

# OPTIMIZACIÓN CON SOLVER

SÁNCHEZ ALVAREZ, ISIDRO ([isanchez@econo.uniovi.es](mailto:isanchez@econo.uniovi.es))

LÓPEZ ARES, SUSANA ([slopez@econo.uniovi.es](mailto:slopez@econo.uniovi.es))

*Departamento de Economía Cuantitativa - Universidad de Oviedo*

La relevancia de los problemas de optimización en el mundo empresarial ha generado la introducción de herramientas de optimización cada vez más sofisticadas en las últimas versiones de las hojas de cálculo de utilización generalizada. Estas utilidades, conocidas habitualmente como «solvers», constituyen una alternativa a los programas especializados de optimización cuando no se trata de problemas de gran escala, presentado la ventaja de su facilidad de uso y de comunicación con el usuario final. Frontline Systems Inc es la empresa que desarrolla el «solver» de Excel, si bien existen asimismo versiones para Lotus y Quattro Pro con ligeras diferencias de uso. En su dirección de internet ([www.frontsys.com](http://www.frontsys.com)) se puede obtener información técnica sobre las diferentes versiones de dicha utilidad y diversos aspectos operativos del programa, algunos de los cuales se comentan en este trabajo.

## I. CONSTRUCCIÓN DE UN MODELO DE OPTIMIZACIÓN.

La introducción de un modelo de optimización, un programa lineal en nuestro ejemplo, se puede sintetizar en cuatro fases:

1. *Organizar los datos del modelo en la hoja de trabajo.* Si bien son múltiples las posibles formas de diseñar el formato y colocación de los datos de entrada, es recomendable seguir los mismos principios que en toda aplicación con hoja de cálculo: pensar en la hoja como un informe que explique el problema, identificar los datos introducidos, colocar comentarios, introducir todos los datos iniciales del problema y construir a partir de los mismos el modelo de optimización con el objeto de facilitar el análisis de sensibilidad, utilizar técnicas de diseño para presentar el modelo, etc. Por otra parte, interesa organizar el programa según el formato del gráfico I con el objeto de ilustrar la propia estructura del modelo.
2. *Reservar una celda para cada variable de decisión.* Siguiendo el esquema de un programa matemático, es recomendable que inicien la hoja de trabajo. Deberán estar vacías o con datos numéricos, nunca fórmulas, y a ser posible con notas o comentarios.
3. *Crear una celda para la función objetivo próxima a las que recogen las variables.* La fórmula que incorpora deberá crearse a partir de las celdas descritas en el punto anterior.

4. Para cada restricción, crear una celda que recoja la fórmula de su parte izquierda, y a la derecha de dicha celda colocar el término independiente. La estructura recomendable es la que se recoge en el gráfico I dado que permite reducir el trabajo en la fase de introducción del problema, facilita la detección de errores y simplifica su resolución con el «solver».

GRÁFICO I

	A	B	C	D	E	F
1						
2	<b>PROBLEMA DE PRODUCCIÓN</b>					
3						
4		PTO.1	PTO.2			
5	COSTE UNITARIO:	21	24		COSTE TOTAL	
6	UNIDADES:	0	0		0	Objetivo
7						
8	RESTRICCIONES:			UTILIZADO	RECURSOS	
9	MATERIA PRIMA	2	1	0	100	
10	MANO DE OBRA	3	2	0	120	
11	UNIDADES	1	1	0	50	
12						

Programa lineal

## II. PROGRAMACIÓN LINEAL.

Una vez introducidos los datos, el programa se resuelve ejecutando el comando «Solver» situado dentro del menú de Herramientas. Para ello es preciso tener en cuenta la siguiente equivalencia de términos:

Celda objetivo	Función objetivo
Celdas cambiantes	Variables de decisión

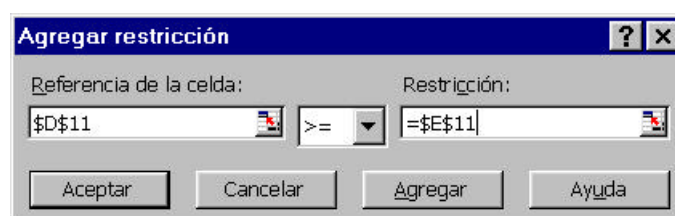
En el gráfico II se recogen los parámetros del «Solver» para el problema del gráfico I. La función objetivo (celda E6) se coloca como celda objetivo, señalando la opción «Min» que indica que el programa es de mínimo. Las variables de decisión se señalan recogiendo el rango de celdas que ocupan (B6:C6) en el cuadro de celdas cambiantes. El botón «Estimar» sugiere automáticamente las variables de decisión ya que introduce en el recuadro de celdas cambiantes todas aquellas de las que depende la celda objetivo. En ciertos casos coincidirá con las variables de decisión - tal como ocurre con el ejemplo que utilizamos de ilustración - mientras que en otros casos será preciso eliminar algunos de los rangos de celdas sugeridos por el programa.

GRÁFICO II



Para introducir las restricciones se presiona el botón de «Agregar» generando el despliegue de una nueva pantalla (gráfico III). En la parte izquierda (Referencia de celda) se introduce la celda que recoge la fórmula matemática que refleja la parte izquierda de la restricción (la celda D11 en el caso de la tercera restricción). En la parte derecha se recoge la celda, o directamente el valor numérico, del término independiente de la restricción. En el cuerpo central se selecciona el signo de la restricción. Excel 97 permite los tres posibles signos permitidos en la programación lineal ( $\leq$ ,  $=$ ,  $\geq$ ).

GRÁFICO III



Obsérvese en el gráfico II cómo la segunda de las restricciones introducida representa realmente dos restricciones del programa matemático. Debido a que las dos primeras restricciones del problema son del mismo signo, y a que se han diseñado en celdas cercanas las partes izquierda y derecha de ambas restricciones, se pueden condensar en una sola restricción tal como se indica en el gráfico citado. Los botones «Cambiar» y «Eliminar» permiten modificar y borrar respectivamente alguna de las restricciones.

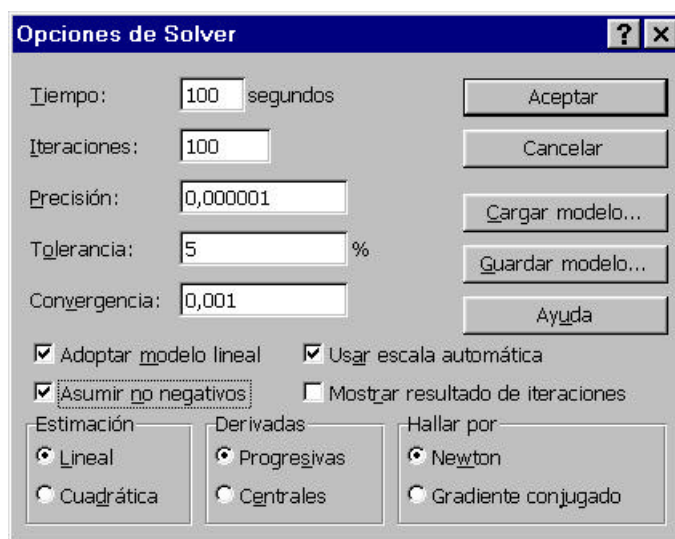
Una vez indicados todos los elementos del programa, el botón «Opciones» permite señalar una serie de características que afectan al modo de resolución del programa (gráfico IV). Las opciones relacionadas con los modelos lineales son las siguientes:

- «Adoptar modelo lineal». Si se señala este cuadro de diálogo, el «Solver» utiliza el método simplex. En caso contrario, utilizaría el método del gradiente reducido, lo cual produciría

una resolución más lenta, sería más susceptible de presentar problemas de degeneración y escala, y produciría sólo valores duales, sin rangos de variación, en el análisis de sensibilidad.

- «Asumir no negativos». Por defecto, las celdas cambiantes - variables - pueden tomar valores cualesquiera. Para que tomen valores no negativos es preciso señalar esta opción.
- «Usar escala automática». Debido a la precisión finita, cuando los cálculos se realizan con valores de magnitudes muy diferentes se producen errores de redondeo que pueden generar soluciones erróneas. Para evitar problemas de escala será preciso replantear el problema de forma que los órdenes de magnitudes no difieran en exceso. La opción de escala automática permite compensar los efectos producidos por problemas de escala. No obstante, es preciso tener en cuenta que en la versión básica de «Solver» que incorpora Excel 97 esta opción sólo es efectiva en problemas no lineales. Versiones más avanzadas del mismo extienden su utilización a todo tipo de modelos.
- Las opciones «Tiempo» e «Iteraciones» determinan el máximo de tiempo y esfuerzo dedicado a resolver el programa matemático. Las opciones indicadas por defecto suelen ser suficientes, y deberían ser modificadas sólo si no se logra resolver el modelo. Su valor debe ser un entero comprendido entre 1 y 32767.
- La «Precisión» hace referencia al grado de cercanía entre los valores calculados en la parte izquierda de las restricciones y los correspondientes a la parte derecha. Con el grado de precisión por defecto (1.0E-6), un valor calculado en la parte izquierda de -1.0E-7 satisfaría una restricción como  $A1 \geq 0$ . La precisión puede tomar valores entre 1.0E-4 y 1.0E-8. A mayor número de decimales, menor será la diferencia permitida entre ambas partes de las restricciones, y por tanto menor tolerancia en los errores.

GRÁFICO IV



- Por último, «Mostrar resultado de las iteraciones» proporciona los valores de las distintas variables del modelo en cada iteración. El resto de opciones son aplicables a modelos que se describen en epígrafes posteriores.

La solución del programa matemático se obtiene ejecutando la opción «Resolver». Al final del proceso, el programa indica que se ha obtenido la solución, solicitando al usuario:

- Si desea que la hoja de trabajo mantenga los valores iniciales o la solución óptima obtenida.
- Si desea los diferentes informes de sensibilidad, los cuales incluyen: solución, precios duales, costes reducidos, rangos de incrementos y disminuciones permisibles de los términos independientes de las restricciones y de los coeficientes de la función objetivo. No proporciona ningún análisis paramétrico.

### **III. PROGRAMACIÓN ENTERA.**

En el caso de que en el programa existan variables enteras o binarias es preciso indicarlo como restricciones. Para ello, la parte izquierda de la pantalla del gráfico III (Referencia de la celda) recogerá las celdas que representan variables enteras o binarias, y en el signo de las restricciones se seleccionará la opción «int» para enteras o «bin» para binarias.

El método utilizado para resolver este tipo de programas es el de ramificación y acotamiento (Branch and Bound). Cada vez que el «Solver» encuentra una solución entera mejorada, calcula el máximo porcentaje de diferencia entre el valor de la función objetivo de esta solución y el mejor valor disponible de la función objetivo obtenido hasta dicho momento:

$$\frac{\text{Valor objetivo de la nueva solución mejorada} - \text{Valor objetivo de la mejor solución actual}}{\text{Valor objetivo de la mejor solución actual}}$$

Si el valor absoluto de este máximo porcentaje de diferencia es igual o menor que la «Tolerancia», el «Solver» parará el proceso y señalará la solución entera actual como el óptimo del problema. Si se fija un nivel de tolerancia nulo, seguirá buscando hasta que todas las alternativas hayan sido exploradas y sea encontrada la solución entera óptima. En definitiva, la tolerancia es una especificación de la suboptimalidad admitida, medida en porcentaje de la mejor solución óptima disponible.

En el caso de programas enteros y binarios, la «Precisión» también juega un papel especialmente relevante para determinar las restricciones del tipo A1=integer. Si la diferencia entre el valor de la variable de decisión y el valor entero más cercano es menor que la precisión, el valor de la variable se considera como entero.

### III. PROGRAMACIÓN NO LINEAL

El algoritmo utilizado por el «Solver» es el *Gradiente Reducido Generalizado (GRG)*, en la versión GRG2, cuya estructura matemática puede ser analizada en Abadie(1978); Lasdon, Waren, Jain y Ratner(1978); Lasdon y Waren(1978); y Ríos(1988).

Básicamente, al igual que otros algoritmos de programación no lineal, parte de una solución factible conocida como punto inicial. El algoritmo intenta entonces moverse, a partir de este punto, en una dirección a través de la región factible, de tal forma que el valor de la función objetivo mejore. Tomando un salto o movimiento determinado en dicha dirección factible, se pasa a una nueva solución factible mejorada. De nuevo, el algoritmo identifica una nueva dirección factible, si existe, y un salto determinado avanzando hacia una nueva solución factible mejorada. El proceso continúa hasta que el algoritmo alcanza un punto en el cual no existe una dirección factible para moverse que mejore el valor de la función objetivo. Cuando no hay posibilidad de mejora, o el potencial para tal mejora es arbitrariamente pequeño, el algoritmo finaliza. Ahora bien, en ese momento la solución es un *óptimo local*, y por tanto no necesariamente *global*. En este sentido, es preciso tener en cuenta dos características de las soluciones obtenidas al resolver un programa no lineal con «Solver»:

- El algoritmo puede finalizar en un óptimo local que puede no ser el óptimo global del problema.
- El óptimo local en que finaliza el algoritmo depende del punto inicial.

Si bien la posibilidad de terminar en un óptimo local no es deseable, en el caso de la programación entera ya habíamos comentado la posibilidad de aceptar soluciones subóptimas si estaban dentro del grado de tolerancia aceptable. Desgraciadamente, en los programas no lineales no se puede determinar fácilmente el grado de alejamiento entre el óptimo local y el global, dado que no existe un método genérico para obtener cotas del valor de la función objetivo.

Sin embargo, muchos programas no lineales tienen óptimos locales únicos que, por definición, necesariamente deben ser globales. Por ejemplo, las siguientes condiciones garantizan, si existe, que el óptimo es global:

- Función objetivo de máximo y cóncava, o el logaritmo de la función objetivo cóncava, con restricciones lineales.
- Función objetivo de mínimo y convexa, con restricciones lineales.

No obstante, en general, no conoceremos si la solución obtenida es un óptimo global. Como consecuencia, se suele intentar la prueba de iniciar el algoritmo desde diferentes puntos para determinar si el problema tiene diferentes soluciones óptimas. Este procedimiento suele

revelar la existencia de un determinado óptimo global, si existe, pero no es un método de total fiabilidad.

Dado el carácter de las soluciones de los programas no lineales es importante tener en cuenta los mensajes que proporciona el «Solver»:

- *Solver ha encontrado la solución. Todas las restricciones y condiciones de optimalidad están satisfechas.* En este caso habrá encontrado un óptimo local, que no necesariamente será global. Matemáticamente, este mensaje indica que las condiciones de Karush-Kuhn-Tucker para óptimos locales han sido satisfechas. Salvo en un problema con un solo óptimo global, se debería ejecutar el «Solver» desde diferentes puntos iniciales para incrementar la seguridad sobre la globalidad del óptimo.
- *Solver ha convergido hacia la solución actual. Todas las restricciones están satisfechas.* En este caso el valor de la función objetivo cambia muy lentamente en las últimas iteraciones. La opción «Convergencia» controla este proceso. El algoritmo termina si el cambio relativo en el valor de la función objetivo durante varias iteraciones es menor que el factor de convergencia. Si se intuye que «Solver» finaliza demasiado rápido o que el punto obtenido no es óptimo, será preciso reducir la convergencia para evitar soluciones subóptimas.
- *Solver no puede mejorar la solución actual. Todas las restricciones están satisfechas.* Este mensaje indica que el modelo presenta degeneración y que el algoritmo ha entrado en un ciclo. La degeneración puede ser evitada en muchos casos eliminando restricciones redundantes.

También es importante tener en cuenta que «Solver» presenta dificultades en muchos casos para empezar a aplicar el algoritmo cuando se inicializa en un punto de valor nulo para todas las variables. Por tanto, es aconsejable comenzar por una solución no nula. Además, en la mayoría de los casos, cuanto más cercanos sean los valores iniciales al óptimo más rápido será el proceso de resolución.

El proceso de solución del GRG, al igual que otros muchos algoritmos de programación no lineal, calcula valores de la primera derivada parcial de la función objetivo y de las restricciones en cada iteración. La opción «Derivadas» fija cómo se realiza dicho cálculo. La alternativa «progresivas» considera conjuntamente el punto de la iteración anterior y el actual, con lo cual reduce el tiempo de computación requerido por la diferenciación finita (este tiempo se estima que puede llegar a suponer el 50 por ciento del tiempo total de resolución). La opción «centrales» tan solo considera el punto actual, lo cual conlleva un mayor tiempo de cálculo que puede sin embargo resultar adecuado si las derivadas cambian rápidamente ya que permite realizar un menor número de iteraciones. En problemas cuadráticos, la diferenciación central

produce valores de las derivadas exactos, lo cual permite mejorar la exactitud del resultado y reducir el número de iteraciones, aunque éstas tendrán un tiempo de ejecución que puede llegar a duplicar el de diferenciación progresiva.

El método del GRG realiza asimismo una reducción del problema original a otro sin restricciones resolviendo un sistema de ecuaciones para ciertas variables - básicas - en términos del resto - no básicas -. Entonces, se elige una dirección de búsqueda (un vector  $n$ -dimensional donde  $n$  es el número de variables no básicas) a lo largo de la cual se establece una mejora de la función objetivo. La opción «Hallar por» fija el criterio para determinar esta dirección de búsqueda. El método de Newton consiste realmente en el método cuasi-Newton BFGS (Broyden-Fletcher-Goldfarb-Shanno). En lugar de utilizar la matriz hessiana, utiliza una aproximación de dicha matriz, lo cual requiere una importante capacidad de almacenaje que sin embargo se compensa por los buenos resultados que genera. La alternativa es el método del «gradiente conjugado», que no requiere el almacenamiento de la matriz hessiana sino tan solo de algunos vectores. Normalmente requiere de más iteraciones que el método cuasi-Newton, siendo recomendable en el caso de problemas de gran tamaño. En el caso del «Solver», la elección de una u otra opción no resulta relevante dado que es capaz de cambiar automáticamente de uno a otro método en función de la capacidad de almacenamiento disponible, sea cual sea la opción elegida.

Por último, una vez elegida la dirección, el algoritmo realiza una búsqueda a través de dicha dirección variando la amplitud del desplazamiento para la mejora del objetivo reducido. Las estimaciones iniciales de los valores de las variables que experimentan un cambio tienen un impacto significativo sobre la efectividad del método. La opción «Estimación» indica cómo se realiza dicho proceso. La alternativa «lineal» utiliza una extrapolación lineal a partir de la tangente a la función objetivo reducida. La alternativa «cuadrática» extrapola a través de un ajuste cuadrático de dicha función en el punto actual. Salvo que la función objetivo reducida se ajuste a un modelo cuadrático, y si no se tiene ninguna información especial a cerca del comportamiento de la misma, la alternativa lineal es la más segura, si bien la más lenta.

#### **IV. LIMITACIONES, EFICIENCIA Y OBSERVACIONES SOBRE LA UTILIZACIÓN DEL SOLVER EN LA OPTIMIZACIÓN EMPRESARIAL**

- Aunque no presente la estructura que hemos propuesto en los epígrafes anteriores, se puede optimizar una hoja de trabajo con el «solver» tanto si el modelo es lineal como no. Si no se ajusta a dicha estructura, no suele ser fácil detectar si el modelo es lineal, y por tanto no se conoce la técnica a utilizar para su optimización. En este caso, resulta adecuado señalar la



opción «Asumir modelo lineal» ya que así se llevará a cabo un test de linealidad que nos indicará el carácter del modelo.

- Existen muchas funciones propias de la hoja de cálculo que presentan discontinuidades y que como consecuencia no pueden ser tratadas por el «solver». Una lista parcial de dichas funciones, que deben ser evitadas en la formulación de modelos a optimizar, incluye: ABS, MIN, MAX, ENTERO, REDONDEAR, SI, ELEGIR, CONTAR. En caso de duda sobre la continuidad de una función es recomendable su representación gráfica en el rango de valores considerado.
- Aunque la parte derecha de las restricciones puede ser cualquier expresión numérica, para evitar posibles errores es adecuado utilizar siempre constantes, o referencias a celdas que contienen valores constantes. Si la parte derecha depende de una de las variables de decisión, el programa transforma internamente la restricción pasando dicha expresión a su parte izquierda.
- El «Solver» reconoce el caso en que la parte izquierda de la restricción es una variable de decisión y la parte derecha una constante, tratando a las mismas como cotas superiores o inferiores, lo cual requiere menores tiempos de computación.
- No existen diferencias en términos de eficiencia entre una restricción del tipo  $A1 \leq 10$  ó  $A1 \leq A2$  donde A2 contiene el valor 10, dado que reconoce a la celda A2 como una constante. La segunda alternativa tiene ventajas para construir el modelo de forma más operativa. Tampoco existen diferencias en la eficiencia por el hecho de definir nombres para las celdas en lugar de utilizar referencias.
- Las fórmulas introducidas en la parte derecha de las restricciones (incluso expresiones con constantes como 2+1) incrementan los tiempos de solución del modelo. En este caso, el programa crea internamente una nueva restricción en la que pasa la parte derecha a la parte izquierda de la restricción con signo contrario. En estos casos, lo recomendable es realizar los cálculos en otra zona de la hoja y referenciar dicha celda en la parte derecha de la restricción. En ese caso, la hoja de trabajo ya habrá analizado esa expresión en la celda citada y el «Solver» puede determinar si depende de las variables de decisión.
- En el caso de funciones lineales, con el objeto de mantener el modelo más manejable y fácil de estructurar, es recomendable utilizar la función SUMAPRODUCTO, tal como se recoge en la figura I para obtener la función objetivo.
- La versión estándar del «Solver» que viene incorporada en la Excel 97 existe un límite de 200 variables de decisión en las celdas cambiantes. En función del tipo de modelo también existen limitaciones respecto al número de restricciones. Si el modelo es lineal y se señala la opción «Asumir modelo lineal» no existe límite en el número de restricciones. Si el

modelo no es lineal existe un límite de 100 restricciones, además de las correspondientes a cotas y a variables enteras.

- Debido a las limitaciones de la versión estándar existen disponibles versiones con mayor capacidad, incluso para programas de gran tamaño. Sus características se pueden contrastar en la dirección [www.frontsys.com](http://www.frontsys.com). En general se trata de versiones que incluyen una opción especial para programación cuadrática, y que permiten la escala automática también en los programas lineales. Por otra parte, su velocidad de ejecución multiplica por cien la capacidad estándar, los test de linealidad indican donde se quiebra dicha condición, poseen indicadores de progresividad sobre el tiempo total estimado de resolución, etc.
- Por defecto, la opción «Asumir modelo lineal» no está señalada, por lo que el método de optimización utilizado será el GRG. Si bien esto puede permitir llegar a la solución de un programa lineal, siempre será más rápido y seguro su resolución por el método simplex. Por tanto, se deberá ejecutar siempre dicha opción ya que, por otra parte, ello supone que los informes de sensibilidad sean más completos.
- En el proceso de solución, el «Solver» ajusta sólo los valores de las variables de decisión, permaneciendo constantes las celdas que no dependen de dichos valores. Dado que Excel 97 realiza un chequeo previo para conocer qué celdas tendrán cambios, se pueden lograr ahorros de tiempo si se eliminan celdas que no tienen relación directa con el programa. Una opción puede ser la de copiar el modelo en otra hoja, sin referencias de celdas ni hojas. Este proceso se puede automatizar mediante macros.

## REFERENCIAS BIBLIOGRÁFICAS

ABADIE, J.(1978): «The GRG Method for Nonlinear Programming» en Greenberg, H.J.(Ed), *Design and Implementation of Optimization Software*, Sijthoff and Noordhoof, pp. 325-363.

GOULD, F.J., G.D. EPPEN y C.P.SCHMIDT (1992): *Investigación de operaciones en la ciencia administrativa*. Prentice-Hall. México.

LASDON, L.S. y A.D. WAREN(1978): «Generalized Reduced Gradient Software for Linearly and Nonlinearly Constrained Problems» en Greenberg, H.J.(Ed), *Design and Implementation of Optimization Software*, Sijthoff and Noordhoof, pp. 363-397.

LASDON, L.S., A.D. WAREN, A. JAIN y M. RATNER(1978): «Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Constrained Programming», *ACM Transactions on Mathematical Software*, vol. 4, pp. 34-50.

PLANE, D.R.(1994): *Management Science. A spreadsheet Aproach*. Body and Fraser. Danvers (Massachussetts).

RÍOS, S. (1988): *Investigación operativa*. Centro de Estudios Ramón Areces. Madrid.