Tutorial: Solving Linear Programming Problems (LPP) in Python

Department of Computer Science and Engineering

1 Introduction

Linear Programming Problems (LPP) are mathematical optimization models where we want to maximize or minimize a linear objective function subject to a set of linear constraints. These problems appear in resource allocation, scheduling, logistics, and many engineering applications.

Python provides libraries like PuLP that make it easy to formulate and solve LPPs. In this tutorial, we demonstrate how to model, solve, and inter pret an LPP using PuLP.

2 Problem Statement (Primal)

We want to solve the following LPP:

$$\begin{aligned} \text{Maximize } Z &= 2x_1 + 3x_2 \\ \text{subject to } x_1 + x_2 &\leq 4 \\ x_1 + 2x_2 &\leq 6 \\ x_1, x_2 &\geq 0 \end{aligned}$$

3 Python Implementation using PuLP

The following Python code solves the above problem.

```
import pulp

# 1. Define the model ( maximization problem )
model = pulp . LpProblem (" LPP_Example ", pulp . LpMaximize ) 1

```

```
# 2. Define decision variables
x1 = pulp . LpVariable ('x1 ', lowBound =0) # x1 >= 0
x2 = pulp . LpVariable ('x2 ', lowBound =0) # x2 >= 0

# 3. Define the objective function
model += 2* x1 + 3* x2 , " Cost "

# 4. Define the constraints
model += x1 + x2 <= 4 , " Constraint1 "
model += x1 + 2* x2 <= 6 , " Constraint2 "

# 5. Solve the problem
model . solve ()

# 6. Print the results
print (" Status :", pulp . LpStatus [ model . status ])
print ("x1 =", x1 . varValue )
print ("x2 =", x2 . varValue )
print (" Optimal Value of Z =", pulp . value ( model . objective ) )
```

Listing 1: Solving LPP using PuLP

4 Explanation of Code

1. Model Creation:

```
model = pulp.LpProblem("LPP_Example", pulp.LpMaximize)
```

   This creates an LP model named LPP Example with the goal of maxi mization.

2. Decision Variables:

```
x1 = pulp.LpVariable('x1', lowBound=0)
x2 = pulp.LpVariable('x2', lowBound=0)
```

   Here, x1 and x2 are non-negative real variables.

3. Objective Function:


```
model += 2*x1 + 3*x2, "Cost"
```

The += operator is used to add expressions to the model. This state ment sets the objective function: maximize $2x_1 + 3x_2$. The string "Cost" is just a label.

4. Constraints:

```
model += x1 + x2 <= 4, "Constraint1"
model += x1 + 2*x2 <= 6, "Constraint2"
```

   These represent the inequalities of the problem.

5. Solve:

```
model.solve()
```

   The model is solved using PuLP's default solver (CBC).

6. Results: After solving, we can query the status, variable values, and optimal objective value.

5 Output

If we run the program, we get:

```
Status: Optimal
x1 = 2.0
x2 = 2.0
Optimal Value of Z = 10.0
```

Thus, the optimal solution is $x_1 = 2$, $x_2 = 2$ with maximum objective value $Z = 10$.

6 Points to be noted

Here we demonstrated how to:

- Formulate an LPP mathematically.
- Model it in Python using PuLP.
- Solve and interpret the results.

With PuLP, one can extend this framework to larger problems such as trans portation, assignment, and scheduling.

7 Introduction to the PuLP model

Once an optimization model is built in PuLP, we can do far more than simply solve and print results. This handout outlines useful capabilities of a PuLP model for teaching and research.

8 Capabilities of a PuLP Model

8.1 1. Print Model Formulation

```
print ( model )
```

This prints the complete LP (objective and constraints).

8.2 2. Export Model

```
model . writeLP (" my_model .lp")
```

The model is exported in LP format, usable by external

solvers. 8.3 3. Access Variable Values

```
for var in model . variables () :
print ( var . name , "=", var . varValue )
```

8.4 4. Access Objective Value

```
print (" Objective =", pulp . value ( model . objective ) )
```

8.5 5. Constraint Slack

```
for name , c in model . constraints . items () :
print ( name , ":", c , " Slack =", c . slack )
```

8.6 6. Modify the Model Dynamically

```
model += x1 <= 10 , " NewConstraint "
model . setObjective (5* x1 + 2* x2 )
```

8.7 7. Use Different Solvers

```
model . solve ( pulp . PULP_CBC_CMD ( msg = True ) )
# model . solve ( pulp . GUROBI_CMD ())
# model . solve ( pulp . CPLEX_CMD ())
```

8.8 8. Larger Applications

PuLP supports binary, integer, and loop-generated variables, enabling mod eling of transportation, assignment, and scheduling problems.

9 Points to be noted

PuLP is not only a tool to solve LPPs but also a framework for experimen tation, analysis, and industrial-scale optimization models.

10 Solving Primal and Dual Linear Programs using PuLP: Example 2:

10.1 Example Problem 01: Primal

$$\begin{aligned}
\text{Maximize } Z &= 50x_1 + 40x_2 \\
\text{subject to } 3x_1 + 2x_2 &\leq 240 \\
x_1 + 4x_2 &\leq 200 \\
x_1, x_2 &\geq 0
\end{aligned}$$

10.2 Dual

$$\begin{aligned}
\text{Minimize } W &= 240y_1 + 200y_2 \\
\text{subject to } 3y_1 + y_2 &\geq 50 \\
2y_1 + 4y_2 &\geq 40 \\
y_1, y_2 &\geq 0
\end{aligned}$$

11 Python Implementation

11.1 Primal in PuLP

```
import pulp

# Define the model
primal = pulp . LpProblem (" Primal ", pulp . LpMaximize )

# Variables
x1 = pulp . LpVariable ('x1 ', lowBound =0)
x2 = pulp . LpVariable ('x2 ', lowBound =0)

# Objective
primal += 50* x1 + 40* x2 , " Profit "
```

```
# Constraints
primal += 3* x1 + 2* x2 <= 240 , " Constraint1 "
primal += x1 + 4* x2 <= 200 , " Constraint2 "

# Solve
primal . solve ()
print (" Primal Optimal :", pulp . value ( primal . objective ) ) 6

for v in primal . variables () :
print ( v . name , "=", v . varValue )
```

11.2 Dual in PuLP

```
# Define the model
dual = pulp . LpProblem (" Dual ", pulp . LpMinimize )

# Variables
y1 = pulp . LpVariable ('y1 ', lowBound =0)
y2 = pulp . LpVariable ('y2 ', lowBound =0)

# Objective
dual += 240* y1 + 200* y2 , " Cost "

# Constraints
dual += 3* y1 + y2 >= 50 , " Constraint1 "
dual += 2* y1 + 4* y2 >= 40 , " Constraint2 "

# Solve
dual . solve ()
print (" Dual Optimal :", pulp . value ( dual . objective ) )
for v in dual . variables () :
print ( v . name , "=", v . varValue )
```

12 Results

• Primal Optimal Solution: $x_1 = 40, x_2 = 40$, with $Z = 3600$. • Dual Optimal Solution: $y_1 = 20, y_2 = 5$, with $W = 3600$. The primal and dual objectives are equal, verifying strong duality.

Example Problem 2: (Primal)

A manufacturer produces two products x_1 and x_2 with profits 50 and 40 (per unit). Production consumes two limited resources:

7

$$\text{Resource 1: } 3x_1 + 2x_2 \leq 240$$

$$\text{Resource 2: } x_1 + 4x_2 \leq 200$$

The primal maximization problem is:

$$
\begin{aligned}
(P)\ \max\ Z = 50x_1 + 40x_2 & \\
\text{s.t. } 3x_1 + 2x_2 &\leq 240 \text{ (resource 1)} \\
x_1 + 4x_2 &\leq 200 \text{ (resource 2)} \\
x_1, x_2 &\geq 0
\end{aligned}
$$

Deriving the Dual

Since primal is a maximization with ≤-constraints, the dual is a minimiza tion. Dual variables $y_1, y_2 \geq 0$ correspond to the two resource constraints.

$$
\begin{aligned}
(D)\ \min\ W = 240y_1 + 200y_2 & \\
\text{s.t. } 3y_1 + y_2 &\geq 50 \text{ (for } x_1) \\
2y_1 + 4y_2 &\geq 40 \text{ (for } x_2) \\
y_1, y_2 &\geq 0
\end{aligned}
$$

Analytic Solution

Primal

Solve by intersection of constraints:

$$
\begin{cases}
3x_1 + 2x_2 = 240 \\
x_1 + 4x_2 = 200
\end{cases}
$$

Solution: $x_1 = 56$, $x_2 = 36$. Objective: $Z = 50 \cdot 56 + 40 \cdot 36 = 4240$.

Dual

Solve equalities:

$$\begin{cases} 3y_1 + y_2 = 50 \\ 2y_1 + 4y_2 = 40 \end{cases}$$

8

Solution: $y_1 = 16,\ y_2 = 2$. Objective: $W = 240 \cdot 16 + 200 \cdot 2 = 4240$.
Strong duality holds: $Z = W = 4240$.

Interpretation

- $y_1 = 16$: shadow price of resource 1 (per unit increase in RHS of first constraint increases profit by 16).
- $y_2 = 2$: shadow price of resource 2 (less valuable resource).
- Both primal constraints bind (slack = 0). Both dual constraints bind as well.

Python Implementation (PuLP)

```
import pulp

# ----------------------
# Solve the Primal (P)
# ----------------------
primal = pulp . LpProblem (" Primal_Max_Profit ", pulp . LpMaximize )

# Variables
x1 = pulp . LpVariable ('x1 ', lowBound =0)
x2 = pulp . LpVariable ('x2 ', lowBound =0)

# Objective
primal += 50* x1 + 40* x2 , " Profit "

# Constraints
primal += 3* x1 + 2* x2 <= 240 , " Resource1 "
```

```
primal += x1 + 4* x2 <= 200 , " Resource2 "

# Solve
primal . solve ()
print ("x1 =", pulp . value ( x1 ) , ", x2 =", pulp . value ( x2 ) )
print (" Optimal Z =", pulp . value ( primal . objective ) )
```



```

# ----------------------
# Solve the Dual (D)
# ----------------------
dual = pulp . LpProblem (" Dual_Min_Cost ", pulp . LpMinimize )

y1 = pulp . LpVariable ('y1 ', lowBound =0)
y2 = pulp . LpVariable ('y2 ', lowBound =0)

dual += 240* y1 + 200* y2 , " DualCost "
dual += 3* y1 + 1* y2 >= 50 , " DualConstr_x1 "
dual += 2* y1 + 4* y2 >= 40 , " DualConstr_x2 "

dual . solve ()
print ("y1 =", pulp . value ( y1 ) , ", y2 =", pulp . value ( y2 ) )
print (" Optimal W =", pulp . value ( dual . objective ) )

# ----------------------
# Sensitivity check
# ----------------------
primal . constraints [" Resource1 "]. changeRHS (241)
primal . solve ()
print (" Z_new =", pulp . value ( primal . objective ) )
```

Explanation & Teaching Points

Distinct Shadow Prices

The dual solution $y_1 = 16,\ y_2 = 2$ shows that resource 1 is much more valuable than resource 2. Why? Because producing uses the first resource more intensively for profitable items: the coefficients and profit structure make resource 1 a bottleneck.

Complementary Slackness and Binding Constraints

Both primal constraints have slack 0 (both are tight), and both dual con straints are tight (slack 0). This is consistent with the complementary slackness theorem, given that both x and y variables are positive.

Sensitivity Interpretation (Shadow Price = Marginal Value)

When you increase the RHS of resource 1 by 1 (from 240 → 241), the optimal objective increases by exactly 16 (the dual variable y_1). The code confirms this: objective goes from 4240 → 4256 (increase of 16). This provides a con crete way for students to appreciate that dual variables represent marginal values.

Economic / Physical Meaning

If resource 1 is, for example, machine hours or a scarce chemical, the dual tells you how much extra profit you could earn by acquiring one more unit of that resource. A high y_1 indicates you would gladly pay up to 16 per unit to obtain more of resource 1.

Why Solve the Dual Explicitly?

- It gives the shadow prices directly.
- Some large problems or decomposition methods are easier to solve in the dual.
- Dual solutions guide column generation and cutting plane strategies.

Classroom Activities

1. Exercise 1: Change the profit coefficients (e.g., make x_1 less prof itable) and observe how y_1, y_2 change. Discuss why.
2. Exercise 2: Fix resource 2 RHS to a smaller value and observe when y_2 increases — show which resource becomes the new bottleneck.
3. Exercise 3: Convert the primal to standard form and derive the dual manually, then validate with PuLP.

4. Exercise 4 (Advanced): Solve the primal with integer constraints (e.g., x_1, x_2 integer) and discuss how duality breaks for ILPs (no exact dual with same properties). This motivates the use of LP relaxation.

Important Points

1. Duality Insight: Distinct shadow prices indicate relative scarcity of resources.
2. Complementary Slackness: Tight constraints correspond to posi tive dual variables.
3. Sensitivity: Increasing RHS of Resource 1 by 1 increases profit by 16 (confirms dual price).
4. Economic Meaning: Dual variables give the marginal value of extra resources.

Case Study Assignments – Linear Programming & Duality

Instructions for Students

1. For each case study, carefully read the problem statement and formulate the Linear Programming Problem (LPP).
2. Where applicable, construct the dual problem. Clearly state both the primal and dual formulations.
3. Solve the problem using an appropriate solver (Python PuLP, MAT LAB, R, or any other tool).
4. Analyze the shadow prices (dual variables) and interpret them in the given context. Discuss their managerial or policy implications.
5. Provide both numerical results and qualitative interpretations.

6. Submit a typed report (PDF) including mathematical formulation, solver output, and analysis. Handwritten solutions will not be ac cepted.

Problem 1: Diet Optimization for Athletes

RGIPT is collaborating with the Sports Authority of India to design cost effective diet plans for young athletes in a training camp. Each athlete must consume sufficient macronutrients and vitamins daily, and the camp seeks to minimize cost while ensuring nutrition.

Foods available (per serving):

• Chicken (100g): 25g protein, 5g carbs, 2g vitamins, Cost

50 • Milk (200ml): 10g protein, 12g carbs, 4g vitamins, Cost

30 • Rice (150g): 4g protein, 40g carbs, 1g vitamins, Cost

20 • Fruits (200g): 2g protein, 20g carbs, 6g vitamins, Cost

25

Daily Requirement: At least 100g protein, 200g carbs, and 20g vita mins.

Analysis Questions:

• What is the minimum cost per athlete per day?

• Which food items are part of the optimal diet? Are any items redun dant?

• What is the shadow price of protein? How does it affect the daily cost if protein requirement increases by 10g?

Problem 2: Workforce Allocation in a Refinery

A refinery at RGIPT operates in three shifts per day. Each shift requires a minimum number of workers, and wages differ by shift due to allowances. The refinery has a pool of permanent staff but may also hire additional contract workers.

Shift requirements and costs:

- Morning shift: At least 6 workers, Cost 500/worker
- Afternoon shift: At least 5 workers, Cost 450/worker

- Night shift: At least 4 workers, Cost 600/worker

The refinery has 12 permanent workers available. Any additional require ment must be met using contract workers at 700/worker. Analysis Questions:

- What is the minimum daily wage expenditure?
- If one more permanent worker is added, how much cost saving is pos sible (using dual solution)?
- Which shift has the highest shadow price, and what does it imply?

Problem 3: Energy Distribution in Solar Grid Network

RGIPT is piloting a solar power distribution project with two solar plants (P1 and P2) supplying electricity to four demand centers (A, B, C, D). Trans mission costs vary depending on distance and infrastructure. Data:

- Supply: P1 = 200 units, P2 = 180 units
- Demand: A = 120, B = 100, C = 80, D = 60
- Transmission costs (/unit):
  - P1 → A = 2, B = 3, C = 4, D = 5
  - P2 → A = 3, B = 1, C = 2, D = 4

Analysis Questions:

- What is the minimum transmission cost?
- Which demand center has the highest implicit value (shadow price)?
- If Plant P1's supply is increased by 20 units, how does the total cost change?

Problem 4: Airline Crew Scheduling

An airline operating from Delhi must assign crew to 3 flights in a day. Each flight has a minimum crew requirement. Two categories of crew are available: Type A (senior crew, can work on any flight) and Type B (junior crew, restricted to certain flights).

Data:

- Flight 1 requires at least 5 crew
- Flight 2 requires at least 6 crew
- Flight 3 requires at least 4 crew
- Type A: Available = 10, Cost = 12,000/crew
- Type B: Available = 8, Cost = 9,000/crew (can only work on Flights 2 and 3)

Analysis Questions:

- What is the minimum crew cost for the day?
- What is the shadow price for Type A crew availability?
- If Flight 2's requirement increases by 1 crew, how much will the cost rise?

Problem 5: Emergency Fuel Allocation (Ad

vanced)

During a fuel crisis, RGIPT has been asked by the state government to allocate limited fuel stock to different cities in Uttar Pradesh to maximize profit while ensuring fairness in distribution.

Data:

- Total available fuel stock at RGIPT depot = 500 units
- City X: Demand = 200 units, Profit = 40/unit, Pipeline capacity = 180

- City Y: Demand = 220 units, Profit = 50/unit, Pipeline capacity = 200
- City Z: Demand = 180 units, Profit = 60/unit, Pipeline capacity = 150

Analysis Questions:

- What is the maximum achievable profit?
- What is the shadow price of fuel stock at the depot? Interpret its meaning for policymakers.
- If the government increases depot supply by 50 units, how much addi tional profit can be realized?
- Which pipeline capacity is the most binding, and what does its shadow price imply?