
In this session, your task is to get one of software/libraries installed properly on your personal computer (or on your home directory of the CSLinux server).

1. Linear Programming

The software to be set up in this lab is a mathematics library to solve linear programming problems that we shall study in **Lec02** next week. You may preview the slides to have some initial understanding of the problem and the algorithms. For students who prefer using Python, you may skip the rest of sections and install another commercial math programming solver, Gurobi, on your computer. Gurobi's package `gurobipy` can be found from <https://support.gurobi.com/hc/en-us/articles/360044290292-How-do-I-installGurobi-for-Python#section:Python>.

2. LP_Solve Software

LPSolve is a **free** (see [LGPL](#) for the GNU lesser general public license) linear (integer) programming solver based on the revised **simplex method** and the **Branch-and-bound method** for the integers. The solver can be called through most major programming languages, including C/C++, Java, .Net, Delphi, etc. It can also be called from scripting languages like R, Python, MATLAB, and others. The software is freely downloadable from <https://sourceforge.net/projects/lpsolve/>. For Windows users, the solver can also be called through LPSolve IDE (<http://lpsolve.sourceforge.net/5.5/>). A complete reference document is available from <http://lpsolve.sourceforge.net/5.5/>.

3. Installation

In this lab, you need to get the library installed properly on your home directory of the CSLinux server. In this module we shall use C to interact with LPSolve. To do this, you need to download the library pack (`lpsolve55.tar`) from module Moodle page. If you want to install it on a different system or using a different programming language, please refer to the software online documentation.

Once downloaded, please copy the file a directory where you want to work on the exercises in Lab1 and Lab2. Extract the files using the following command on a Linux terminal:

```
tar -xf lpsolve55.tar
```

Now you should see a number of files extracted to a newly created directory `lpsolve55`.

Among the extracted files, I included a `demo.c` file that provides an example on how to call the solver from C to solve the following simple linear programming problem:

max: $143x + 60y$;

subject to:

$120x + 210y \leq 15000$

$110x + 30y \leq 4000$

$x + y \leq 75$

The program can be compiled using the following:

```
gcc demo.c -lm -ldl liblpsolve55.a
```

or

```
g++ demo.c -lm -ldl -no-pie liblpsolve55.a
```

Both `-lm` and `-ldl` are `gcc/g++` options to load necessary library to support `LPSolve` call. `liblpsolve55.a` is the pre-compiled `LPSolve` library. After compilation, you should be able to run the program. It would generate the following output:

```
/* Objective function */
max: +143 x +60 y;

/* Constraints */
+120 x +210 y <= 15000;
+110 x +30 y <= 4000;
+x +y <= 75;
Objective value: 6315.625000
x: 21.875000 y: 53.125000
```

Problem:

A company produces three types of products: A, B, and C. The company has limited resources, and the goal is to maximize its profit while adhering to the following constraints:

Inputs:

1. Resource availability:
 - The company has 100 units of Material 1, 150 units of Material 2, and 120 units of Material 3.
2. Resource consumption for each product (per unit produced):
 - Product A requires 2 units of Material 1, 3 units of Material 2, and 1 unit of Material 3.
 - Product B requires 3 units of Material 1, 2 units of Material 2, and 2 units of Material 3.
 - Product C requires 1 unit of Material 1, 2 units of Material 2, and 3 units of Material 3.
3. Profit per unit of each product:
 - Product A: \$10 per unit
 - Product B: \$12 per unit
 - Product C: \$8 per unit

Hints:

Add below lines after 'set_maxim(lp);' to limit A, B, C can only be integers.

```
/* Set the variables as integers */  
set_int(lp, 1, TRUE); /* Set Product A (x) as an integer */  
set_int(lp, 2, TRUE); /* Set Product B (y) as an integer */  
set_int(lp, 3, TRUE); /* Set Product C (z) as an integer */
```