoria, recopila información confidencial sobre empresas e individuos sobre la que debe garantizar su confidencialidad.

Hay que alcanzar un compromiso entre la creciente demanda de información económica y social, por medio de una amplia y precisa información estadística, y la obligación legal y ética de la Agencia Tributaria de proteger la privacidad de los individuos y las empresas que son la fuente de dichos datos estadísticos.

No se trata, por tanto, de un problema de confidencialidad exclusivamente, sino de maximizar la información proporcionada manteniendo las restricciones que impone la protección de los datos estadísticos confidenciales.

El conjunto de métodos que intentan dar solución a este problema se denominan técnicas de control de la confidencialidad estadística (Statistical Disclosure Control, SDC).

1.2 Ámbito legal de la confidencialidad estadística

El control de la confidencialidad estadística en la difusión de datos estadísticos está regulado, a nivel estatal, por medio de la Ley de la Función Estadística Pública [1]. En el ámbito europeo es el Reglamento 233/2009 del Parlamento Europeo [13] el que establece la normativa relativa a la función estadística.

La Ley de la Función Estadística Pública, Ley 12/1989, en su capítulo III, contiene la normativa relativa al tratamiento de la confidencialidad de los datos personales que obtengan los servicios estadísticos, tanto directamente de los informantes como a través de otras fuentes administrativas.

Por otra parte, el Reglamento (CE) 233/2009 del Parlamento Europeo, en su capítulo V, artículo 20, trata la protección de datos confidenciales. Según el Reglamento se considera "dato confidencial" aquel que permite identificar, directa o indirectamente, a las unidades estadísticas y divulgar, por tanto, información sobre particulares.

Según dicho Reglamento, la confidencialidad de los datos debe satisfacer los mismos principios en todos los Estados miembros. Establece las normas y medidas que se aplicarán para garantizar que los datos confidenciales se utilicen exclusivamente con fines estadísticos, y para evitar su revelación ilegal.

1.3 Técnicas de control de la confidencialidad estadística

El problema del control de la confidencialidad estadística consiste en la no divulgación de datos confidenciales directos así como de datos que permitan la inferencia de datos confidenciales.

El principal objetivo de dichas técnicas consiste en aportar la máxima protección de los datos confidenciales con la menor pérdida de información.

En la actualidad existe una gran variedad de técnicas de *control de la confidencialidad estadística*[3]:

- **Métodos de Restricción**: Este tipo de métodos consiste en la limitación de la información publicada. Podemos subdividirlos a su vez en:
  - Métodos de recodificación: agrupación de valores de variables de clasificación para ocultar los datos confidenciales de determinados colectivos minoritarios. Incluyendo la eliminación de algunos valores de la variable o variables elegidas.
  - Supresión de celdas (CSP): se suprimen tanto las celdas que contienen in formación confidencial, llamadas supresiones primarias, así como aquellas que puedan servir para inferir las primeras, supresiones secundarias.
  - Publicación de intervalos: en lugar de publicar datos individuales, se muestran los márgenes del intervalo seguro en el que se encuentra el verdadero valor.
- **Métodos de perturbación**: Los métodos de control de la confidencialidad estadística basados en la perturbación de datos se emplean la alteración de los datos originales proporcionando unos nuevos datos sintéticos protegidos. Dentro de este conjunto de métodos se pueden citar los siguientes:
  - Redondeo aleatorio (CRP): se trata de redondear de forma aleatoria los valores de los datos confidenciales para mantener su confidencialidad.
  - Ruido (CTA): emplea la introducción de pequeñas variaciones en los valores o en las variables de clasificación de modo que desaparezcan o queden ocultas las celdas deseadas.
  - Sustitución: intercambio de valores dentro de una variable de clasificación para su tratamiento estadístico sin posibilitar su identificación.
  - Microagregación: consiste en sustituir los valores confidenciales por la media de los valores dentro del grupo al que pertenecen.

1.4 Metodología elegida

En el contexto de la difusión de datos estadísticos de la Agencia Tributaria, la metodología elegida ha sido la supresión de celdas. Se realiza una recodificación previa de algunas de las variables de clasificación para reducir la cantidad global de datos confidenciales.

[3] Para una exposición detallada y ejemplos de los principales métodos de control de la confidencialidad estadística ver Willenborg y De Wall (1996) [2].

Se ha primado la fidelidad de los datos en la difusión de datos tributarios destinados al público general, sin fines científicos. La principal razón para emplear la metodología de supresión de celdas es que garantiza la fidelidad de los datos publicados. A pesar de que los métodos perturbativos ofrecen mayor información (por ejemplo, el valor aproximado o el intervalo en el que se encuentra el dato) la fidelidad de los datos no está garantizada.

Para su uso en estudios económicos y estadísticos, tal y como contempla el Reglamento 223/2009 del Parlamento Europeo y del Consejo sobre la función estadística, tanto Eurostat como los Institutos Nacionales de Estadística podrán conceder el acceso a datos confidenciales a investigadores que lleven a cabo análisis estadísticos con fines científicos. En cuyo caso no es necesario el control de la confidencialidad estadística.

1.5 Modelo entero del problema de supresión de celdas

Cuando se realiza una publicación de datos estadísticos se eligen unos parámetros de seguridad a partir de los cuales se determina qué celdas[4] deben ser ocultadas por contener información confidencial. Estas celdas se denominan supresiones primarias.

En ocasiones, dichos parámetros de configuración son tan simples como el mínimo número de declarantes agrupados de los que puede mostrarse una cierta información sensible. En otros casos, los requisitos de confidencialidad pueden ser más complejos (regla de dominancia, intervalo de confianza…).

Los niveles de protección se establecen definiendo, para cada celda: los valores mínimo (lpl), máximo (upl) y el tamaño del intervalo (spl) en el que puede hallarse el valor de una celda que se quiere proteger. Los requisitos de confidencialidad de un dato se definen generalmente como el valor máximo, mínimo y el tamaño del intervalo deducibles por un lector de la publicación a partir de la relaciones tabulares publicadas (marginales, subtotales, etc.).

En el caso de que la información mostrada contenga marginales, subtotales, tablas enlazadas u otras operaciones sobre celdas publicadas, surge el problema de que algunas de las celdas, suprimidas en la publicación, puedan ser deducidas. En determinadas situaciones pueden invertirse algunas de las fórmulas para obtener el valor exacto o aproximado de celdas que han sido ocultadas.

Por este motivo, se deben establecer nuevas supresiones (secundarias) cuyo objetivo es que las supresiones primarias no puedan ser deducidas. El problema de obtener una colección de supresiones secundarias que garanticen la confidencialidad y que minimicen la pérdida de información es conocido como problema CSP.

Una persona ajena a la organización y cuyos datos no forman parte de la estadística (atacante[5] externo) puede intentar, no sólo deducir el valor exacto de los datos suprimidos, sino también, en qué intervalo puede encontrarse. Por lo tanto, la

[4] En adelante, llamaremos *celda* a cualquier elemento individual que aparezca en la publicación estadística.

[5] Se entiende por *atacante* un sujeto que tiene en su poder los datos publicados y empleando métodos matemáticos (por ejemplo, optimización lineal) intenta desvelar valores que han sido suprimidos en la publicación por razones de confidencialidad.

publicación estadística requiere una protección extra que también debe incluirse entre los requisitos de confidencialidad.

La pérdida de información se establece numéricamente para cada celda de la publicación. En el caso de la publicación estadística de la AEAT, se establece que el coste de la supresión de una celda es igual al dato que representa aunque esto no tiene porqué ser necesariamente así.

A partir de todas estas premisas se establece un modelo de optimización lineal entera cuya resolución permite obtener el conjunto óptimo (en el sentido de pérdida de información antes indicado) de celdas a suprimir para garantizar la confidencialidad de la solución. En el *Apéndice 1. Modelo entero del problema de supresión de celdas* puede verse un desarrollo más extenso de este modelo entero.

1.6 Motivación del proyecto

El volumen del problema de la protección de la confidencialidad de datos tabulares se hace muy grande cuando aumenta el número de variables de clasificación. Resolver este problema empleando métodos manuales resulta imposible en la práctica.

Entre las alternativas existentes para el cálculo automatizado de las supresiones secundarias se encuentra el método del hipercubo de Sarah Giessing, ver [6]. Este método implica una gran pérdida de información ya que, aunque la confidencialidad está asegurada, el resultado no es óptimo.

En el momento en el que se planteó este proyecto, el principal software existente para la resolución del problema de la supresión de celdas era Tau Argus. Dicho aplicativo requiere del uso de un optimizador lineal comercial (XPress y CPlex). Ambos con un precio de licencia por usuario bastante elevado.

Por otra parte, al no disponer del código fuente de la aplicación, era imposible realizar la adaptación al sistema de difusión de la AEAT.

Finalmente, forma parte de la política de desarrollo del servicio la participación en proyectos de código abierto, porque garantizan la naturaleza colaborativa del desarrollo, promueven la evolución de las aplicaciones y son objetivo marcado por las decisiones europeas en materia de desarrollo de software en la Administración Pública y por la normativa española correspondiente[6].

2. Aplicación JCSP

En este apartado se describen algunos de los aspectos técnicos de la implementación de la aplicación informática de control de la confidencialidad estadística (JCSP). Por una parte, se describe algunas cuestiones prácticas del desarrollo informático y, por otra, se

[6] Decision 87/95/CEE, Decision 2004/387/CE (IDABC *Interoperable Delivery of European e Government to public Administrations, Business and Citizens*),
Decision 922/2009/CE (ISA, *Interoperability Solutions for Administrations*).
Esquemas Nacionales de Seguridad e Interoperabilidad (RD 3/2010 y RD 4/2010)

incluyen los principales resultados computacionales del uso de la herramienta en el contexto de la difusión estadística de la AEAT.

2.1 Implementación en GLPK

El núcleo del sistema de supresión de celdas ha sido programado en lenguaje Java. Se eligió este lenguaje por razones de compatibilidad en la integración con las aplicaciones del sistema de difusión de estadísticas de la AEAT, que se encuentra programado en el mismo lenguaje. El uso de Java no ha significado una pérdida de rendimiento considerable ya que la mayoría de los cálculos de optimización lineal recaen sobre el motor de optimización lineal enteramente programado en lenguaje C. En el siguiente apartado se describen, en detalle, las particularidades del sistema de cálculo de las supresiones óptimas que garantizan la confidencialidad de los datos publicados.

En primer lugar, se estudió el uso de diversos optimizadores lineales. La principal condición establecida era que el sistema no obligara a emplear optimizadores lineales comerciales. Esta es una diferencia importante respecto al software existente hasta este momento, como TauArgus [6] que requiere un optimizador lineal de pago como XPress o CPlex para la resolución de problemas CSP.

El optimizador lineal seleccionado para la presente implementación ha sido GLPK 4.43, ver [12]. Entre las razones para su elección se encuentran las siguientes:

- Se trata de un optimizador lineal abierto con interfaces de programación (APIs) en distintos lenguajes, C, C++, Java, etc.
- Soporta distintos sistemas operativos, principalmente aquellos basados en Unix (Linux, Solarix, OSX), Windows 32 y 64 bits.
- Dispone de una API específica para la resolución de problemas empleando el método de branch-and-cut que ha facilitado en gran medida el desarrollo del programa.
- Su rendimiento medio en la resolución de los problemas de confidencialidad planteados ha sido aceptable, como se puede observar en el apartado de resultados computacionales.

2.2 Resultados computacionales

Se han realizado dos estudios del comportamiento de la aplicación. Por una parte, se han empleado las tablas de prueba públicas de la conocida librería "CSPlib", ver [15]. Por otra parte, se ha trabajado con el caso práctico de la publicación estadística relativa al Impuesto de la Renta de la Personas Físicas del año 2008, ver [17].

Los test han sido realizados en un ordenador con procesador Intel core i5 3.2GHz, sistema operativo Windows Vista x32 y 1.5Gb de memoria RAM destinados a la resolución del problema.

2.2.1 Resultados CSPlib

Problema	TauArgus	JCSP
1000	504,93	3223,17
100x100	6,04	435,81
100x100conZ	5,71	442,72
100x2	0,38	0,44
100x20	0,77	4,90
100x3	0,11	0,78
10x10	0,00	0,15
182x60	11,04	4371,11
20x20	0,28	0,16
2x10x10	1,15	12,17
2x2	0,06	0,16
2x2x2	0,05	0,31
2x2x2x2	0,11	0,15
3x3	0,06	0,16
3x3x3	0,22	0,31
3x3x3rb1	0,22	0,31
3x3x3rbv	0,22	0,32
3x3x3x3	11,54	9,12
3x4x5	4,34	1,16
3x4x5rbv	1,32	1,77
3x4x5sinZ	7,64	3,20
50x40	1,54	7,85
50x40conZ	1,26	7,29
5x5x5	4,67	2,16
60x40	1,65	13,64
9x9x9	3.629,26	117861,12
cbs	11,42	4218,06
cox	0,00	0,81
dale	81,23	34947,12
demo1	0,00	0,22
demo2	0,06	0,4
demo3	0,22	0,38
demo4	0,06	0,24
demo5	0,06	0,18
demo6	0,06	0,15
demo7	0,22	0,15
demo8	0,17	0,16
demo9	0,00	0,17
demo10	0,05	0,18
ejemplo	35,92	342,24
ejemplo2	0,54	0,76
freq1	0,00	0,65
freq2	1,04	0,13
lisboa	0,05	0,14
mops2c	0,05	0,18

Problema	TauArgus	JCSP
mops70	0,17	0,27
mops80	0,05	0,12
osorio	5,05	4,24
prueba	0,05	0,14
sdc40x30	2,19	2,65
tabla	0,06	0,2
tabla0	0,06	0,14
tabla3	0,00	0,16
table7	51,74	509,99
table8	2,36	4,81
vb01	0,11	0,9
vb02	0,00	0,19
vb03	0,11	0,21
vb04	1,10	0,51
vb05	1,10	0,29
vb06	0,00	0,32
vb07	0,11	0,19
vb09	0,22	0,88
vb10	0,00	0,14
vb11	0,00	0,13
vb12	0,00	0,2
vb13	0,00	0,17
vb14	0,05	0,43
vb15	0,06	0,25
vb16	1,05	0,28
vb17	0,00	0,18
vb18	0,05	0,2
vb19	0,05	0,49
vb20	0,06	0,14
vb21	0,00	0,15
vb22	0,00	0,14
vb23	0,06	0,15
vb24	0,11	0,25
vb25	0,00	0,16
vb26	0,00	0,2
vb27	0,00	0,14
vb28	0,16	0,33
vb29	0,06	0,24
vb30	0,00	0,15
vb31	0,06	0,37
vb32	0,00	0,38
vb33	0,00	0,19
vb34	0,05	0,22
vb35	0,05	0,14

La tabla anterior muestra los tiempos de ejecución comparando entre el nuevo desarrollo JCSP y los referidos en [15]. Estos últimos fueron obtenidos con un equipo de prestaciones muy inferiores (Pentium III 866MHz) pero con un optimizador lineal de muy alta calidad CPlex 7.0. Los tiempos vienen indicados en segundos y se refieren al tiempo empleado por el programa para conseguir la solución óptima del problema CSP sin incluir el tiempo de cálculo de los heurísticos iniciales.

A pesar de la enorme diferencia entre los equipos sobre los que se han realizado los test es más determinante la elección del optimizador lineal. Para ver hasta qué punto el optimizador lineal puede influir en el tiempo de resolución, se puede consultar una comparativa entre los optimizadores empleados en la resolución de problemas lineales generales en [16].

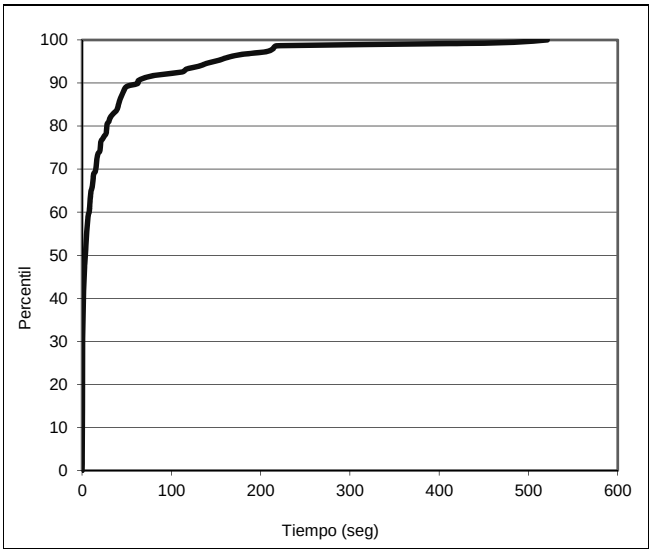
2.2.2 Resultados datos AEAT

En este apartado se muestran los resultados obtenidos para la publicación estadística relativa al Impuesto de la Renta de la Personas Físicas del año 2008. La dimensión de los 184 problemas tratados tiene un rango aproximado de 700 celdas por 300 ecuaciones.

En el siguiente gráfico se muestra cómo el 90% de los problemas se resolvieron en menos de un minuto.

Figura 1

Distribución de tiempos de resolución



En la Figura 2 se puede ver el tiempo de resolución frente al porcentaje de primarios, que contenía el problema, respecto al volumen del mismo (producto del número de celdas por el número de ecuaciones). Se puede concluir que el tiempo de resolución de

los problemas no es directamente proporcional al número de celdas primarias ni al número de ecuaciones que lo componen.

Figura 2

Tiempo de resolución frente al porcentaje de primarios del problema

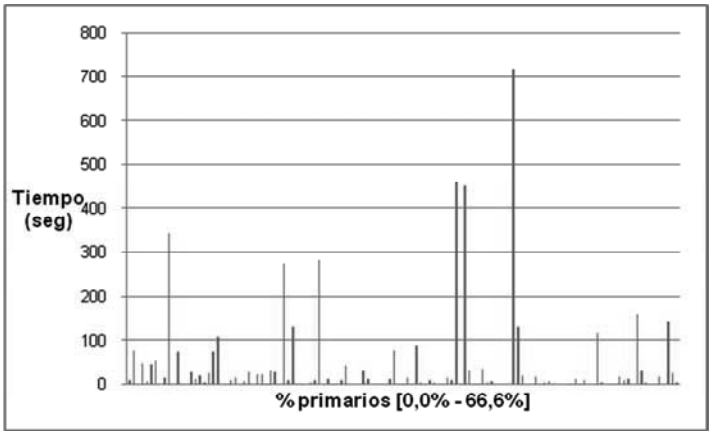
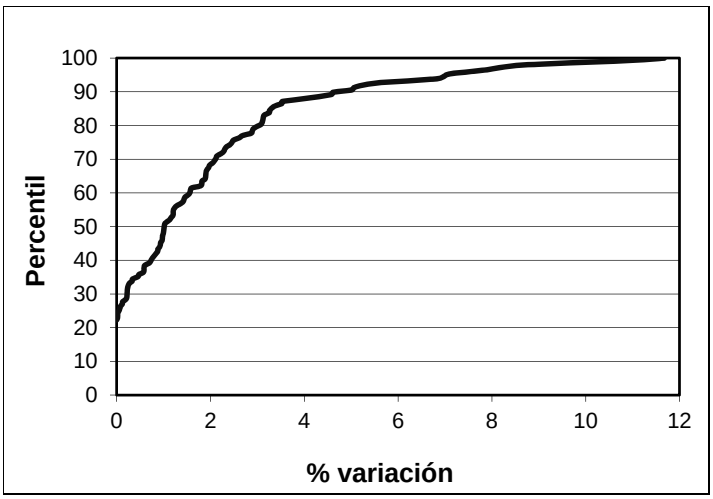


Figura 3

Variación heurístico con solución óptima



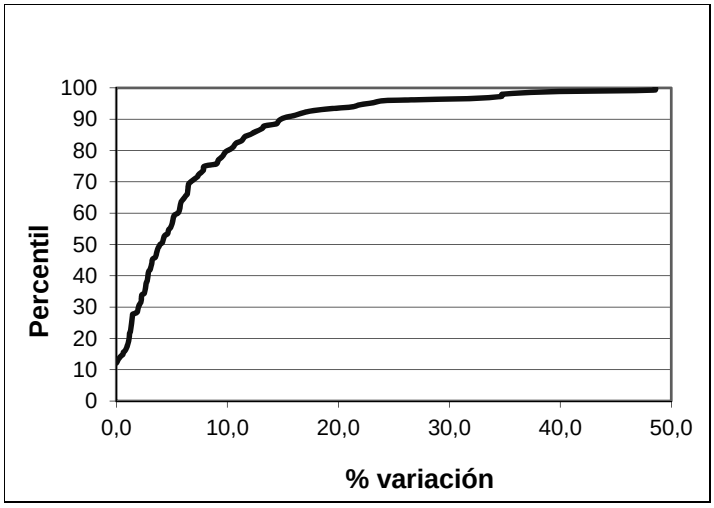
En la Figura 3 se puede observar cómo en el 22% de los casos estudiados el valor de la solución heurística coincide con la solución óptima. Para el 90% de los casos de estudio

el margen de error, de la solución heurística respecto a la solución óptima, se encuentra por debajo del 5%. Para todos los casos el margen de error no supera el 12%.

Después de estos resultados prácticos, podemos concluir que las soluciones heurísticas tienen un grado de confianza alto y el tiempo en encontrarlo es muy rápido en comparación con el cálculo de la solución óptima.

Figura 4

Variación cota inferior y solución óptima



La cota inferior corresponde al valor de la solución al problema relajado (LP, problema sobre los números reales). La solución real es óptima y cumple con todas las restricciones impuestas al problema. Además, esta solución siempre es menor o igual que la solución óptima al problema de programación entera (MIP, problema sobre los números enteros).

Como se explica en el apartado relativo al *Algoritmo empleado en la resolución* del Apéndice 1, el método de ramificación comienza en este punto. Una vez hallada la solución real, comienza el método de branch-and-cut para la búsqueda de solución óptima sobre los números enteros.

En la Figura 4 se observa que para el 12% de los casos de estudio, la cota inferior coincide con el valor óptimo calculado. Esto quiere decir que, para estos casos, la solución real óptima era también la solución entera óptima y; por tanto, no ha sido necesaria ramificación alguna.

3. Integración con el sistema de difusión de estadísticas

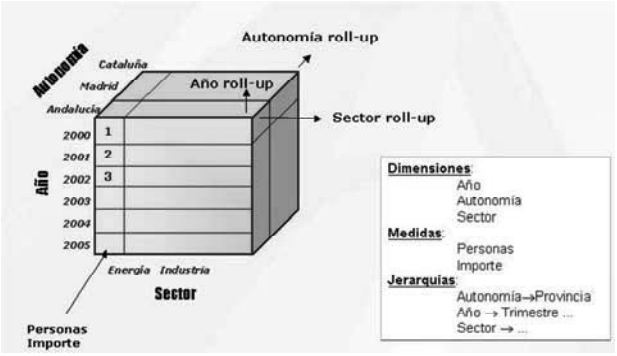
El objetivo de transparencia en la difusión de datos estadísticos afecta, no solamente a la cantidad de información estadística publicada, sino sobre todo, a la calidad de la misma. Para que esta información resulte útil, tanto para el ciudadano como para los distintos grupos de estudio, debe ser publicada en soportes accesibles, basados en estándares abiertos y con técnicas modernas que permitan su estudio y análisis.

3.1 Modelo OLAP de la difusión de estadísticas en la AEAT

El modelo empleado en la difusión de estadísticas de la AEAT se basa en la conceptualización de los datos realizada a través del modelo de datos OLAP (Online Analytical Processing). Bajo este paradigma, los datos individuales o microdatos de cada impuesto (que son el componente principal del DataWarehouse de la AEAT) son depurados y agregados en agrupaciones útiles (también llamados Datamarts) para su estudio o publicación.

Figura. 1

Modelo OLAP de difusión de estadísticas



En el modelo OLAP, los datos se representan mediante hipercubos multidimensionales donde, las dimensiones están definidas por las variables de clasificación, y las celdas contienen los valores de las variables de explotación o medidas.

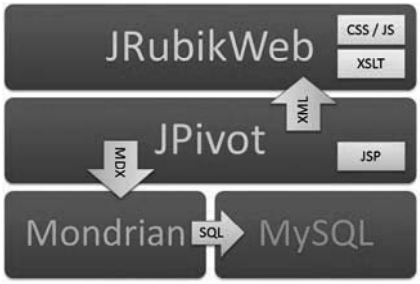
Cada celda de una tabla, en una publicación estadística, corresponde a una celda del hipercubo. Esta celda se puede identificar de forma única, usando para ello el valor que tiene en cada una de sus dimensiones (variables de clasificación) y la medida que representa (variable de explotación).

3.2 Arquitectura del sistema OLAP

En cuanto a la realización informática, el sistema de difusión web consiste en una aplicación desarrollada a medida, ***JRubikWeb***. Esta aplicación está basada en componentes JPivot [8] y utiliza un motor ROLAP de código abierto (Pentaho Mondrian [9]).

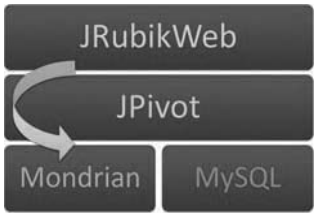
Figura. 2

Arquitectura del sistema OLAP sin tratamiento de la confidencialidad



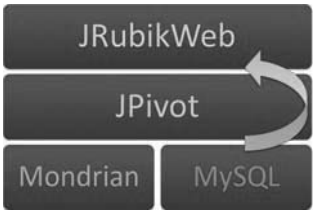
La publicación estadística de la AEAT se organiza siguiendo una estructura arbórea compuesta de hojas y carpetas. Cada hoja muestra una o varias consultas. Cada una de las consultas son visualizadas por distintos componentes; tablas, gráficos, mapas, etc. Dichas consultas están definidas en el lenguaje multidimensional MDX, ver [10].

3.2.1 Flujo de ejecución para la solicitud de datos



- En JRubikWeb se define una consulta en un fichero que contiene una consulta MDX dentro de la etiqueta de JPivot *<jp:MondrianQuery>*.
- JPivot envía la consulta a Mondrian y obtiene los datos para construir la tabla.
- Mondrian traduce la consulta MDX en una consulta sql y solicita los datos a la base de datos MySql.

3.2.2 Flujo de ejecución para mostrar los datos solicitados



- Mondrian, con la respuesta de la base de datos, construye un objeto OLAP que envía a JPivot.
- JPivot con los datos recibidos construye un fichero XML, que contiene los datos para rellenar la tabla o gráfico solicitado, y lo envía a JRubikWeb.
- JRubikWeb, haciendo uso de XSLT, CSS y JavaScript , construye la página HTML que muestra el resultado de una consulta; es decir, la tabla, gráfico, mapa, etc. que vemos en el navegador.

3.3 Confidencialidad en la publicación estadística

Como parte del proceso de publicación de una estadística, se realiza un primer estudio de los problemas de confidencialidad que conlleva la publicación de las variables de clasificación seleccionadas, y de los tramos empleados en las mismas. En ocasiones es necesario reagrupar o cambiar algunos intervalos.

Posteriormente, se definen las consultas de datos que darán soporte a las tablas, gráficos y mapas que compondrán la publicación estadística. Se realiza un segundo estudio de la confidencialidad estadística, teniendo en cuenta, en esta ocasión, el conjunto completo de las consultas que compondrán la publicación. De ello se puede derivar la no publicación de determinadas tablas o la modificación de algunas consultas que muestran mayoritariamente datos confidenciales.

3.4 Integración del procedimiento CSP en la publicación estadística

El proceso de protección de la confidencialidad de una estadística está constituido por las siguientes tareas:

Paso 1: Generación CSP.

– Guardar celdas.

- Determinar qué celdas forman parte de una publicación y cómo identificarlas.
  - Una celda que aparece en la publicación viene determinada por los miembros que representan su valor en cada dimensión (variables de clasificación) y el valor de la medida que se muestra en esa celda (variable de explotación).

- Se almacena en una base de datos la información de las celdas.

- Considerar aquellas celdas que emplean fórmulas basadas en el valor de otras celdas (celdas derivadas: totales, subtotales, ...).
- Establecer los umbrales de protección de cada celda y los parámetros generales de la publicación (por ejemplo, el número mínimo de individuos que permitirá considerar la celda como confidencial).
- Generar los ficheros con la información de las celdas y sus relaciones para el cálculo de las supresiones secundarias óptimas.

Paso 2: Resolución del problema CSP.

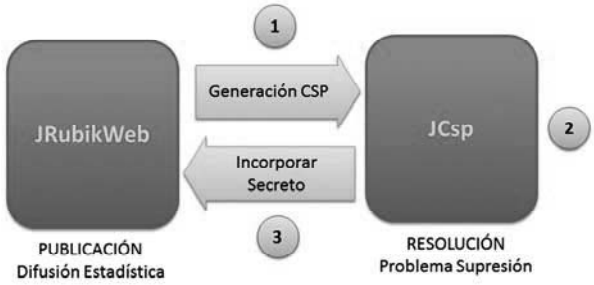
- Calcular las supresiones secundarias para cada una de las medidas sensibles. El resultado de este cálculo es el conjunto de celdas que se deben ocultar en la publicación.

Paso 3: Incorporar las supresiones a la publicación.

- Almacenar en una tabla de la base de datos los resultados obtenidos. El sistema de difusión de estadísticas consultará esta tabla cada vez que vaya a mostrar una celda.
- Controlar la salida de datos en cada una de las tablas publicadas. Dependiendo de su estatus, segura o insegura, se muestra o no. De manera adicional, se pueden suprimir otras celdas asociadas a las celdas ocultadas. Por ejemplo, si tratamos de proteger la variable de explotación "número de declarantes" podemos no querer mostrar otra variable de explotación asociada a ésta, "importe declarado".

Figura 3

Gráfico del procedimiento de integración



A continuación se detalla cómo se han implementado las distintas tareas expuestas en el apartado anterior:

Paso 1: Generación CSP.

- Guardar celdas.
  - Se ha desarrollado una extensión del módulo JPivot para grabar todas las celdas confidenciales de la publicación en una tabla de la base de datos, para su posterior tratamiento. (extensión JPivot: *secretWriter* de la librería *jpSecret*).
    - El mecanismo consiste en recorrer la estadística con una araña web. De este modo se solicitan todas las celdas que componen la estadística y se van guardando. En este punto es importante además procesar las posibles fórmulas que definen cada celda porque que pueden generar nuevas ecuaciones.
- El programa *GeneraCSP*, incluido en la librería *jpSecret*.
  - Recibe como parámetros el nombre de la estadística y la ruta a la base de datos que contiene la tabla con todas las celdas sensibles.
  - Este programa recorre la tabla de celdas, generada en el paso anterior, y crea los ficheros en formato CSPlib (generalmente, uno por cada variable de explotación, a no ser que existan ecuaciones que relacionen varias de estas variables).
  - Estos ficheros servirán como entrada para el programa de control de la confidencialidad estadística. Los ficheros contienen; por una parte, las celdas y por otra, las ecuaciones lineales que las relacionan. En la especificación de las celdas se incluye su identificador, valor, peso, grado de confidencialidad (primario, nulo o publicable), intervalo conocido en el que se encuentra el valor (LB, UB) y sus márgenes de confidencialidad (LPL, UPL y SPL).

Paso 2: Resolución del problema CSP.

- Programa de control de la confidencialidad estadística, *JCsp*.
  - Toma como entrada el fichero en formato CSPlib (o una colección de ellos, uno por cada variable de explotación independiente).
  - Genera un fichero (o colección de ficheros) con el listado de celdas que deben ocultarse en la publicación para cada variable de explotación.

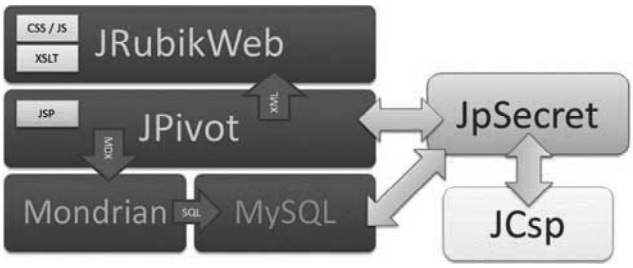
Paso 3: Incorporar las supresiones a la publicación.

- Los resultados son incorporados a una tabla de la base de datos, que almacena, por cada publicación y cubo de datos, las celdas que no deben publicarse. Para llevar a cabo esta grabación se utiliza el programa *CargaResultadosSecreto*.
- Finalmente, se activa otra extensión de JPivot, llamada *hideSecret*, incluido en la librería *jpSecret*, que ocultará las celdas que correspondan a celdas confidenciales o primarias, celdas secundarias y celdas asociadas a una celda confidencial.

3.5 Arquitectura del sistema con protección de la confidencialidad.

Figura 4

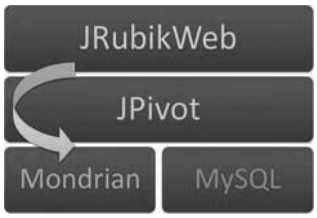
Arquitectura del sistema OLAP con tratamiento de la confidencialidad



Después de la integración del tratamiento de la confidencialidad, la nueva arquitectura del sistema se puede observar en la Figura 4. En rojo y verde aparecen los nuevos módulos, programas o extensiones añadidas al sistema original (JpSecret y JCsp).

3.5.1 Flujo de ejecución para la solicitud de datos con tratamiento de la confidencialidad.

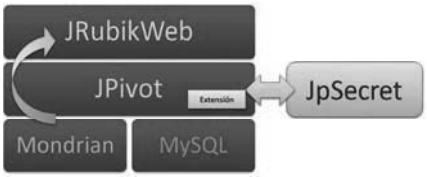
El flujo de ejecución para la solicitud de los datos por parte de la aplicación JRubikWeb no varía con respecto al flujo de ejecución anterior al tratamiento de la confidencialidad; es decir:



- En JRubikWeb se define una consulta en un fichero que contiene una consulta MDX dentro de la etiqueta de JPivot *<jp:MondrianQuery>*.
- JPivot envía la consulta a Mondrian y obtiene los datos para construir la tabla.
- Mondrian traduce la consulta MDX en una consulta sql y solicita los datos a la base de datos MySql.

3.5.2 Flujo de ejecución para mostrar los datos con tratamiento de la confidencialidad

Cuando en el proceso se incluye el tratamiento de la confidencialidad, el flujo de ejecución varía con respecto al establecido en la aplicación original.



- Mondrian, con la respuesta de la base de datos, construye un objeto que envía a JPivot.
- JPivot, con los datos recibidos, construye un fichero XML que contiene los datos para rellenar la tabla solicitada y lo envía a JRubikWeb.
  - Si la extensión de guardar celdas está activada
    - Se conecta directamente a la base de datos y guarda las celdas que contengan una medida sensible
  - Si la extensión que oculta celdas confidenciales está activada
    - Modifica el xml de respuesta incluyendo información en la tabla que indica si una celda contiene una medida que se debe ocultar
- JRubikWeb haciendo uso de XSLT, CSS y JavaScript construye la página HTML que muestra el resultado de una consulta; es decir, la tabla, gráfico, mapa, etc. que vemos en el navegador.
  - La transformada que construye la tabla se ha modificado para que utilice la nueva información incluida para el secreto estadístico

5. Conclusiones y futuras mejoras

5.1 Conclusiones

El tratamiento de la confidencialidad en la publicación de estadísticas de la AEAT ha sido satisfactorio haciendo uso de las herramientas descritas en este documento. El desarrollo se puede considerar que ha sido un éxito puesto que los tiempos de cálculo de las supresiones secundarias para estas publicaciones son razonables y, además, se han conseguido reducir tanto los plazos temporales como los recursos necesarios para la elaboración de las publicaciones.

En la comparativa de la aplicación con la herramienta tauArgus, comprobamos que la velocidad de resolución de la nueva aplicación es netamente inferior, principalmente por

el uso de un optimizador lineal con capacidades inferiores al utilizado por tauArgus. Sin embargo, estos tiempos son totalmente aceptables para el uso al que están destinados y la utilización de un optimizador de código abierto reduce considerablemente el coste de la producción de las publicaciones.

La integración de la herramienta, con el sistema OLAP de difusión, ha supuesto un gran ahorro de tiempo, tanto en la generación de los problemas CSP asociados a las publicaciones, como en la incorporación de las supresiones que garantizan la confidencialidad de los datos de dichas publicaciones.

5.2 Futuras mejoras

5.2.1 Resolución del problema lineal

Entre las mejoras del algoritmo de resolución del problema de optimización, asociado a la supresión de celdas, se encuentra el fortalecimiento de las restricciones de capacidad y el uso de desigualdades de cobertura (lifted cover inequalities). También se plantea la incorporación de desigualdades de eliminación de puentes. Para una descripción completa sobre estas desigualdades ver [3].

En cualquier caso, se trata de crear restricciones de capacidad mejoradas que aceleren la convergencia del problema. La mayoría de los optimizadores lineales actuales incluyen sistemas de fortalecimiento de las restricciones de un problema. En el caso de GLPK, el optimizador empleado, se espera que en las siguientes versiones se implemente el uso de las cover inequalities para la resolución del problema.

El algoritmo que se ha descrito emplea un heurístico general al comienzo del proceso de branch-and-cut, después de haber realizado el preproceso. Este heurístico se emplea con carácter general para todos los nodos. Es posible incluir el cálculo de un heurístico en cada nodo del algoritmo iterativo de branch-and-cut, tal y como se describe en [3]. Esto permite eliminar ramas cuyos valores óptimos de las funciones objetivo superan al heurístico del nodo del que dependen.

5.2.2 Optimizador lineal

El sistema ha sido diseñado sobre el optimizador lineal GLPK cuyo rendimiento ha sido suficiente para el uso en la difusión estadística de la AEAT pero que tiene algunas limitaciones en las mejoras de optimización y problemas de rendimiento. Para ver una comparativa del rendimiento ver en enlace [7].

Una posible mejora podría ser la migración desde las llamadas al API del optimizador GLPK al API de Coin-OR. El motivo de este cambio sería que dicho interfaz permite el uso de múltiples optimizadores lineales, abiertos o comerciales, que podrían ofrecer un mejor rendimiento y funcionalidad que GLPK en algunos casos.

El principal problema en este punto es la compatibilidad de las rutinas del método branch-and-cut del optimizador GLPK que han sido usadas en el desarrollo.

5.2.3 Integración con el sistema de difusión

La versión actual tiene algunas limitaciones en las fórmulas que es capaz de analizar (por ejemplo, fórmulas de celdas del cubo) sería deseable cubrir todo el espectro MDX, siempre que las fórmulas sean lineales.

El sistema de control de la confidencialidad estadística es una colección de programas que deben ejecutarse, de manera secuencial, seleccionando determinadas opciones. Existe la posibilidad de integrarlas en una sola herramienta que resultaría más sencilla para el usuario.

5.3 Agradecimientos

Nos gustaría agradecer el apoyo de la subdirección de Estadísticas del SETE y la confianza puesta en el equipo de trabajo. También la importante aportación de Juan José Salazar en la explicación del modelo teórico en el que se basa el programa de control de la confidencialidad estadística.

Apéndice 1. Modelo entero del problema de supresión de celdas

A continuación, se muestra el problema de la supresión de celdas y su modelización como problema de optimización lineal.

Finalmente, se detalla el método de resolución del problema entero que permitirá obtener la supresión óptima; es decir, aquella que garantiza la confidencialidad con la mínima pérdida de información.

Modelo entero

A continuación se realiza una descripción formal del problema de supresión de celdas (CSP). En esta sección, se introduce el modelo matemático de programación lineal entera expuesto en detalle en el artículo de Fischetti y Salazar [3].

Los elementos de partida para el planteamiento del problema son los siguientes:

- Colección de celdas que se desea publicar.
- Serie de restricciones lineales que satisfacen las celdas. Éstas vienen fijadas por los marginales, subtotales, tablas enlazadas u otras operaciones entre celdas.
- El intervalo en el que se encuentran los valores publicados ($[lb_i, ub_i]$).

Matemáticamente estas relaciones se pueden expresar como:

$$\left.\begin{array}{l} \sum_{i=1}^{n} M_{ij}\, y_j = b \\ lb_i \le y_i \le ub_i \quad \forall i \in \{1, \ldots n\} \end{array}\right\}$$

donde,

i es el número de variables (celdas)

j es el número de ecuaciones del problema

M_{ij} es la matriz que expresa las relaciones aritméticas de las variables

y_j corresponde al valor de la celda i en la publicación,

b_i son los marginales y

lb_i, ub_i define el intervalo conocido en el que se encuentra el valor de la celda.

Para la resolución del problema CSP se introduce una variable de decisión binaria por cada celda publicada; es decir, $\boldsymbol{x_i} \in \{0,1\}$ para todo ***i*** desde 1 hasta el número de celdas. Si en la solución del problema CSP su valor es cero, la celda será publicada; en caso contrario, la celda debe ser suprimida.

Debe definirse el ***umbral de confidencialidad estadística***[7] que determina qué cedas deben suprimirse (supresiones primarias) y debe determinarse si el valor cero será ocultado o no. Así mismo, se establecen los niveles de protección por cada celda. Se trata del valor máximo deducible[8] de la celda, el mínimo y la diferencia entre ambos (estos valores son denominados *lpl, upl, spl*; y provienen de los términos ingleses lower protection level, upper protection level y slide protection level).

En las publicaciones estadísticas del SETE se opta por mostrar las celdas con valor cero y el intervalo de protección de cada celda (lpl, upl) se define desde el valor uno hasta umbral de confidencialidad elegido.

Para representar el problema CSP asociado a la búsqueda de las supresiones secundarias que minimizan la pérdida de información manteniendo la confidencialidad existen exencialmente dos modelos de programación entera.

El primer modelo, llamado modelo clásico, propuesto por Kelly (1990), expresa el problema CSP como un problema de programación lineal entera (ver [3] para una estimación del tamaño del problema según el tamaño de las tablas, el número de celdas marginales y el número de supresiones primarias). Aun tratándose de un problema de programación lineal relajado[9], implica un número muy alto de variables auxiliares y de restricciones asociadas que hacen en la práctica inviable la resolución del problema, en un tiempo razonable, utilizando los optimizadores lineales con los que se realizaron pruebas: GLPK, CPLEX, XPress...

El segundo planteamiento, propuesto por Fischetti y Salazar detalladamente en [3], lleva al siguiente problema de programación entera:

[7] El umbral de confidencialidad es un número entero que marca el límite por debajo del cual una celda, que represente un número de declarantes, se considera sensible y no se debe mostrar en la publicación.

[8] Si no se introducen determinadas supresiones secundarias, a partir de los marginales, por ejemplo, publicados es posible desvelar datos que hayan sido ocultados por cuestiones de confidencialidad.

[9] Relajar el problema consiste en trabajar con números reales en lugar de con números enteros. Por lo tanto; para nuestro caso, las variables binarias $xi \in \{0, 1\}$ pasan a cumplir la siguiente condición: $0 \leq xi \leq 1$.

$$\min \sum_{i=1}^{n} w_i \, x_i \qquad [2]$$

donde los elementos de esta función objetivo son:

- $\boldsymbol{x_i}$ son las variables de decisión que definen si la celda i se oculta o se publica.
- $\boldsymbol{w_i}$ establece el coste en información de ocultar la celda i. En el contexto de las publicaciones de la AEAT este valor es idéntico al número de declarantes que representa. En ocasiones también puede establecerse como los importes declarados por los individuos.

Las restricciones que se imponen a la anterior función objetivo son los niveles de protección de cada celda confidencial (supresión primaria). En el caso de un único atacante externo estos niveles (lpl, upl, spl), pueden establecerse como un problema lineal para cada celda.

Por ejemplo, la condición del límite superior (upl), para cada celda i_k, se corresponde con el siguiente problema lineal:

$$\left.\begin{array}{l} \overline{y}_{i_k} = \max y_{i_k}, \\ \text{Sujeto a} \\ M_y = b, \\ \quad y_i \le a_i + UB_i x_i \quad \forall i \in \{1,...n\} \\ \quad -y_i \le -a_i + LB_i x_i \quad \forall i \in \{1,...n\} \end{array}\right\} \qquad [3]$$

Si el valor del óptimo, $\overline{y}_{i_k}$ es superior al valor de la celda $i_k + upl_{i_k}$ la celda confidencial se encuentra protegida para las supresiones, $\mathbf{x_i}$, propuestas. En caso contrario, el conjunto de supresiones no cumple los niveles de protección superior requeridos para la celda i_k. Si no se cumplen los niveles de protección se deberá añadir, al problema inicial, una restricción adicional que garantice la confidencialidad exigida. Estas restricciones adicionales se denominan restricciones de capacidad (*capacity constraints*) y posteriormente, en el apartado relativo al *Algoritmo empleado en la resolución*, se detallará cómo se calculan.

El tipo de problema definido en (3) es conocido como *subproblema del atacante*. El número de estos problemas crece en el caso de considerarse múltiples atacantes externos.

Los problemas que representan los límites inferior (lpl) y de tamaño de intervalo (spl), para la celda i_k, son análogos al expuesto anteriormente, ver [3].

El problema completo CSP consiste, por lo tanto, en optimizar la función objetivo (2), sujeta a las restricciones definidas en (1) y a las restricciones adicionales derivadas de los niveles de protección definidos por los subproblemas del atacante (3) para cada celda confidencial.

En definitiva, el modelo entero que representa al problema CSP es un modelo binivel entero, con un número exponencial de restricciones lineales establecidas para garantizar los niveles de protección requeridos.

Algoritmo empleado en la resolución

El proceso de resolución que se expone en este apartado, consiste en obtener una solución que optimice la función objetivo definida en (2); es decir, encontrar aquel conjunto de celdas a suprimir que garanticen la mínima pérdida de información. Posteriormente, se intenta vulnerar la confidencialidad de la solución generada y se comprueban cuáles son las restricciones de confidencialidad que un atacante externo puede encontrar para el máximo, el mínimo y el intervalo en el que se encuentra una celda. Si dichos valores cumplen los niveles de protección exigidos, para todas y cada una de las celdas consideradas supresiones primarias, el problema ha finalizado; en caso contrario, las restricciones que no satisfacen los niveles de protección sirven para definir nuevas condiciones que se deben añadir al problema y repetir el proceso.

La solución de problemas binivel se suele obtener empleando un esquema iterativo en el que se van añadiendo planos de corte. Este procedimiento se denomina método de ramificación y poda (Branch-and-Cut) ver [13]. Aplicando este procedimiento de ramificación, se llega a conseguir la solución del problema binivel en un tiempo polinómico.

El algoritmo se inicia con un problema principal (***problema master***) definido por la siguiente función objetivo:

$$\min\left\{\sum_{i=1}^{n} w_i x_i : x_{i_1} = \mathrm{L} = x_{i_p} = 1, x \in [0,1]^n\right\} \qquad [4]$$

donde:

- $\boldsymbol{x_i}$ son las variables de decisión que definen si la celda i se oculta o se publica.
- $\boldsymbol{w_i}$ establece el coste en información de ocultar la celda i.

A la que se imponen las siguientes condiciones:

- Las celdas consideradas supresiones primarias; es decir, su valor está por debajo del umbral elegido, se suprimen de la publicación. Esto se expresa matemáticamente:

$$x_{i_1}, \ldots, x_{i_p} = 1$$

- Por cada ecuación en la que participe una celda de supresión primaria, se debe suprimir además al menos otra celda (supresión secundaria).

Proceso de ramificación

La resolución del problema inicial sobre los enteros (en nuestro caso variable binaria 0,1) tiene una mayor complejidad que su resolución sobre los reales (problema relajado). Por este motivo; en primer lugar, se encuentra la solución real del problema que cumple todas las restricciones de confidencialidad y; posteriormente, se va ramificando la solución real.

El proceso de ramificación (fase Branch) consiste en asignar un valor 0 ó 1 a cada elemento de la solución real descartando ramas de forma inteligente. Todas las posibles ramas representan las diferentes combinaciones de valores enteros (binarios en nuestro caso) cercanos a la solución real.

Proceso de poda

A continuación, se detallan los pasos a seguir para añadir restricciones al problema (fase Cut). Este proceso se repite en cada una de las ramas incluso en la rama principal del árbol constituida por el problema inicial.

En primer lugar, se obtiene una solución óptima, $\boldsymbol{x}^*$, del problema inicial y se comprueba si las supresiones propuestas, $\boldsymbol{x}^*$, cumplen los requisitos de confidencialidad definidos:

- En caso afirmativo, el problema está resuelto y tenemos una solución con pérdida mínima de información que cumple los criterios de seguridad en la publicación.
- En caso contrario, tenemos un problema de separación asociado a las restricciones de capacidad. Este problema se resuelve siguiendo el algoritmo que se expone a continuación.

Por cada supresión primaria, se realizan estos pasos:

1) Se soluciona el subproblema del atacante, para el nivel de protección superior (upl) similar a (3), y se chequea si se cumple la restricción de confidencialidad superior para la celda:

   a) En caso afirmativo, continuamos con el siguiente paso.

   b) En caso negativo, generamos una restricción de capacidad, a partir de la condición violada por la solución propuesta $\boldsymbol{x}^*$. Esta restricción se obtiene a partir de la solución del *problema dual al subproblema del atacante* (4) que incumple la protección exigida.

      El dual del subproblema del atacante se formula en el siguiente problema lineal:

$$\left.\begin{array}{l} \bar{y}_{i_k} = \min \Upsilon^t b + \sum_{i=1}^{n} \left(\alpha_i \left(a_i + UB_i x_i \right) - \beta_i \left(a_i - LB_i x_i \right) \right), \\ \text{Sujeto a} \\ \alpha^t - \beta^t + \Upsilon^t M = e_{i_k}^t \\ \alpha \geq 0, \beta \geq 0, \Upsilon \quad \text{sin restricción en el signo} \end{array}\right\} \quad [5]$$

La nueva restricción de capacidad generada tiene la forma:

$$\sum_{i=1}^{n} \left(\alpha_i \, UB_i + \beta_i \, LB_i \right) x_i \geq UPL_k$$

Para todos los puntos extremos (α, β, γ) que satisfacen el problema dual (5) al subproblema del atacante.

2) Se soluciona el subproblema del atacante para el nivel de protección inferior (lpl) y se chequea si cumple todas las restricciones de confidencialidad. Se procede del mismo modo que en el anterior apartado:

   a) En caso afirmativo, continuamos con el siguiente paso.

   b) En caso negativo, generamos una restricción de capacidad violada por la solución propuesta x^* de la forma:

$$\sum_{i=1}^{n} \left(\alpha_i \, UB_i + \beta_i \, LB_i \right) x_i \; ³ \; LPL_k$$

   Para todos los puntos extremos (α, β, γ) que satisfacen el problema dual al subproblema del atacante asociado a la restricción de confidencialidad inferior (lpl).

3) Empleando las soluciones de los problemas del atacante anteriores se chequea el nivel de protección del intervalo de confidencialidad (spl):

   a) En caso afirmativo, continuamos con el siguiente paso.

   b) En caso negativo, generamos una restricción de capacidad violada por la solución propuesta x^*. Esta restricción tiene la forma:

$$\sum_{i=1}^{n} \left(\left(\alpha_i + \alpha_i' \right) UB_i + \left(\beta_i + \beta_i' \right) LB_i \right) x_i \; ³ \; SPL_k$$

   Para todos los puntos extremos $(\alpha, \alpha', \beta, \beta', \gamma, \gamma')$ que satisfacen los problemas duales a los subproblemas del atacante anteriores.

4) Después de haber recorrido todas las celdas primarias, se comprueba si se han añadido nuevas restricciones de capacidad:

a) En caso negativo, la solución propuesta x*, es la supresión que buscamos y cumple todas las restricciones de confidencialidad.

b) En caso de haberse generado alguna nueva restricción debemos realizar una nueva optimización del problema inicial. De este modo se generará una nueva solución **x**** a partir de la cual se repetirá el proceso.

Optimización del algoritmo

El algoritmo incluye mejoras basadas en el fortalecimiento de las restricciones del problema lineal. Principalmente, se trata de desigualdades de cobertura, eliminación de desigualdades puente, etc. Algunas de estas mejoras son directamente implementadas por el optimizador lineal empleado y otras han sido añadidas en el desarrollo (para una exposición detallada de estas técnicas consultar el artículo [3]).

Para aumentar la velocidad de resolución del problema se realiza un preproceso del mismo, en el que se eliminan del problema celdas primarias que obtienen protección automáticamente del resto de celdas primarias. De esta manera, se consigue reducir la dimensión del problema.

Uso de soluciones heurísticas

La velocidad de convergencia del algoritmo iterativo de branch-and-cut puede incrementarse si se halla antes una solución heurística del problema. Esta solución, generalmente distinta de la óptima, se emplea para desestimar ramas del árbol de decisión.

El heurístico implementado se basa en las ideas de Nelly y Robertson [4], [5], tal y como se describe en [3].

El procedimiento se ejecuta por pasos:

- Se comienza con un conjunto de supresiones inicial que coincide con las supresiones primarias.
- Para todas las celdas con supresiones primarias:
  - Se encuentra un conjunto de supresiones que garantizan los niveles de protección para cada celda; upl, lpl y spl (resolviendo los problemas del atacante correspondientes).
  - Las nuevas supresiones se añaden al conjunto inicial. Por lo tanto, para cada nueva celda primaria analizada lo normal es que se haya ampliado el conjunto de supresiones secundarias.
- Se realiza un procedimiento de limpieza de las supresiones halladas para eliminar redundancias dado que algunas supresiones pueden sobreproteger celdas que ya se encontraban protegidas. De forma iterativa se intenta eliminar las supresiones redundantes de mayor peso.

Referencias

[1] «Ley de la Función Estadística Pública (9 de Mayo de 1989)», Título 1, Capítulo III "Del secreto estadístico".

[2] WILLENBORG Y DE WALL (1996), «Statistical Disclosure Control in practice», *Lecture notes in Statistics,* 111 Springer. New York.

[3] FISCHETTI Y SALAZAR (2001), «Solving the cell suppression problem on tabular data with linear constraints», *Management Science*.

[4] ROBERTSON D.A. (1995), «Cell Suppression at Statistics Canada», *Proc. Second International Conference in Statistical Confidentiality*

[5] KELLY J.P., GOLDEN L.A. (1992), "Cell Suppression Disclosure protection for sensitive tabular data".

[6] *TauArgus, NuArgus,* http://neon.vb.cbs.nl/casc/

[7] *Comparativa optimizadores*, http://scip.zib.de/

[8] JPIVOT, http://sourceforge.net/projects/jpivot/

[9] MONDRIAN, http://mondrian.pentaho.com/documentacion/index.php

[10] *Lenguaje MDX*, http://en.wikipedia.org/wiki/MultiDimensional_eXpressions

[11] JRUBIK, http://sourceforge.net/projects/rubik/

[12] *GLPK (GNU Linear Programming Kit),* http://www.gnu.org/software/glpk/glpk.html

[13] NEMHAUSER, G. L., L. A. WOLSEY, (1988), «Integer and combinatorial optimization», *John Wiley & Sons*.

[14] «Reglamento (CE) No 223/2009 del Parlamento Europeo y del Consejo de 11 de marzo de 2009», Capítulo V, Art. 20.

[15] *Librería de test CSPlib*, http://webpages.ull.es/users/casc/#CSPlib:

[16] *Comparativa optimizadores*: http://www.coin-or.org/GAMSlinks/benchmarks/MIP/allSolver_080601/1-CPLEX-1.trc4-GLPK-1.trcsqr.htm

[17] «Estadística sobre el IRPF año 2008» http://www.agenciatributaria.es/AEAT/Contenidos_Comunes/La_Agencia_Tributaria/Estadisticas/Publicaciones/sites/irpf/2008/home.html