CONJUNTOS ESPECIALES EN PROGRAMACION MATEMATICA

L.F. ESCUDERO

Se describen las nociones fundamentales y operativa de ciertos conjuntos especiales en programación matemática, en concreto los conjuntos S1, S2, SOS1, SOS2, LS1 y LS2 tal que permiten tratar funciones no-lineales obteniendo un óptimo que también es global. Se analizan las fase y seudo-fase branch and - and bound en que se basa su operativa y se describen algunos campos de aplicación.

1. INTRODUCCION Y CAMPO DE APLICACION

La noción de special ordered sets fue introducida por Beale y Tomlin /6/ y desarrollada por Tomlin /20/ y Forrest, Hirst y -- Tomlin /17/. Beale y Forrest /4/ introducen modificaciones muy sustanciales sobre todo al determinar los pseudo-shadow-costs D^+ y D^- de los special ordered sets. Ver también Beale /2/ y /3/ y Land y Powell /19/ entre otros muchos trabajos de este equipo de investigación. Los special ordered sets son - de dos tipos. En un problema de programación matemática se denominará special ordered -- set de tipo 1, o conjunto S1, al conjunto - de variables no-negativas en el cual sólo - una puede tomar un valor distinto de cero. Se denominará special ordered set de tipo 2, o conjunto S2, al conjunto de variables no-negativas en el que se permite que haya hasta dos variables distintas de cero, pero -- con la condición de que éstas sean consecutivas. Si los conjuntos S1 ó S2 tienen la condición adicional de que la suma de las - variables que forman el conjunto debe ser - igual a 1, entonces estos conjuntos se denominarán SOS1 (Benichou et al., /9/, Escudero /11/, Gauthier y Ribiere /18/) y SOS2 respectivamente. En el caso de SOS1, las variables que integran el conjunto deben ser binarias. Se puede apreciar que el orden en - que las variables aparecen en el conjunto - S2 afecta al significado del mismo. El orden también afecta a la forma en que el algoritmo trata los conjuntos S1; si no pueden ordenarse de una forma lógica, no sería útil tratar estas variables como conjuntos S1.

La introducción de los conjuntos S1 y S2 en los algoritmos generales de programación lineal (PL) o, en concreto, de programación - lineal entera (PLE) (ver Escudero, /12/) ha incrementado fuertemente la aplicabilidad - de los mismos, constituyendo una de sus innovaciones más importantes. El campo de -- aplicación de los conjuntos S1 se encuentra en los casos de selección de una alternativa entre varias posibles: construir una -- planta o no, decidir una entre varias localizaciones, entre varios procesos, etc. (Es el caso en que $\Sigma X_j = 1$; $X_j \in \{0,1\}$) o e.g. - decidir entre varias capacidades de una -- planta, etc., tal que si una variable $X_j>0$, el resto tienen que ser nulas, pero sin exigir que el valor positivo de la variable -- sea uno, sino que puede tener cualquier valor contínuo positivo, con o sin límite. -- Asimismo, el campo de aplicación de los conjuntos S1 se encuentra en los denominados - problemas con estructura especial: set packing, partitioning y knapsack binario (Escudero, /12/), así como en programación separable no-convexa (Escudero, /13/ y /14/). - El campo de aplicación de los conjuntos S2 se encuentra fundamentalmente en la programación separable no-convexa obteniendo el - óptimo global en problemas no-convexos (Escudero, /13/ y /14/), en ciertos problemas no-lineales, para los que también obtiene - el óptimo global, y en problemas multiperíodo (e.g. en los que una determinada activi-

- L.F. Escudero de IBM, S.A.E., Paseo de la Castellana, 4. Madrid 1.
- Artículo recibido en Enero de 1978.

dad sólo se puede producir en un período o a lo sumo en dos, pero siendo éstos consecutivos).

2. CONJUNTOS ESPECIALES SOS1

Para ilustrar la forma en cómo operan los métodos branch-and-bound con los special ordered sets, nos vamos a basar en los conjuntos SOS1 (conjunto de variables binarias entre las cuales una y sólo una debe valer 1). Barceló y Companys, /7/, presentan una panorámica de las técnicas branch-and-bound. Ver también Escudero /11/ y /12/ y Escudero y Vázquez-Muñiz /15/ y /16/.

Si la determinación de la capacidad de una unidad de producción está limitada, por razones técnicas, a la elección de, por ejemplo, 750, 500, 250, 100 y 0 (es decir, no se construye la unidad), se puede efectuar este planteamiento afectando a cada posible capacidad de producción una variable binaria de tal forma que $Y_1=1$ si se selecciona el tamaño $W_1=750$ y en caso contrario $Y_1=0$; $Y_2=1$ si se selecciona el tamaño $W_2=500$ y en caso contrario $Y_2=0$; $Y_3=1$ si se selecciona el tamaño $W_3=250$ y en caso contrario $Y_3=0$; $Y_4=1$ para la elección $W_4=100$ y en caso contrario $Y_4=0$; $Y_5=1$ para $W_5=0$, es decir, si no se construye la unidad, y en caso contrario $Y_5=0$. Evidentemente $Y_1+Y_2+Y_3+Y_4+Y_5=1$. Se denomina conjunto SOS1 a este conjunto de variables, siendo variables SOS1 cada una de las variables Y_j que integran el conjunto. En cada conjunto SOS1, a cada variable le corresponde un <u>peso (weight) o ponderación</u>. En el ejemplo anterior, el peso de cada variable Y_j podría ser la capacidad afectada W_j, de tal forma que la condición $X=750\ Y_1 + 500\ Y_2 + 250\ Y_3 + 100\ Y_4 + 0\ Y_5$ (que recoge la capacidad seleccionada en la unidad de producción) sería la condición ponderativa de acuerdo con el orden decreciente de sus ponderaciones. Esta ordenación es muy importante en el tratamiento especial que reciben las variables SOS1 en la fase branch-and-bound de un problema de programación lineal entera mixta. Es preciso considerar que la misma variable puede pertenecer a más de un tipo de conjunto SOS1. El tiempo de CPU que se ahorra con este tratamiento especial es considerable.

El campo de aplicación de los conjuntos SOS1 es muy extenso (tal como hemos indicado para los conjuntos S1) ya que es muy frecuente que en un problema de optimización haya la posibilidad de una elección múltiple, tal que una alternativa excluya las otras. Así ocurre en la planificación de la producción, en problemas de inversiones, en problemas de distribución, en problemas de localización de plantas y almacenes, en problemas multiperíodo en los que si se toma una decisión en un período se excluye esa misma alternativa en los demás, en problemas de asignación de trabajos a tareas, de órdenes a factorías, en problemas de selección de rutas óptimas, etc. Siempre que, ante diversas alternativas, una de ellas excluya las demás se puede plantear el problema como un conjunto SOS1.

Se recogen a continuación las diversas posibilidades de ponderar un conjunto SOS1, de bifurcar sobre un conjunto SOS1, de obtener sus correspondientes seudo-costos, y de incidir en el orden de prioridad junto con el resto de las variables enteras para obtener la próxima variable o conjunto SOS1 sobre qué bifurcar.

<u>Ponderación de las variables SOS1</u>

Una vez que un conjunto SOS1 ha sido seleccionado, en el orden de prioridad (ver más abajo) entre los demás conjuntos SOS1 y variables enteras, para efectuar sobre él la correspondiente bifurcación y así obtener a partir del nudo <u>k</u> los dos nudos sucesores (n+1) y (n+2), adquiere especial importancia la ponderación atribuida a cada variable SOS1.

Esta ponderación debe depender del grado de influencia que la variable SOS1 tenga en el modelo. Así, en el caso anterior la variable Y_1 tiene más importancia que, por ejemplo, la variable Y_4, ya que está afectada a una capacidad de producción superior ($W_1=750$) con la consiguiente influencia (ya sea $Y_1=1$ ó 0) en la función objetivo. Por tanto, una buena ponderación de Y_1 podría ser W_1 y W_4 la correspondiente a Y_4.

Hay, básicamente, tres formas de indicar la ponderación de las variables SOS1. Primero, indicar qué condición del problema será <u>condición ponderativa</u> de un determinado conjunto o condición SOS1. Es el caso recogido en el párrafo precedente. Así, la ponderación de cada variable Y_j es precisamente la capacidad W_j. Segundo, atribuir explícitamente un peso, a cada variable S0S1. Como la ponderación es utilizada por la fase branch-and-bound (ver más abajo) para bifurcar sobre la condición S0S1 de que forma parte la variable S0S1 a la que se está atribuyendo un peso, es preciso adecuar este peso a la importancia de la variable en el modelo, ya que en el tratamiento de una condición S0S1 se clasifican las variables S0S1 de acuerdo con el orden decreciente de estos pesos. -- Tercero, si el usuario no ha utilizado ninguna de las dos formas anteriores de indicar la ponderación de las variables S0S1, - el código de PLE supone generalmente que, - en una condición S0S1, se han introducido - las variables S0S1 en la matriz del modelo en orden decreciente de importancia, asignándoles la secuencia de pesos 0, 1, 2, 3, ..., de tal forma que el peso 0 corresponde a la variable menos importante, que será la introducida en último lugar, el peso 1 corresponde a la variable introducida en la - matriz en el anteúltimo lugar, etc.

Bifurcación sobre una condición S0S1

Cuando en un problema coinciden condiciones S0S1 y variables enteras, los pasos del método de branch-and-bound siguen siendo los mismos, excepto los referentes a la elección de la variable sobre qué bifurcar y a la definición de los subproblemas (n+1) y (n+2). A todos los efectos, un conjunto S0S1 no es considerado como un conjunto de variables, sino como una entidad. Si la variable a seleccionar es una condición S0S1, la bifurcación de la misma se efectúa como sigue. Sea el ejemplo recogido anteriormente en el -- cual

$$\sum_{j=1}^{5} Y_j = 1 \ \underline{\text{(convexity row)}} \qquad (1)$$

$$X = \sum_{j=1}^{5} W_j Y_j \ \underline{\text{(reference row)}} \qquad (2)$$

Como Y_j es binaria (0,1), la solución entera exige que el valor de X sea discreto tomando el valor W_j correspondiente a Y_j=1. - En el nudo seleccionado como nudo a bifurcar <u>k</u>, la condición S0S1 vale 1 pero si alguna variable Y_j es distinta de 0 ó 1, significa que la variable X no ha tomado valores discretos. Sea por ejemplo, $Y_1=0{,}2$; $Y_2=0$; -- $Y_3=0{,}3$; $Y_4=0{,}5$ e $Y_5=0$. El valor de X será - $\bar{X}=0{,}2 * 750 + 0 * 500 + 0{,}3 * 250 + 0{,}5 * 100 + 0 = 275$. Si <u>r</u> es el subíndice para el que $W_r > \bar{X} > W_{r+1}$, aquí será r=2 ya que $W_2=500 > 275 > W_3=250$. Por tanto, la bifurcación sobre la variable X se efectuará considerando que $X \geq 500$ o bien $X \leq 250$, lo -- que en el primer caso equivale a indicar -- que $Y_5=Y_4=Y_3=0$, y en el segundo $Y_1=Y_2=0$.

Como resultado de esta dicotomía se generan dos nuevos nudos (n+1 y n+2) sucesores del nudo <u>k</u>. Tienen las mismas función objetivo, restricciones y limitaciones de las variables que integran la condición S0S1.

En un subproblema las variables Y_3, Y_4 e Y_5 se han fijado a cero, y en el otro subproblema las variables fijadas a cero son Y_1 e Y_2. Por tanto, en una rama se han efectuado simultáneamente tres bifurcaciones y dos en la otra rama, en comparación con las que se hubieran efectuado con variables enteras -- normales. Se puede observar la importancia que tienen las ponderaciones, ya que si éstas hubiesen sido distintas, la dicotomía - se hubiese efectuado de una forma diferente.

Seudocostos de una condición S0S1

La noción y la utilidad de los seudocostos de una condición S0S1 son análogas a las de cualquier variable entera. Dado que una condición S0S1 está considerada como una sola entidad, no tienen significado los posibles seudocostos de una variable S0S1. Ver Benichou et al. /9/, Escudero /11/ y sobre todo Gauthier y Ribiere /18/.

Al bifurcar sobre una condición S0S1 una bifurcación se forma haciendo cero $Y_1, Y_2, \ldots, Y_r$ y en la otra bifurcación se anulan $Y_{r+1}, Y_{r+2}, \ldots$ Por tanto, al bifurcar entre r y r+1 resultan los siguientes seudocostos inferior $\left(PCL_i^{(r)}\right)$ y superior $\left(PCU_i^{(r)}\right)$ de la condición S0S1 <u>i</u>:

$$PCL_i^{(r)} = \left| \frac{F_k - F_{n+1}}{\sum_{j \leq r} Y_j} \right| \qquad (3)$$

$$PCU_i^{(r)} = \left| \frac{F_k - F_{n+2}}{\sum_{j > r} Y_j} \right| = \left| \frac{F_k - F_{n+2}}{1 - \sum_{j \leq r} Y_j} \right| \qquad (4)$$

De donde resulta que si, posteriormente, en otro nudo, sea p, se decidiese establecer - la dicotomía en la condición S0S1 i entre r y r+1, la estimación del valor de la función objetivo al final de cada una de las dos ramas generadas sería:

$$F_p + s * PCL_i^{(r)} * \sum_{j \le r} Y_j \qquad (5)$$

$$F_p + s * PCU_i^{(r)} * \left(1 - \sum_{j \le r} Y_j\right) \qquad (6)$$

donde Y_j recoge el valor de la variable S0S1 j de la condición S0S1 i en el nudo p y -- donde s = +1 si la optimización es un mínimo y s = -1 si es un máximo.

De acuerdo con una terminología parecida a la empleada para las variables enteras normales, $\sigma_i^{(r)}$ será la más pequeña de las dos degradaciones y $\Delta_i^{(r)}$ será la mayor. Por tanto, para la bifurcación entre r y r+1 en el nudo generante k por lo que respecta a la - condición S0S1 i será:

$$\sigma_i^{(r)} = \min \left\{ PCL_i^{(r)} * \sum_{j \le r} Y_j, PCU_i^{(r)} * \left(1 - \sum_{j \le r} Y_j\right) \right\} \qquad (7)$$

$$\Delta_i^{(r)} = \max \left\{ PCL_i^{(r)} * \sum_{j \le r} Y_j, PCU_i^{(r)} * \left(1 - \sum_{j \le r} Y_j\right) \right\} \qquad (8)$$

donde Y_j es el valor que en el nudo k ha tomado la variable S0S1 j. De forma análoga a las variables enteras normales, ΣY_j $j \le r$ recoge la parte fraccional de una variable entera (ΣY_j).

La estimación de la mejor solución entera a obtener a partir del nudo, sea p, se efectúa de la forma siguiente. Se analizan las variables enteras normales que todavía no - han tomado valores enteros, y se analizan - las condiciones S0S1 en las que algún valor Y_j no es 1 ni 0. Se estudia cuál es la dicotomía a efectuar en cada condición S0S1, de tal forma que si ésta se efectúa en la bifurcación r y r+1, previamente se han calculado los correspondientes seudocostos, el valor estimado E_p será:

$$E_p = F_p + s \left(\sum_{g=1}^{G} \sigma_g + \sum_{i=1}^{I} \sigma_i^{(r)} \right) \qquad (9)$$

donde G es el número de variables enteras - normales que en el nudo p no han tomado valores enteros, σ_g es el mínimo de degeneración de la variable g, I es el número de -- condiciones S0S1 incumplidas, r es el índice de bifurcación de la condición S0S1 i y $\sigma_i^{(r)}$ es la menor de las correspondientes degeneraciones (7). Los valores σ_g se obtienen a base de los seudocostos PCU_g y PCL_g - de las variables enteras normales, obtenidos en el nudo generante, sea k, y a base - de la parte fraccional f_g de dichas variables correspondiendo ésta al nudo p.

En cambio, los valores $\sigma_i^{(r)}$ se calculan en el nudo generante k y ya no sufren modificación en los demás nudos, salvo que debido a ciertas circunstancias se modifiquen los -- seudocostos $PCL_i^{(r)}$ y $PCU_i^{(r)}$. Por tanto, el valor ΣY_j $j \le r$ sólo se utiliza en el nudo p para determinar el índice r sobre el que habría que bifurcar para obtener una solución entera. Determinado el índice r, se selecciona el valor correspondiente $\sigma_i^{(r)}$ (7) previamente calculado en el nudo generante k. Ver Benichou et al. /8/ y /9/, Escudero /11/, /12/, /13/ y /14/, Barceló y Companys /7/, Escudero y Vázquez-Muñiz /15/ y /16/ y Gauthier y Ribiere /18/ en la estimación de -- los seudocostos de una variable entera.

Selección de la variable sobre qué bifurcar

Una vez seleccionado un nudo a bifurcar -- (ver referencias anteriores), es preciso seleccionar la variable entera o condición -- S0S1 sobre qué bifurcar entre aquellas que en el nudo a bifurcar sean incumplidas. En las referencias anteriores se recogen los - criterios seguidos para bifurcar sobre las variables enteras normales, y la operativa sirve también para las condiciones S0S1. -- Así el usuario puede indicar el orden de -- prioridad, procesándose primero las variables enteras normales y condiciones S0S1 -- más importantes en el modelo, o aquellas cuyo coeficiente tenga mayor valor absoluto - en la función objetivo, etc. Barceló y Companys /7/ y Escudero /12/ recogen los criterios más importantes para seleccionar el -- próximo nudo a bifurcar, y en éste, la variable sobre qué bifurcar.

Como no se puede bifurcar sobre una variable S0S1 ya que la condición S0S1 forma una sóla entidad, si para seleccionar la variable entera normal o condición S0S1 a sepa--

rar se elige la estrategia del máximo valor absoluto del coeficiente en la función objetivo, por lo que respecta a las S0S1 éste será el mayor coeficiente en valor absoluto en la función objetivo de las variables S0S1 -- que la integran.

La obtención de los seudocostos de las variables enteras normales y de las condiciones - S0S1 es fundamental para seleccionar el nudo a bifurcar, la variable sobre qué bifurcar, el orden de optimización de las dos ramas a generar, la detención de una optimización, - etc. Es muy probable que en las primeras iteraciones de la formación arborescente, no haya muchas variables enteras con seudocostos fiables. El problema se agrava en las condiciones S0S1, ya que precisan seudocostos para cada posible bifurcación entre r y r+1. - Si, por ejemplo, en una condición S0S1 hay - 10 variables, puede haber 10-1 posibles bi-- furcaciones, por tanto es posible que se necesiten 9 tipos de seudocostos inferior y superior, de tal forma que los seudocostos entre r=1 y r+1=2 corresponden al caso en que una rama $Y_1=0$ y en la otra $Y_2=Y_3=\ldots=Y_{10}=0$, los seudocostos entre r=2 y r+1=3 corresponden al caso en que una rama $Y_1=Y_2=0$ y en -- otra $Y_3=Y_4=\ldots=Y_{10}=0$, y así sucesivamente -- hasta que en la bifurcación entre r=9 y -- r+1=10 los seudocostos corresponden al caso en que en una rama $Y_1=Y_2=\ldots=Y_9=0$ y en la otra $Y_{10}=0$. No obstante, la utilidad de las - condiciones S0S1 es evidente, constituyendo una de las características más sobresalientes de los métodos branch-and-bound.

Resultados numéricos

Benichou et al. /9/, Escudero /11/, /13/ y - /14/ y Gauthier y Ribiere /18/, recogen análisis comparativos con o sin conjuntos S0S1.

Uno de los casos más evidentes de la utilidad de las condiciones S0S1, lo constituye - un modelo de programación lineal entera mixta cuyo objetivo consiste en efectuar a mínimo costo la planificación para el decenio -- 1974/83 de las dimensiones de parques, muelles y flota marítima de una fuerte siderurgia española. Las dimensiones del modelo son M=1000 condiciones y N=1000 variables de las cuales I=450 variables son enteras.

Sin condiciones S0S1 se obtuvo una solución "buena" en 100 horas de CPU de un ordenador 360/65. En cambio, con condiciones S0S1 utilizando un ordenador S/370 modelo 158, se - transforma el problema agrupado en 72 condiciones S0S1 las 400 variables S0S1 que, junto con el resto de las 50 variables enteras normales, componen el total de las variables enteras del problema. Dado que el modelo es multiperíodo en el sentido de que una alternativa (selección de una determinada dimensión de parques, muelles o tipo de barco) - que se tome en un período anual excluye la posibilidad de tomar la misma decisión en - los otros años y que además las diversas capacidades de las unidades de parques, muelles y barcos son exclusivas, no es de extrañar que la utilización de las condiciones S0S1 acelere la convergencia de la optimización. Se precisó sólo de 9 minutos de - CPU para obtener la primera solución entera, partiendo de la solución óptima continua, - pero sin conceder explícitamente pesos a -- las diversas variables S0S1, dependiendo de su orden de inclusión en la matrix de condiciones del modelo. En una nueva prueba, en la que se ponderaron las diversas variables S0S1 con la capacidad del parque, muelle o tipo de barco a que fueron afectadas, la -- primera solución entera se obtuvo a los -- 7 minutos de CPU representando una mejora - del 10% en la función objetivo con respecto a la solución obtenida sin ninguna ponderación, y del 15% con respecto a la solución obtenida sin condiciones S0S1. Aunque limitaciones exteriores impidieron probar la optimalidad de la solución entera, ésta sólo difería en un 2% de la solución óptima del mejor nudo candidato.

Prácticamente todos los códigos generados - de PL con métodos branch-and-bound tienen - la posibilidad de tratar conjuntos S0S1. No obstante, Bricker /10/ simula el tratamiento de un conjunto S0S1 a base de crear nuevas variables binarias cada una de las cuales restringe el número de variables S0S1 - libres en el conjunto.

3. CONJUNTOS ESPECIALES S0S2

El método a aplicar en el caso de conjuntos S2 es muy parecido al caso de S1 y, por supuesto, de S0S1. A efectos ilustrativos recogemos aquí una técnica a aplicar para los conjuntos S0S2. Su operativa y los conceptos que utiliza son análogos a los empleados para los conjuntos S0S1, con las diferencias que aquí se indican. También a este tipo de conjuntos se le asocia una condición ponderativa que puede formar parte del modelo o no. Sea el conjunto S0S2:

$$\sum_{j=1}^{5} Y_j = 1 \qquad 1 \geq Y_j \geq 0 \tag{10}$$

y su condición ponderativa:

$$\sum_{j=1}^{5} W_j Y_j = X \qquad W_j > W_{j+1} \tag{11}$$

de tal forma que W_j representa la importancia atribuida a la variable Y_j. La utilización del índice de bifurcación r para formar la dicotomía de (10) y así crear los dos nudos sucesores al nudo estudiado, tiene una operativa ligeramente diferente al caso del conjunto S0S1. También en este caso se observa el valor de X y si el conjunto S0S2 no cumple la condición de que si hay dos variables $Y_j \neq 0$, éstas deben ser consecutivas, se considera que el índice de bifurcación r tiene que ser tal que $W_r > \bar{X} > W_{r+1}$. Por tanto, se crean nudos sucesores que teniendo el mismo planteamiento que el nudo contemplado, se diferencian en que en uno se anulan las variables Y_j para j=1,2,...,r-1 (es decir, todas las variables anteriores a Y_r deben ser cero) y en el otro se anulan las variables Y_j para j=r+1, r+2,...,5 (es decir, todas las variables posteriores a Y_r deben ser cero).

Se puede observar que en esta dicotomía (y en ello estriba la gran diferencia con la operativa en el conjunto S0S1), la única variable que es común a los dos nudos es precisamente Y_r. A medida que se avance en este tipo de fase branch-and-bound, el número de variables libres va siendo menor y se considera que el nudo tiene la condición satisfecha cuando no haya más de dos variables $Y_j \neq 0$ y sean en tal caso consecutivas. Un caso extremo sería cuando en una de las dos ramas de la dicotomía sólo esten las variables Y_r e Y_{r-1} o Y_r e Y_{r+1}.

No todos los códigos generales de PL con métodos branch-and-bound tratan los conjuntos S0S2. Land y Powell /19/ indican que sólo son soportados por los códigos APEX-III, SCICONIC y XDLA. No obstante, su formulación se puede efectuar a base de variables continuas con límites cero y uno, combinadas con conjuntos S0S1.

4. CONJUNTOS ESPECIALES S1 y S2

Beale y Forrest /4/ y Beale /2/ y /3/ operan con los conjuntos S1 y S2 de una forma análoga a como operan Benichou et al. /9/, Escudero /11/, /13/ y /14/, y Gauthier y Ribiere /18/, respectivamente, con los conjuntos S0S1 y S0S2 (para este último, ver sólo Escudero /13/ y /14/); pero utilizando la noción del seudo-shadow-cost en lugar de la noción del seudo-cost y utilizando la noción del valor estimado seudo-shadow-cost en lugar de la noción del valor estimado. Land y Powell /19/ indican que sólo los códigos APEX-III y SCICONIC soportan los conjuntos S1 y S2 sin exigir que la suma de las respectivas variables sea uno. Al final de este trabajo, se recoge la referencia de los códigos comerciales de PL y PLE más utilizados.

5. CONJUNTOS ESPECIALES LS1 y LS2

Beale /3/ introduce la noción de linked ordered sets, cuyo objetivo consiste en formular funciones no-lineales de tal forma que se pueden tratar en códigos generales de PL con métodos branch-and-bound que, lógicamente, soporten los conjuntos LS (linked ordered sets).

Se pueden representar un gran número de funciones no-lineales a base de sumas de productos de funciones no-lineales con una única variable de argumento. Beale y Forrest /5/ reformulan en este sentido un gran número de funciones no-lineales ya clásicas en la literatura sobre esta materia. En principio, cualquier producto de funciones no-lineales de argumento único puede ser aproximado tomando logaritmos y utilizando conjuntos S0S1 o S0S2 en la resolución de problemas de programación separable no-convexa, dado que e.g.

$$Y = \prod_j X_j^{a_j} = \exp\left\{\sum_j a_j Z_j\right\} \tag{12}$$

$$\ln Y = \sum_j a_j \; Z_j \qquad (13)$$

$$Z_j = \ln X_j \qquad (14)$$

Esta forma de operar puede tener problemas, salvo que Y y X_j sean estrictamente positivas y se conozca que los valores de Y y X_j tienen un adecuado orden de magnitud.

Por otro lado, las funciones cuadráticas del tipo XY pueden ser aproximadas a base de sumas y diferencias de cuadrados de funciones lineales, e.g.

$$XY = \left(\frac{X + Y}{2}\right)^2 - \left(\frac{X - Y}{2}\right)^2 = \frac{1}{4} Z_1^2 - \frac{1}{4} Z_2^2 \quad (15)$$

donde $Z_1 = X+Y$ y $Z_2 = X-Y$.

Utilizando la técnica de programación separable (Beale /1/) se puede aproximar perfectamente el problema (15). Si el problema fuese Xf(Y) también se podría utilizar la técnica de programación separable si f(Y) pudiera llegar a representarse por sumas de cuadrados de funciones lineales, pero se precisaría un gran número de convexity rows y reference rows de programación separable. Por tanto, esta posibilidad es interesante cuando la función cuadrática es del tipo XY y no hay una gran profusión de ellas.

La noción de linked ordered sets es una aproximación a la optimización global de problemas con funciones no-lineales, sin tener que utilizar programación no-lineal y utilizando sólo códigos de PL con una pequeña modificación en el método branch-and-bound. Se pueden clasificar las variables de un problema en variables lineales y variables no-lineales, de tal forma que el problema se convierte en PL si se fijan las variables no-lineales. La técnica de linked ordered sets aproxima la resolución del problema si las funciones no-lineales, e.g. $\gamma(Z,Y)$, se pueden representar por sumas o diferencias de productos de funciones no-lineales con argumento único. Ejemplo, $\gamma(Z,Y) = \Sigma(f(Z)\; f(Y))$. En este caso, cada uno de los productos f(Z) f(Y) se sustituye por X=f(Z) (programación separable) y Xf(Y). La técnica de linked ordered sets aproxima el valor de Xf(Y).

Para representar de esta forma Xf(Y) es preciso conocer los valores máximo X_M y mínimo X_m de la variable X. Si Y sólo puede tomar valores finitos $Y_1, Y_2, \ldots, Y_m$ entonces la técnica es de linked ordered sets de tipo 1 (conjuntos LS1) y si el conjunto de valores posibles de Y no es finito, entonces la técnica es de linked ordered sets de tipo 2 (conjuntos LS2).

Comencemos con el conjunto LS1. Este será el conjunto de variables $\lambda_{11}, \lambda_{12}, \ldots, \lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{m1}, \lambda_{m2}$, $0 \leq \lambda_{i1}, \lambda_{i2} \leq 1$ y continuo, tal que

$$\sum_{i=1}^{m} (\lambda_{i1} + \lambda_{i2}) = 1 \qquad (16)$$

con la condición adicional de que si $Y=Y_j$, entonces $\lambda_{i1}=\lambda_{i2}=0$ para $i=1,2,\ldots,m$, $i\neq j$. Por tanto, $\lambda_{j1}+\lambda_{j2}=1$. La condición se denomina convexity row de forma análoga al conjunto SOS1. La condición ponderativa (reference row) del conjunto LS1 será

$$Y = \sum_{i=1}^{m} Y_i(\lambda_{i1}+\lambda_{i2}) = Y_1\lambda_{11}+Y_2\lambda_{21}+ +\ldots+Y_m\lambda_{m1}+Y_1\lambda_{12}+Y_2\lambda_{22}+\ldots+Y_m\lambda_{m2} \qquad (17)$$

la variable X tiene la representación:

$$X = \sum_{i=1}^{m} (X_m \; \lambda_{i1} + X_M \; \lambda_{i2}) \qquad (18)$$

Por tanto, si $A_i=f(Y_i)$, la función Xf(Y) tiene la siguiente representación lineal,

$$\sum_{i=1}^{m} A_i \; (X_m \; \lambda_{i1} + X_M \; \lambda_{i2}) \qquad (19)$$

estando (19) en función del conjunto LS1.

La operativa en el tratamiento del conjunto LS1 en el método branch-and-bound es similar al caso de los special orderes sets de tipo SOS1. Así en un nudo determinado, sea k, para bifurcar sobre el conjunto LS1 se observa si las condiciones de este conjunto (16) se han satisfecho. Será el caso en el que $\lambda_{i1}=\lambda_{i2}=0$ y $\lambda_{j1}+\lambda_{j2}=1$, $i=1,2,\ldots,m$, $i\neq j$. Si no fuese así se analiza la condición ponderativa. Considerando que $Y_i>Y_{i+1}$, el índice de bifurcación r será aquel para el que $Y_r>\bar{Y}>Y_{r+1}$. Por tanto, se efectúa una dicotomía en la que se crean los nudos suce sores (n+1) y (n+2) cuyas función objetivo y condiciones son las mismas que en el nudo k, pero en el nudo (n+1) se sustituye la -- condición (16) por

$$\sum_{i=1}^{r} (\lambda_{i1} + \lambda_{i2}) = 1 \tag{20}$$

y en el nudo (n+2) se sustituye la condición (16) por

$$\sum_{i=r+1}^{m} (\lambda_{i1} + \lambda_{i2}) = 1 \tag{21}$$

La operativa a partir de aquí es similar a la desarrollada en (1) a (9) para el conjunto SOS1. Así los seudocostos serán:

$$PCL^{(r)} = \left| \frac{F_k - F_{n+1}}{\sum_{i\leq r} (\lambda_{i1}+\lambda_{i2})} \right| \tag{22}$$

$$PCU^{(r)} = \left| \frac{F_k - F_{n+2}}{1- \sum_{i\leq r} (\lambda_{i1}+\lambda_{i2})} \right| \tag{23}$$

Beale /3/ opera con los conjuntos LS1 de -- una forma análoga a como operan Beale y -- Forrest /5/ con los conjuntos S1; por tanto, sustituye la noción de seudocostos por la de seudo-shadow-cost y la noción de valor estimado por la de valor estimado seudo-shadow-cost.

Ahora bien, si el conjunto de posibles valores de la variable Y es infinito, entonces hay que utilizar la noción del conjunto LS2 (similar a la del conjunto SOS2) en lugar - de la noción del conjunto LS1. En este caso, sólo se efectúa una aproximación de la función Xf(Y) al igual que en programación separable (ver Escudero /13/ y /14/) se efectúa con la aproximación de la función f(Y). Se parte también de las convexity row (16) y reference row (24). La diferencia estriba en que además de las condiciones adicionales a (16) se permite que a lo máximo dos - subconjuntos $(\lambda_{i1},\lambda_{i2})$ sean distintos de cero, pero en este caso deben ser consecutivos. Por tanto, se satisface la condición - si $\lambda_{j1}+\lambda_{j2}=1$ y $\lambda_{i1}=\lambda_{i2}=0$ para $i=1,2,\ldots,m$, $i\neq j$ o también si $\lambda_{j1}+\lambda_{j2}+\lambda_{j+1,1}+\lambda_{j+1,2}=1$ y $\lambda_{i1}=\lambda_{i2}=0$ para $i=1,2,\ldots,m$, $i\neq j$, $i\neq j+1$. En este caso, (17) y (18) serían:

$$Y = \sum_{j=1}^{m-1} \left[Y_j(\lambda_{j1}+\lambda_{j2}) + Y_{j+1}(\lambda_{j+1,1}+\lambda_{j+1,2}) \right] \tag{24}$$

$$X = \sum_{j=1}^{m-1} \left[X_m(\lambda_{j1}+\lambda_{j+1,1}) + X_M(\lambda_{j2}+\lambda_{j+1,2}) \right] \tag{25}$$

con lo cual no se impide que Y y X tomen el valor apropiado. Considerando que

$$B_i = A_i \left[X_m(\lambda_{j1}+\lambda_{j+1,1}) + X_M(\lambda_{j2}+\lambda_{j+1,2}) \right] \tag{26}$$

para $i=1,2,\ldots,m$, la representación de Xf(Y) no sería lineal, ya que

$$Xf(Y) = B_j(\lambda_{j1}+\lambda_{j2}) + B_{j+1}(\lambda_{j+1,1}+\lambda_{j+1,2}) \tag{27}$$

Ahora bien, e.g. $B_j\lambda_{j1}$ es un producto de dos variables que al igual que en (15) puede sustituirse por $1/4(Z_1^2-Z_2^2)$ donde $Z_1=B_j+\lambda_{j1}$ y - $Z_2=B_j-\lambda_{j1}$, de tal forma que Z_1^2 y Z_2^2 pueden - aproximarse por los métodos tradicionales de programación separable convexa. Ver Beale -- /1/ y Escudero /13 y /14/.

La operativa del tratamiento del conjunto -- LS2 es similar a la correspondiente al conjunto SOS2. Si las condiciones del conjunto LS2 no se han satisfecho en el nudo k considerado, se analiza la condición ponderativa (24) y el índice de bifurcación r será aquel para el que $Y_r>\bar{Y}\geq Y_{r+1}$. Por tanto, se crean - dos nudos sucesores que teniendo el mismo -- planteamiento que el nudo contemplado, se diferencian en que en uno se anulan las variables λ_{i1}, λ_{i2} para $i=1,2,\ldots,r-1$ y en el -- otro se anulan las variables λ_{i1}, λ_{i2} para - $i=r+1, r+2, \ldots$, de forma análoga a cómo se actúa en el conjunto SOS2. Beale /3/ presenta un tratamiento ligeramente diferente de - los conjuntos LS1 y LS2. Beale y Forrest /5/ muestran problemas de programación no-lineal tratados como conjuntos SOS2 y LS1. Aunque - todavía no se puede obtener conclusiones definitivas, la utilización de los linked ordered sets puede ser uno de los grandes instrumentos para abordar los problemas de optimización global.

6. CONCLUSIONES

Los conjuntos especiales SOS1, SOS2 (dado -- que los conjuntos S1 y S2 pueden transformarse en los anteriores e incluso los conjuntos SOS2 se pueden sustituir por modelizaciones de conjuntos SOS1), LS1 y LS2 constituyen una de las grandes innovaciones en programación matemática en los últimos años. Permiten obtener óptimos globales en programación separable así como en una gran variedad de problemas de programación no-lineal, en lugar de alcanzar sólo óptimos locales. Asimismo, reducen enormemente el tiempo de CPU en la optimización no solo de problemas típicos de programación lineal entera mixta con variables binarias, sino también en problemas combinatorios cuya estructura especial exige muchas veces algoritmos especialmente adaptados y que, sin embargo, utilizando la técnica de conjuntos especiales pueden ser, si no eliminados, si al menos complementarios.

7. RECONOCIMIENTOS

Agradezco a IBM España la autorización concedida para incluir en este trabajo la operativa de los conjuntos SOS1, SOS2, LS1 y LS2 tiene incorporada en su sistema de programación matemática.

8. BIBLIOGRAFIA

/1/ BEALE, E.M.L. "Mathematical Programming in Practice". Pitman, Londres, 1968.

/2/ BEALE, E.M.L. "Integer programming", en: D. JACOBS (ed.). "The state-of-the-art in Numerical Analysis", Academic Press, Londres, 1977, pp. 409-448.

/3/ BEALE, E.M.L. "Branch and bound methods for mathematical programming systems". Discrete Optimization Conference DO'77. Vancouver, 1977.

/4/ BEALE, E.M.L. y FORREST, J.J.H. "Global Optimizacion using special ordered sets". Mathematical Programming 10, 1976, pp. 52-69.

/5/ BEALE, E.M.L. y FORREST, J.J.H. "Global Optimization as an extension of integer programming", en: L.C.W. DIXON y G.P. SZEGO (eds.), "Towards global optimization 2". North-Holland, AMsterdam, 1977.

/6/ BEALE, E.M.L. y TOMLIN, J.A. "Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables", en: J. LAWRENCE (ed.), "Operational Research '69". Tavistock Publ., Londres, 1969, pp. 447-454.

/7/ BARCELO, J. y COMPANYS, R. "Panorámica actual de las técnicas branch-and-bound en programación en números enteros", Seminario de Programación Matemática PM'77, Madrid, 1977. Publicado en BARCELO, J. "Panorámica actual de los métodos enumerativos", QÜESTIIÓ, v. 2, parte I en n° 1 y parte II en n° 2, 1978.

/8/ BENICHOU, M., GAUTHIER, J.M., GIRODET, P., HENTGES, G., RIBIERE, G. y VINCENT, O. "Experiments in mixed integer programming", Mathematical Programming 1, 1971, pp. 76-94.

/9/ BENICHOU, M., GAUTHIER, J.M., HENTGES, G. y RIBIERE, G. "The efficient solution of large scale linear programming problems. Some algorithmic techniques and computational results". IX International Symposium on Mathematical Programming, Budapest, 1976.

/10/ BRICKER, D.L. "Reformulation of Special Ordered Sets for Implicit Enumeration algorithm with application in Non-convex Separable Programming". AIIE Transactions 9, 1977, pp. 1975-203.

/11/ ESCUDERO, L.F. "Programación lineal: continua, entera, mixta y bivalente". Ediciones Deusto, Bilbao, 1976.

/12/ ESCUDERO, L.F. "Panorámica actual de Programación Matemática". Seminario de Programación Matemática PM'77, Madrid, 1977.

/13/ ESCUDERO, L.F. "A comparative study of linear fitting for non-lineal functions on optimization. A case study: air pollution problems". European Journal of Operational Research (aparecerá en 1978).

/14/ ESCUDERO, L.F. "Análisis de estrategias en el ajuste lineal de funciones no-li neales en problemas de Optimización". Trabajos de Estadística e Investigación Operativa. (Aparecerá en 1978).

/15/ ESCUDERO, L.F. y VAZQUEZ-MUÑIZ, A. "The air pollution abatement. A mathematical model", en: M. ROUBENS (ed.), "Advances in Operations Research. North-Holland, Amsterdam, 1977, pp. 659-667.

/16/ ESCUDERO, L.F. y VAZQUEZ-MUÑIZ, A. "Un modelo matemático para la reducción de la contaminación atmosférica". Trabajos de Estadística e Investigación Operativa. (Aparecerá en 1977).

/17/ FORREST, J.J.H., HIRST, J.P. y TOMLIN, J.A. "Practical solution of large mixed integer programming problems with UMPIRE". Management Science 20, 1974. pp. 736-776.

/18/ GAUTHIER, J.M. y RIBIERE, G. "Experiments in mixed integer programming using pseudo-costs". Mathematical Programming 12, 1977. pp. 26-47.

/19/ LAND, A. y POWELL, S. "Computer codes for problems of integer programming". Discrete Optimization Conference DO'77. Vancouver, 1977.

/20/ TOMLIN, J.A. "Branch-and-bound methods for integer programming and non-convex programming" en: J. ABADIE (ed.), "Integer and non-linear programming". North-Holland, AMsterdam, 1970. pp. 437-450.

9. ANEXO. REFERENCIAS DE CODIGOS COMERCIALES DE PL Y PLE

APEX-III, Reference manual, CDC Services Publications HQC02C, P.O. Box 0, Minneapolis, Minnesota 55440, 1976.

FMPS, Sperry Univac 1100 Series Functional Mathematical Programming System, Programmer Reference UP-8198, 1975.

Haverly-MIP, LP360/370 Linear Programming System: User and Operating manual, 10 Haverly Systems Inc., 78 Broadway, Denville, N.J. 07834, USA.

LP400, Linear Programming 400, System 4, ICL, ICL House, Putney, Londres SW15, 1970.

MIP/370, IBM Mathematical Programming System Extended/370 (MPSX/370), Mixed Integer Programming 370 (MIP/370). Program Reference Manual Program Product 5740-XM3 (OS/VS), 5746-SM2 (DOS/VS), IBM Corporation Technical Publications Dept., 1133 Westchester Av., White Plains, N.Y. 10604, 1975.

MPS, Mathematical Programming System. Mixed Integer Programming User's Guide, DD62, Honeywell Information Systems, 60 Walnut St., Wallesley Hills, Massachussets 02181, 1974.

OPHELIE, LP User Manual SIA (Service in Informatics and Analysis Ltd.), Ebury Gate, 23 Lower Belgrave St., London SWIW ONW, 1972.

SCICONIC, Users Guide to Sciconic (3.2.), Scicon Computer Services Ltds. Brick Close, Kiln Farm, Milton Keynes MK11 3EJ, UK, 1976.

TEMPO, Mathematical Programming System, User's Manual, Burroughs Corporation B 7700/ /B Systems, Documentation Dept., Business Management and Scientific Systems, Burroughs Place, Detroit, Michigan 48232, 1975.

XDLA, Linear Programming Mark 3, Series 1900, ICL, ICL House, Putney, Londres SW15, 1967.