

# Herramienta gráfica para el aprendizaje del algoritmo simplex

Victoria Fernández, Izaskun Urdangarin, Ana Zelaia  
Dpto. de Ciencia de la Computación e IA  
Facultad de Informática  
Universidad del País Vasco  
victoria.fernandez@ehu.es, izaskunu@hotmail.com, ana.zelaia@ehu.es

## Resumen

Gran parte de los contenidos de Investigación Operativa, dentro de las titulaciones de Informática, consiste en el estudio de algoritmos de programación lineal. El trabajo que presentamos muestra una herramienta gráfica construida para el aprendizaje del algoritmo simplex. Dicha aplicación puede ser utilizada tanto por el profesorado en el aula como por el alumnado de forma autónoma, ya que dispone de una interfaz gráfica de manejo sencillo.

## 1. Motivación

En las titulaciones de Ingeniería Técnica en Informática de Sistemas e Ingeniería Informática de nuestro centro se imparte como asignatura obligatoria en el segundo curso la Investigación Operativa. La mayoría de los tópicos estudiados en esta asignatura son de programación lineal.

La programación lineal es una técnica exacta de resolución de problemas lineales de optimización con restricciones. Un modelo de este tipo surge en distintos contextos prácticos en la formulación de un sistema real. Aunque modelos de este tipo fueron estudiados anteriormente, se atribuye el desarrollo de la programación lineal a George B. Dantzig, que en 1949 publicó el método simplex, primer método construido para resolver problemas de estas características.

La programación lineal se basa en dos resultados importantes. El primero de ellos es que el conjunto de soluciones factibles es un conjunto convexo con un número finito de puntos extremos, y que la solución óptima se encuentra en uno de los puntos extremos. Por tanto, es suficiente calcular dichos puntos para poder dar una solución al problema. Sin embargo, para problemas grandes, sería muy costoso considerar todos los puntos extremos, y de ahí la importancia del método simplex que reduce significativamente el número de puntos a chequear. El segundo

resultado es el que establece que existe una correspondencia biunívoca entre el conjunto de puntos extremos y el de soluciones factibles básicas del sistema de restricciones en forma estándar. Precisamente, podremos observar que el algoritmo simplex va de una solución factible básica (punto extremo) a otra (punto extremo adyacente) que da a la función a optimizar un valor mejor hasta conseguir el óptimo, sin necesidad de recorrer todas las soluciones básicas factibles [1] [2] [4].

En nuestra experiencia como docentes hemos ido observando la dificultad que el alumnado tiene para comprender estos resultados teóricos, convirtiéndose en muchos casos el proceso de resolución en una sucesión de cálculos incomprensibles. Una manera de hacer comprensible el proceso consiste en realizar la solución gráfica de modelos lineales. Cuando el modelo consta de solo dos variables, cada una de las restricciones del modelo puede ser representada como un semiespacio en el plano y la función a optimizar como una recta dibujada en función del valor de  $z$ . Resolver un problema lineal de dos variables gráficamente significa desplazar la función a optimizar sobre los puntos que pertenecen simultáneamente a todos los semiespacios, hasta alcanzar el valor óptimo para  $z$ .

Es evidente que la resolución gráfica de problemas lineales no es un método práctico, debido al elevado número de variables de los problemas lineales que surgen en la práctica. Sin embargo, su utilización resulta de gran ayuda a la hora de interpretar la resolución algebraica. Todos los textos de programación lineal que conocemos dedican una parte a la solución gráfica de modelos lineales que sirve para familiarizar al alumnado con los distintos tipos de soluciones: única, múltiples, no acotada, infactibilidad. Pero en ninguno de ellos hemos visto que se resuelva al mismo tiempo gráficamente y algebraicamente un

modelo lineal para poder observar la correspondencia biunívoca que existe entre el conjunto de puntos extremos y el conjunto de soluciones factibles básicas. En las clases magistrales solemos resolver un ejercicio simultáneamente desde ambos puntos de vista, pero se ha visto que el tiempo que se le puede dedicar en el aula a este tipo de ejercicios es insuficiente. Además, trabajar con gráficas en la pizarra resulta poco atractivo para el alumnado.

Por ello, con el objetivo de paliar las deficiencias observadas, se pensó que sería de gran ayuda utilizar algún software de apoyo adecuado. En realidad, hay abundante software disponible para resolver modelos lineales (ver por ejemplo [3] y [5]), pero no hemos encontrado ninguno que muestre gráficamente el procedimiento algorítmico, como hacemos en esta aplicación.

Además de las motivaciones expresadas, valoramos importante el haber desarrollado la herramienta en los dos idiomas en los que la asignatura se imparte en nuestro centro: euskara y castellano.

## 2. La aplicación

La aplicación informática que presentamos ha sido diseñada específicamente para trabajar con el algoritmo simplex y dispone de una interfaz de usuario/a sencilla y cómoda de manejar. La pantalla principal (Figura 1) nos muestra las dos partes principales de las que consta la aplicación: repaso de conceptos teóricos y solución de modelos lineales.



Figura 1. Pantalla principal

Queremos destacar que la integración de los conceptos teóricos en la aplicación resulta muy atractiva para el alumnado, dado que en todo momento puede realizar consultas tanto para observar el algoritmo simplex, como para comprender los cálculos realizados sobre la tabla. Esto le proporciona una gran autonomía en el aprendizaje de la programación lineal.

Pero, sin duda, la aportación más importante de la aplicación que presentamos es poder observar gráficamente el recorrido del algoritmo simplex en la búsqueda de la solución óptima para un modelo lineal con dos variables. Esto permite comprobar que a cada solución factible básica de la tabla le corresponde un punto extremo en la región de factibilidad. Observémoslo a través de un ejemplo (ver Figura 2).

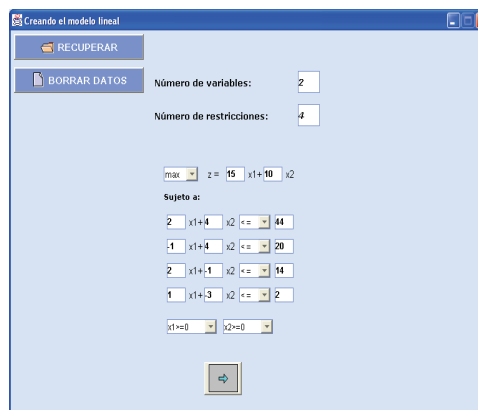


Figura 2. Ejemplo de modelo lineal

Como el modelo introducido tiene sólo dos variables, la aplicación muestra automáticamente la tabla del simplex y la gráfica que le corresponde. Así, en la Figura 3 se puede observar la tabla inicial, y la representación gráfica que corresponde a dicha tabla. Las rectas delimitan semiespacios y la zona sombreada corresponde al conjunto de soluciones del problema. La recta que aparece marcada en negro representa la función a optimizar. Se puede observar que en esta representación gráfica dicha recta pasa por el punto extremo  $x_1=0$ ,  $x_2=0$ , que es el punto extremo que corresponde a la solución factible básica de la tabla. Concretamente, en esa solución las variables  $x_1$  y  $x_2$  toman el valor cero por no

ser básicas. El valor de la función en este punto es  $z=0$ .

Progresivamente, en las Figuras 4, 5 y 6 la función objetivo aparece desplazada en el sentido de la optimización, pasando por los valores  $z=30$ ,  $z=140$  y  $z^*=210$ , respectivamente. Las flechas de las figuras muestran el recorrido de puntos extremos. Las soluciones factibles básicas correspondientes a cada uno de ellos se pueden encontrar en las respectivas tablas. En la Figura 6 se observa el punto extremo óptimo y en la tabla la solución factible básica óptima  $x_1^*=10$ ,  $x_2^*=6$ .

De esta manera, la interpretación del proceso algebraico de resolución resulta más intuitiva y de gran ayuda para lograr una completa comprensión del algoritmo simplex.

Hay que decir que la aplicación no se limita a resolver modelos de dos variables. Así, cuando el/la usuario/a introduce un modelo de más variables, se pueden ver las tablas de cada iteración, repasar los cálculos realizados, etc. pero sin solución gráfica.

La aplicación presentada ha sido desarrollada en el entorno JDeveloper 9.0.3 [6] para los sistemas operativos Windows 2000, Windows XP y Linux. Se ha creado el archivo simplex.jar que se puede ejecutar disponiendo de un interprete de Java. De la página web de la asignatura se puede descargar el fichero simplex.jar y se encuentran instrucciones más detalladas para su ejecución (<http://www.sc.edu/ccwikera/principal.html>).

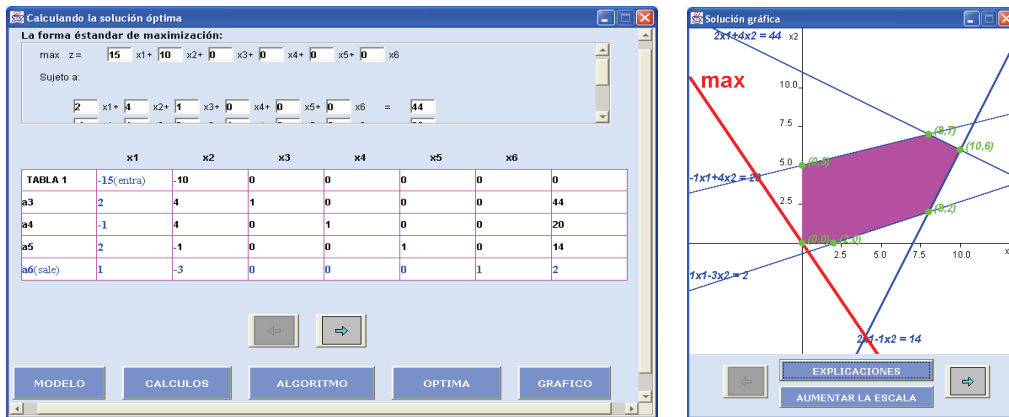


Figura 3. Tabla inicial del simplex y su correspondiente representación gráfica

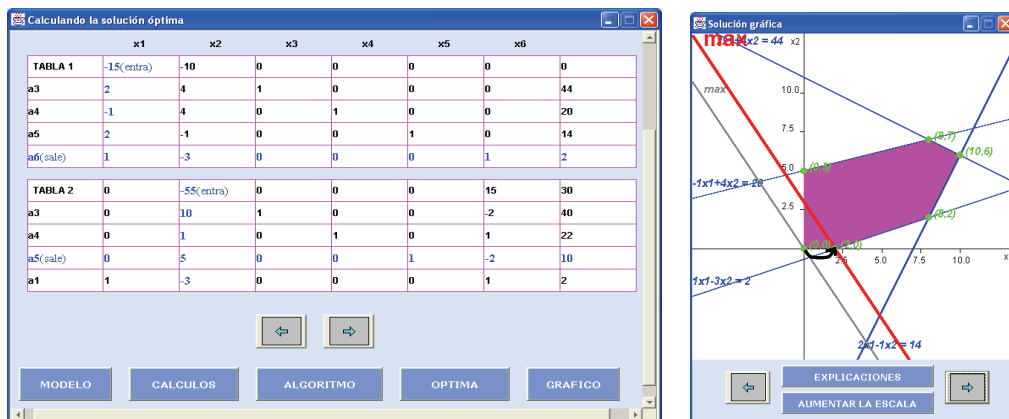


Figura 4. Primera iteración del simplex y su correspondiente representación gráfica

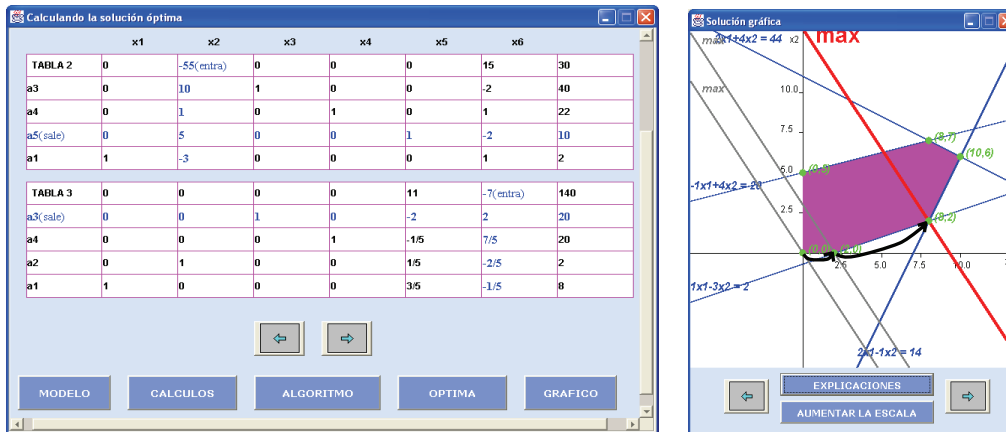


Figura 5. Segunda iteración del simplex y su correspondiente representación gráfica

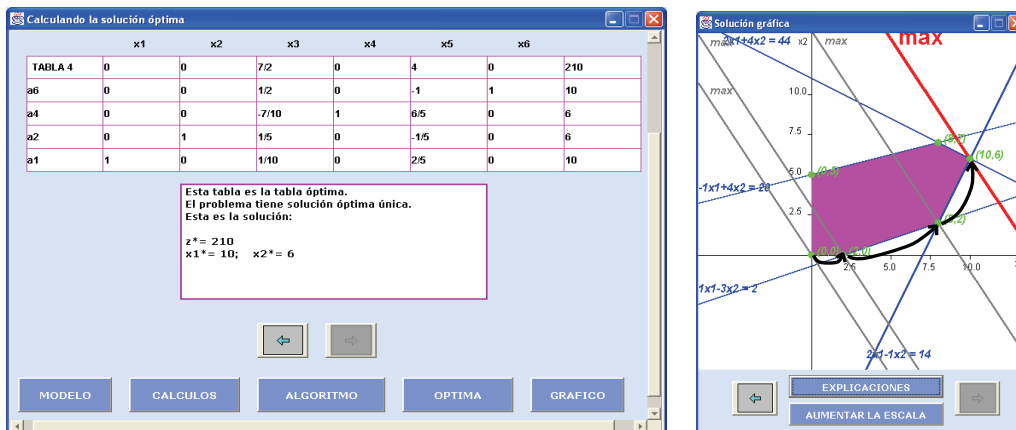


Figura 6. Tabla óptima del simplex y su correspondiente representación gráfica

### 3. Conclusiones y futuras líneas de trabajo

La aplicación ha sido desarrollada por la alumna Izaskun Urdangarin como proyecto fin de carrera en Ingeniería Técnica en Informática de Sistemas. Se ha utilizado por primera vez en el laboratorio durante el curso 05-06 y ha tenido una buena acogida por parte del alumnado. Aunque no disponemos todavía de datos objetivos como podrían ser las calificaciones de las correspondientes preguntas de examen, hemos podido observar, tanto en las clases magistrales como en las tutorías y laboratorios, que un importante porcentaje del alumnado ha llegado a

comprender con más facilidad los conceptos estudiados en las clases teóricas.

Esta primera experiencia nos ha permitido tanto a nosotras como al alumnado detectar posibles mejoras a la aplicación. Entre ellas, permitir a el/la usuario/a la elección manual de pivote, así como el cálculo de los puntos extremos de la solución gráfica, la construcción de la forma estándar del modelo, etc. En concreto en la selección manual del pivote, una elección errónea permitirá observar a el/la usuario/a que o bien retrocede a un punto extremo anterior (empeora) o bien la solución básica correspondiente sería infactible (fuera de la región).

Teniendo en cuenta la buena acogida por parte del alumnado, en nuevos proyectos fin de carrera estamos planteando el desarrollo de aplicaciones gráficas para el simplex dual, algoritmos de ramificación y acotación y para las técnicas de análisis de sensibilidad. El objetivo final es conseguir un software para programación lineal que permita al alumnado abordar de forma bastante autónoma el aprendizaje de estas técnicas, evitando el excesivo cálculo y pudiendo dedicar las clases magistrales al desarrollo y comprensión de los aspectos teóricos.

### Referencias

- [1] Bazaraa, M.S., Jarvis, J.J. and Sherali, H.D., *Linear Programming and Network Flows*. John Wiley & Sons, Inc., 1990.
- [2] Calvert, J.E. and Voxman, W.L., *Linear programming*. Harcourt Brace Jovanovich Inc., 1989.
- [3] Chang, Y. *QSB+ Quantitative Systems for Business Plus*. Prentice Hall, 1994.
- [4] Ríos Insua, S. *Investigación Operativa. Optimización*. Centro de Estudios Ramón Areces, 1988.
- [5] Scharage, L. *Lindo. An Optimization Modeling System*. The Scientific Press, 1991.
- [6] Tutoriales de Java:  
<http://www.cica.es/formacion/JavaTut>,  
<http://java.sun.com/docs/books/tutorial>

