

## SOLVERS COMERCIALES: ¿LA MEJOR ALTERNATIVA PARA LA ENSEÑANZA?

### *PAID SOLVERS: ARE THOSE THE BEST CHOICE FOR THE TEACHING WORK?*

**Juan Pablo Caballero Villalobos, Eliana María González Neira y Olga Lucía Araoz Cajiao**  
Pontificia Universidad Javeriana, Bogotá (Colombia)

**Rafael Guillermo García Cáceres**  
Escuela Colombiana de Ingeniería Julio Garavito, Bogotá (Colombia)

#### Resumen

Este artículo presenta un análisis exploratorio sobre la conveniencia y factibilidad del uso de *software* libre o *software* comercial como herramienta de apoyo en el proceso de enseñanza de asignaturas de programas de ingeniería en donde se desarrollan competencias en modelación matemática. En particular, este trabajo presenta elementos para motivar la discusión respecto a la escogencia de *solvers* comerciales o libres para la enseñanza de modelos lineales y mixtos y su solución, como fase de iniciación al campo de modelado matemático. Se desarrolló una fase experimental usando instancias de la librería NETLIB para evaluar el desempeño en tiempo de ejecución y capacidad para hallar solución a las instancias como posible criterio fuerte al momento de sustentar la escogencia de las opciones comerciales sobre las de código libre. Se evaluaron dos aplicaciones de software libre y tres de software comercial encontrándose que estadísticamente no hay diferencias significativas en la calidad de las soluciones reportadas por los *solvers* comerciales y uno de los *solvers* libres, apoyando la idea del *software* libre como alternativa apropiada para el proceso de enseñanza y solución de los mencionados modelos en el entorno académico.

**Palabras claves:** optimización, programación lineal, software libre, educación ingeniería

#### Abstract

This paper presents an exploratory analysis on the convenience and feasibility of using free software or commercial software as a support tool in the teaching of subjects in engineering programs develop competence in mathematical modeling. In particular, this work presents elements to encourage discussion about the choice of commercial or free solvers for teaching linear mixed models and their solution, as initiation phase to the field of mathematical modeling. The experimental stage used the linear instances

provided by the NETLIB library and its quality outcome was compared through the CPU time and the ability to provide solution to the instances in order to evaluate it as a strong criterion that will support the choice of commercial solvers instead of the open source counterparts. Two open source solvers and three well known commercial solvers were tested finding out there is no significant difference between the performances reported by the commercial application and one of the open source counterpart. As a result, the use of open source solvers should be considered as a real alternative in the teaching and solving process in the academic context.

**Keywords:** *open source solver, linear programming, optimization, engineering education*

## Introducción

La modelación matemática de fenómenos y situaciones reales permite mejorar su comprensión, facilita la comunicación de conceptos entre diversos actores y provee un marco sobre el cual se especifican los recursos disponibles, sus limitaciones e interacciones. Estas bondades de los modelos matemáticos son potenciadas, si adicionalmente se cuenta con algoritmos exitosos al proveer soluciones numéricas que satisfacen sus requerimientos. Debido a estos beneficios mencionados, cada vez es más frecuente el empleo de modelos matemáticos como herramienta de soporte en procesos de toma de decisiones en diversas áreas del conocimiento; en parte debido a la amplia oferta de algoritmos eficientes, a las presiones competitivas propias de la globalización y a las mejoras obtenidas en hardware en las últimas décadas (Winston, 2004); (Hillier et al., 1997); (Bazaraa et al., 1977); (Lasdon, 1970).

Como es de esperar, ante la gran diversidad de fenómenos y situaciones reales existentes, también existe una amplia gama de familias de modelos matemáticos, tales como lineales, mixtos, cuadráticos, combinatorios, entre otros. Estas familias agrupan modelos matemáticos similares con respecto a tipos de variables e interacciones empleadas, así como a tipos de relaciones usadas para expresar restricciones o funciones objetivo.

Dentro de la amplia gama de modelos matemáticos existentes, los lineales, los mixtos enteros-lineales, los mixtos binarios-lineales y los enteros-binarios revisten especial importancia en el desarrollo de competencias para estudiantes de programas de ingeniería, debido a que permiten formular un buen número de situaciones de su interés y adicionalmente afianza las bases teóricas necesarias para modelar situaciones más complejas usando otra

clase de modelos. Por este motivo, es frecuente que en los planes de estudio de carreras de ingeniería se encuentren asignaturas específicamente diseñadas para alcanzar estas competencias, contemplando el uso de estrategias pedagógicas que involucran el uso de *solvers* especializados para obtener soluciones numéricas a los modelos matemáticos planteados. Estas prácticas buscan que el estudiante modele, experimente y encuentre la solución a problemas planteados, refuerce conceptos teóricos previamente impartidos, y verifique y dimensione la importancia práctica de las técnicas tratadas en el aula de clase.

A pesar de la importancia manifiesta de afianzar conocimientos a través de la experimentación, esta oportunidad -bajo ciertas condiciones- se limita o se niega debido a la creación involuntaria de barreras de acceso a las herramientas de apoyo al aprendizaje. Ejemplos de este fenómeno surgen cuando se emplea software licenciado para apoyar el proceso educativo, software que en ocasiones sólo está disponible para los estudiantes en ciertas salas de cómputo de su institución y casi siempre con imposibilidad de usarse desde fuera de estas instalaciones.

En el caso de los programas de ingeniería en Colombia, y para efecto de soportar el proceso de aprendizaje ya mencionado, se encuentra una amplia gama de *solvers* tales como GAMS®, XPRESS®, LINGO®, GUROBI®, GLPK, entre otros. Ante tal diversidad y las complicaciones implícitas, este trabajo no pretende ser exhaustivo en cuanto a la comparación de todos los posibles *solvers* encontrados en nuestro entorno, pero si pretende aportar elementos de juicio al debate sobre la posibilidad académica de uso de software libre para apoyar el proceso de enseñanza en asignaturas que requieren de la solución de modelos lineales y mixtos de programación matemática. Para ello el artículo explora, a través de la medición del tiempo de solución y la capacidad para solucionar

el conjunto de instancias lineales bien conocidas (librería NETLIB), el desempeño de una muestra conformada por dos *solvers* libres: GLPK (GNU Linear Programming Kit) y Lp\_solve, y tres comerciales: LINGO® en su versión 6.0 SUPER®; XPRESS® versión 7.2 y GAMS® versión 23.5. Las instancias de prueba fueron tomadas y las soluciones obtenidas de la ejecución de los *solvers* fueron analizadas con pruebas estadísticas t para determinar posibles diferencias en sus desempeños que pudieran soportar de manera contundente el uso de *solvers* comerciales como única alternativa para apoyo al proceso de enseñanza.

Es tan abundante el número de estudios sobre el desempeño de algoritmos, heurísticas y hardware utilizado para la solución de problemas específicos de optimización, que resulta imposible realizar una revisión en un artículo. Sin embargo, los resultados de la revisión bibliográfica realizada, no permitieron referenciar trabajos que trataran sobre el desempeño entre diferentes *solvers* en el contexto de apoyo a la enseñanza en programas de educación superior.

### **Desarrollo**

La organización de este artículo comprende las siguientes secciones: 1. La oferta de *solvers*. 2. Concepto de *software* libre y características de los *solvers* contemplados. 3. Fase de experimentación y de comparación de características y 4. Conclusiones y recomendaciones.

#### **1. Acerca de los solver de PL**

La oferta de *solvers* de programación lineal es relativamente amplia en la actualidad. Dentro de la gama de software comercial se destacan marcas como CPLEX®, AMPL®, GAMS®, LINDO® y XPRESS®, mientras que GLPK (*Gnu Linear Programming Kit*) y Lp\_solve se reconocen como las opciones de *software* libre más populares. Los solvers libres tratados aquí se reconocen como las aplicaciones de código abierto más prominentes en este campo (Fourer, 2005) y han sido empleados por diversos autores en sus investigaciones (Mashaei et al., , 2013); (Yi, 2013); (Bachouch et al., 2012); (Bergamaschi et al., 2012); (Rossi et al., 2012); (Montoya-Torres et al., 2010); (Han et al., 2010); (Bohle et al., 2010); (Mitsutoshi et al., 2009); (Salvietti et al., 2009); (Fu et al., 2009); (El-Sayed et al., 2009);

(Ang et al., 2007) entre otros. Por su parte, los costos de las licencias comerciales difieren del tipo de uso siendo significativamente menor el costo de las versiones educativas, a su vez los costos de las versiones pueden también diferir según el tamaño de los problemas a tratar. Opciones de punto interior se encuentran disponibles en el mercado para resolver problemas de programación lineal, aunque estas no fueron tratadas en este artículo por sus limitados ambientes de uso y oferta. Es conveniente comentar que la aplicación de los algoritmos de punto interior ha probado ser especialmente útil en contextos de alta escala de optimización donde muestra un mejor desempeño que el algoritmo simplex.

Tradicionalmente, los criterios tenidos en cuenta para la evaluación de software de optimización incluyen: a. el análisis de desempeño de los *solvers* estudiados, b. la estabilidad numérica del *solver*, c. amigabilidad de la interface visual, d. facilidad de manejo, e. flexibilidad y facilidad del lenguaje de programación utilizado por el *software*, f. flexibilidad y facilidad para utilizar interfaces de entrada y salida de datos, g. aceptabilidad de desarrollo de aplicaciones independientes, h. si permite ser usado mediante una invocación de librería dinámica en tiempo de ejecución, i. si cuenta con la posibilidad de manipular el código fuente para darle una aplicación especial, j. su flexibilidad de funcionar en diferentes sistemas operativos y k. posibilidad de tener acceso a la programación de los tipos de algoritmos que se utiliza para resolver el modelo matemático.

Los anteriores criterios son diferentes en su naturaleza a los relacionados con el desempeño algorítmico pues son de tipo cualitativo, cuyo análisis tradicionalmente se relaciona con el uso de técnicas de toma de decisiones; con todo, el desarrollo de trabajos en este sentido se manifiestan como perspectivas de investigación sugeridas de este trabajo. La selección de los tres primeros criterios para el estudio se debe a que son los criterios más utilizados para la comparación de algoritmos de optimización (Ahuja et al, 1993), y porque la naturaleza cualitativa de algunos de los otros criterios sobrepasa los propósitos de análisis de este trabajo.

#### **2. Solvers contemplados y sus características**

El concepto de *software* libre (*Free Software Foundation*, 2009) se refiere a la libertad de los

usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el *software*. De forma más concreta, este término hace referencia a cuatro libertades de los usuarios del *software*:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades (libertad 1). Esta libertad implica que el acceso al código fuente esté disponible sin ninguna restricción.
- La libertad de distribuir copias, con lo que se puede ayudar a otras personas o instituciones (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). Esta libertad implica que el acceso al código fuente esté disponible sin ninguna restricción.

#### a. *LP\_Solve*

*Lp\_solve* es un *solver* para modelos de programación lineal, entera y mixta que es distribuido a través de la licencia LGPL (*Lesser General Public Licence*, 1999) del proyecto GNU iniciado en 1984 (GNU, 2009) y fue desarrollado inicialmente por Michel Berkelaar en Eindhoven University of Technology, Holanda. Las características de este licenciamiento propenden por garantizar la posibilidad de compartir y cambiar *software*, asegurando que sea libre para los usuarios. Adicionalmente, no impone límites explícitos en el tamaño de los modelos – número de restricciones y columnas - e interpreta modelos escritos en diferentes lenguajes de modelamiento matemático tales como: LP®, MPS®, MATHPROG® versión 4.7, CPLEX® versión 1, LINGO® versión 1.

El mecanismo de colaboración y documentación que utiliza la comunidad de usuarios de *Lp\_Solve* se basa en listas de correo, y un grupo de interés en yahoo (*Lp\_Solve*, 2009), donde se encuentra una amplia gama de recursos, manuales, foros y respuestas a preguntas frecuentes. Algunas características de *Lp\_Solve*, adicionales a las presentadas en la tabla 1, son: permite escoger la regla de pivoteo; puede utilizar una solución básica especificada por el usuario como punto de partida de los algoritmos implementados, o utilizar la anterior solución encontrada a un modelo que fue solucionado y posteriormente modificado; permite convertir un modelo matemático a los diferentes

lenguajes de modelado matemático que interpreta y muestra la información completa para realizar los análisis de sensibilidad post óptimo.

#### b. *GLPK (GNU Linear Programming Kit)*

Este Kit es un conjunto de rutinas escritas en *C ANSI* que han sido organizadas en forma de librería y que permite solucionar problemas numéricos provenientes de modelos de programación lineal de gran tamaño y modelos lineales mixtos. *GLPK* es mantenido gracias a la colaboración de Andrew Makhorin del Departamento para matemáticas aplicadas del Instituto de Aviación de Moscú. *GLPK* puede interpretar modelos matemáticos escritos en *GNU MATHPROG®*, que es un subconjunto del lenguaje de modelado *AMPL®*.

La distribución actualizada y la documentación de *GLPK* está conformada de un manual de referencia y la descripción del lenguaje de modelado matemático *GNU MathProg*. Los dos documentos se encuentran disponibles en varios formatos, *LaTeX®*, *DVI®* y *PostScript®*. Una traducción al español del manual del lenguaje de modelado *GNU MathProg* se encuentra disponible en la página web del grupo de investigación *Zentech* del departamento de Ingeniería Industrial de la Pontificia Universidad Javeriana. Adicionalmente es posible suscribirse a las listas de distribución de este proyecto, las cuales permiten estar en permanente conocimiento de su desarrollo.

#### c. *LINGO®*

*LINGO®* es uno de los más populares paquetes comerciales de optimización desarrollado a través de versiones periódicas por *LINDO SYSTEMS INC* que cuenta hoy con su versión 11.0. El *solver* PL permite definir de manera relativa o absoluta el error de redondeo. Las versiones de *LINGO®* dependen del número de variables y restricciones que manejan que van desde la versión libre *DEMO* desarrollada para el aprendizaje del lenguaje de modelamiento, hasta las versiones comerciales *SUPER* para 200 variables, *INDUSTRIAL* para 1000 variables y *EXTENDIDA* sin límite de variables y restricciones en los modelos; para este último caso las restricciones reales tienen que ver con la capacidad de procesamiento (tanto en tiempo *CPU* como de memoria *RAM*) y de la complejidad misma del problema. (*LINGO SYSTEMS*, 2006).

## d. XPRESS®

XPRESS® es uno de los paquetes de optimización más grandes del mercado, desarrollado por FICO. Permite resolver problemas de programación lineales, enteros, no lineales y estocásticos. El modelamiento se hace en el lenguaje MOSEL® que incluye un editor drag-and-drop que permite construir rápida e interactivamente sistemas visuales de soporte a la toma de decisiones fáciles de interpretar. Tiene tres motores de solución que cuentan con interfaz de programación de aplicaciones en lenguajes como C, C++, VB, Java, .NET. La versión comercial tiene una capacidad de 2 billones de coeficientes (“nonzero elements”), lo cual implica que es prácticamente ilimitada. FICO® ofrece a las instituciones académicas una membresía de dos tipos para que dichas instituciones obtengan, con fines académicos, la licencia completa de XPRESS® a ningún o bajo costo, dependiendo del servicio requerido, y con permanencia dependiente del reporte de uso enviado por la institución.

## e. GAMS®

GAMS® fue el primer lenguaje de modelamiento algebraico creado, cuya versión más reciente es la 23.9.5. GAMS® está conectado con un grupo de solvers de optimización tales como BARON®, COIN®, CONOPT®, CLPlex®, DICOPT®, GUROBI®, MOSEK®, SNOPT®, y XPRESS®. El programa es ofrecido en versión demo, completa y académica. La primera de ellas tiene límite para 300 variables y 300 restricciones, 2000 coeficientes y 50 variables discretas (incluyendo semi-enteras, semi-continuas). La versión completa es ilimitada. La versión académica cuenta con un módulo básico ilimitado que incluye el solver LOGMIP®, pero los demás solvers se comportan como la versión demo, a no ser que la institución pague por la versión completa en cada uno de ellos dependiendo del número de usuarios.

La tabla 1 presenta un compendio de las características más importantes de los cinco solvers comparados.

Tabla 1. Características de los solvers comparados

Características Generales del Software										
Producto	Solver disponible		Ambiente de Modelado		Aplicación Independiente		Procedimiento / Librería invocable	Código Fuente	Complementos para:	
GLPK	sí		no		sí		sí	sí	no	
LINGO®	no		sí		sí		sí	no	Excel	
LPSOLVE	sí		sí		sí		sí	sí	no	
XPRESS®			sí		sí		sí	no	Excel	
GAMS®			sí		sí		sí	no	no	
Características E/S del Software								Adquisición de datos ( lectura/escritura )		
Producto	PC / Windows		PC / Linux		Unix y variantes		Otros S.O.	a hoja de cálculo	a bases de datos	a archivos de texto
	32-bits	64-bits	32-bits	64-bits	32-bits	64-bits		a hoja de cálculo	a bases de datos	a archivos de texto
GLPK	sí	sí	sí	sí	sí	sí	sí	no	sí	sí
LINGO®	sí	sí	sí	sí	no	no	no	sí	sí	sí
LPSOLVE	sí	sí	sí	sí	sí	sí	no	no	no	sí
XPRESS®	sí	sí	sí	sí	sí	sí	no	sí	sí	sí
GAMS®	sí	sí	sí	sí	sí	sí	sí	sí	sí	sí
Características Específicas del Software										
Producto	Programación lineal						Utilidad			
	Simplex Primal		Simplex Dual		Punto Interior		Presolución		Diagnóstico de infactibilidad	
GLPK	sí		sí		sí		sí		no	
LINGO®	sí		sí		sí		sí		sí	
LPSOLVE	no		No		no		sí		no	
XPRESS®	sí		Sí		sí		sí		sí	
GAMS®	sí		Sí		sí		sí		sí	

### 3. Desempeño comparativo LINGO®, GLPK, LP\_SOLVE, GAMS® y XPRESS®

La comparación del desempeño de los 5 software estudiados se realizó mediante la solución de las instancias de la librería *NETLIB Linear Programming Test Set*, que agrupa una colección de problemas (de la vida real y fabricados en laboratorio) recopilados a través de diferentes fuentes, y que han sido ampliamente utilizados con el fin de establecer puntos de comparación en el desempeño de los diferentes métodos de solución. Todas las soluciones de los problemas de la librería se obtuvieron de un computador Pentium IV de 1.8 MHz, con 1Gb de memoria RAM y sistema operativo Windows XP. Un requerimiento que se consideró indispensable en el desarrollo del experimento consistió en no manipular los parámetros por defecto de los *solvers* de manera que sea más apropiada la comparación para el usuario final promedio -estudiante-.

El esquema propuesto por (Hock et al., 1981) utilizado en NETLIB permite clasificar los problemas de la librería. En particular las instancias usadas en este estudio comparativo comparten las siguientes características: 1. función objetivo lineal, las restricciones son lineales; 2. existe la primera y segunda derivada y son continuas en cualquier valor; 3. el grado de la derivada más alta que es proporcionada analíticamente en la descripción del problema es 2; 4. el origen o interés del problema puede ser académico (ha sido construido por investigadores con el fin de medir el desempeño de uno o más algoritmos), es parte de un ejercicio

de modelado donde el valor real de la solución no es usada en una aplicación práctica real o la solución del problema es o ha sido usada en aplicaciones reales para propósitos diferentes a la medición del desempeño de algoritmos; y 5. no hay variables internas explícitas (combinaciones lineales de variables elementales). NETLIB comprende una amplia y variada gama de instancias cuya configuración va desde la más pequeña, de 32 variables y 27 restricciones hasta la más grande que tiene 22.275 variables y 16.675 restricciones.

Debido a las restricciones de capacidad de las versiones establecidas por las casas de software comercial, no fue posible solucionar todos los problemas de la librería NETLIB mediante LINGO® SUPER® V6.0, así como con GAMS® V23.5. Con el propósito de registrar este hecho se presentan las figuras 1 y 2 que muestran de manera separada, los conjuntos de instancias que pudieron (y no) ser solucionadas por cada uno de los cinco *solvers*. Adicionalmente, la figura 1 muestra el porcentaje de problemas que pudieron o no ser solucionados por cada uno de los *solvers*. *Lp\_Solve*, no encontró solución óptima en algunos de los problemas PL de la librería en tiempos razonables, un hecho extraño relacionado con la estabilidad del *solver*, si se tiene en cuenta que es una implementación del algoritmo simplex (incluso este hecho se presentó en instancias de pequeña y mediana escala de optimización) ya que los problemas de ciclado pueden ser corregidos con una etapa adicional de código de programación. Finalmente, la totalidad de las instancias pudieron ser resueltas con las versiones empleadas de GLPK y XPRESS®.

Desempeño en instancias de librería NETLIB

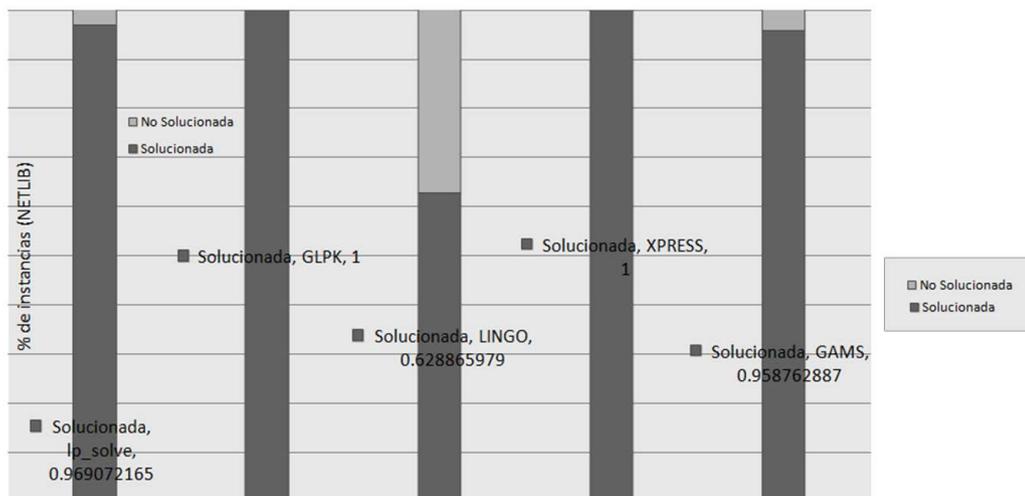


Figura 1 Comparativo de desempeño de los *solvers* usando la librería NETLIB

La tabla 2 muestra las diferencias absolutas en los tiempos de solución de las instancias de los *solvers* con respecto a los tiempos de solución obtenidos por GLPK, debido a que esta opción libre permitió

obtener la solución óptima de todas las instancias de la librería. Las casillas marcadas con n.a. corresponden a las instancias que no fueron solucionadas por el *solver* empleado en la comparación.

Tabla 2. Diferencias de tiempos de solución (segundos) con respecto a GLPK

<i>Instancia</i>	<i>Clasificación</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>	<i>d4</i>	<i>Instancia</i>	<i>Clasificación</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>	<i>d4</i>
25FV47	LLR2-AN-1571-822	0	0	1	-3	PILOT-WE	LLR2-AN-2789-722	0	0	n.a.	-4
80BAU3B	LLR2-AN-9799-2263	2	3	n.a.	-12	PILOT	LLR2-AN-3652-1441	8	11	n.a.	-5
ADLITTLE	LLR2-AN-97-56	0	0	0	0	PILOT4	LLR2-AN-1000-410	0	0	0	0
AFIRO	LLR2-AN-32-27	0	0	0	0	PILOT87	LLR2-AN-4883-2030	19	25	n.a.	-1028
AGG	LLR2-AN-163-488	0	0	0	0	PILOTNOV	LLR2-AN-2172-975	1	1	n.a.	-1
AGG2	LLR2-AN-302-516	0	0	0	0	QAP12	LLR2-AN-8856-3192	133	198	n.a.	-81
AGG3	LLR2-AN-302-516	0	0	0	0	QAP15	LLR2-AN-22275-6330	2465	3131	n.a.	n.a.
BANDM	LLR2-AN-472-305	0	0	0	0	QAP8	LLR2-AN-1632-912	0	1	2	-54
BEACONFD	LLR2-AN-262-173	0	0	0	0	RECIPELP	LLR2-AN-180-91	0	0	0	0
BLEND	LLR2-MN-83-74	0	0	0	0	SC105	LLR2-AN-103-105	0	0	0	0
BNL1	LLR2-AN-1175-643	0	0	0	0	SC205	LLR2-AN-203-205	0	0	0	0
BNL2	LLR2-AN-3489-2324	1	1	n.a.	-2	SC50A	LLR2-AN-48-50	0	0	0	0
BOEING1	LLR2-MN-384-351	0	0	0	0	SC50B	LLR2-AN-48-50	0	0	0	0
BOEING2	LLR2-MN-143-166	0	0	0	0	SCAGR25	LLR2-AN-500-471	0	0	0	0
BORE3D	LLR2-AN-315-233	0	0	0	n.a.	SCAGR7	LLR2-AN-140-129	0	0	0	0
BRANDY	LLR2-AN-249-220	0	0	0	0	SCFXM1	LLR2-AN-457-330	0	0	0	0
CAPRI	LLR2-AN-353-271	0	0	0	0	SCFXM2	LLR2-AN-914-660	0	0	0	0
CYCLE	LLR2-AN-2857-1903	0	0	n.a.	-10	SCFXM3	LLR2-AN-1371-990	0	0	0	-1
CZPROB	LLR2-AN-3523-929	0	0	n.a.	-2	SCORPION	LLR2-AN-358-388	0	0	0	0
D2Q06C	LLR2-AN-5167-2171	2	5	n.a.	-41	SCRS8	LLR2-AN-1169-490	0	0	0	0
D6CUBE	LLR2-MN-6184-415	3	8	n.a.	4	SCSD1	LLR2-AN-760-77	0	0	0	0
DEGEN2	LLR2-AN-534-444	0	0	0	-2	SCSD6	LLR2-AN-1350-147	0	0	0	0
DEGEN3	LLR2-AN-1818-1503	0	1	n.a.	n.a.	SCSD8	LLR2-AN-2750-397	0	0	n.a.	-1
DFL001	LLR2-MN-12230-6071	114	n.a.	n.a.	n.a.	SCTAP1	LLR2-AN-480-300	0	0	0	0
E226	LLR2-AN-282-223	0	0	0	0	SCTAP2	LLR2-AN-1880-1090	0	0	n.a.	0

<i>Instancia</i>	<i>Clasificación</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>	<i>d4</i>	<i>Instancia</i>	<i>Clasificación</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>	<i>d4</i>
ETAMACRO	LLR2-AN-688-400	0	0	0	0	SCTAP3	LLR2-AN-2480-1480	0	0	n.a.	-1
FFFFF800	LLR2-AN-854-524	0	0	0	0	SEBA	LLR2-AN-1028-515	0	0	0	0
FINNIS	LLR2-MN-614-497	0	0	0	0	SHARE1B	LLR2-AN-225-117	0	0	0	0
FIT1D	LLR2-AN-1026-24	0	0	0	0	SHARE2B	LLR2-AN-79-96	0	0	0	0
FIT1P	LLR2-AN-1677-627	0	0	0	-1	SHELL	LLR2-AN-1775-536	0	0	0	0
FIT2D	LLR2-AN-10500-25	19	20	n.a.	3	SHIP04L	LLR2-AN-2118-402	0	0	n.a.	0
FIT2P	LLR2-AN-13525-3000	6	9	n.a.	-39	SHIP04S	LLR2-AN-1458-402	0	0	0	0
FORPLAN	LLR2-AN-421-161	0	n.a.	n.a.	-1	SHIP08L	LLR2-AN-4283-778	0	0	n.a.	-1
GANGES	LLR2-AN-1681-1309	0	0	n.a.	-1	SHIP08S	LLR2-AN-2387-778	0	0	n.a.	-1
GFRD-PNC	LLR2-AN-1092-616	0	n.a.	0	0	SHIP12L	LLR2-AN-5427-1151	0	0	n.a.	-2
GREENBEA	LLR2-AN-5405-2392	2	3	n.a.	-39	SHIP12S	LLR2-AN-2763-1151	0	0	n.a.	-1
GREENBEB	LLR2-AN-5405-2392	1	1	n.a.	-34	SIERRA	LLR2-AN-2036-1227	0	n.a.	n.a.	0
GROW15	LLR2-AN-645-300	0	0	0	0	STAIR	LLR2-AN-467-356	0	0	0	0
GROW22	LLR2-AN-946-440	0	0	0	-2	STANDATA	LLR2-AN-1075-359	0	0	0	0
GROW7	LLR2-AN-301-140	0	0	0	0	STANDGUB	LLR2-AN-1184-361	0	0	0	0
ISRAEL	LLR2-AN-142-174	0	0	0	0	STANDMPS	LLR2-AN-1075-467	0	0	0	0
KB2	LLR2-AN-41-43	0	0	0	0	STOCFOR1	LLR2-MN-111-117	0	0	0	0
LOTFI	LLR2-MN-308-153	0	0	0	0	STOCFOR2	LLR2-MN-2031-2157	0	0	n.a.	-2
MAROS-R7	LLR2-RN-9408-3136	9	11	n.a.	-110	STOCFOR3	LLR2-MN-15695-16675	6	6	n.a.	n.a.
MAROS	LLR2-RN-1443-846	0	0	0	-2	TRUSS	LLR2-MN-8806-1000	2	-1	n.a.	-11
MODSZK1	LLR2-MN-1620-688	0	0	0	0	TUFF	LLR2-AN-587-333	0	0	0	0
NESM	LLR2-AN-2923-662	1	1	n.a.	-1	VTP-BASE	LLR2-AN-203-198	0	0	0	0
PEROLD	LLR2-AN-1376-625	1	1	1	-1	WOOD1P	LLR2-AN-2594-244	0	0	n.a.	-1
PILOT-JA	LLR2-AN-1988-940	1	2	3	-1	WOODW	LLR2-AN-8405-1098	0	0	n.a.	-9
<i>d1=GLPK-XPRESS    d2=GLPK-GAMS    d3=GLPK-LINGO    d4=GLPK-LP_SOLVE (diferencias de tiempos expresados en segundos)</i>											

Estos resultados permiten observar que: aunque en todas las instancias el tiempo de solución obtenido por XPRESS® fue inferior o igual al proporcionado por el mejor *solver* libre (GLPK), sólo en el 21% de

los casos fue estrictamente menor y sólo en el 3% de los casos la diferencia en los tiempos de solución proporcionado por la opción libre es superior a 20 segundos; un tiempo poco relevante en términos

académicos y en muchas de las aplicaciones prácticas empresariales.

La figura 2, muestra la proporción de las instancias que fue solucionada en el menor tiempo por cada uno de los *solvers*, sin que exista indicio de la existencia de un único solver dominante para todas las instancias o

de indicios fuertes de la conveniencia de uso de *solver* de origen comercial. Observaciones que se validan mediante las pruebas estadísticas aplicadas. Es notorio el dominio de desempeño de la versión GLPK sobre la versión Lp\_Solve, esto es aún más significativo si se tiene en cuenta que varios problemas no pudieron ser resueltos por Lp\_Solve.

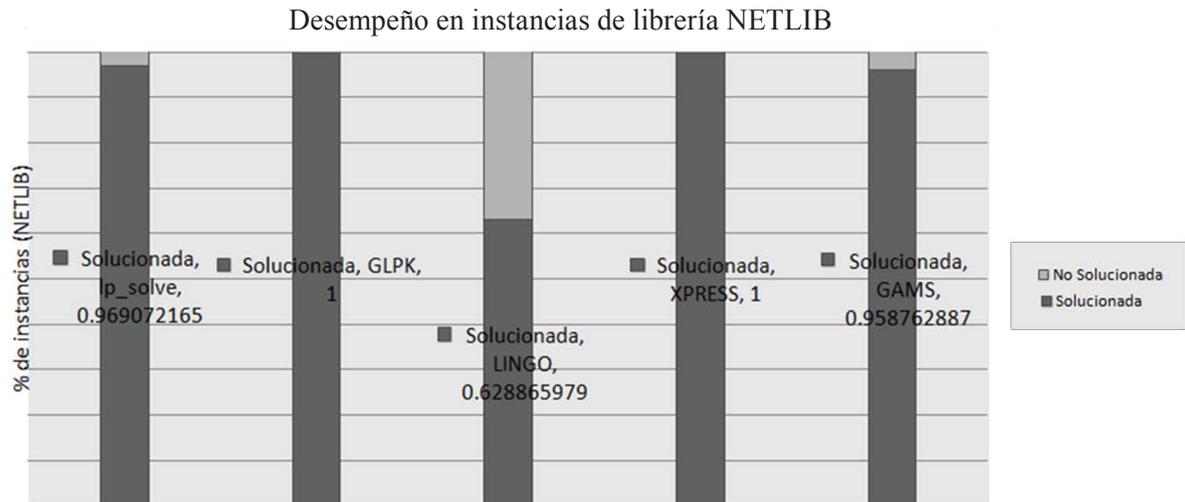


Figura 2. Proporción de instancias solucionadas en tiempo mínimo por cada uno de los *solvers*.

Como análisis final se realizó un ANOVA de un factor para determinar si existen diferencias estadísticas en los tiempos promedio de solución de aquellas instancias que fueron solucionadas de manera óptima por los 5 *solvers*. Si bien el ANOVA se rechazó con un *p-value* de 0.044, al ser comparado con un nivel de confianza fijado en el 95%, indicando que hay diferencias significativas en los tiempos de ejecución entre algunos pares de medias, el supuesto de homogeneidad de varianzas no se cumplió. Por ello se seleccionó como método de comparación la prueba de *Tamhane*, la cual asume varianzas diferentes, con el fin de hacer la comparación de los 10 pares de medias. Los *p-values* obtenidos fueron: GLPK-LINGO® (0.279); GLPK-Lp\_Solve (0.730); GLPK-XPRESS® (1.0); GLPK-GAMS® (0.910); LINGO-Lp\_Solve (0.677); LINGO-XPRESS® (0.008); LINGO-GAMS® (0.000); Lp\_Solve-XPRESS® (0.733); Lp\_Solve-GAMS® (0.760); XPRESS®-GAMS® (1.0). Basados en estos resultados y fijando un nivel de confianza del 95%, se rechazó la hipótesis nula de diferencia de medias en el tiempo de desempeño entre los *solvers* LINGO® y XPRESS®, y LINGO® y GAMS®; en tanto que para las demás diferencias no se encontró diferencias estadística significativa en el tiempo promedio.

En términos generales los tiempos de desempeño promedio de Lp\_Solve y de GLPK frente a los otros tres *solvers* no son significativamente diferentes.

## Conclusiones y recomendaciones

En gran medida la motivación para escribir este artículo yace en nuestro interés por promover al interior de las instituciones académicas, y desde la academia, la discusión respecto a la conveniencia o no del uso de software comercial como herramienta tecnológica dominante para la enseñanza de optimización lineal, y otras áreas del conocimiento relacionadas con sus aplicaciones. Creemos que este es un paso inicial para desencadenar una discusión similar al interior del sector empresarial quien es finalmente el ente donde se concreta el proceso permanente de innovación necesario para alcanzar la competitividad.

Los resultados del estudio de la muestra de *solvers* mostraron que las opciones de *software* libre son alternativas merecedoras de ser tenidas en cuenta en los procesos de adaptación tecnológica, tanto en

sectores académicos como empresariales. Lo anterior no implica, particularmente en el caso empresarial, que no se hagan las evaluaciones técnicas y los estudios de rigor previos, con los cuales se resuelvan interrogantes propios de la decisión de compra de un *solver* de PL, esto debido principalmente a que nuestro estudio no incluyó todos los criterios tenidos en cuenta por usuarios expertos.

Algunas ventajas previstas que se generarían al promover el uso de software de código abierto especializado en los programas formales de educación en instituciones educativas, incluyen:

- Descongestionar las instalaciones de cómputo de las instituciones
- Promocionar el trabajo autónomo de los estudiantes en sus hogares para profundizar sus conocimientos sobre el tema de estudio
- Disminuir los gastos de licenciamiento de software comercial
- Desincentivar entre la comunidad educativa la práctica nociva de compra/intercambio de copias ilegales – no licenciadas – de software comercial
- Incentivar la utilización de procesos óptimos, debido a que se eliminan las barreras de acceso – económicas y legales - a las implementaciones funcionales de los modelos teóricos enseñados.

## Bibliografía

- Ahuja, R.K., Magnanti, T.L., & Orlin, J.B. (1993). *Network Flows*. New Jersey: Prentice Hall.
- Ang, J.S.K., Cao, Ch. & Ye, Heng-Qing. (2007). Model and algorithms for multi-period sea cargo mix problem. *European Journal of Operational Research*, 180(3), 1381-1393.
- Bachouch, R.B., Guinet, A., & Hajri-Gabouj, S. (2012). An integer linear model for hospital bed planning. *International Journal of Production Economics*, 140(2), 833-843.
- Bazaraa, M.S., Jarvis, J.J., & Sherali, H.D. (1977). *Linear Programming and Network Flows*. New York: Jhon Wiley & Sons.
- Bergamaschi, R.A., Piga, L., Rigo, S., Azevedo, R. & Araújo, G. (2012). Data center power and performance optimization through global selection of P-states and utilization rates. *Sustainable Computing: Informatics and Systems*, 2(4), 198-208.
- Bohle, C., Maturana, S. & Vera, J. (2010). A robust optimization approach to wine grape harvesting scheduling. *European Journal of Operation Research*, 200(1), 245-252.
- El-Sayed, H., Ibrahim, W., & Amer, H. (2009). An adaptive heuristic algorithm for VLSI test vectors selection. *European Journal of Operational Research*, 199(3), 630-639.
- Fourer, Robert. (2005). *Software Survey – Linear Programming*. Eighth in a series of LP surveys highlights recent trends in profession's most popular software. ORMS today. Recuperado el 13-02-2013 de <http://www.lionhrtpub.com/orms/orms-6-05/fsurvey.html>
- Free Software Foundation (2009). *The free software definition*. Recuperado el 6-10-2009 de <http://www.fsf.org/licensing/essays/free-sw.html>
- Fu, L., Trudel, M. & Kim, V. (2009). Optimizing winter road maintenance operations under real-time information. *European Journal of Operational Research*, 196(1), 332-341.
- GNU. (2009). Recuperado el 6-10-2009 de <http://www.gnu.org>
- Han, B., Leblet, J. & Simon G. (2010). Hard multidimensional multiple choice knapsack problems, an empirical study. *Computers and Operational Research*, 37(1), 172-181.
- Hillier, F.S., & Lieberman G.J.. *Introducción a la Investigación de Operaciones*, Mc Graw Hill, México, 1997.
- Hock, W & Schittkowski, K. (1980). *Test Examples for Nonlinear Programming Codes*. Journal of Optimization Theory and Applications, 30(1), 127, 1980.
- Lasdon, L. (1970). *Optimization Theory for large Systems*. New York: Macmillan.
- Lesser General Public License. (1999). *GNU LESSER GENERAL PUBLIC LICENSE* recuperado el 6-10-2009 de <http://lpsolve.sourceforge.net/5.5/LGPL.htm>
- LINGO SYSTEMS. (2006). *LINGO: The modeling language and optimizer*. Chicago: LINGO SYSTEMS.
- Lp\_Solve. (2009). Recuperado el 6-10-2009 de [http://groups.yahoo.com/group/lp\\_solve/](http://groups.yahoo.com/group/lp_solve/)
- Mashaei, M. & Lennartson, B. (2013). Energy Reduction in a Pallet-Constrained Flow Shop Through On-Off

- Control of Idle Machines. *Automation Science and Engineering, IEEE Transactions*, 10(1), 45-56.
- Mitsutoshi, K., Takashi, I., Koji, N., Mutsunori, Y., & Hiroshi., N. (2009). Exact algorithms for the two-dimensional strip packing problem with and without rotations. *European Journal of Operational Research*, 198(1), 73-83.
- Montoya-Torres, J.R., Aponte, A., Rosas, P. & Caballero-Villalobos, J.P. (2010). Applying GRASP meta-heuristic to solve the single-item two-echelon uncapacitated facility location problem. *International Journal of Applied Decision Sciences*, 3(4), 297-310.
- NETLIB (2009). *NETLIB Linear Programming test set* [base de datos en línea]. Recuperado el 6-10-2009 de <http://cuter.rl.ac.uk/cuter-www/Problems/netlib.shtml>
- Rossi, A., Singh, A. & Sevaux, M. (2012). An exact approach for maximizing the lifetime of sensor networks with adjustable sensing ranges. *Computers & Operations Research*, 39(12), 3166-3176.
- Salvietti, L. & Smith, N.R. (2008). A profit-maximizing economic lot scheduling problem with price optimization. *European Journal of Operational Research*, 184(3), 900-914.
- Winston W. L. (2004). *Operations Research: Applications and Algorithms*. (4ta. Ed.). Thompson.
- Yi, W. (2013). Optimally Removing Intercore Communication Overhead for Streaming Applications on MPSoCs. *IEEE Transactions on Computers*, 62(2), 336-350.

## Sobre los autores

---

### Juan Pablo Caballero Villalobos

Pontificia Universidad Javeriana, Bogotá (Colombia)  
 Magíster en Ingeniería Industrial de la Universidad de los Andes (Bogotá, Colombia, 2005). Ingeniero Industrial de la Pontificia Universidad Javeriana (Bogotá, Colombia, 1999). Profesor Asistente y CoDirector del Centro de Investigaciones en Optimización y Logística – CIOL -. Pontificia Universidad Javeriana Departamento de Ingeniería Industrial. Calle 40 5-50. Bogotá (Colombia).  
[juan.caballero@javeriana.edu.co](mailto:juan.caballero@javeriana.edu.co)

### Eliana María González Neira

Pontificia Universidad Javeriana, Bogotá (Colombia)  
 Magíster en Ingeniería Industrial (Bogotá, Colombia, 2011) e Ingeniera Industrial de la Pontificia Universidad Javeriana (Bogotá, Colombia, 2004). Profesor Instructor Pontificia Universidad Javeriana Departamento de Ingeniería Industrial. Calle 40 5-50. Bogotá (Colombia).  
[eliana.gonzalez@javeriana.edu.co](mailto:eliana.gonzalez@javeriana.edu.co)

### Olga Lucía Araoz Cajiao

Pontificia Universidad Javeriana, Bogotá (Colombia)  
 Magíster en Sistemas de Calidad y Productividad del Tecnológico de Monterrey (Guadalajara, México, 2013) e Ingeniera Industrial de la Pontificia Universidad Javeriana (Bogotá, Colombia, 2003). Profesor Instructor Pontificia Universidad Javeriana Departamento de Ingeniería Industrial. Calle 40 5-50. Bogotá (Colombia).  
[olga.araoz@javeriana.edu.co](mailto:olga.araoz@javeriana.edu.co)

### Rafael Guillermo García Cáceres

Escuela Colombiana de Ingeniería Julio Garavito, Bogotá (Colombia). Doctor en Ingeniería y Magíster en Ingeniería Industrial de la Universidad de los Andes (Bogotá, Colombia, 2008; 2000). Ingeniero Industrial de la Universidad Pedagógica y Tecnológica de Colombia (Sogamoso, Colombia, 1998). Profesor Asociado y CoDirector del Centro de Investigaciones en Optimización y Logística –CIOL-. Escuela Colombiana de Ingeniería Julio Garavito, Departamento de Ingeniería Industrial. Carrera 45 205-59, Bogotá (Colombia).  
[rafael.garcia@escuelaing.edu.co](mailto:rafael.garcia@escuelaing.edu.co)

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.