# Solution to LPP using Dynamic Programming.

Submitted in partial fulfilment of the requirements of the degree

### BACHELOR OF ENGINEERING IN Artificial Intelligence and Data Science

By

**Drishti Bathija 06**

**Harsh Bhatia 09**

**Meet Brijwani 14**

**Vinay Dawani 22**

**Supervisor. : Dr. G.T.Thampi**



**Artificial Intelligence and Data Science Department**

**Thadomal Shahani Engineering College**

**Bandra(w), Mumbai - 400 050**

## University of Mumbai

**(AY 2023-24)**

# CERTIFICATE

This is to certify that the Mini Project entitled **“Solutions to LPP using Dynamic Programming”** is a bona fide work of

1. Drishti Bathija 06
2. Harsh Bhatia 09
3. Meet Brijwani 14
4. Vinay Dawani 22

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **“Bachelor of Engineering”** in **“Artificial Intelligence and Data Science Engineering” .**

#### (Dr. G. T.Thampi)

Supervisor

#### (Prof. Madhuri Rao) (Prof. Dr.G.T.Thampi)

Head of Department Principal

# Mini Project Approval

This Mini Project entitled “ **Solutions to LPP using Dynamic Programming ”** by

1. Drishti Bathija 06
2. Harsh Bhatia 09
3. Meet Brijwani 14
4. Vinay Dawani 22

is approved for the degree of **Bachelor of Engineering** in **“Artificial Intelligence and Data Science Engineering”**

**Examiners**

**1………………………………………**

(Internal Examiner Name & Sign)

#### 2…………………………………………

(External Examiner name & Sign)

Date:

Place:

# Contents

#### Acknowledgments i

#### 

#### Introduction 1

* 1. Motivation
  2. Problem Statement & Objectives
  3. Organization of the Report

1. **Literature Survey 2** 
   1. Survey of Existing System
   2. Limitation Existing system or research gap
   3. Mini Project Contribution

#### Proposed System (eg New Approach of Data Summarization) 3

* 1. Introduction
  2. Architecture/ Framework
  3. Algorithm and Process Design
  4. Details of Hardware & Software
  5. Experiment and Results
  6. Conclusion and Future work.

**References 7**

# 

# Acknowledgement

We would like to express our gratitude and thanks to **Prof. Himani Deshpande** for her valuable guidance and help. We are indebted for her guidance and constant supervision as well as for providing necessary information regarding the project. We would like to express our greatest appreciation to our principal **Dr. G.T. Thampi** and head of the department **Prof.Madhuri Rao** for their encouragement and tremendous support. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of the project.

1. Drishti Bathija 06
2. Harsh Bhatia 09
3. Meet Brijwani 14
4. Vinay Dawani 22

# Chapter 1

## Introduction

This chapter explains the aim, objectives and scope of the proposed system.

### Introduction

Our topic Linear Programming is a powerful mathematical method used to find the best possible outcome in a given mathematical model for a set of constraints. It involves optimizing a linear objective function, subject to linear inequality and equality constraints. The Solution of Linear Programming Problems (LPP) has wide applications in various fields such as economics, engineering, operations research, and more. The simplex method is one of the primary and widely used algorithms to solve LPP. However, in this mini-project, we explore an alternative approach by leveraging Dynamic Programming to solve Linear Programming Problems.

### Motivation

The motivation of the simplex method is to provide an efficient and systematic approach for solving linear programming problems by iteratively moving from one vertex to another in the feasible region, optimizing a linear objective function while ensuring mathematical rigor and practical applicability in various fields, including operations research, economics, and engineering.

### Problem Statement & Objectives

**Problem :**

**Solution of Linear Programming Problem (LPP) using Dynamic Programming**

Develop a Java program

for the following problem

description the project work shall consist of coding for the functional requirement of the problem description and shall demonstrate various Object-oriented programming constructs and methodologies for example

constructors, сору constructors, destructors, inheritance, polymorphism, object as function parameters, exception handling, container classes and GUI with **AWT** components

**Objective:**

The primary objective of this project is to design, implement, and evaluate a Java-based solution that applies Dynamic Programming concepts to solve Linear Programming Problems. By utilizing Dynamic Programming, we aim to create a computational method that efficiently optimizes the objective function within the constraints defined by the problem. This implementation seeks to showcase the applicability and effectiveness of Dynamic Programming in solving Linear Programming Problems.

### Following are examples of problems that can be solved:

* **Minimization**

Q) A dietician wishes to mix two types of foods in such a way that vitamin contents of the mixture contain atleast 8 units of vitamin A and 10 units of vitamin C. Food ‘I’ contains 2 units/kg of vitamin A and 1 unit/kg of vitamin C. Food ‘II’ contains 1 unit/kg of vitamin A and 2 units/kg of vitamin C. It costs Rs 50 per kg to purchase Food ‘I’ and Rs 70 per kg to purchase Food ‘II’. Formulate

this problem as a linear programming problem to minimise the cost of such a mixture.

Z = 50x + 70y

2x + y ≥ 8

x + 2y ≥ 10

x, y ≥ 0

* **Maximization**

Q) A store has requested a manufacturer to produce pants and sports jackets. For materials, the manufacturer has 750 m^2 $ of cotton textile and 1000 m^2 $ of polyester. Every pair of pants (1 unit) needs 1 m^2 $ of cotton and 2 m ^2$ of polyester. Every jacket needs 1.5 m ^2 $ of cotton and 1 m^2 $ of polyester. The price of the pants is fixed at $50 and the jacket, $40. What is the number of pants and jackets that the manufacturer must give to the stores so that these items obtain a maximum sale?

Z = 50x + 40y

x + 1.5y <=750

2x+3y <= 1500

2x + y <= 1000

 x ≥ 0, y ≥ 0

### Organization of the Report

This report consists of three chapters. The first chapter deals with introduction of the topic, problem statement, motivation behind the topic and objectives. The second chapter is the Literature Survey. It includes all the research work done related to this topic. All information related to study of existing systems as well as learning of new tools is mentioned in this chapter. The third chapter is about the proposed system which is used in this project. The block diagram, techniques used, hardware and software used screenshots of the project are presented in this chapter. All the documents related to development of this project are mentioned in Reference

# Chapter 2

## Literature Survey

This chapter explains the concepts used in this project, study of existing system and contribution of this project

### Survey of Existing System

### There are several online linear programming calculators and software tools available on the internet that can help you solve Linear Programming Problems (LPP). Here are a few examples of LPP calculators and software:

### 1. Wolfram Alpha: Wolfram Alpha is a powerful computational engine that can solve a variety of mathematical problems, including linear programming. It allows you to input the objective function, constraints, and variables to find the optimal solution.

### 2. OR-Tools by Google: Google's OR-Tools is an open-source software suite for optimization problems, including linear programming. It can be used with various programming languages like Python, Java, and others.

### There are several websites that offer online LPP solvers where you can input your problem's data and constraints, and the solver will find the optimal solution. Some of these may not be as feature-rich as dedicated software but are simple to use for basic problems.

### Limitation of existing system

Limitation of these applications is-

They are not dynamic in regards of number of constraints and number of variables and they do not provide SIMPLEX Solution in step wise showing each Iteration.

### Mini Project Contribution

Our simplex method program can contribute significantly to various industries and fields by providing solutions to complex linear programming problems. It helps organizations make informed decisions, maximize efficiency, and improve resource utilization, ultimately leading to cost savings, better service delivery, and more sustainable practices.

# Chapter 3

## Proposed System

This chapter consists of detailed description about the methodology used, the hardware and software components, the tools used and also the screenshots of the project

### Introduction

The simplex method is a widely used mathematical optimization technique for solving linear programming problems. It is particularly valuable in a variety of real-world applications, such as supply chain management, resource allocation, and production planning. This program is designed to implement the simplex method algorithm, allowing users to efficiently find optimal solutions to linear programming problems.

### Architecture

The program is written in Java and a graphical user interface (GUI) application for solving linear programming problems using the Simplex method. The architecture of the program can be broken down into the following components:

1. **Main Class (Main.java):**

- This class serves as the entry point of the program.

- It creates and configures the graphical user interface (GUI) for the linear programming problem solver.

- The GUI includes components like labels, combo boxes, buttons, text areas, and panels.

2. **GUI Components:**

- The program uses various Swing components for the user interface, such as JFrame, JLabel, JComboBox, JButton, JTextArea, and JTextField. These components are placed on the frame and panels to create the user interface.

3. **Event Handling:**

- The program uses event listeners and lambda expressions to handle user interactions. For example, it handles button clicks to generate the input fields based on the number of variables and constraints, as well as to perform the simplex calculation.

4. **Tableau and Simplex Logic**

- The program contains logic for creating and maintaining the tableau, which is a matrix used in linear programming to solve optimization problems.

- It uses the Simplex method to perform iterations to find the optimal solution.

- The tableau and Simplex-related methods are included in the code.

5. **Displaying Results:**

- The program displays the results of the linear programming problem, such as the objective value (Z) and decision variables (x1, x2, etc.), in a text area within the GUI.

6. **Utility Methods:**

- The program includes utility methods for tasks like finding non-basic variables, counting non-zero entries in columns, and checking for negative entries in the tableau.

7. **Data Storage:**

- The program uses several class-level fields to store data, including the tableau, the number of variables and constraints, the type of the objective function (maximization or minimization), and the decision variables.

8. **Objective Type:**

- The code handles both maximization and minimization objective functions.

9. **User Interaction:**

- The program takes user input to specify the number of variables and constraints, coefficients, and constraint types (≤ or ≥).

10. **Graphical User Interface:**

- The program presents the user with a visually appealing interface to enter the linear programming problem.

Overall, the program has a modular structure with separate sections for GUI handling, Simplex logic, and utility methods. It aims to provide users with an interactive tool to solve linear programming problems and visualize the results. The GUI components are used to collect input data, perform calculations, and display the results of the optimization problem.

### Algorithm and Process Design

1. Formulating the Problem statement:

By reading our problem statement we can understand that we have to make inequalities into equation using slack/surplus variable.

1. Understanding the framework and requirements:

The framework goes as follows:

* Taking number of variables, constraints and coefficients from the user
* Converting the user's inequation input to an equation which can be used to evaluate results
* Creating dynamic GUI based on user's input
* Performing simplex algorithm on the equations
* Showing optimal output in GUI

1. Identifying tools/technology to be used:

Tools that were used to create the solution:

* Java programming language
* Visual Studio Code editor(VS code).

1. Finalizing the features to be included:

* Input and Data Representation
* Sensitivity Analysis
* Solving LPP
* User-Friendly Interface

1. Development:
2. Testing
3. Evaluation

### Details of Hardware & Software

RAM 🡪 8GB DDR5

GPU 🡪 4GB Nvidia RTX3050

OS 🡪 Microsoft 11

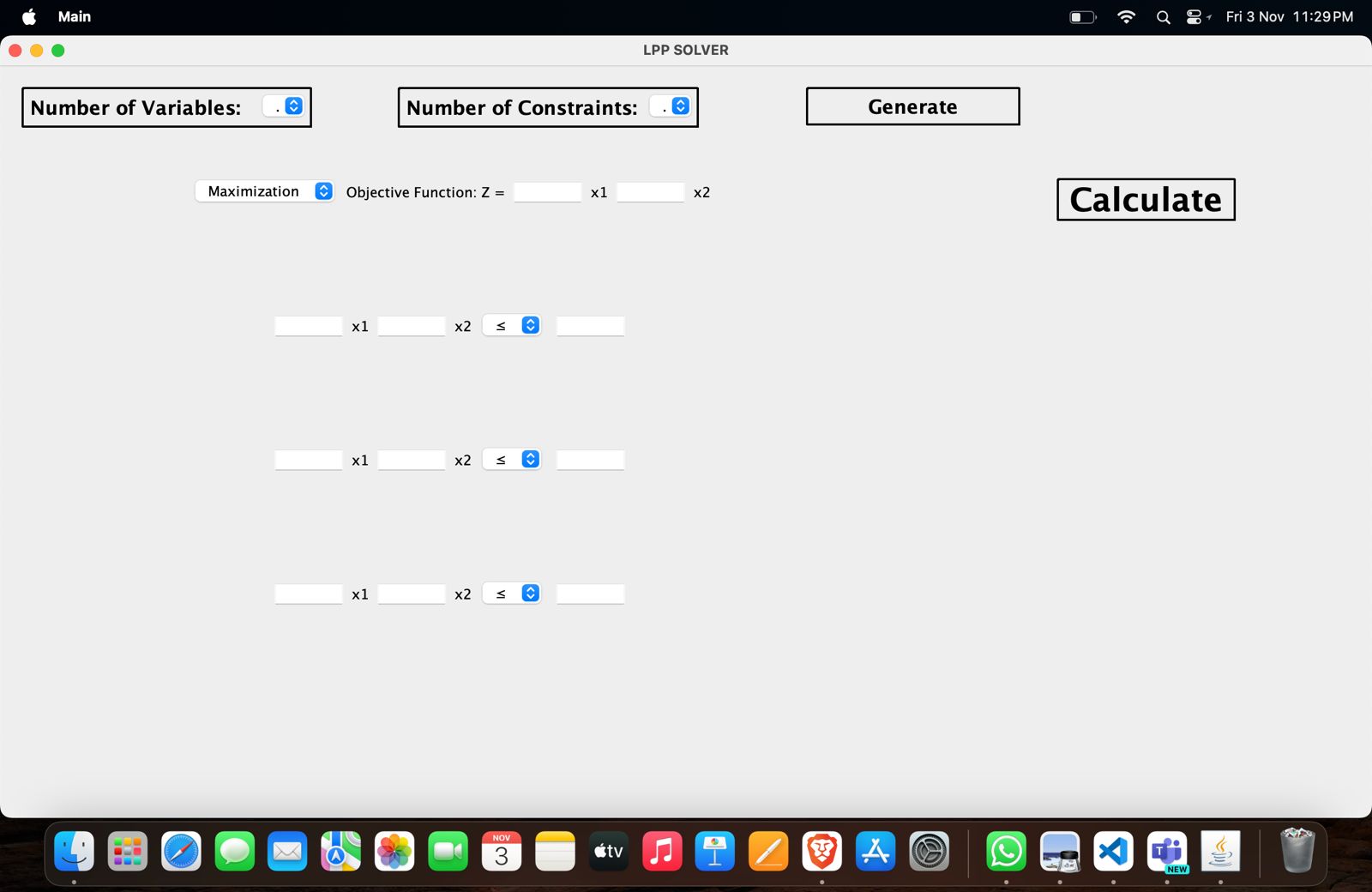
Processor 🡪 Ryzen 5 5600H

Memory 🡪 512GB SSD

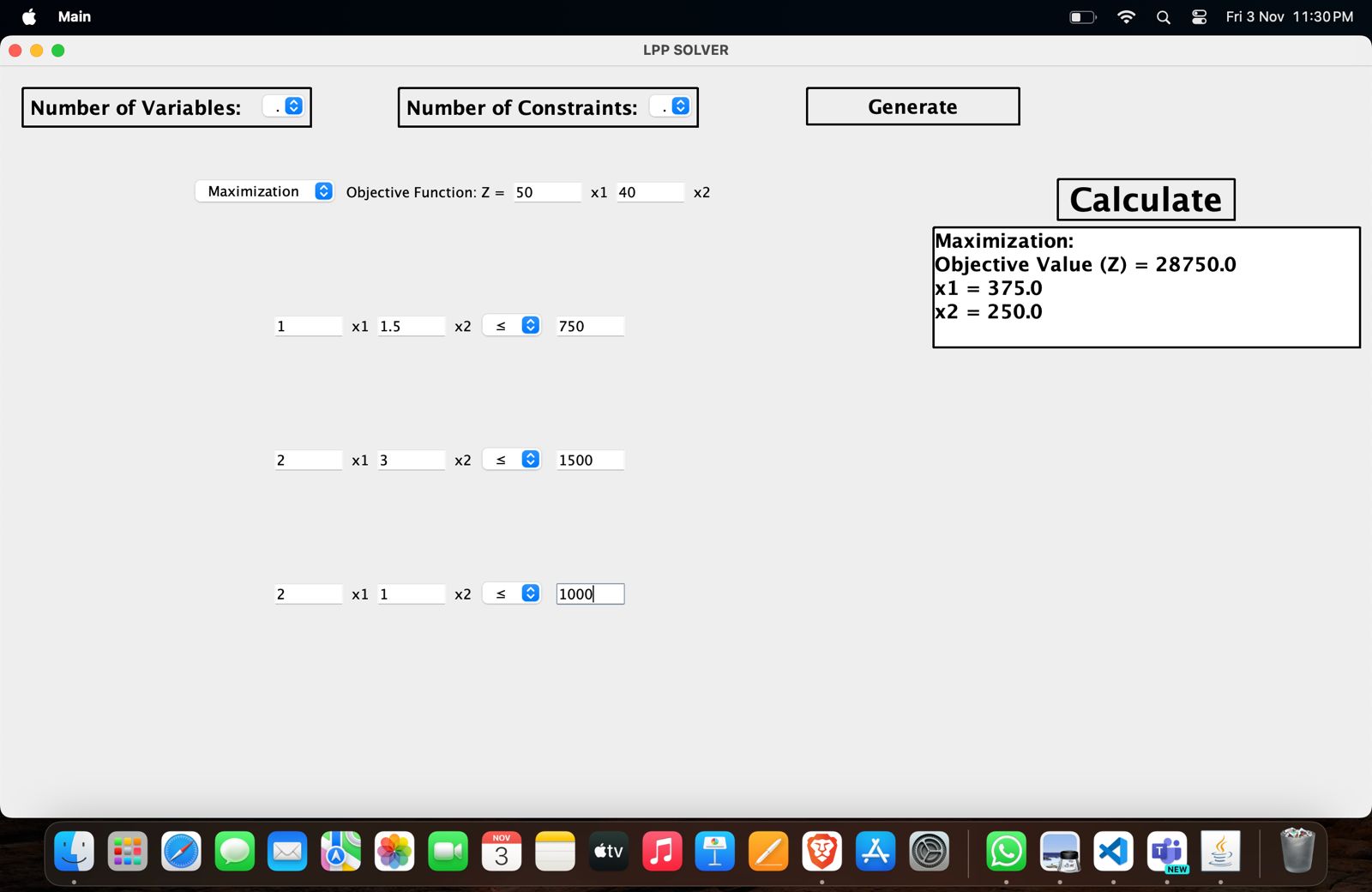
### Results

### 

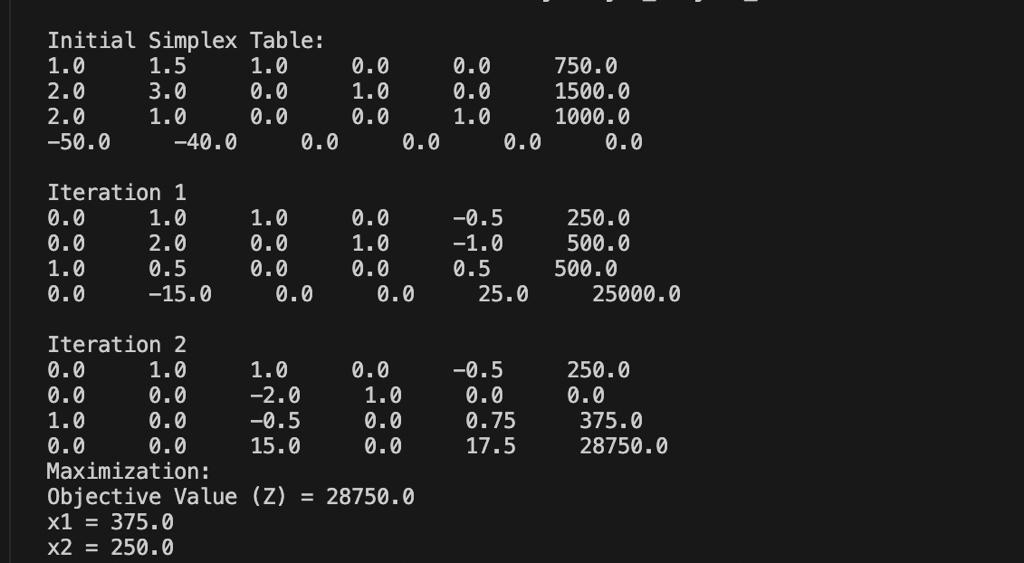
**GUI:**



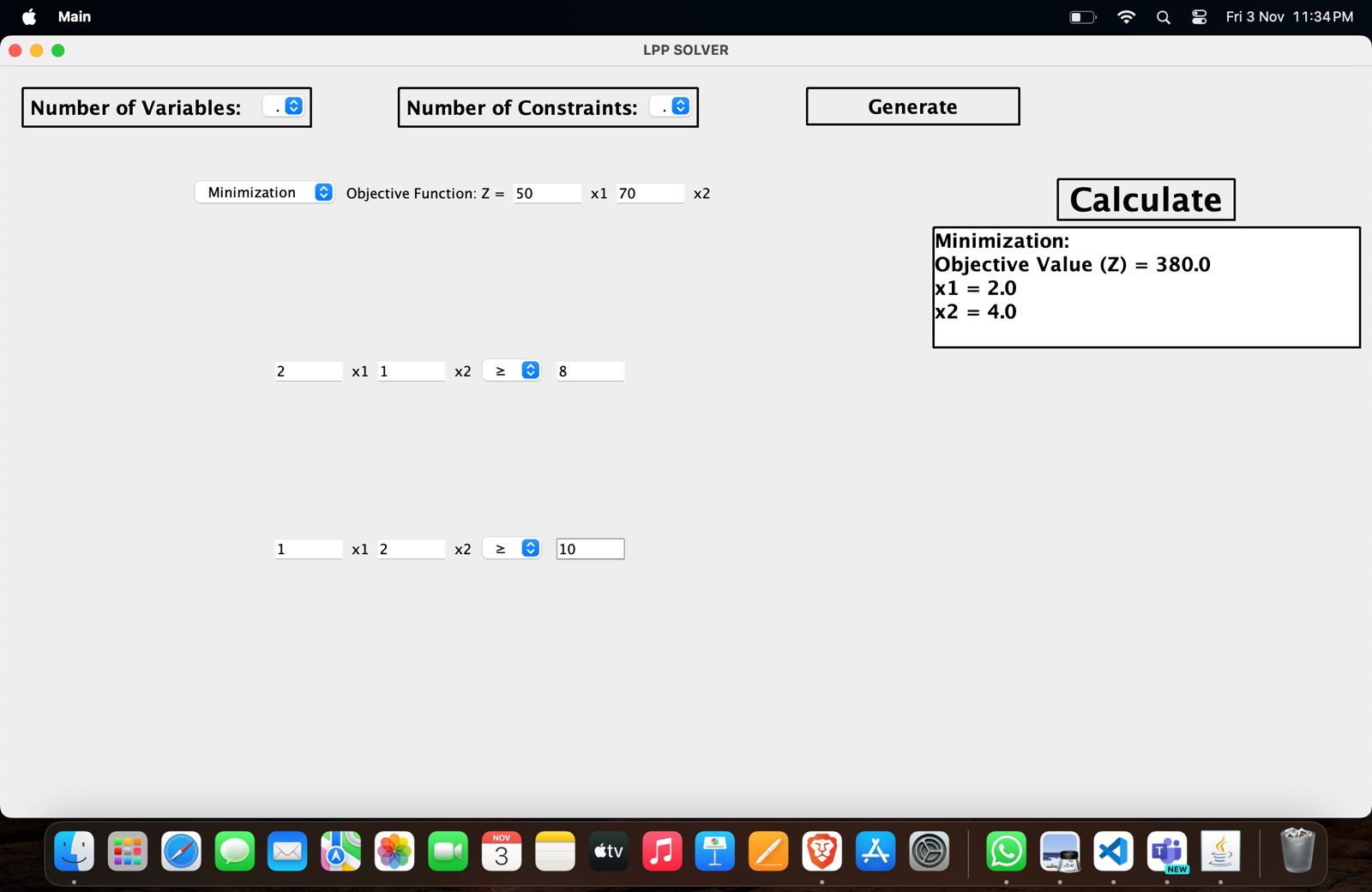
**Maximization Problem:**



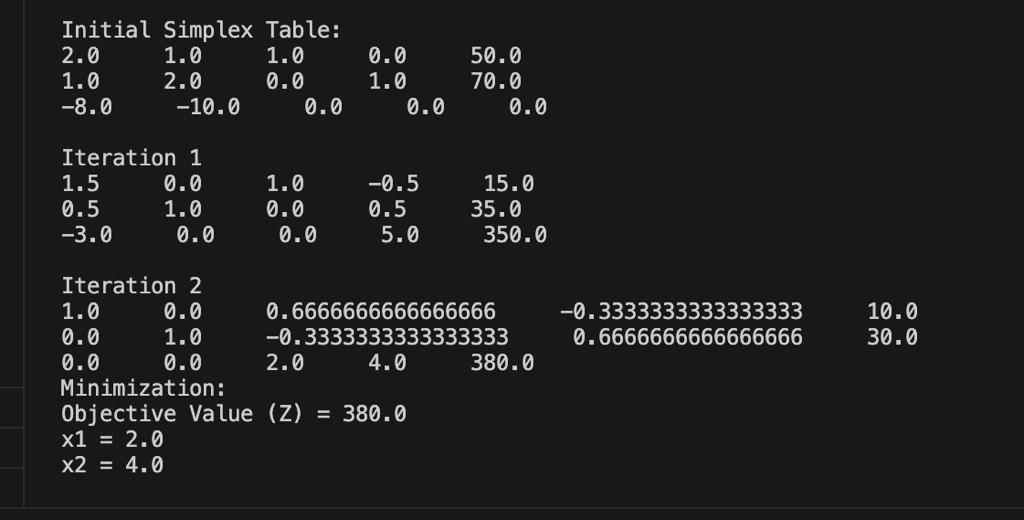
**Terminal :-**



**Minimization Problem:**



**Terminal :-**



### Conclusion and Future Work

Hence, we implemented a solution to LPP using Dynamic Programming.

Next, we will try to Integrate machine learning and AI techniques for dynamic decision support.

Focus on industry-specific adaptations and user-friendly interfaces for practical use.

**Reference**

**Internet Documentation**

**Material type In-text citation Reference list entry**

Documentation https://math.libretexts.org/

### Similar Systems [1] https://www.wolframalpha.com/

### 

[2] https://developers.google.com/optimization

**Books**

**Material type Reference list entry**

Book: Single Author Operations-Research-An-Introduction-10th-Ed.- Hamdy-A-Taha