

Laboratorio 4: librería "lpSolveAPI"

MARSL

April 1, 2020

En este laboratorio emplearemos la librería("package" de R) **lpSolveAPI**. La podemos instalar desde las pestañas de R-studio o desde la ventana de comandos:

```
>install.packages("lpSolveAPI")
```

Desde el programa principal, leeremos los datos de un archivo de datos, con las funciones y formato definidos en el laboratorio 2:

```
lista <- leer_datos (archivo)
```

En este objeto, que es tipo lista, tendremos distintos campos, que corresponden a las matrices y vectores necesarios para resolver el problema lineal:

`lista$n` : número de variables de decisión

`lista$m` : número de restricciones

`lista$opt` : cadena de caracteres, que puede ser "min" o "max"

`lista$matrizA` : matriz $m \times n$ de coeficientes tecnológicos

`lista$costes` : vector $1 \times n$ de costes

`lista$recursos` : vector $1 \times m$ de recursos

`lista$direccion` : vector $1 \times m$ de caracteres, que pueden ser ">=", "=", o "<="

Se le pueden añadir campos, desde el principio, en la función "genera.lista" para almacenar los resultados.

Una vez leídos los datos tenemos que crear una estructura adecuada para usar las funciones de *lpSolveAPI*, mediante la función *make.lp*, con m filas y n columnas.

```
make.lp(nrow=m, ncol=n) <- objeto
```

El resultado lo guardaremos en un objeto (nombre "objeto", en el ejemplo), que estará vacío. Para ver su estructura, simplemente *print(objeto)*

Lo primero hay que añadir, por columnas, al matriz de coeficientes tecnológicos $\forall j = 1 \dots n$:

```
set.column (objeto, j, lista$matrizA[,j])
```

A continuación los vectores de costes, de recursos y las desigualdades de las restricciones.:

```
set.objfn (objeto, lista$costes)
```

```
set.constr.value (objeto, rhs=problema$recursos, constraints=1:problema$m)
```

```
set.constr.type (objeto, lista$direccion)
```

Finalmente se resuelve el problema con la función *solve*

CUIDADO !! Esta función está implementada para mínimos.

```
solve(objeto )
```

Esta función nos devuelve un *cero* si se ha resuelto el problema con éxito.
Obtenemos los resultados con las funciones *get.variables* y *get.objective*

```
print(get.variables( objeto ))
```

```
print(get.objective(objeto ))
```

Finalmente, si queremos restringir alguna variable a entero o binario:

```
set.type(opjeto , 2, "integer")  
set.type(objeto , 3, "binary")
```

Más funciones de *lpSolveAPI* en

<https://cran.r-project.org/web/packages/lpSolveAPI/lpSolveAPI.pdf>