Operations Research - Assignment 1 Report

Alexandria University

Faculty of Engineering
Computer and Systems Department

Team members

- Name: عبدالرحمن سعيد رمضان جمعه
- ID: 22010879

- Name: ايمن ابراهيم محمد قطب
- ID: 22010656

- Name: فاروق اشرف فاروق
- ID: 22011012

1. Introduction

This report presents the implementation of a Linear Programming (LP) solver using the Simplex Method and its variations. The solver is designed to handle different types of LP problems efficiently and provides a stepby-step solution process.

---

2. Code Flow and Architecture Report

2.1. Main Application Flow (main.py)

The GUI application (`main.py`) serves as the entry point, built using PyQt5. It provides a user interface for defining LP problems and selecting solution methods.

Key components:

- **Tabs**: Problem Definition, Solution, Iteration Steps.
- **Input Handling**: Collects variables, constraints, objective type (max/min), and method selection.
- **Solver Dispatch**: Calls appropriate solver based on user selection (Simplex, Big-M, Two-Phase, Goal Programming).

Data Structures:

2.2 Solver Methods

The solver was implemented using **Python**.

2.2.1 Standard Simplex Method (`Simplex.py`)

Purpose: Solve LP problems with `<=` constraints and non-negative variables.
Key Functions:

- `simplex_method(c, A, b, isMax)`:
  - Initializes a tableau with slack variables.
  - Iteratively selects entering/leaving variables using pivot operations.
  - Returns the optimal solution and iteration steps.

Steps:

1. Convert inequalities to equations using slack variables.
2. Iterate via pivoting to maximize/minimize the objective.

2.2.2 Big-M Method (`Big_M.py`)

Purpose: Handle problems with `>=` or `=` constraints using artificial variables and a penalty term (M).
Key Functions:

- `big_m_method(c, A, b, constraint_types, isMax, variable_types)`:
  - Splits unrestricted variables into `x+` and `x-`.
  - Adds artificial variables with large penalty coefficients (M).
  - Uses simplex iterations to drive artificial variables out of the basis.

Steps:

1. Add artificial variables with penalty coefficient M.
2. Solve using simplex, penalizing solutions where artificial variables are non-zero.

2.2.3 Two-Phase Method (`Two_phase.py`)

Purpose: Another way to Handle problems with `>=` or `=` constraints using two phases.
phase 1 : To minimize the effect of artificial variables and find a feasible solution .
phase 2 : Use the feasible basis to optimize the original objective. **Key Functions**:

- **Phase 1**: `__execute_phase1()` minimizes the sum of artificial variables.
- **Phase 2**: `__execute_phase2()` optimizes the original objective after removing artificial variables.
- `make_vars_zeros_Linearly()`: Adjusts the objective row to eliminate basic variables coefficients from it.

2.2.4 Goal Programming (`Goal_Programming.py`)

Purpose: Achieve multiple prioritized goals by minimizing deviations.
Key Functions:

- `goal_method()`:
  - Adds deviation variables (`d+`, `d-`) for each goal.
  - Modifies the tableau to prioritize goals.
  -

Checks goal satisfaction via `isDone` list.

Steps:

1. Convert goals into constraints with deviation variables.
2. construct the objective function by summation of multiplyed priorities by deviation variables responsible for its Goal
3. Optimize goals sequentially based on priority using lexicographic simplex.

---

3. Key Data Structures

- **Numpy Arrays**: For constraint matrices (`A`), objective coefficients (`c`), RHS values (`b`) and 2D numpy array representing coefficients, slack/artificial variables, and RHS (`tableau`).
- **Lists**: Track variable types (unrestricted/non-negative), constraint types (`<=`, `=`, `>=`), `main_row`: Column headers (e.g., `x1`, `s1`, `a1`) , `basic_var`: Current basic variables in the solution and goal priorities.
- **Dictionaries**: Store solutions and iterations for display.

---

4. Function Interactions

1. **GUI Inputs** → Converted to matrices (`A`, `b`, `c`) and constraint lists.
2. **Solver Selection** → Dispatches to `simplex_method()`, `big_m_method()`, etc.
3. **Tableau Initialization** → Built based on variable types and constraints.
4. **Pivoting** → `__execute_simplex()` performs iterations across all methods.
5. **Solution Extraction** → Post-processing (e.g., merging `x+`/`x-` for unrestricted variables).

---

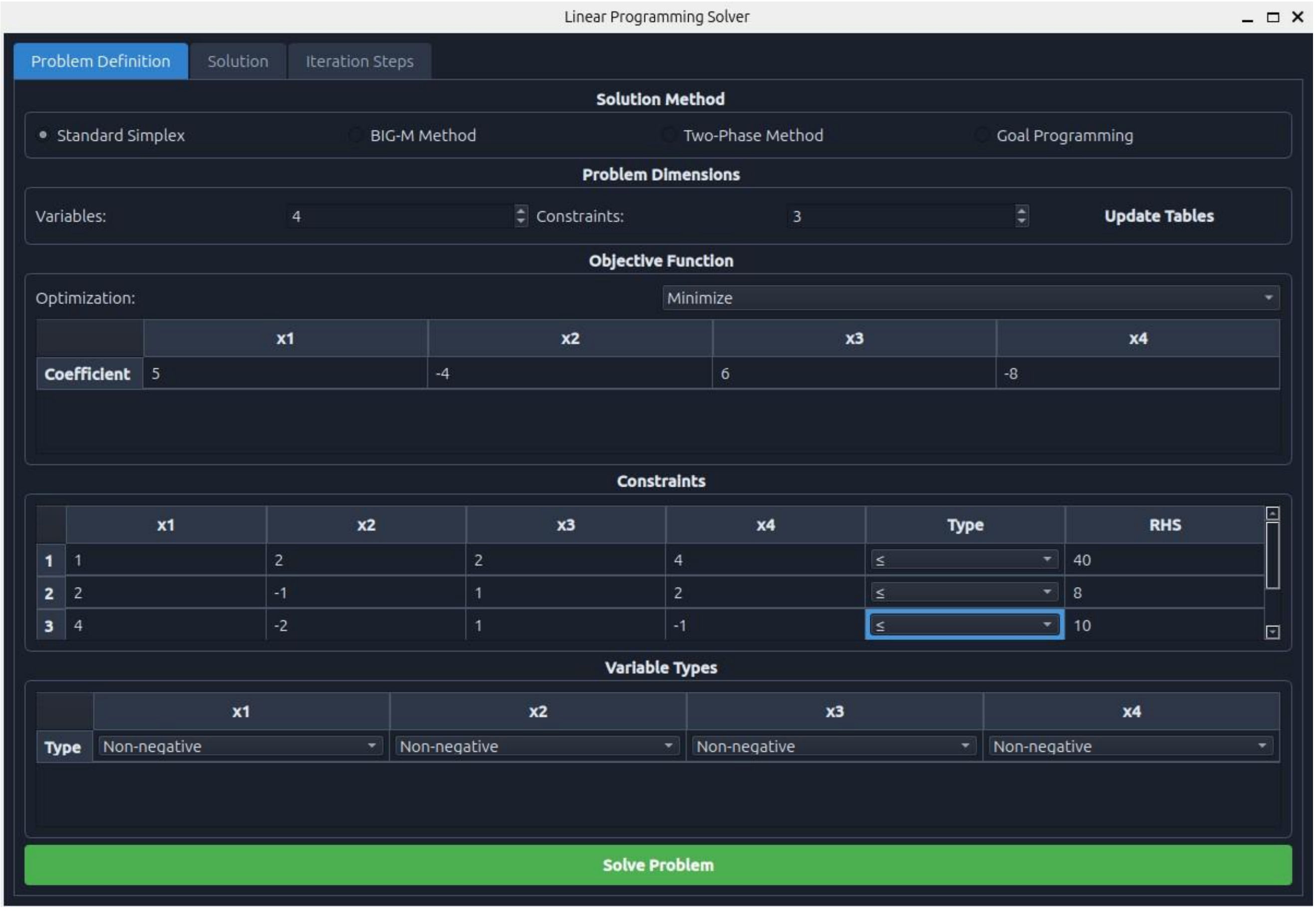
5. Sample Runs

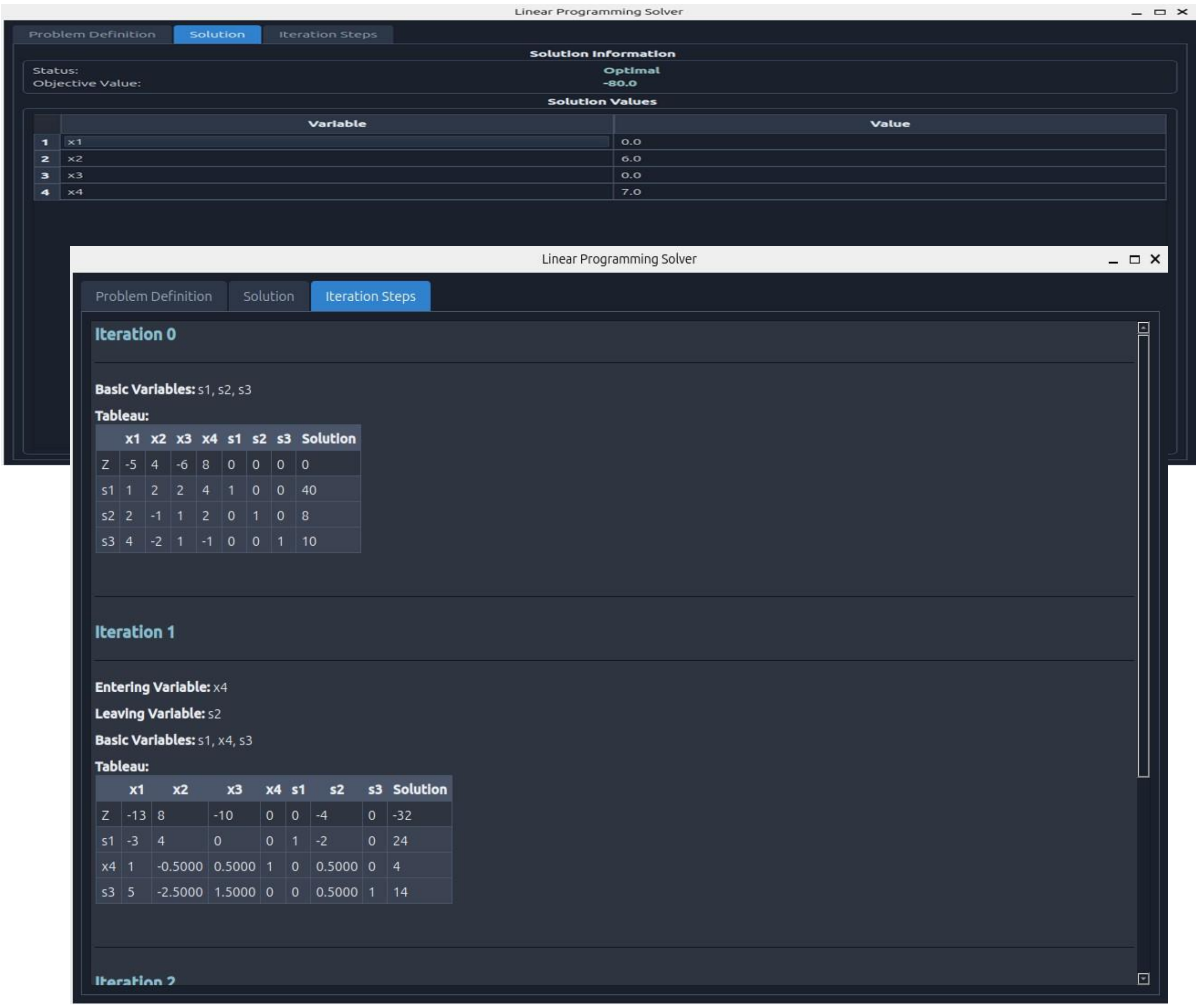
Below are example cases solved using our program:

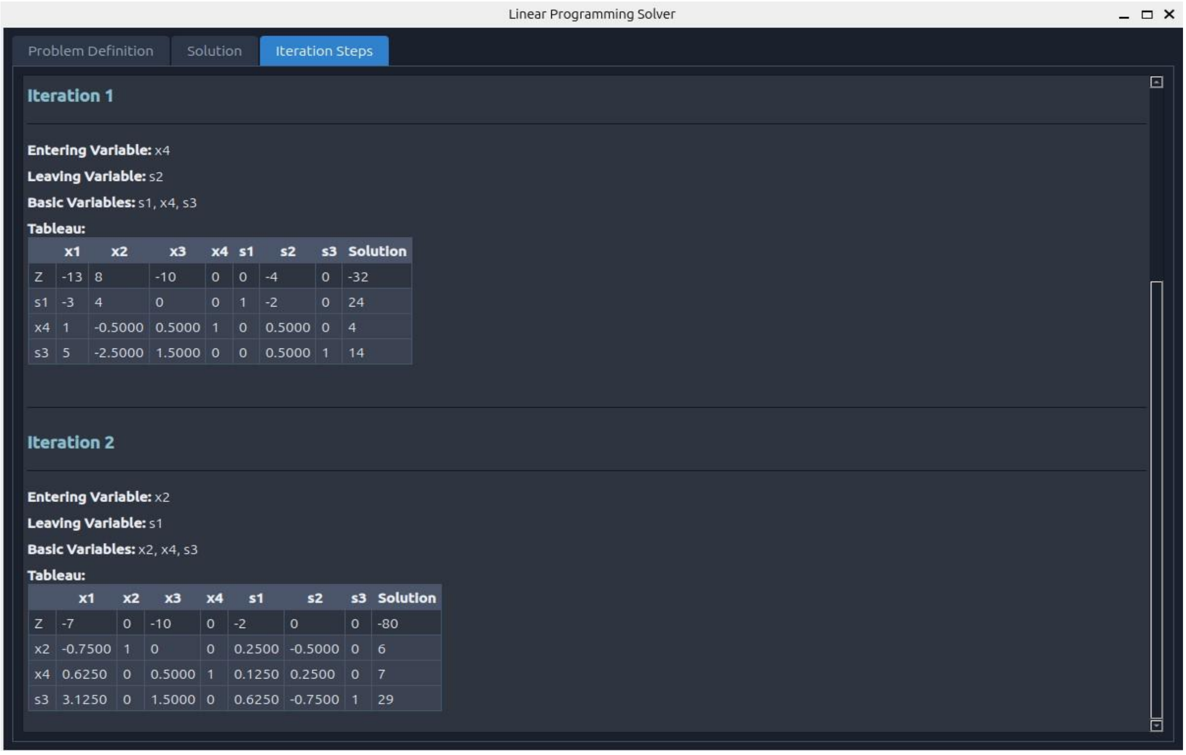
5.1 Example 1 - Simplex Method

```
Maximize Z = 5x1 - 4x2 + 6x3 -8x4

Subject to:

  x1 + 2x2 +2x3 +4x4 ≤ 40

  2x1 – x2 +x3 +2x4 ≤ 8

  4x1 – 2x2 +x3 -x4 ≤ 10

   x1, x2, x3, x4 ≥ 0
```

Output:


Linear Programming Solver
Problem Definition
Solution
Iteration Steps
Solution Method
Standard Simplex
BIG-M Method
Two-Phase Method
Goal Programming
Problem Dimensions
Variables:
4
Constraints:
3
Update Tables
Objective Function
Optimization:
Minimize
x1 x2 x3 x4
Coefficient 5 -4 6 -8
Constraints
x1 x2 x3 x4 Type RHS
1 1 2 2 4 ≤ 40
2 2 -1 1 2 ≤ 8
3 4 -2 1 -1 ≤ 10
Variable Types
x1 x2 x3 x4
Type Non-negative Non-negative Non-negative Non-negative
Solve Problem

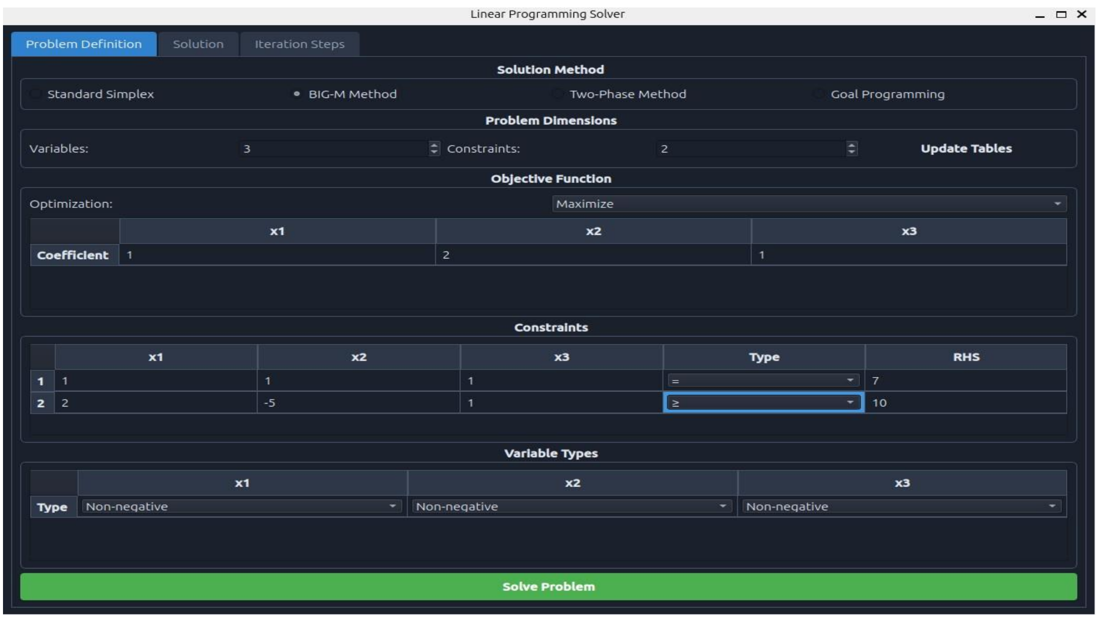

Linear Programming Solver
Problem Definition
Solution
Iteration Steps
Solution Information
Status:
Objective Value:
Optimal
-80.0
Solution Values
Variable Value
1 x1 0.0
2 x2 6.0
3 x3 0.0
4 x4 7.0
Linear Programming Solver
Problem Definition
Solution
Iteration Steps
Iteration 0
Basic Variables: s1, s2, s3
Tableau:
x1 x2 x3 x4 s1 s2 s3 Solution
Z -5 4 -6 8 0 0 0 0
s1 1 2 2 4 1 0 0 40
s2 2 -1 1 2 0 1 0 8
s3 4 -2 1 -1 0 0 1 10
Iteration 1
Entering Variable: x4
Leaving Variable: s2
Basic Variables: s1, x4, s3
Tableau:
x1 x2 x3 x4 s1 s2 s3 Solution
Z -13 8 -10 0 0 -4 0 -32
s1 -3 4 0 0 1 -2 0 24
x4 1 -0.5000 0.5000 1 0 0.5000 0 4
s3 5 -2.5000 1.5000 0 0 0.5000 1 14
Iteration 2

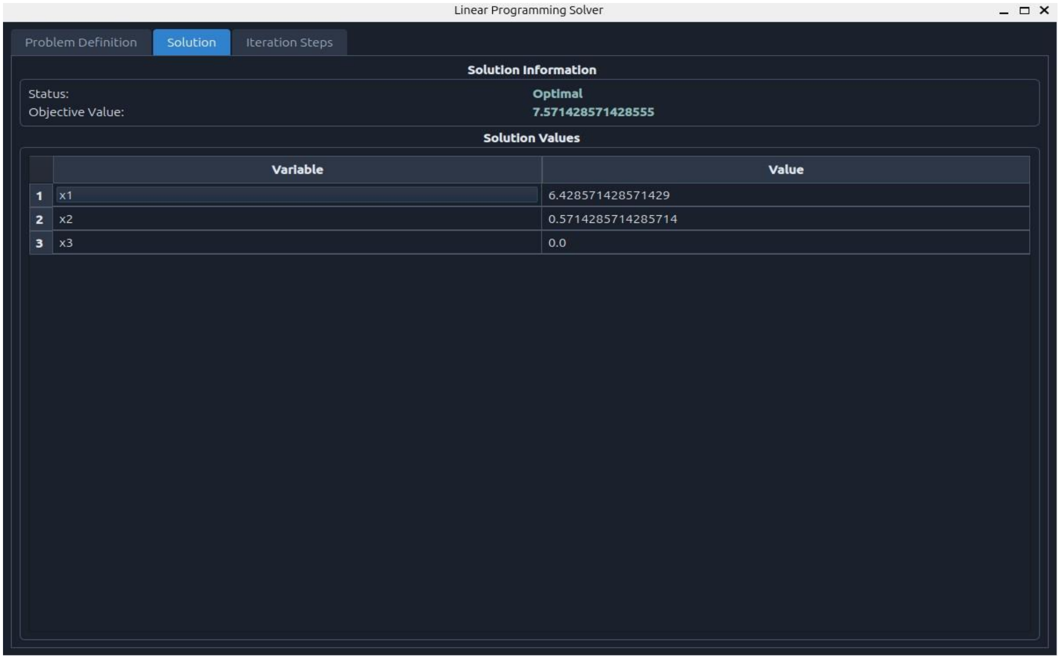


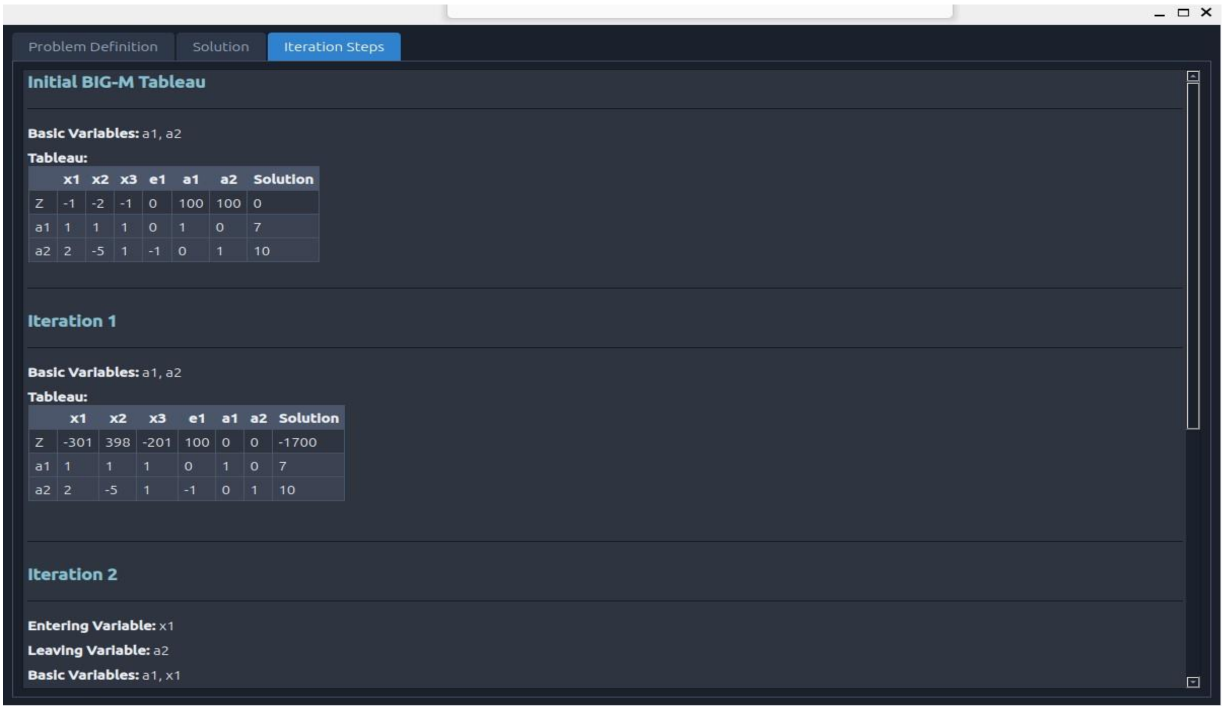
- • Optimal Solution: x1 = 0, x2 = 6, x3 = 0, x4 = 7
- •Optimal Objective Value: Z = -80

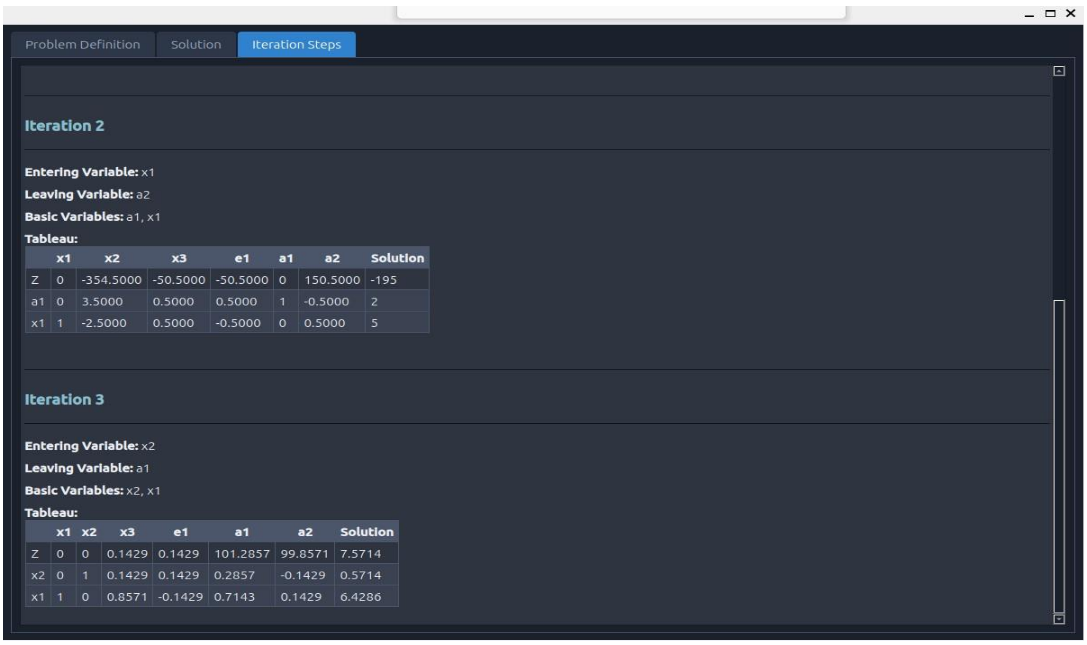
5.2 Example 2 - Big-M Method

```
Maximize Z = x1 + 2x2 +x3
Subject to:
  x1 + x2 +x3 = 7
  2x1 - 5x2 +x3 ≥ 10
  x1, x2 , x3 ≥ 0
```




Linear Programming Solver
Problem Definition
Solution
Iteration Steps
Solution Information
Status: Optimal
Objective Value: 7.571428571428555
Solution Values
Variable | Value
1 x1 6.428571428571429
2 x2 0.5714285714285714
3 x3 0.0


Problem Definition
Solution
Iteration Steps
Initial BIG-M Tableau
Basic Variables: a1, a2
Tableau:
x1 x2 x3 e1 a1 a2 Solution
Z -1 -2 -1 0 100 100 0
a1 1 1 1 0 1 0 7
a2 2 -5 1 -1 0 1 10
Iteration 1
Basic Variables: a1, a2
Tableau:
x1 x2 x3 e1 a1 a2 Solution
Z -301 398 -201 100 0 0 -1700
a1 1 1 1 0 1 0 7
a2 2 -5 1 -1 0 1 10
Iteration 2
Entering Variable: x1
Leaving Variable: a2
Basic Variables: a1, x1

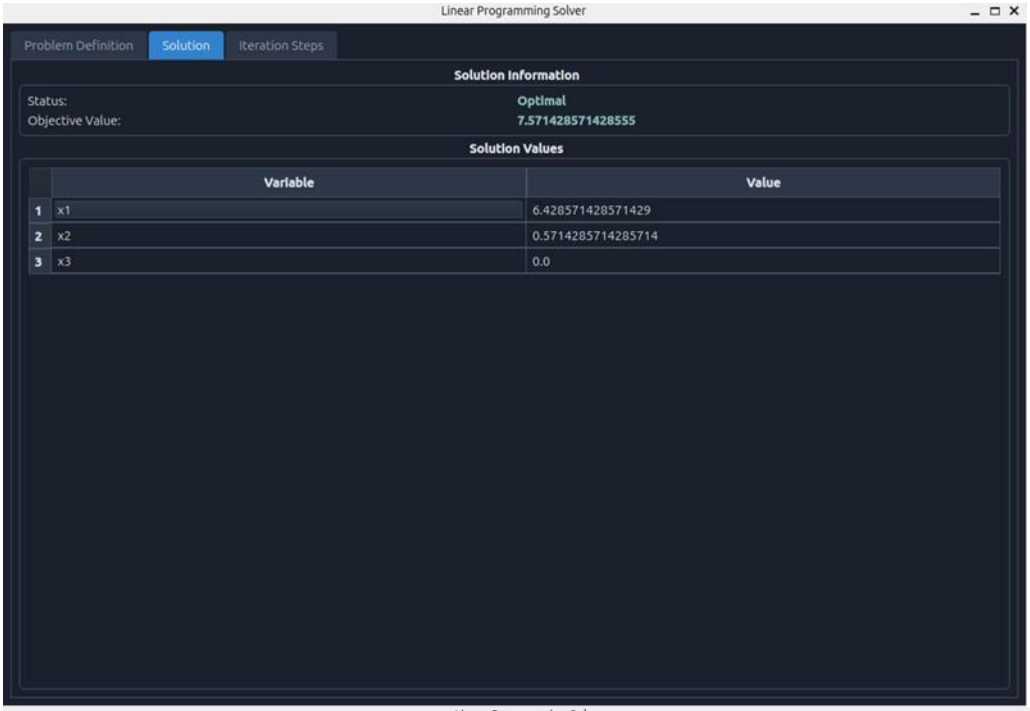

Problem Definition
Solution
Iteration Steps
Iteration 2
Entering Variable: x1
Leaving Variable: a2
Basic Variables: a1, x1
Tableau:
x1 x2 x3 e1 a1 a2 Solution
Z 0 -354.5000 -50.5000 -50.5000 0 150.5000 -195
a1 0 3.5000 0.5000 0.5000 1 -0.5000 2
x1 1 -2.5000 0.5000 -0.5000 0 0.5000 5
Iteration 3
Entering Variable: x2
Leaving Variable: a1
Basic Variables: x2, x1
Tableau:
x1 x2 x3 e1 a1 a2 Solution
Z 0 0 0.1429 0.1429 101.2857 99.8571 7.5714
x2 0 1 0.1429 0.1429 0.2857 -0.1429 0.5714
x1 1 0 0.8571 -0.1429 0.7143 0.1429 6.4286

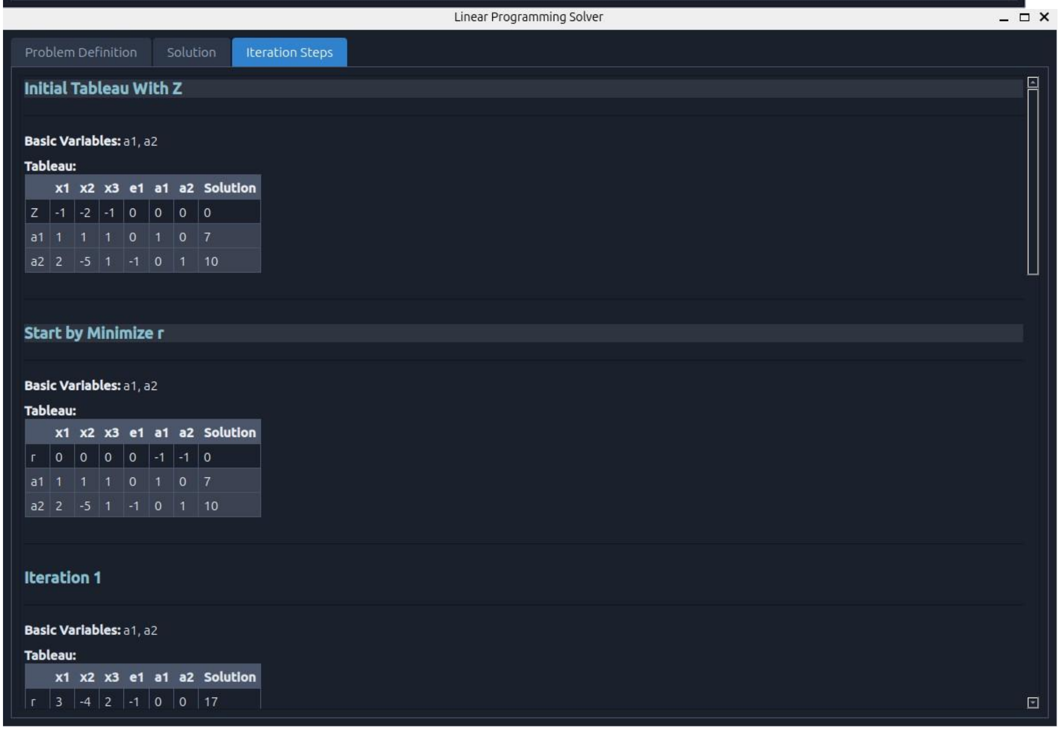
Output:

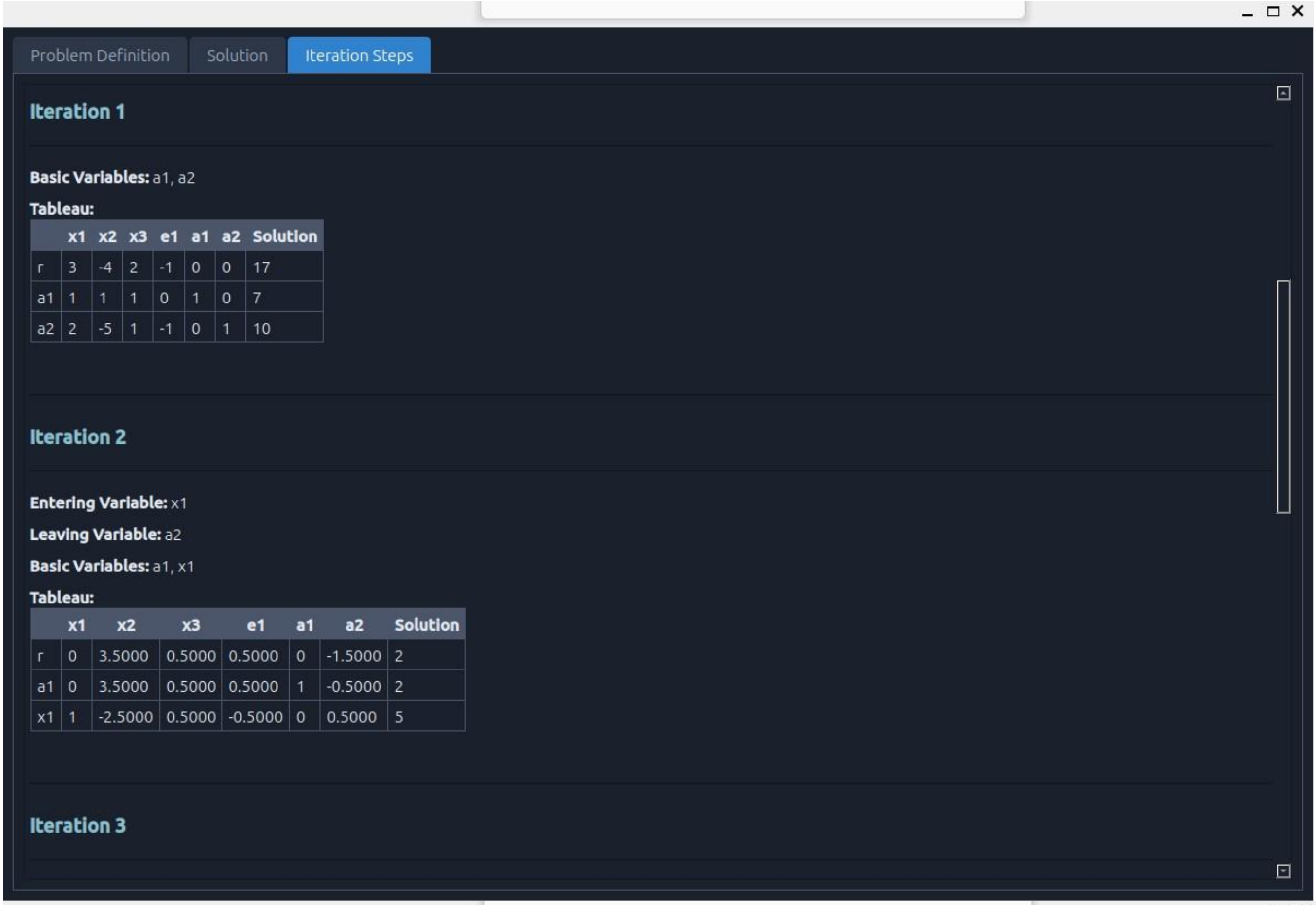
- Optimal Solution: x1 = 45/7, x2 = 4/7
- Optimal Objective Value: Z = 53/7

Example 3 - Two-Phase Method

```
Maximize Z = x1 +2x2+ x3
Subject to:
  x1 + x2 +x3 = 7
  2x1 - 5x2 +x3 ≥ 10
  x1, x2 , x3 ≥ 0
```







Problem Definition | Solution | Iteration Steps

Iteration 1

Basic Variables: a1, a2

Tableau:

	x1	x2	x3	e1	a1	a2	Solution
r	3	-4	2	-1	0	0	17
a1	1	1	1	0	1	0	7
a2	2	-5	1	-1	0	1	10

Iteration 2

Entering Variable: x1

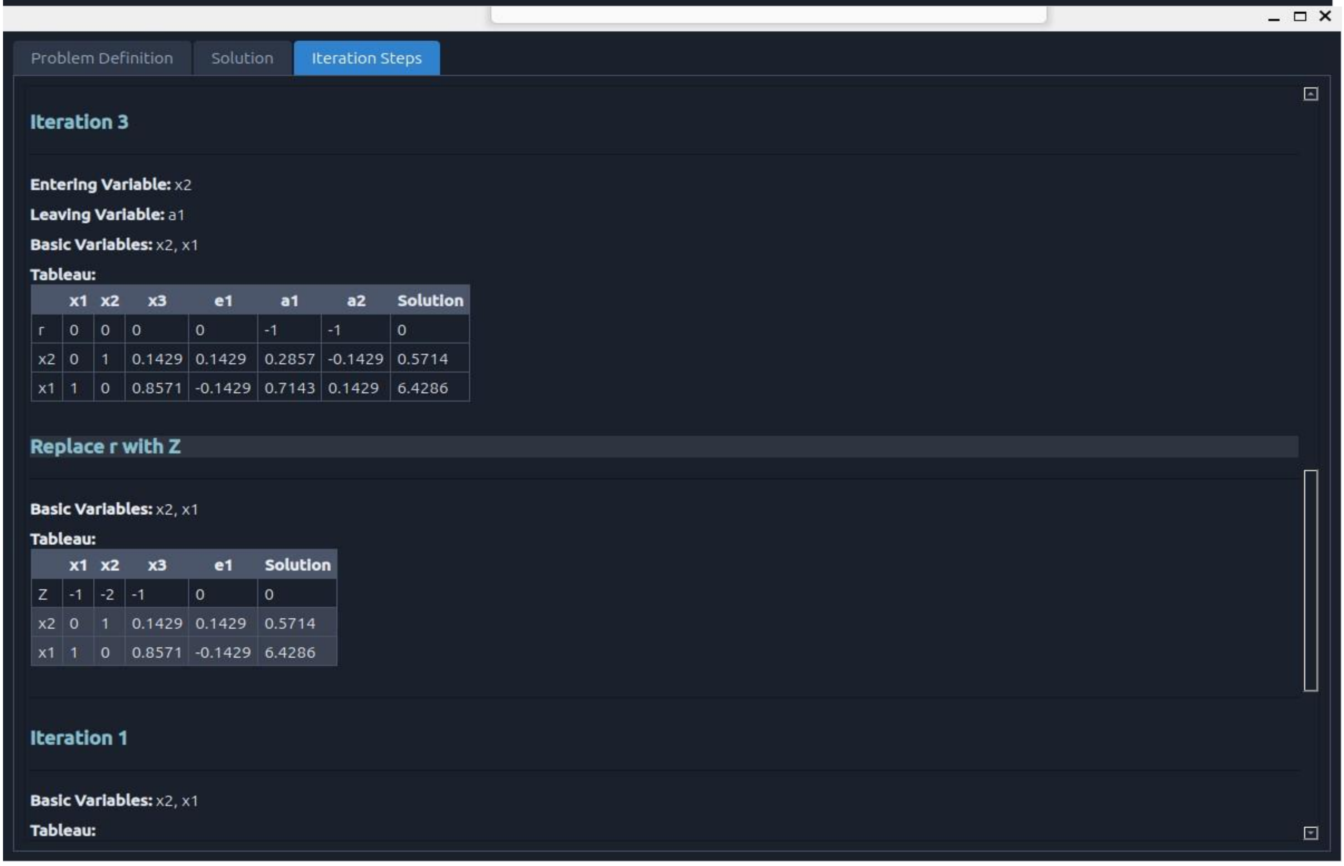
Leaving Variable: a2

Basic Variables: a1, x1

Tableau:

	x1	x2	x3	e1	a1	a2	Solution
r	0	3.5000	0.5000	0.5000	0	-1.5000	2
a1	0	3.5000	0.5000	0.5000	1	-0.5000	2
x1	1	-2.5000	0.5000	-0.5000	0	0.5000	5

Iteration 3



Problem Definition | Solution | Iteration Steps

Iteration 3

Entering Variable: x2

Leaving Variable: a1

Basic Variables: x2, x1

Tableau:

	x1	x2	x3	e1	a1	a2	Solution
r	0	0	0	0	-1	-1	0
x2	0	1	0.1429	0.1429	0.2857	-0.1429	0.5714
x1	1	0	0.8571	-0.1429	0.7143	0.1429	6.4286

Replace r with Z

Basic Variables: x2, x1

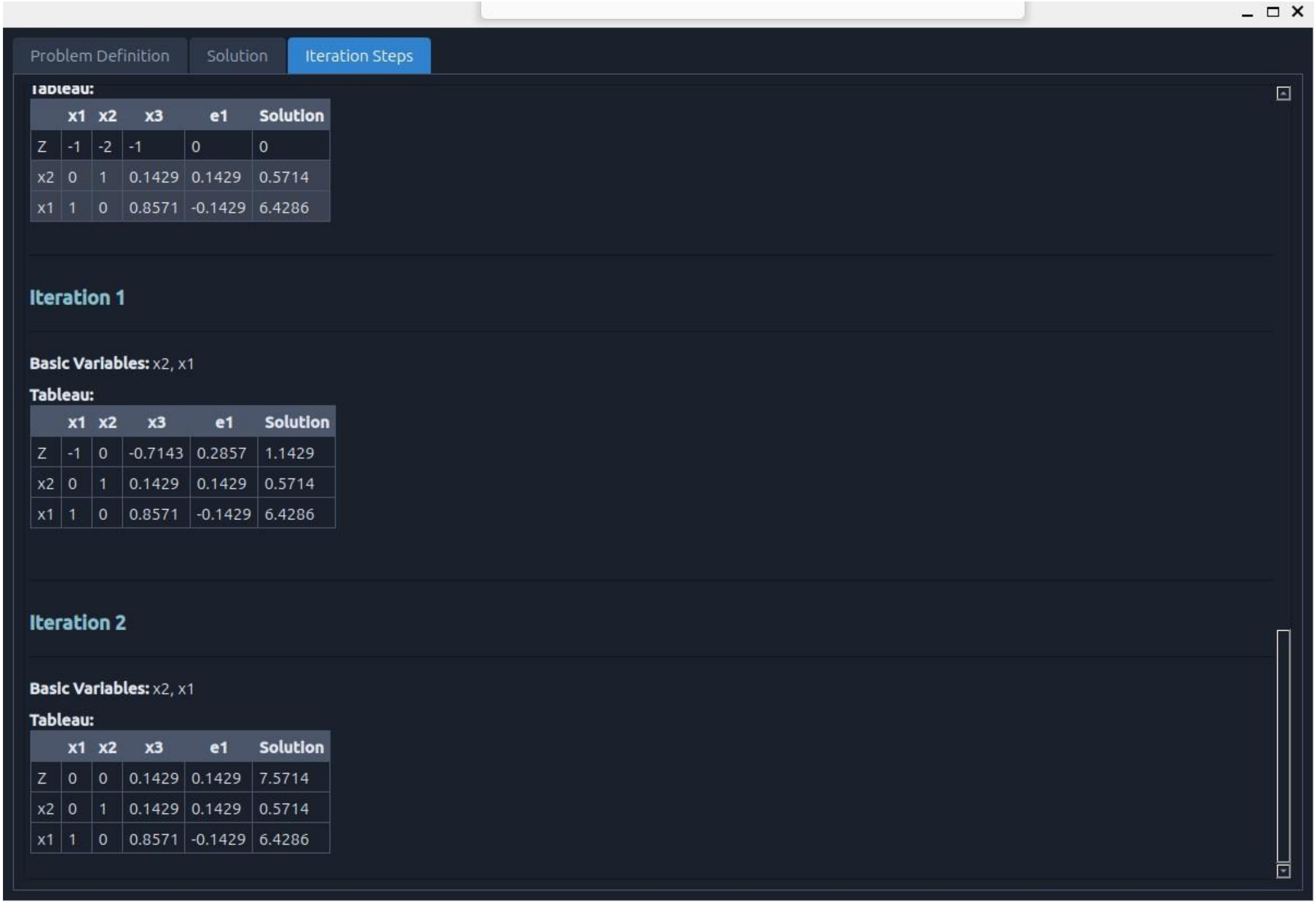
Tableau:

	x1	x2	x3	e1	Solution
Z	-1	-2	-1	0	0
x2	0	1	0.1429	0.1429	0.5714
x1	1	0	0.8571	-0.1429	6.4286

Iteration 1

Basic Variables: x2, x1

Tableau:



	x1	x2	x3	e1	Solution
Z	-1	-2	-1	0	0
x2	0	1	0.1429	0.1429	0.5714
x1	1	0	0.8571	-0.1429	6.4286

	x1	x2	x3	e1	Solution
Z	-1	0	-0.7143	0.2857	1.1429
x2	0	1	0.1429	0.1429	0.5714
x1	1	0	0.8571	-0.1429	6.4286

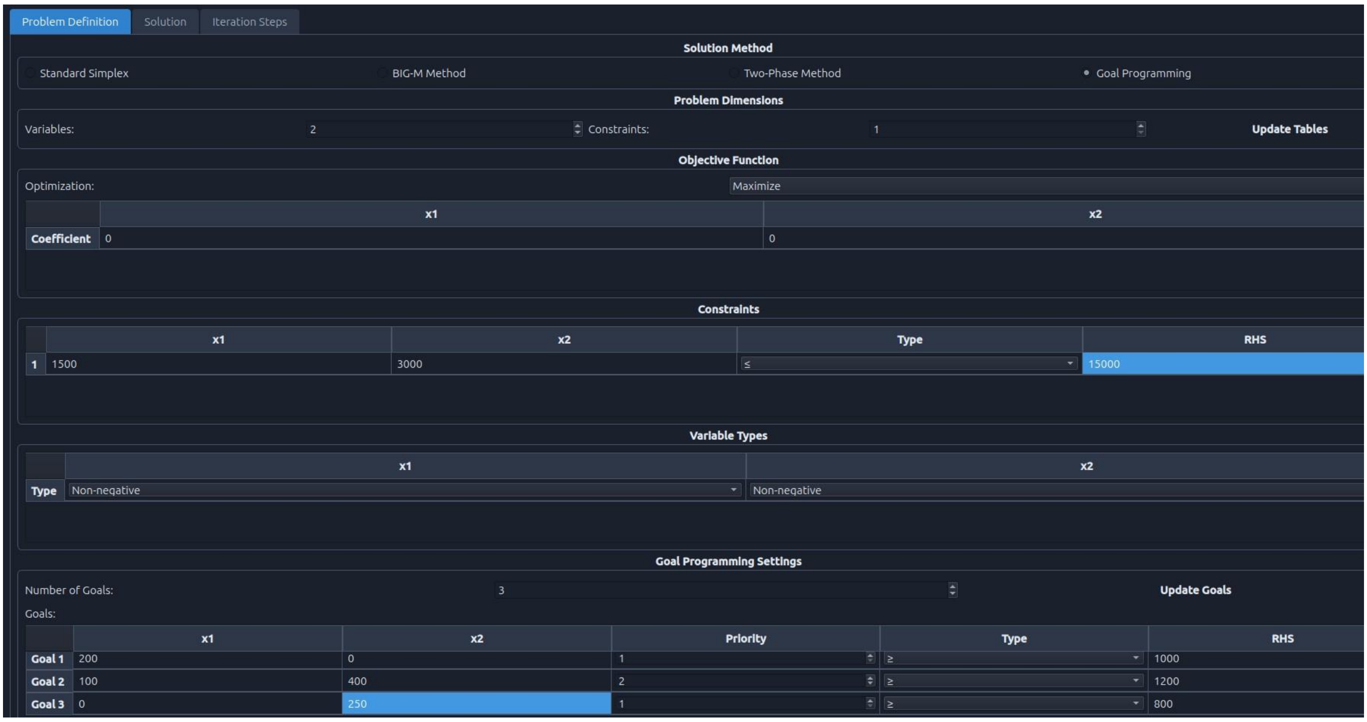
	x1	x2	x3	e1	Solution
Z	0	0	0.1429	0.1429	7.5714
x2	0	1	0.1429	0.1429	0.5714
x1	1	0	0.8571	-0.1429	6.4286

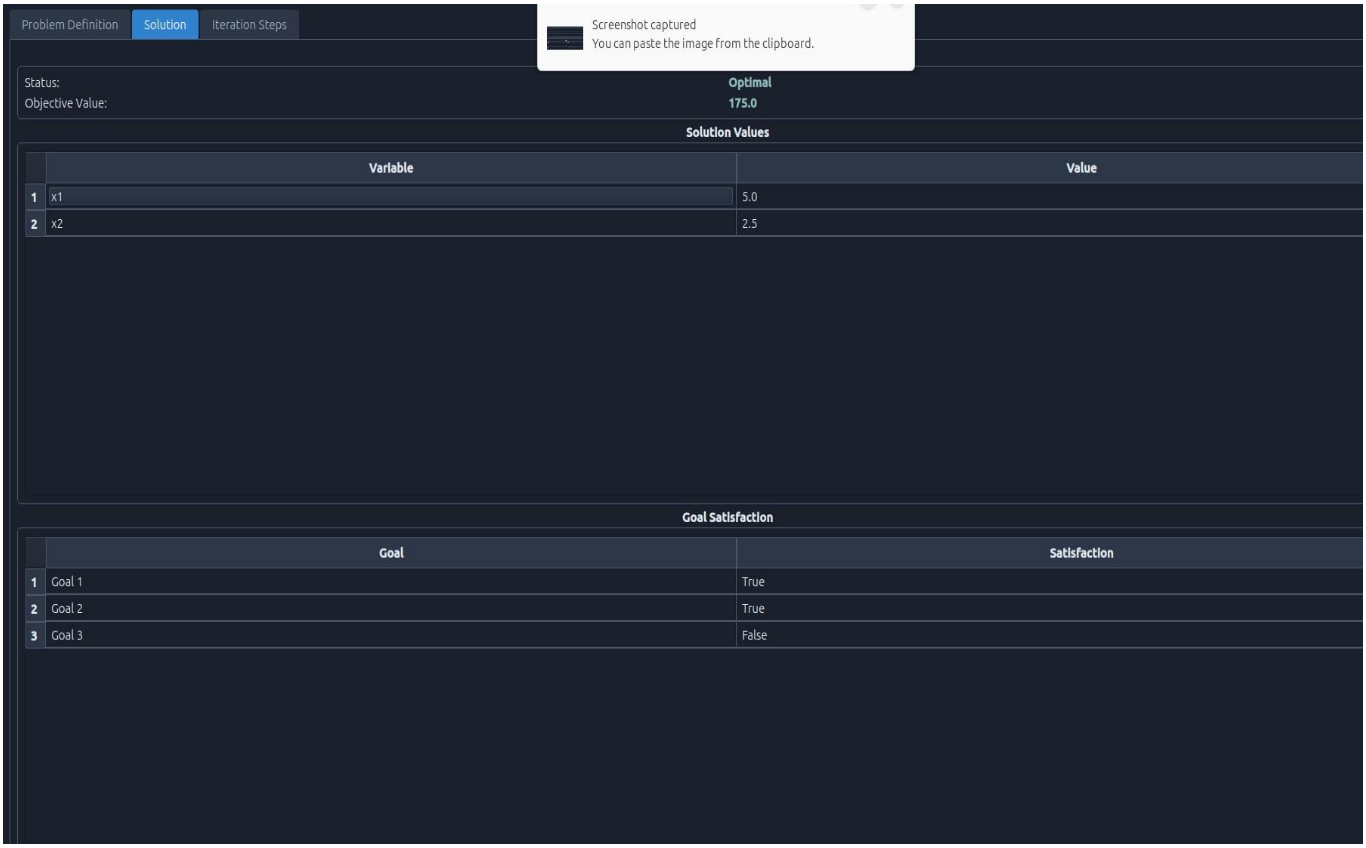
Output:

- Optimal Solution: x1 = 45/7, x2 = 4/7
- Optimal Objective Value: Z = 53/7

5.4 Example 4 - Goal-programming Method

```
Constraints:
1500x1 +3000x2 ≤ 15000

Goals:
200x1>=1000              priority:1
100x1+ 400x2 >=1200      priority:2
250x2 >=800              priority:1
```


Problem Definition
Solution
Iteration Steps
Solution Method
Standard Simplex
BIG-M Method
Two-Phase Method
Goal Programming
Problem Dimensions
Variables: 2
Constraints: 1
Update Tables
Objective Function
Optimization: Maximize
x1
x2
Coefficient 0 0
Constraints
x1 x2 Type RHS
1 1500 3000 ≤ 15000
Variable Types
x1 x2
Type Non-negative Non-negative
Goal Programming Settings
Number of Goals: 3
Update Goals
Goals:
x1 x2 Priority Type RHS
Goal 1 200 0 1 ≥ 1000
Goal 2 100 400 2 ≥ 1200
Goal 3 0 250 1 ≥ 800


Problem Definition
Solution
Iteration Steps
Screenshot captured
You can paste the image from the clipboard.
Status: Optimal
Objective Value: 175.0
Solution Values
Variable Value
1 x1 5.0
2 x2 2.5
Goal Satisfaction
Goal Satisfaction
1 Goal 1 True
2 Goal 2 True
3 Goal 3 False

Problem Definition | Solution | Iteration Steps

Basic Variables: d1-, d2-, d3-, s1

Tableau:

	x1	x2	d1-	d1+	d2-	d2+	d3-	d3+	s1	Solution
Z1	0	0	-1	0	0	0	0	0	0	0
Z2	0	0	0	0	-2	0	0	0	0	0
Z3	0	0	0	0	0	0	-1	0	0	0
d1-	200	0	1	-1	0	0	0	0	0	1000
d2-	100	400	0	0	1	-1	0	0	0	1200
d3-	0	250	0	0	0	0	1	-1	0	800
s1	1500	3000	0	0	0	0	0	0	1	15000

Iteration 1

Basic Variables: d1-, d2-, d3-, s1

Tableau:

	x1	x2	d1-	d1+	d2-	d2+	d3-	d3+	s1	Solution
Z1	200	0	0	-1	0	0	0	0	0	1000
Z2	200	800	0	0	0	-2	0	0	0	2400
Z3	0	250	0	0	0	0	0	-1	0	800
d1-	200	0	1	-1	0	0	0	0	0	1000
d2-	100	400	0	0	1	-1	0	0	0	1200
d3-	0	250	0	0	0	0	1	-1	0	800
s1	1500	3000	0	0	0	0	0	0	1	15000

Iteration 2

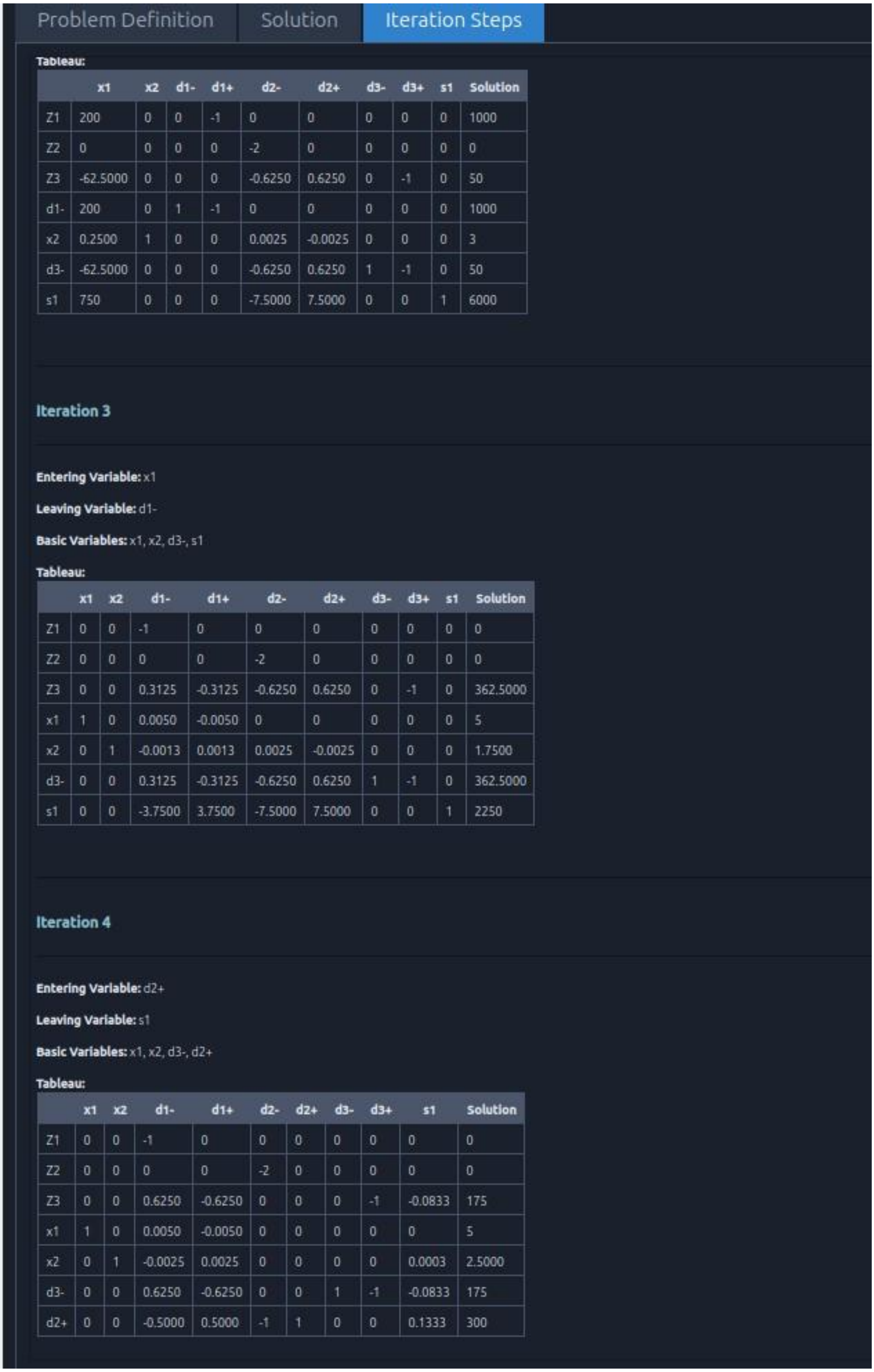
Entering Variable: x2

Leaving Variable: d2-

Basic Variables: d1-, x2, d3-, s1

Tableau:

	x1	x2	d1-	d1+	d2-	d2+	d3-	d3+	s1	Solution
Z1	200	0	0	-1	0	0	0	0	0	1000
Z2	0	0	0	0	-2	0	0	0	0	0
Z3	-62.5000	0	0	0	-0.6250	0.6250	0	-1	0	50
d1-	200	0	1	-1	0	0	0	0	0	1000
x2	0.2500	1	0	0	0.0025	-0.0025	0	0	0	3
d3-	-62.5000	0	0	0	-0.6250	0.6250	1	-1	0	50
s1	750	0	0	0	-7.5000	7.5000	0	0	1	6000



Problem Definition | Solution | Iteration Steps

Tableau:

	x1	x2	d1-	d1+	d2-	d2+	d3-	d3+	s1	Solution
Z1	200	0	0	-1	0	0	0	0	0	1000
Z2	0	0	0	0	-2	0	0	0	0	0
Z3	-62.5000	0	0	0	-0.6250	0.6250	0	-1	0	50
d1-	200	0	1	-1	0	0	0	0	0	1000
x2	0.2500	1	0	0	0.0025	-0.0025	0	0	0	3
d3-	-62.5000	0	0	0	-0.6250	0.6250	1	-1	0	50
s1	750	0	0	0	-7.5000	7.5000	0	0	1	6000

Iteration 3

Entering Variable: x1

Leaving Variable: d1-

Basic Variables: x1, x2, d3-, s1

Tableau:

	x1	x2	d1-	d1+	d2-	d2+	d3-	d3+	s1	Solution
Z1	0	0	-1	0	0	0	0	0	0	0
Z2	0	0	0	0	-2	0	0	0	0	0
Z3	0	0	0.3125	-0.3125	-0.6250	0.6250	0	-1	0	362.5000
x1	1	0	0.0050	-0.0050	0	0	0	0	0	5
x2	0	1	-0.0013	0.0013	0.0025	-0.0025	0	0	0	1.7500
d3-	0	0	0.3125	-0.3125	-0.6250	0.6250	1	-1	0	362.5000
s1	0	0	-3.7500	3.7500	-7.5000	7.5000	0	0	1	2250

Iteration 4

Entering Variable: d2+

Leaving Variable: s1

Basic Variables: x1, x2, d3-, d2+

Tableau:

	x1	x2	d1-	d1+	d2-	d2+	d3-	d3+	s1	Solution
Z1	0	0	-1	0	0	0	0	0	0	0
Z2	0	0	0	0	-2	0	0	0	0	0
Z3	0	0	0.6250	-0.6250	0	0	0	-1	-0.0833	175
x1	1	0	0.0050	-0.0050	0	0	0	0	0	5
x2	0	1	-0.0025	0.0025	0	0	0	0	0.0003	2.5000
d3-	0	0	0.6250	-0.6250	0	0	1	-1	-0.0833	175
d2+	0	0	-0.5000	0.5000	-1	1	0	0	0.1333	300

Output:

Optimal Solution: x1 = 5, x2 = 2.5

Satisfying Goals : G1 and G2

5.5 Example 5 – Unrestricted variables

```
Maximize Z =3 x1 +x2
```

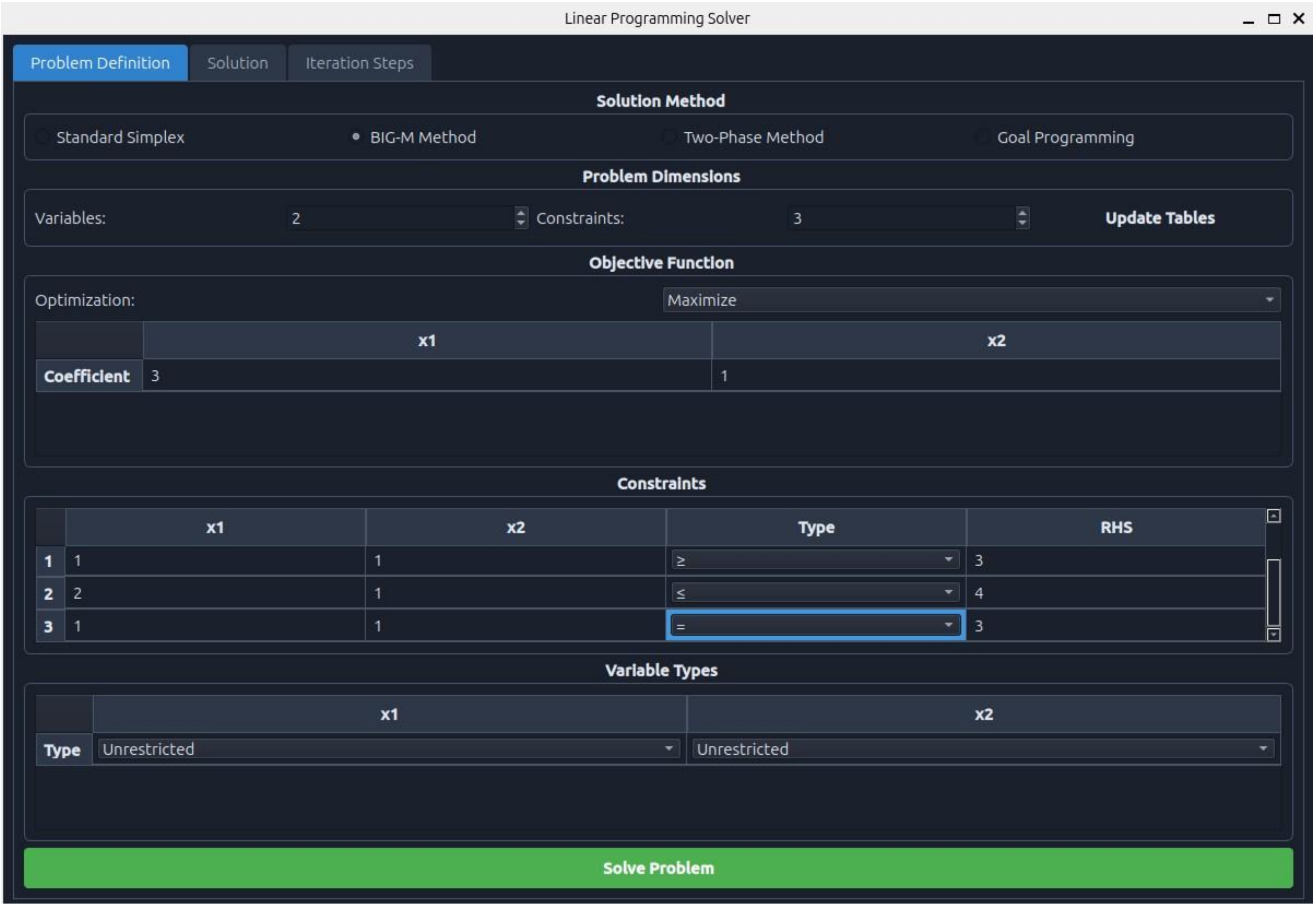
x1 & x2 are unrestricted variables.

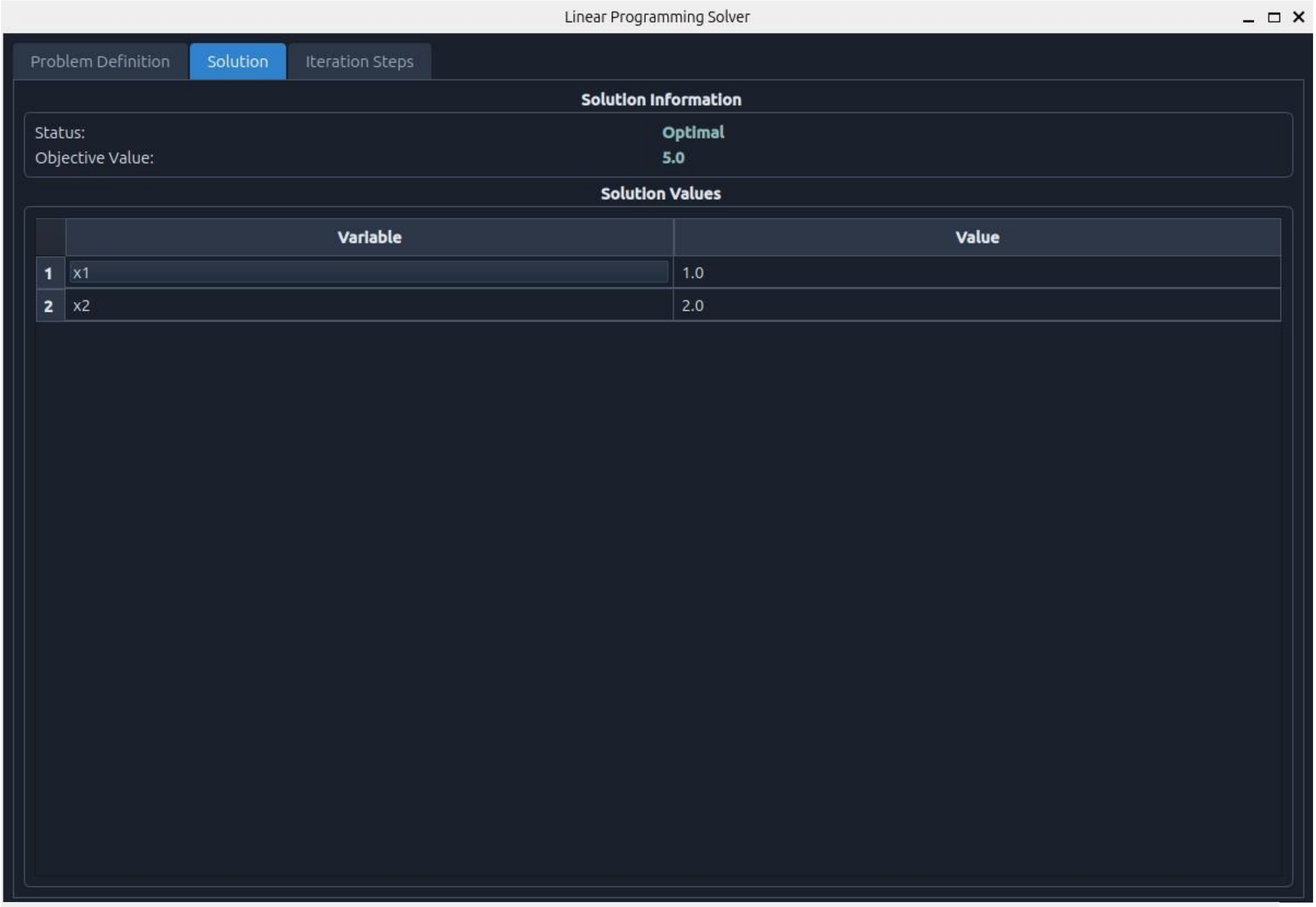
```
Constraints:
```

x1+x2 >=7

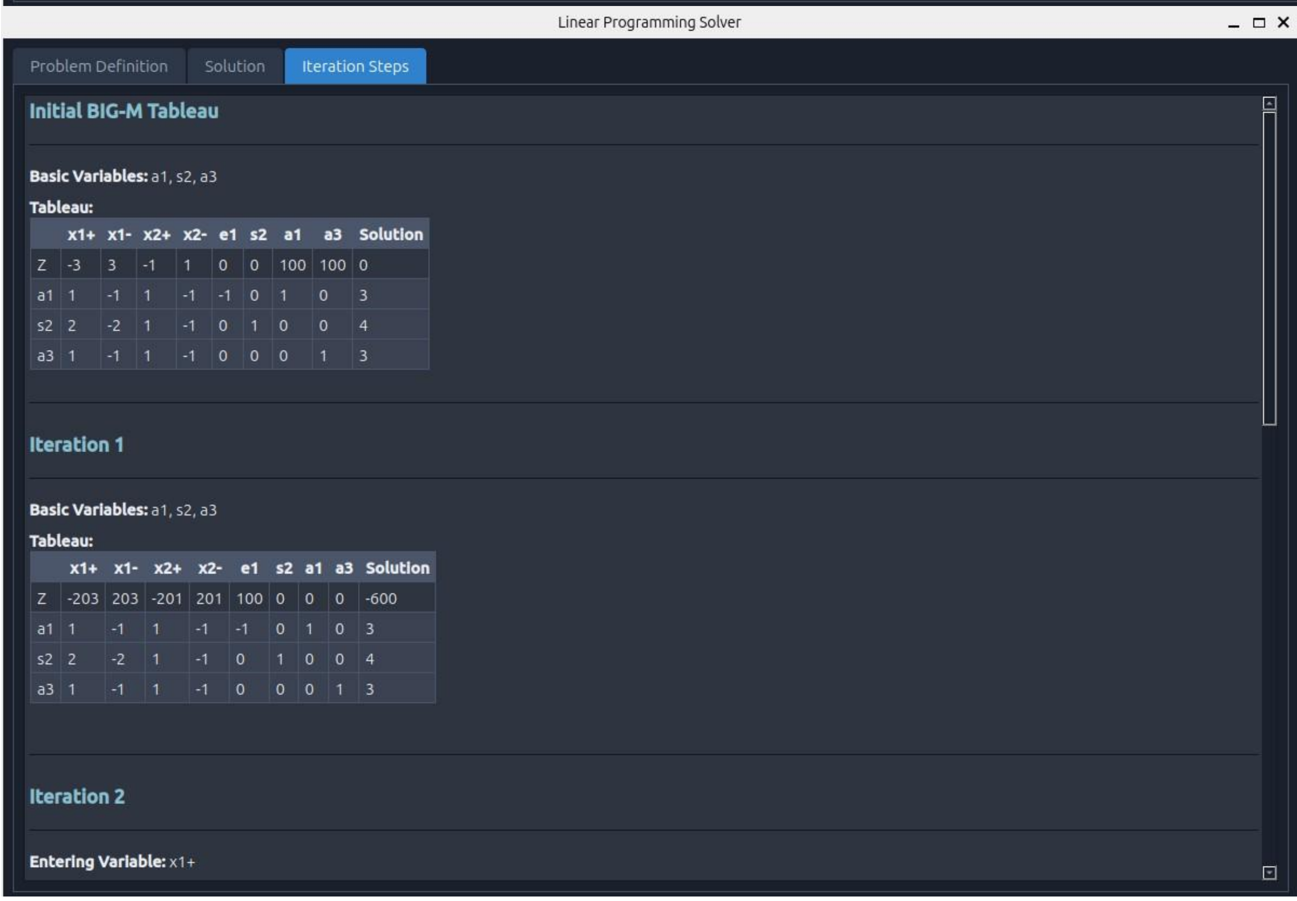
2x1+ x2 <=4

x1 + x2=800




Linear Programming Solver
Problem Definition
Solution
Iteration Steps
Solution Information
Status: Optimal
Objective Value: 5.0
Solution Values

	Variable	Value
1	x1	1.0
2	x2	2.0

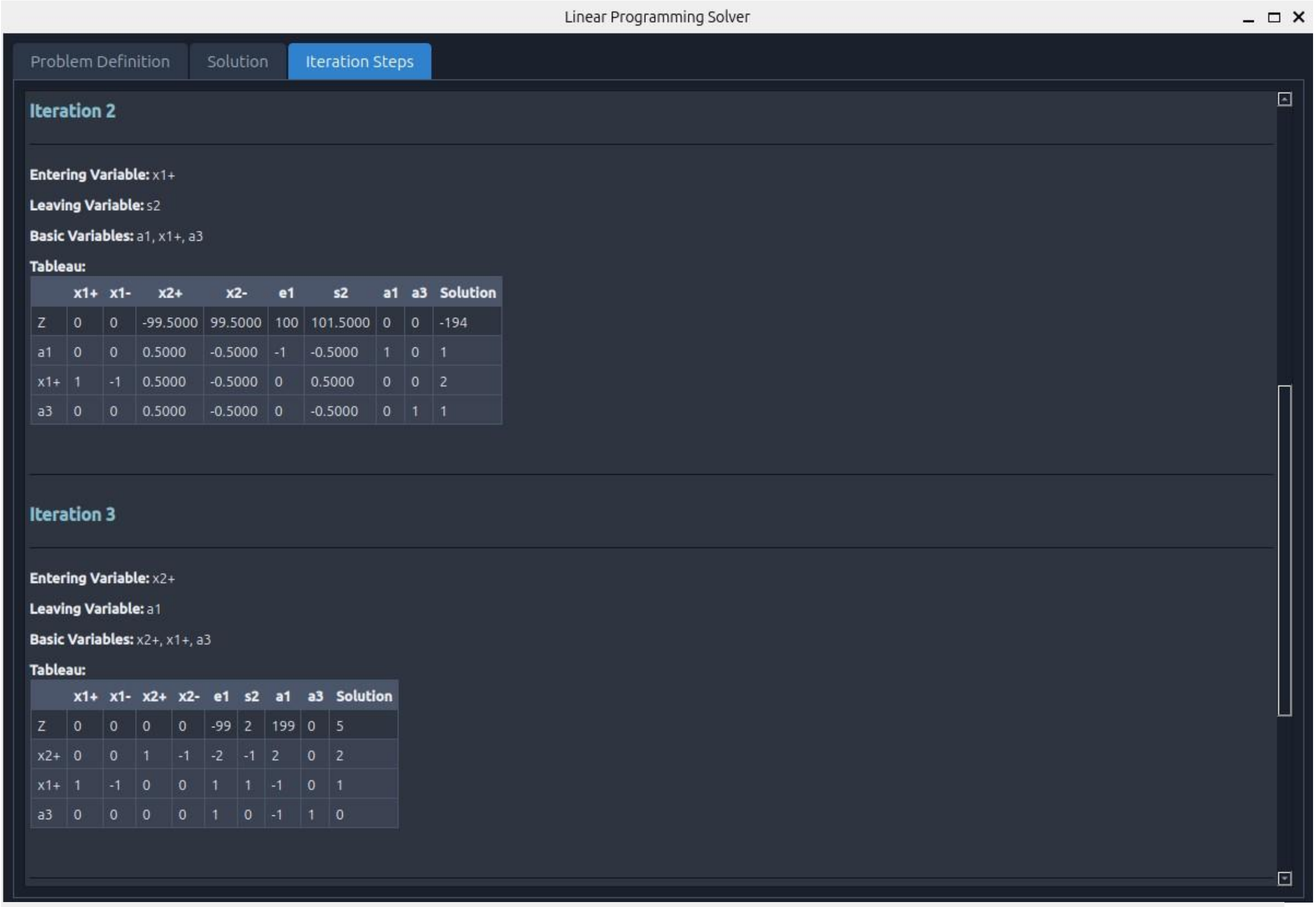

Linear Programming Solver
Problem Definition
Solution
Iteration Steps
Initial BIG-M Tableau
Basic Variables: a1, s2, a3
Tableau:

	x1+	x1-	x2+	x2-	e1	s2	a1	a3	Solution
Z	-3	3	-1	1	0	0	100	100	0
a1	1	-1	1	-1	-1	0	1	0	3
s2	2	-2	1	-1	0	1	0	0	4
a3	1	-1	1	-1	0	0	0	1	3

Iteration 1
Basic Variables: a1, s2, a3
Tableau:

	x1+	x1-	x2+	x2-	e1	s2	a1	a3	Solution
Z	-203	203	-201	201	100	0	0	0	-600
a1	1	-1	1	-1	-1	0	1	0	3
s2	2	-2	1	-1	0	1	0	0	4
a3	1	-1	1	-1	0	0	0	1	3

Iteration 2
Entering Variable: x1+



Linear Programming Solver

Problem Definition | Solution | Iteration Steps

Iteration 2

Entering Variable: x1+

Leaving Variable: s2

Basic Variables: a1, x1+, a3

Tableau:

	x1+	x1-	x2+	x2-	e1	s2	a1	a3	Solution
Z	0	0	-99.5000	99.5000	100	101.5000	0	0	-194
a1	0	0	0.5000	-0.5000	-1	-0.5000	1	0	1
x1+	1	-1	0.5000	-0.5000	0	0.5000	0	0	2
a3	0	0	0.5000	-0.5000	0	-0.5000	0	1	1

Iteration 3

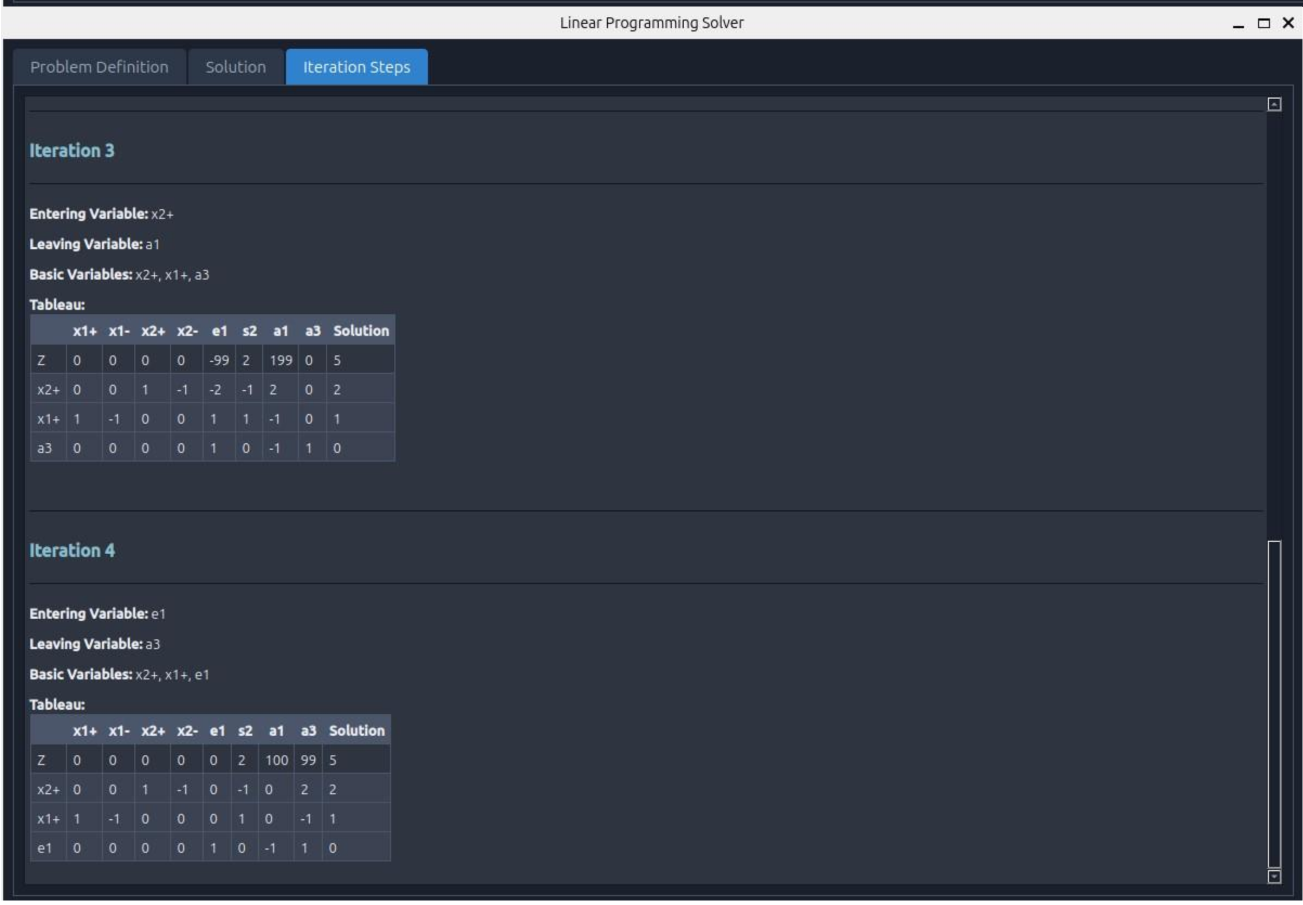
Entering Variable: x2+

Leaving Variable: a1

Basic Variables: x2+, x1+, a3

Tableau:

	x1+	x1-	x2+	x2-	e1	s2	a1	a3	Solution
Z	0	0	0	0	-99	2	199	0	5
x2+	0	0	1	-1	-2	-1	2	0	2
x1+	1	-1	0	0	1	1	-1	0	1
a3	0	0	0	0	1	0	-1	1	0



Linear Programming Solver

Problem Definition | Solution | Iteration Steps

Iteration 3

Entering Variable: x2+

Leaving Variable: a1

Basic Variables: x2+, x1+, a3

Tableau:

	x1+	x1-	x2+	x2-	e1	s2	a1	a3	Solution
Z	0	0	0	0	-99	2	199	0	5
x2+	0	0	1	-1	-2	-1	2	0	2
x1+	1	-1	0	0	1	1	-1	0	1
a3	0	0	0	0	1	0	-1	1	0

Iteration 4

Entering Variable: e1

Leaving Variable: a3

Basic Variables: x2+, x1+, e1

Tableau:

	x1+	x1-	x2+	x2-	e1	s2	a1	a3	Solution
Z	0	0	0	0	0	2	100	99	5
x2+	0	0	1	-1	0	-1	0	2	2
x1+	1	-1	0	0	0	1	0	-1	1
e1	0	0	0	0	1	0	-1	1	0

5.6 Example 6 – Unbounded solution

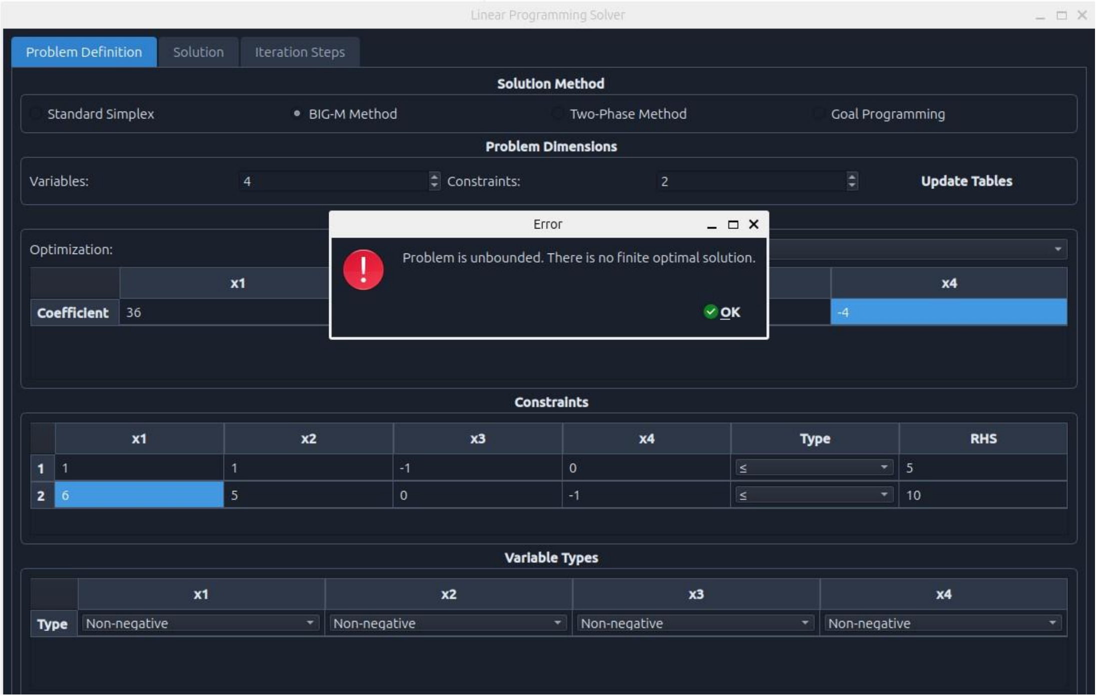
```
Maximize Z =36 x1 +30x2 -3 x3 -4x4
```

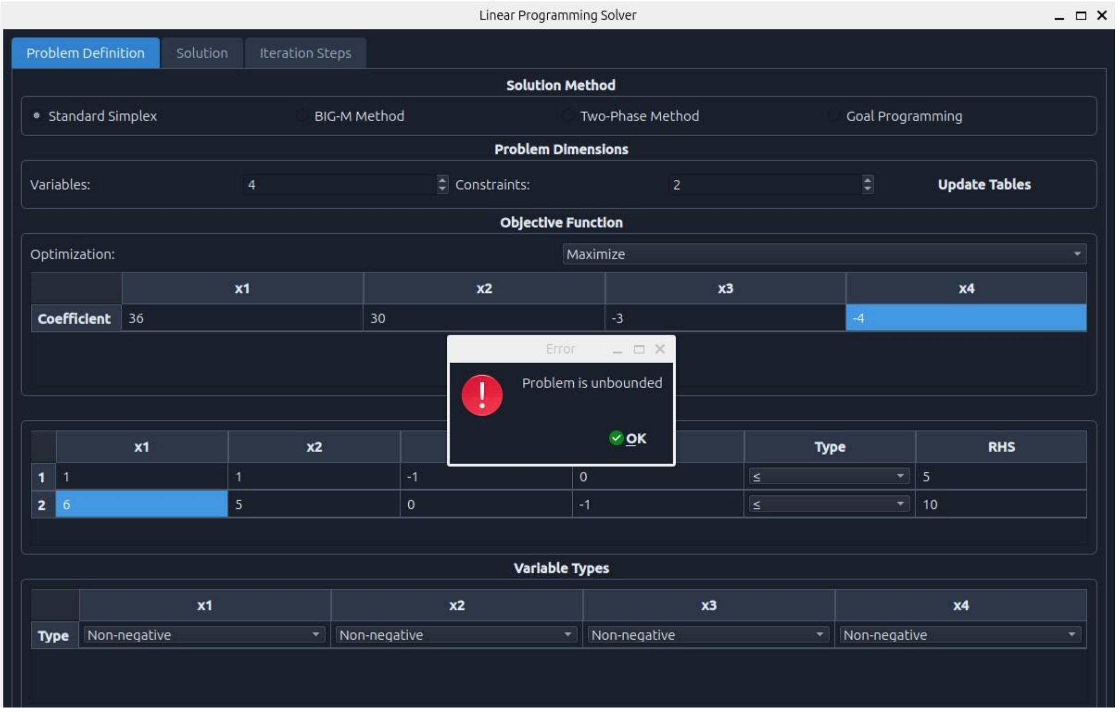
```
Constraints:
```

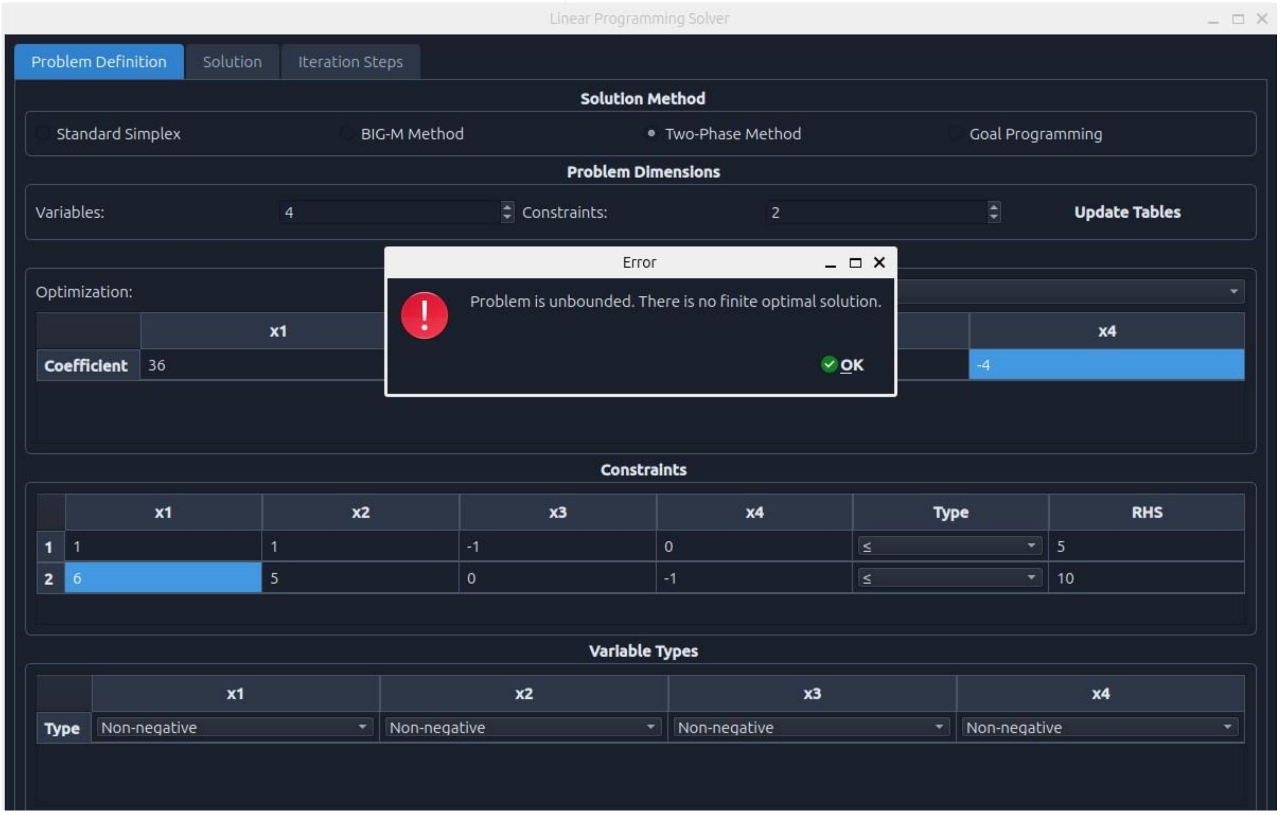
x1+x2 -x3 <=5

6x1+ 5x2 – x4 <=10

x1 , x2 , x3 , x4 > =0

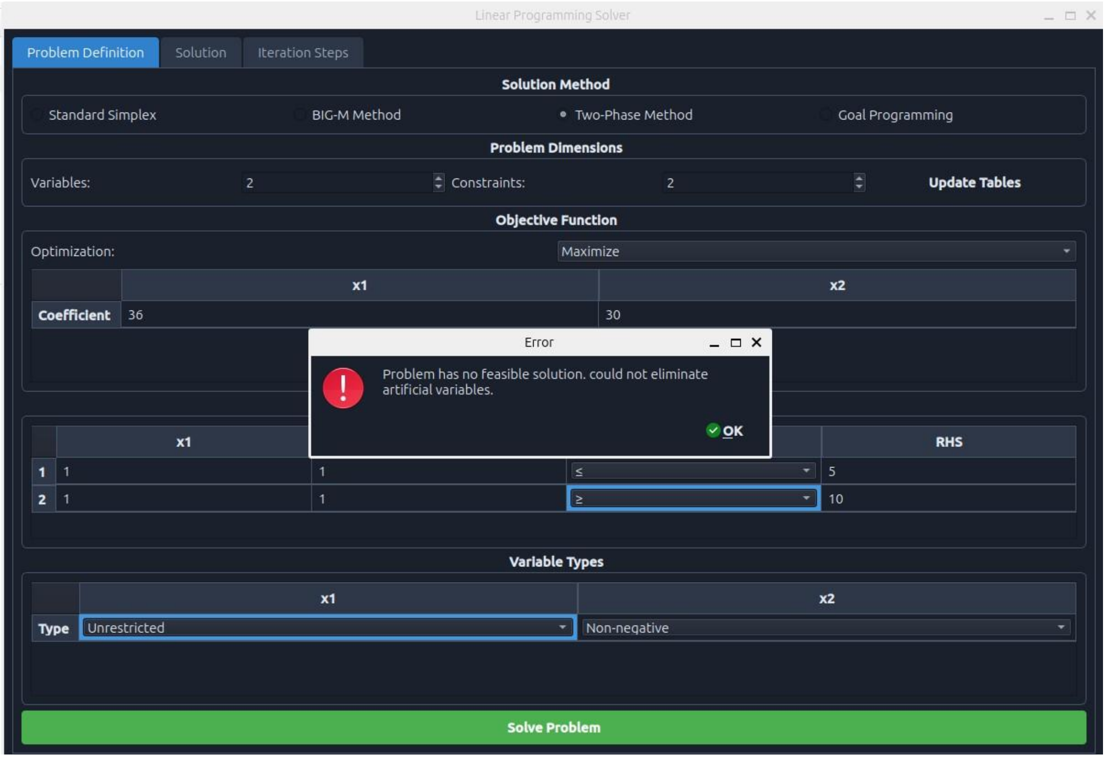






5.7 Example 7 – infeasible solution

```
Maximize Z =36 x1 +30x2                    Constraints:
                                           x1+x2 >=5

x1 , x2 > =0                               x1+ x2 <=5
```



6. Bonus Feature

We developed a **user-friendly interface** by `pyQt5 in python` that allows users to input LP problems easily and view the solution process interactively.

7. Conclusion

This project provided hands-on experience in solving LP problems using different methods. The solver successfully handles various constraints and outputs detailed step-by-step solutions. Future improvements include extending support for additional optimization techniques and graphical visualization of results.