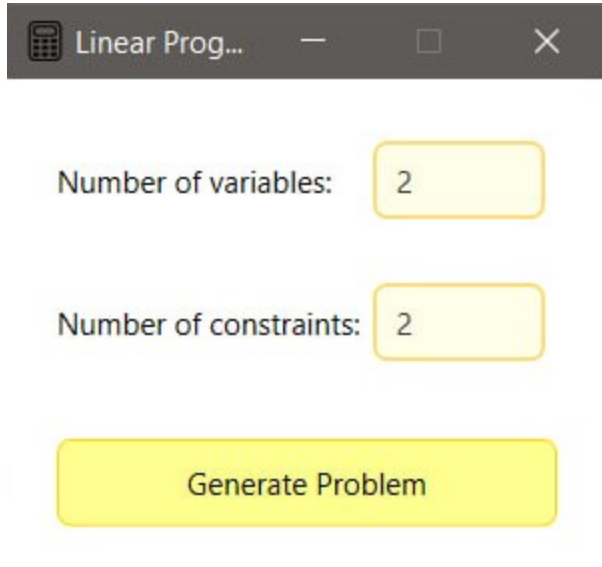


Linear Programming Calculator

Linear Programming Calculator is a product that helps you solve LPPs using the simplex method for constraints that are greater than or equal to, the Big M method for constraints that are less than or equal to or equal to, and finding integer solutions using Gomory cuts.

Quick Start 🏃:

When you start the project, it displays a **Startup window** where you can set the number of variables and constraints.

A screenshot of a software window titled "Linear Prog..." with standard window controls (minimize, maximize, close). The window contains two input fields: "Number of variables:" with a value of "2" and "Number of constraints:" with a value of "2". Below these fields is a large yellow button labeled "Generate Problem".

Linear Prog... — □ ×

Number of variables: 2

Number of constraints: 2

Generate Problem

Then, when you press *Generate Problem*, another window will appear where you can set up all the necessary parameters. You can also **return to the startup window**, press *New Problem*, to edit the number of variables or constraints.

Linear Programming Solver

Objective function:

2

x1

+

-3

x2

→

max

New problem

Solve

Constraints

5

x1

+

2

x2

≥

10

1

x1

+

3

x2

≤

12

☐ x1, x2 are integers

Finally, when you press *Solve* button, the application navigates to the **Result window**, where you can view all steps of the problem's solving process. Additionally, you can edit the current problem by pressing *Edit problem*, or start a new one by pressing *New problem*.



New problem

Edit problem

Mathematical model

$$F(x_1, x_2) = 2x_1 + (-3)x_2 \rightarrow \max$$

$$5x_1 + 2x_2 \geq 10$$

$$1x_1 + 3x_2 \leq 12$$

$$x_1 \geq 0, x_2 \geq 0$$

Introduce slack variables

$$F(x_1, x_2, x_3, x_4) = 2x_1 + (-3)x_2 + 0x_3 + 0x_4 \rightarrow \max$$

$$5x_1 + 2x_2 + (-1)x_3 + 0x_4 = 10$$

$$1x_1 + 3x_2 + 0x_3 + 1x_4 = 12$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

Introduce artificial variables

$$F(x_1, x_2, x_3, x_4, x_5) = 2x_1 + (-3)x_2 + 0x_3 + 0x_4 + (-M)x_5 \rightarrow \max$$

$$5x_1 + 2x_2 + (-1)x_3 + 0x_4 + 1x_5 = 10$$

$$1x_1 + 3x_2 + 0x_3 + 1x_4 + 0x_5 = 12$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0$$

Initial feasible basic solution of the problem

$$x_4 = 12, x_5 = 10$$

Constructing the simplex table

Pivot column: A1

Pivot row: x5

Pivot element: x51

	C	-	2	-3	0	0	-M
	B	A0	A1	A2	A3	A4	A5
0	x4	12	1	3	0	1	0
-M	x5	10	5	2	-1	0	1
Δ		-10M	-5M - 2	-2M + 3	1M	0	0

Linear Programming Solvers.

This class library contains three LPP solvers:

- **Primary Simplex Method;**
- **The Big M method;**
- **Gomory Cutting-Plane method.**

It also contains essential objects for defining a problem:

- **Linear Programming Problem;**
- **Constraint.**

Usage example :

```
var problem = new LinearProgrammingProblem
{
    IsMaximization = true,
    ObjectiveFunctionCoefficients = new List<string> { "1", "1" },
    Constraints = new List<Constraint>
    {
        new Constraint
        {
            Coefficients = new List<string> { "6", "5" },
            RightHandSide = "20",
            Type = ConstraintType.LessThanOrEqual,
        },
        new Constraint
        {
            Coefficients = new List<string> { "2", "3" },
            RightHandSide = "10",
            Type = ConstraintType.LessThanOrEqual,
        },
    }
};

var solver = new SimplexSolver(problem);
solver.Solve();
var gomory = new GomorySolver(_solver.Table, problem);
gomory.Solve();
```