



UNIVERSIDAD  
**COMPLUTENSE**  
MADRID

Proyecto de innovación  
Convocatoria 2021/2022

Número de proyecto: 118

Docencia de optimización en el entorno virtual Moodle  
a partir de ejercicios resueltos

Responsable del proyecto:

Pedro Miranda Menéndez

Facultad de Ciencias Matemáticas

Departamento:  
Estadística e Investigación Operativa

## 1.- Objetivos propuestos en la presentación del proyecto.

El objetivo del proyecto se englobaba en la línea

Innovación en recursos educativos en abierto y enseñanza virtual.

El objetivo principal del proyecto era desarrollar una aplicación que permitiese a los alumnos de Optimización, asignatura del Doble Grado de Matemáticas y Física, Doble Grado de Ingeniería Informática y Matemática y del Grado en Ingeniería Matemática, practicar los algoritmos vistos en clase referentes a la parte de Ampliación de Programación Lineal. Esta era una primera aproximación para ir generando programas que generen problemas en los que aplicar distintos tipos de algoritmos. En este caso nos planteábamos generar tres programas aplicados a tres tipos de algoritmos que aparecen en los contenidos de Optimización y para los cuales no se encuentran problemas resueltos. Estos algoritmos proporcionan mejoras computacionales del método simplex:

- Algoritmo del simplex revisado
- Algoritmo del simplex para variables acotadas
- Algoritmo primal-dual

En cada uno de ellos no sólo se trataba de que los alumnos comprobasen que obtienen la solución correcta, sino que se tenía la posibilidad de plantearles cuestiones para que pudiesen evaluar la comprensión de los entresijos teóricos de los algoritmos.

Para ello, se ha usado el entorno exams del lenguaje de programación R, que permiten la inclusión de los programas en el campus virtual, además de ser un software libre. La bibliografía fundamental en la que nos hemos basado es

- **Flexible Generation of E-Learning Exams in R: Moodle Quizzes, OLAT Assessments, and Beyond.** Zeileis, N. Umlauf, F. Leisch. *Journal of Statistical Software*, 58 (1), (2014).

Un objetivo secundario que permite la aplicación es poder evaluar de forma online los contenidos de la asignatura. Pero este objetivo no se plantea para que el profesor evalúe a los alumnos (aunque es directo hacerlo) sino que se trata de que los alumnos puedan autoevaluarse y ver si han asimilado correctamente los algoritmos.

## 2.- Objetivos alcanzados

Se han desarrollado varios programas (en concreto 8) que generan los datos de problemas de los algoritmos nombrados anteriormente de forma aleatoria.

Hemos planteado 8 ficheros RnW diferentes debido a que ha sido necesario diferenciar en función del algoritmo utilizado, el número de variables artificiales que se usan (salvo en el caso del algoritmo primal-dual) y el tipo de solución del problema (salvo nuevamente en el caso del algoritmo primal-dual). Más concretamente, los ficheros que se han desarrollado son:

- Algoritmo simplex revisado con solución óptima finita sin variables artificiales.
- Algoritmo simplex revisado con solución óptima finita con variables artificiales.
- Algoritmo simplex revisado con solución no acotada.
- Algoritmo simplex revisado infactible.
- Algoritmo del simplex para variables acotadas con solución óptima finita sin variables artificiales.
- Algoritmo del simplex para variables acotadas con solución óptima finita con variables artificiales.
- Algoritmo del simplex para variables acotadas infactible.
- Algoritmo primal-dual.

### 3.- Metodología empleada en el proyecto

El proyecto se estructuraba en dos fases diferenciadas.

1. **FASE 1:** Construir la aplicación que permitiese generar los problemas de los distintos algoritmos de forma aleatoria. Esta parte se había planteado para ser desarrollada en el primer cuatrimestre para que los programas estuviesen terminados debería estar terminada antes del comienzo de las clases de la asignatura, que es una asignatura del segundo cuatrimestre. En esta fase se han fijado también la forma de plantear el problema. Se ha decidido para no desanimar a los alumnos y ya que el objetivo es practicar el funcionamiento de los algoritmos que el enunciado sea lo más sencillo posible y que el número de preguntas sea lo más reducido posible.
2. **FASE 2:** Probar y corregir posibles problemas que pudiesen aparecer en el uso práctico de la aplicación. Esta parte estaba planteada para ser realizada en el segundo cuatrimestre, ya que sería en este momento en el que los alumnos pueden probar los programas y sugerir mejoras o detectar errores que se pudieran haber pasado por alto en la primera etapa.

La metodología que se ha seguido con este proyecto es permitir a los alumnos practicar estos algoritmos de forma autónoma y dedicando el tiempo que cada alumno considere necesario. En ningún momento hemos accedido a las respuestas o número de ensayos que cada alumno ha realizado porque se ha intentado que el alumno esté cómodo y que no se sienta evaluado al utilizar la aplicación.

#### **4.- Recursos humanos**

El equipo que ha realizado este proyecto está formado por:

Responsable: Pedro Miranda Menéndez [pmiranda@ucm.es](mailto:pmiranda@ucm.es)

Equipo: Elena María Castilla González [elecasti@ucm.es](mailto:elecasti@ucm.es)

Equipo: Pedro José Chocano Feito [pedrocho@ucm.es](mailto:pedrocho@ucm.es)

Equipo: María Jaenada Malagón [mjaenada@ucm.es](mailto:mjaenada@ucm.es)

Equipo: Nirian Martín Apaolaza [nimartin@ucm.es](mailto:nimartin@ucm.es)

Equipo: Susana Martínez Suárez [smartinezsuarez@gamil.com](mailto:smartinezsuarez@gamil.com)

Equipo: Leandro Pardo Llorente [lpardo@ucm.es](mailto:lpardo@ucm.es)

## 5.- Desarrollo de las actividades

El proyecto se ha desarrollado según se había explicado en la solicitud.

- En primer lugar se ha tenido que programar los algoritmos en R. Aunque R tiene una librería para resolver programas de Programación Lineal, no tiene una librería que resuelva este tipo de problemas mediante los algoritmos de mejora del funcionamiento del simplex. Por ello, hemos tenido que desarrollar nosotros mismos estos algoritmos. Esta primera etapa estaba prevista en el Plan de Trabajo y nos ha llevado casi los cuatro primeros meses del curso. De esta manera se han generado tres programas de R que resuelven los problemas de Programación Lineal mediante los algoritmos del Simplex Revisado, Simplex para variables acotadas y el algoritmo Primal-Dual.
- Una vez generados los distintos programas, era necesario incluirlos como ficheros de formato RnW para poder utilizarlos con la librería exams de R. Se trataba aquí de combinar los ficheros desarrollados en la etapa anterior con un fichero de tipo .tex en el que se plantean las preguntas que se hacen en cada problema y que proporciona la solución del mismo. Aunque proporcionar la solución es algo opcional en la librería exams, precisamente éste era el objetivo del proyecto, ya que no se trataba de que los alumnos practicasen, sino que pudiesen ver la solución de cada problema completamente desarrollada para comprobar que se ha asimilado el funcionamiento de cada uno de los algoritmos. Por ello, la parte de Latex del programa en RnW es especialmente larga. Esta parte nos ha llevado casi otro mes de trabajo. Un ejemplo de la vista que aparece a los alumnos en el campus virtual al utilizar la aplicación se puede ver en la Figura 2 del anexo.
- Una dificultad que no habíamos previsto aparece con el entorno que proporciona Moodle para mostrar las soluciones. Aunque cuando se generan estas soluciones como un fichero .pdf la solución desarrollada en Latex queda claramente definida, este mismo programa no da una solución que sea tan visualmente clara en el formato Moodle y eso ha llevado también un tiempo ya que es necesario imponer espacios, saltos de línea, etc en el programa RnW (en la parte de Latex) sin poder ver cómo se verán traducidas esas instrucciones en la versión final, ya que no coincide con la que se genera para el formato pdf. Esto puede verse en las diferencias que aparecen entre las Figuras 1 y 3 del anexo, en el que se puede ver la diferencia entre las salidas en formato pdf y la que produce la plataforma Moodle.
- El fichero RnW permite que se escriban variables o datos desconocidos y que luego serán generados por el correspondiente programa. Sin embargo, no permite considerar como algo variable el número de tablas que se van a utilizar en la solución. Es decir, no permite que el número de tablas pueda variar en los distintos problemas, pues necesita fijar de antemano el número de tablas que se van a escribir en la solución. Esto ha supuesto un gran problema a la hora de desarrollar el proyecto, puesto que el objetivo del proyecto consistía en implementar un algoritmo iterativo (en el que no conocemos a priori el número de iteraciones que se van a realizar hasta llegar a la solución óptima) y no en realizar unos cálculos que son siempre los mismos pero variando los números utilizados, que es lo que sucede habitualmente en Estadística.
- El entorno exams está desarrollado pensando en esta última opción y esto nos ha obligado a introducir una instrucción de tipo while en cada programa RnW

que no teníamos en los programas R para fijar de antemano el número de iteraciones. Esta instrucción while fuerza al programa a generar problemas hasta que se dan las condiciones de la instrucción. En nuestro caso, la condición era que el número de tablas del simplex quedaba predeterminada.

- Otra dificultad relacionada con la anterior es que la amplitud de las tablas debe ser fijada de antemano, ya que las dimensiones de las tablas quedan fijadas en el fichero RnW. Esto nos ha forzado a determinar el número de variables que había en cada problema, así como el número de restricciones. La dificultad en este caso ha venido dada porque en un problema generado al azar no se puede controlar el número de variables artificiales que serán necesarias para poder iniciar los distintos algoritmos. Por ello, hemos tenido que imponer condiciones que fijasen el número de dichas variables. Esto ha sido necesario en el simplex revisado y en el simplex para variables acotadas. En el algoritmo primal-dual el número de variables artificiales coincide con el número de restricciones y por lo tanto está fijado por las condiciones del problema. Además, mientras que en el simplex revisado las dimensiones de las tablas son siempre las mismas independientemente del número de variables artificiales y éstas sólo aparecen en los cálculos que realiza el algoritmo, en el caso del simplex para variables acotadas cada variable artificial da lugar a una nueva columna en la tabla. Como es necesario fijar las dimensiones de cada tabla, esto ha hecho que se haya tenido que fijar el número de iteraciones con variables artificiales (es decir, el número de iteraciones de la Fase I del algoritmo) y el número de iteraciones de la Fase II.
- Hemos fijado el tipo de solución (solución óptima finita, solución no acotada, problema infactible) de cada problema introduciendo esta condición como una nueva condición de la instrucción while.
- Finalmente, el entorno exams está desarrollado para ser aplicado en evaluaciones online en la que cada alumno tiene un ejercicio con datos diferentes pero de una dificultad similar. Por ello, no es posible incorporar los programas como aplicaciones en el campus virtual de la asignatura, sino que es necesario incluirlos como cuestionarios.

Hemos planteado 8 ficheros RnW diferentes, de forma que se diferencian entre sí en función del algoritmo utilizado, el número de variables artificiales que se usan (salvo en el caso del algoritmo primal-dual) y el tipo de solución del problema (salvo nuevamente en el caso del algoritmo primal-dual).

Otro aspecto del entorno exams que nos ha sorprendido es que esperábamos generar problemas y que la aplicación permitiese que el alumno realizase un número indefinido de problemas, de forma que en cada caso ejercicio se generasen datos de forma aleatoria. Sin embargo, el entorno exams sólo permite que el alumno realice un ejercicio seleccionado al azar de entre una batería de problemas previamente generados por la aplicación y cuyo número está prefijado en la ejecución del programa. Para solventar este problema y que los alumnos tuviesen a su disposición un número suficientemente grande de problemas, hemos generado una cantidad elevada de problemas, lo que hace que se cumplan las condiciones que se pretendían conseguir inicialmente. La instrucción de ejecución sería de la forma

exams2moodle("../Desktop/R/Final/VariablesAcotadas/Ejercicio3-2Fases.Rnw", 20)

donde el 20 es el número de problemas que queremos que se generen.

Todos estos ficheros han sido puestos a disposición de los alumnos del Doble Grados en Matemáticas y Física y del Doble Grado en Ingeniería Informática y Matemáticas en el campus virtual de la asignatura de Optimización.

La aplicación ha sido utilizada 60 veces por parte de los 56 alumnos del Doble Grados en Matemáticas y Física y del Doble Grado en Ingeniería Informática y Matemáticas matriculados en la asignatura. Esta discrepancia es debida a que, al tener que plantear la aplicación como un cuestionario con distintas preguntas, si el alumno quiere repetir los ejercicios con otros datos es necesario que repita el cuestionario. En total 26 alumnos han hecho uso de la aplicación. Y en principio la impresión después de preguntar a los alumnos sobre su experiencia es que ha sido una herramienta útil para seguir el funcionamiento de estos algoritmos.

## 6.- Anexos

### 1. Problem

Resuelve el siguiente problema de programación lineal mediante el método simplex para variables acotadas de minimizar con los siguientes datos:

$$c = (5, -4, 5)$$

$$b = (-3, 4)$$

$$A = \begin{pmatrix} 1 & -1 & 1 \\ 2 & 3 & 5 \end{pmatrix}$$

$$0 \leq x_1 \leq 5, 0 \leq x_2 \leq 6, 0 \leq x_3 \leq 1.$$

Determinar el tipo de solución del problema: 1- s.o. finita. 2- s. no acotada. 3- Infactible.

### Solution

La solución es 3, es decir, es un problema infactible.

Comenzamos con las variables en su cota inferior. De esta forma, los valores del vector de términos independientes después de hacer las operaciones quedan:

$$-3, 4.$$

Entonces tenemos que aplicar el método de las dos fases.

### FASE I.

La primera tabla es:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	
	0	0	0	0	0	-1	0
$x_6$	-1	1	-1	-1	0	1	3
$x_5$	2	3	5	0	1	0	4

Esta tabla no es una verdadera tabla del simplex. Le sumamos la fila en la que la artificial es básica para que nos quede una tabla del simplex.

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	
	-1	1	-1	-1	0	0	3
$x_6$	-1	1	-1	-1	0	1	3
$x_5$	2	3	5	0	1	0	4

### ITERACION 1:

Entra  $x_2$ .

Entonces:  $\lambda_1 = 4/3, \lambda_2 = Inf, u - l = 6$ .

Sale  $x_5$ .

La nueva tabla es:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	
	-5/3	0	-8/3	-1	-1/3	0	5/3
$x_6$	-5/3	0	-8/3	-1	-1/3	1	5/3
$x_2$	2/3	1	5/3	0	1/3	0	4/3

Y con esto se termina la Fase I. El problema es infactible.

Figura 1: Ejemplo de funcionamiento de la aplicación en formato pdf.

# OPTIMIZACIÓN (único)

[Página Principal](#) / [Mis cursos](#) / [21-653159](#) / [Tema 1](#) / [Cuestionario 2](#) / [Vista previa](#)

Pregunta **1**

Sin responder  
aún

Puntúa como  
1,00

🚩 Marcar  
pregunta

⚙ Editar  
pregunta

Resuelve el siguiente problema de programación lineal mediante el método primal-dual de minimizar con los siguientes datos:

$$c = (5, 4, 2, 2, 2, 4, 4)$$

$$b = (2, 4)$$

$$A = \begin{pmatrix} -3 & -1 & -3 & 1 & 2 & 3 & -3 \\ -3 & -3 & 2 & 3 & -3 & 2 & 0 \end{pmatrix}$$

Determinar el valor óptimo de la función objetivo.

Respuesta:

Figura 2. Interfaz que aparece en el campus virtual y que ven los alumnos.

La solución que aparece en la última tabla es 0.2.

En este caso tenemos que aplicar el método de las dos fases.

Añadimos una variable de holgura, que denotaremos  $x_{10}$ .

Las tablas del simplex para la Fase I, los coeficientes de las variables para estudiar la optimalidad y el vector  $B^{-1}a_i$  en cada iteración son:

PRIMERA ITERACIÓN:

	1	0	0	1						
$x_{10}$	1	0	0	1						
$x_8$	0	1	0	3						
$x_9$	0	0	1	2						
$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	
-3	-3	-3	2	2	-3	-1	0	0	0	

Entra  $x_4$ .

$$B^{-1}a_i = \begin{pmatrix} 2 \\ -2 \\ 5 \end{pmatrix}$$

Sale  $x_9$ . Nos quedaría ahora la tabla:

	1	0	-2/5	1/5						
$x_{10}$	1	0	-2/5	1/5						
$x_8$	0	1	2/5	19/5						
$x_4$	0	0	1/5	2/5						

SEGUNDA ITERACIÓN:

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	
-5	-5	-21/5	0	0	-19/5	-1	0	-2/5	0	

Y entonces ya se termina la Fase I.

La solución es de tipo *Problema – infactible*.

Figura 3. Ejemplo de vista de las respuestas que aparecen en la pantalla tras acabar el intento.