**Prashikshan – 2020**

Project Report

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Reference No. | Name | Discipline | Institute | Dates | Duration | Department |  |
| VT20200408 | Rahul Vishal Mishra | Computer Science | MIT Manipal | 13.04.20 - 11.05.20 | 4 Weeks | Information Technology Services (ITS) |  |

Project Title - Optimization Modelling using Python

Project Guide – Mr. Uttam Mitra [Sr. Manager IT, Information Technology Services (ITS)] (Contact No. 8092189564)

**Acknowledgement:**

First and foremost, I would like to express my gratitude to the company, Tata Steel, for providing such a great opportunity to further my learning and experience, that too during this pandemic period.

I would also like to thank my project guide Mr. Uttam Mitra. His constant support, guidance and keen interest into my work helped me to enhance my understanding of Optimization Modelling.

My gratitude extends to the department of Information Technology Services (ITS), Tata Steel providing me the opportunity to work on my project.

Last but not the least, I would like to thank my family for their constant motivation and support.

Thank You,

Rahul Vishal Mishra.

**Index:**

|  |  |
| --- | --- |
| **Topic** | **Page Number** |
| Initial Problem Statement | 4 |
| Introduction to Optimization Modelling | 5 – 7 |
| Introduction to Idle Freight Problem & related details | 8 – 11 |
| Specific model - mathematical specification for code demonstration | 12 |
| Code Demonstration | 13 – 22 |
| Case Results | 23 |
| Conclusion | 24 |

**Initial Problem Statement**

To **optimize** a given solution to a business problem, in this case, “**The Idle Freight Optimization in Indian Railway Wagons**” and present possible solutions / improvements / hypotheses with respect to different scenarios or constraints **using Python3**.

**Introduction**

### **What is Optimization Modelling ?**

When you’re trying to make tough decisions about questions that involve an inordinate number of factors, Optimization helps you to capture key components to build a mathematical model of the business situation, giving you the confidence to make better decisions more quickly.

An optimization model is a translation of the key characteristics of the business problem you are trying to solve. The model consists of three elements: the Objective Function, Decision Variables and Business Constraints.

* **Components of an Optimization Model –**

An optimization model has three main components:

1. An objective function. This is the function that needs to be optimized.
2. A collection of decision variables. The solution to the optimization problem is the set of values of the decision variables, for which the objective function reaches its optimal value.
3. A collection of constraints that restrict the values of the decision variables.

* **Types of Optimization Models –**

Optimization problems can be classified in terms of the nature of the objective function and the nature of the constraints. From this point of view, there are four types of optimization problems, of increasing complexity.

An *Unconstrained optimization problem* is an optimization problem where the objective function can be of any kind (linear or nonlinear) and there are no constraints.

A *linear program* is an optimization problem with an objective function that is linear in the variables, and all constraints are also linear.

A *quadratic program* is an optimization problem with an objective function that is quadratic in the variables (i.e. it may contain squares and cross products of the decision variables), and all constraints are linear. A quadratic program with no squares or cross products in the objective function is a linear program.

A *nonlinear program* is an optimization problem with an objective function that is an arbitrary nonlinear function of the decision variables, and the constraints can be linear or nonlinear.

* **Decision Variables –**

Finding the optimal values of the decision variables is the goal of solving an optimization model.

* **Constraints –**

Constraints limit the possible values for the decision variables in an optimization model. There are two types of constraints: linear and nonlinear.

Linear constraints express that a linear combination of the decision variables must lie within a certain range.

Nonlinear constraints express that the value of some arbitrary function of the decision variables must lie within a certain range.

* **Linear Programming –**

The goal of linear programming is to optimize a linear function subject to linear constraints. In its most general form, a *linear program* is an optimization problem of the canonical form:

Minimize/Maximize c1x1 + c2x2 + · · · + cnxn = z

Subject to a11x1 + a12x2 + · · · + a1nxn <>= b1

a21x1 + a22x2 + · · · + a2nxn <>= b2

. . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . .

am1x1 + am2x2 + · · · + amnxn <>= bm

x1, x2, . . . , xn ≥ 0.

(P.S. <>= implies greater than / less than / equal to)

**Idle Freight Optimization Problem**

**Business context –**

Tata Steel is a Steel Manufacturing company, which produces Steel in the form of both Flat Products [Steel Slabs, Hot Rolled Coils (HR), Cold Rolled Coils (CR), Sheets etc.] and Long Products (Steel Billets, TMT Bars, Wire Rods etc.). Flat products accounts for ~70% of the total mix. These Steel products are sent to customers using both Rail and Road Transportation, where Rail transportation accounts for ~60% of dispatches. Railway wagons provided by the Indian Railways for steel transportation generally comes in three sizes, 63 Tons, 66 Tons and 69 Tons. In case of Flat Products, the coil sizes generally range from 20 tons to 24 tons. As such, the best combination of wagon and coils has to be used, in order to minimize the Idle Freight.

**Basic Terms & Details –**

* Charge Weight -> The fixed weight capacity per wagon for which Indian Railways charges a fixed fare.
* Railway Receipt Weight -> The actual weight of steel coils / bars (or any other material) loaded in the wagon.
* Idle freight -> The mathematical difference between the values of “Charge Weight” & “Railway Receipt Weight”. (Charge Weight – Railway Receipt Weight).

**Goal –**

Minimise (Idle Freight) depending on the assumed constraints using Python3.

**Need for this –**  The minimization of Idle Freight is a well thought out step towards minimizing unnecessary expenditure, reducing logistical overhead, and improving overall efficiency.

**A brief description of the Tools used for solving this problem –**

**Python** is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

**PuLP** is an LP modeler written in Python. PuLP can generate MPS or LP files and call GLPK, COIN-OR CLP/CBC, CPLEX, GUROBI, MOSEK, XPRESS, CHOCO, MIPCL, SCIP to solve linear problems.

## Installation

The easiest way to install pulp is via PyPi

If pip is available on your system:

pip install pulp

**Assumptions, Constraints & Cases –**

1. Coil weight range = [20 - 24] tons

2. Wagon charge weights (fixed) = 63 tons, 66 tons, 69 tons

Number of wagons available = infinite

**Assumptions for "Idle freight Problem", "Idle Freight subcase 2", "Idle freight subcase 3" -**

1. Number of 20 tonne coils available = 10

Number of 21 tonne coils available = 9

Number of 22 tonne coils available = 8

Number of 23 tonne coils available = 7

Number of 24 tonne coils available = 6

**Assumptions for "Idle freight subcase 4", "Idle freight subcase 4\_2",**

**"Idle freight subcase 4\_3" -**

1. Number of 20 tonne coils available = 6

Number of 21 tonne coils available = 7

Number of 22 tonne coils available = 8

Number of 23 tonne coils available = 9

Number of 24 tonne coils available = 10

63 tonne wagon case ---> "Idle freight Problem", "Idle freight subcase 4"

66 tonne wagon case ---> "Idle freight subcase 3", "Idle freight subcase 4\_3"

69 tonne wagon case ---> "Idle freight subcase 2**", "Idle freight subcase 4\_2”** (code demonstrated from the next page).

**Formal Linear Programming Mathematical Model for “Idle freight subcase 4\_2” problem -**

**Maximize** - **20X1 + 21X2 + 22X3 + 23X4 + 24X5**

**Subject to** - 0 <= X1 <=6

0 <= X2 <=7

0 <= X3 <=8

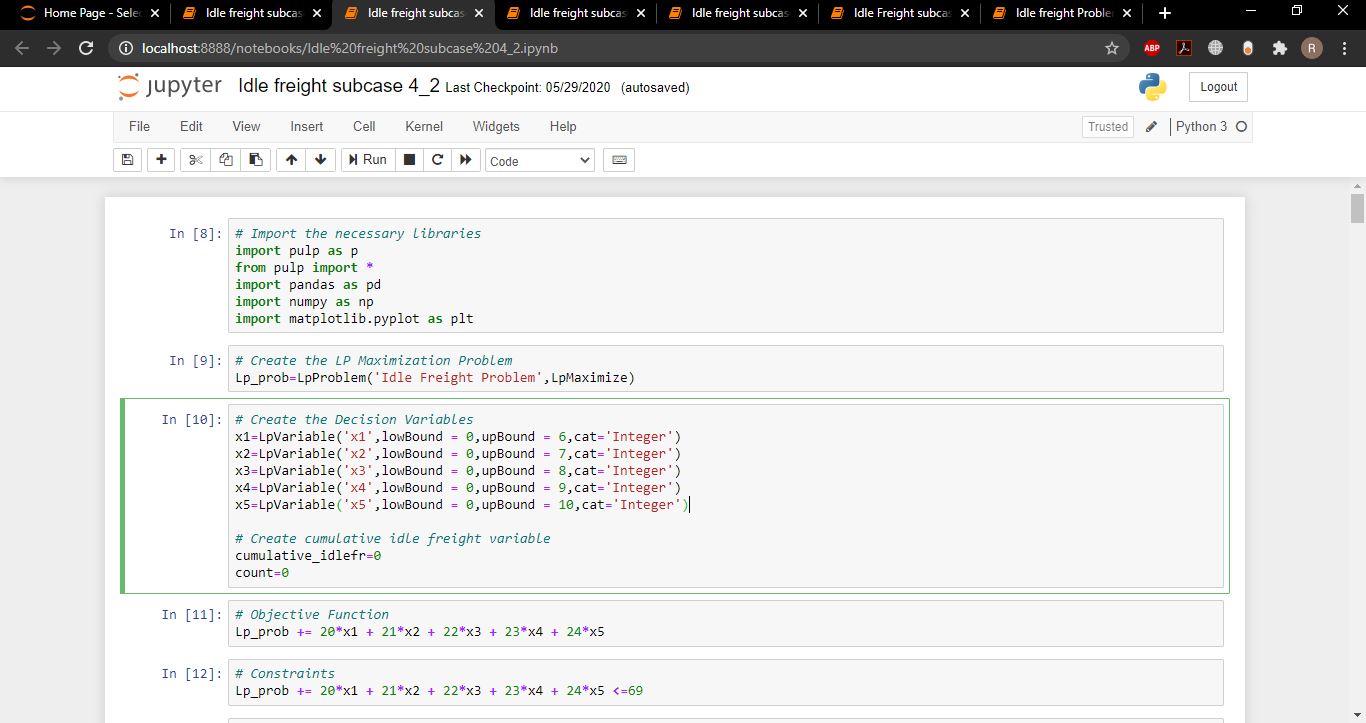
0 <= X4 <=9

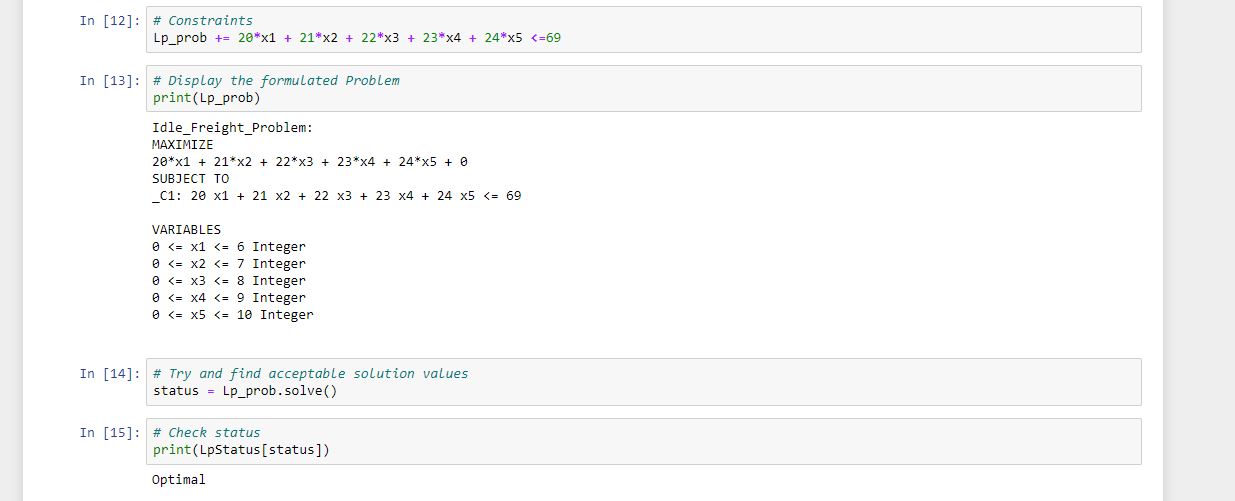
0 <= X5 <=10

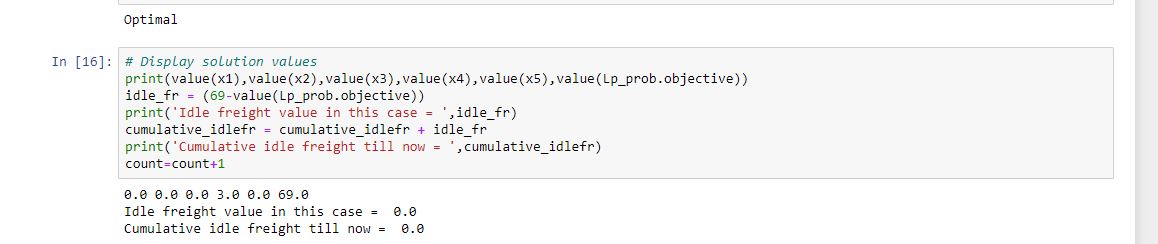
20X1 + 21X2 + 22X3 + 23X4 + 24X5 <= 69

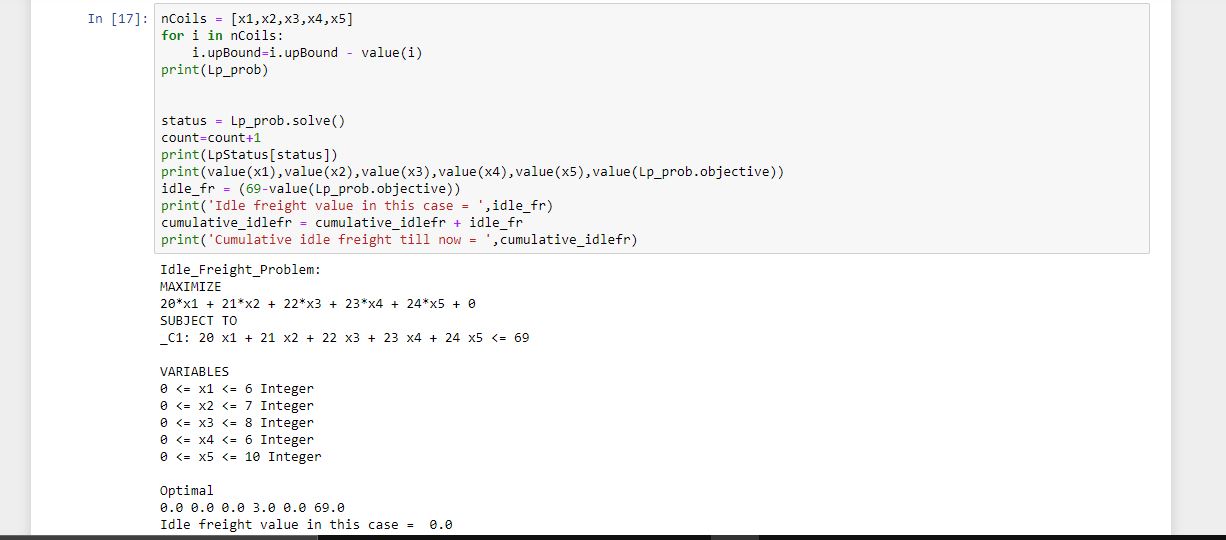
**Code Demonstration –**

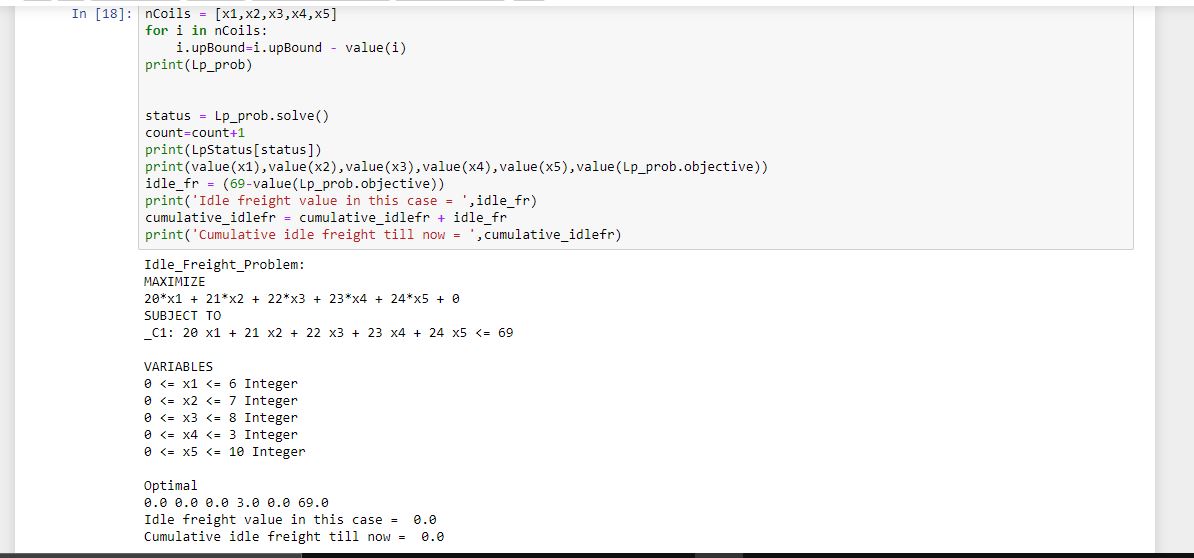
Attached are the screenshots from the code demonstration of **“Idle Freight Subcase 4\_2”.**

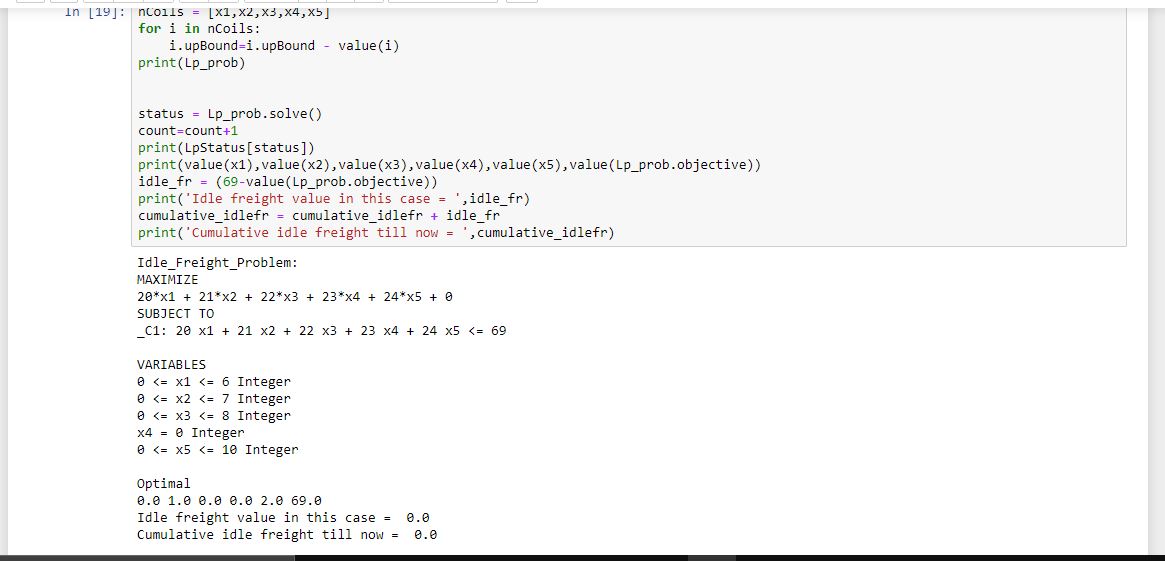


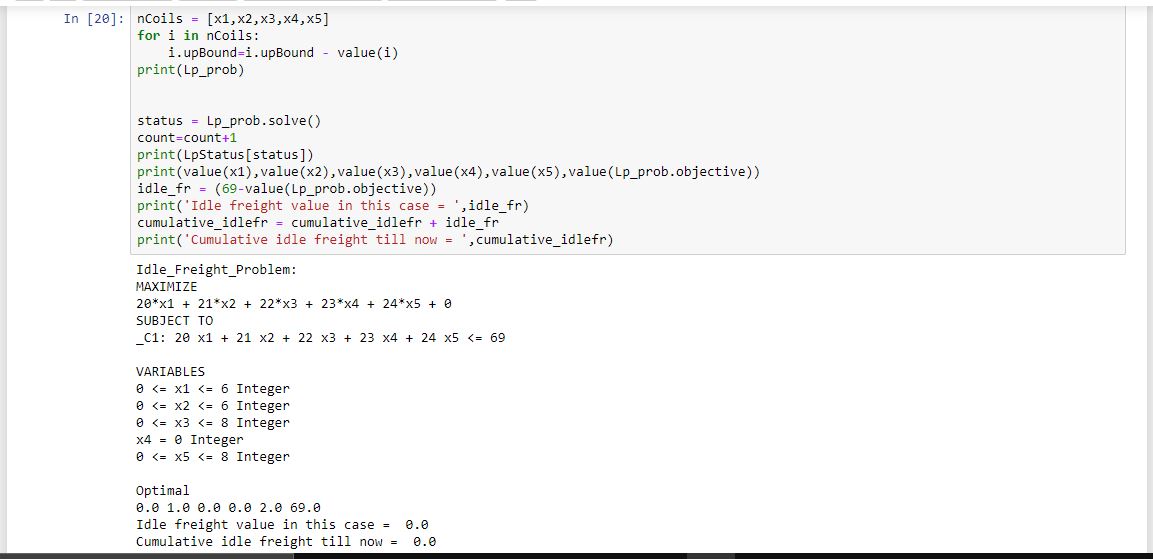


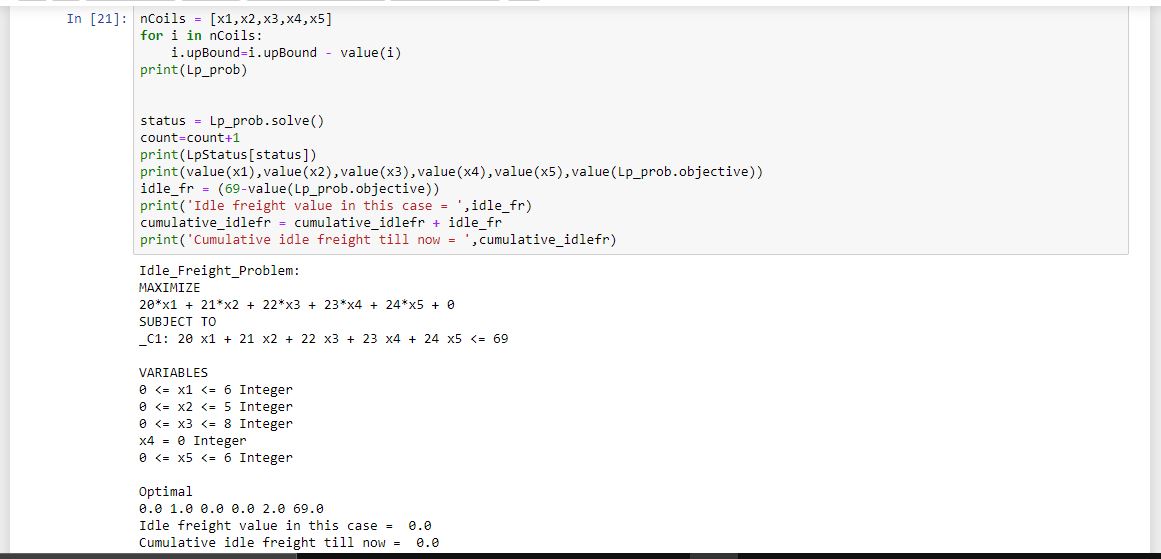


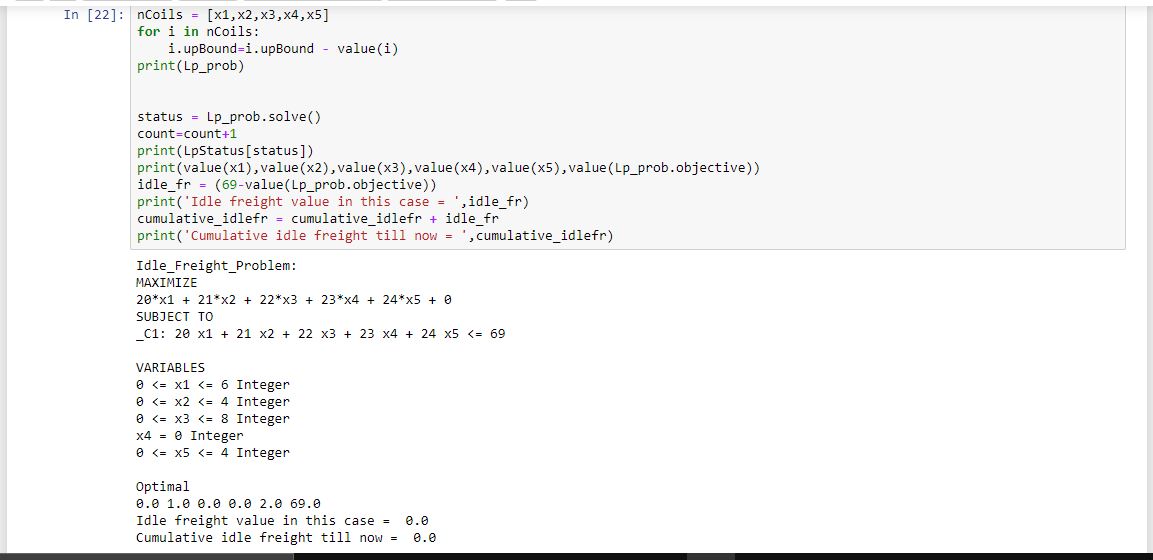


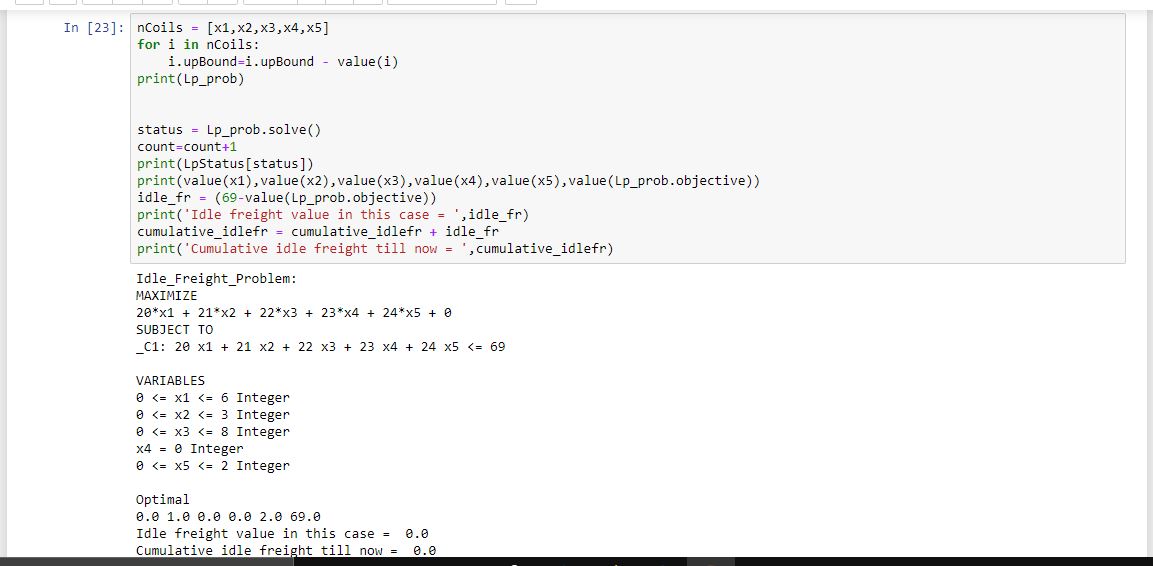


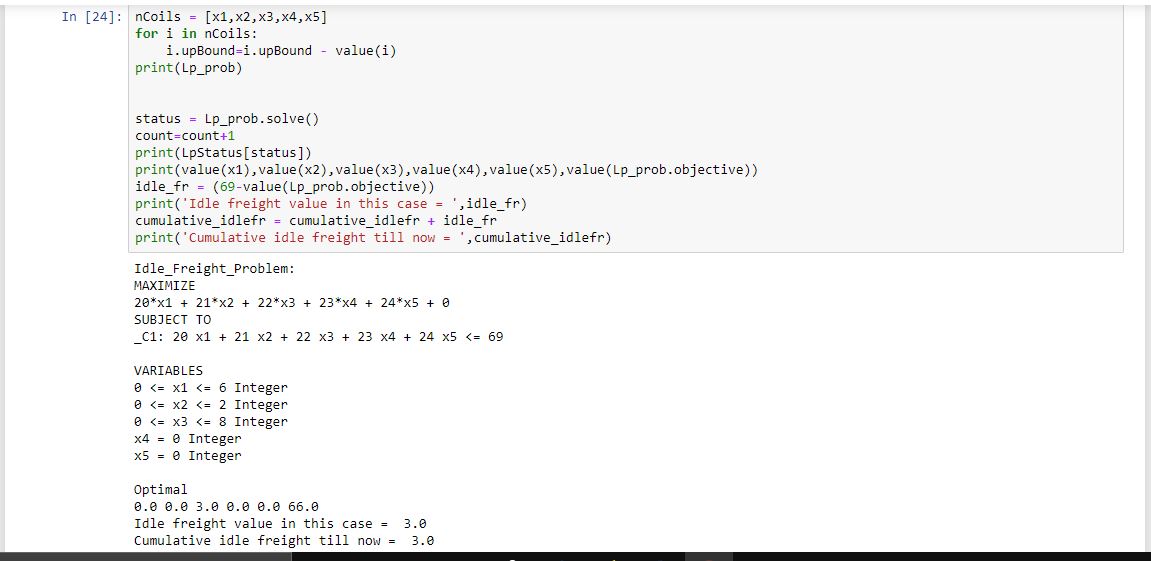


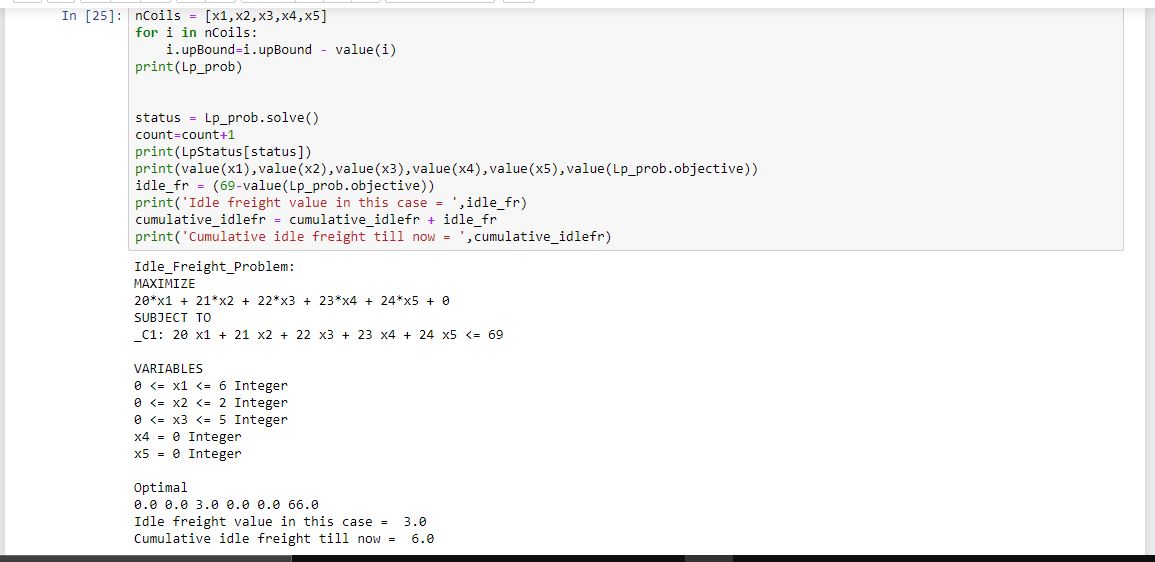


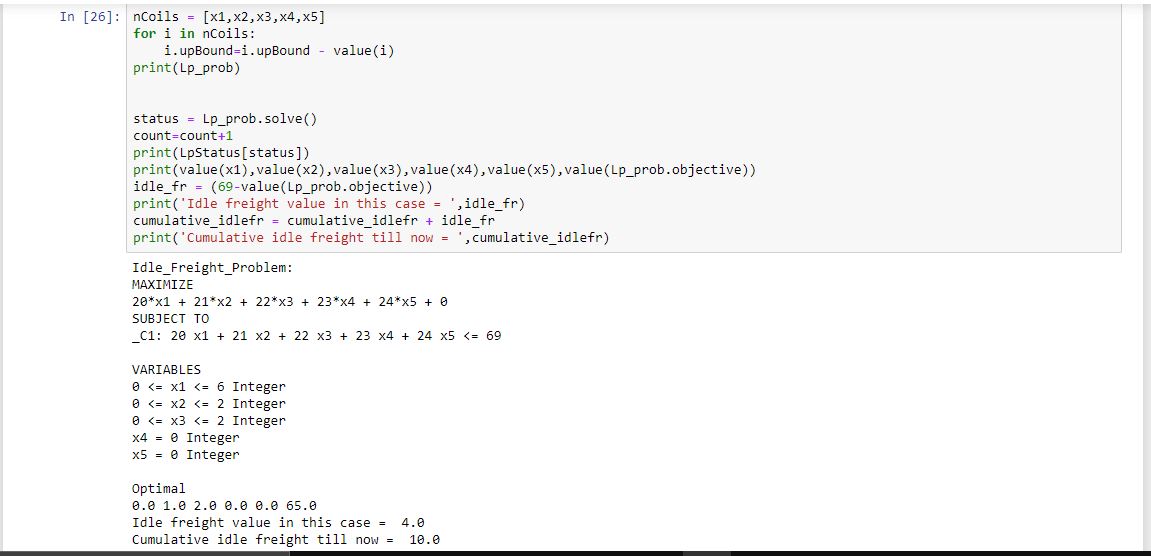


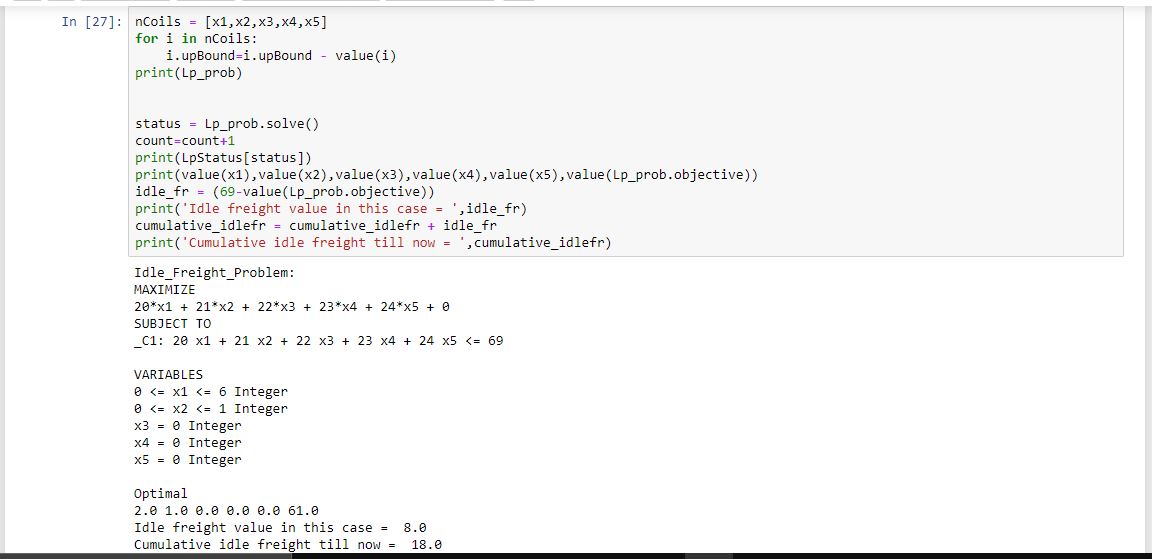


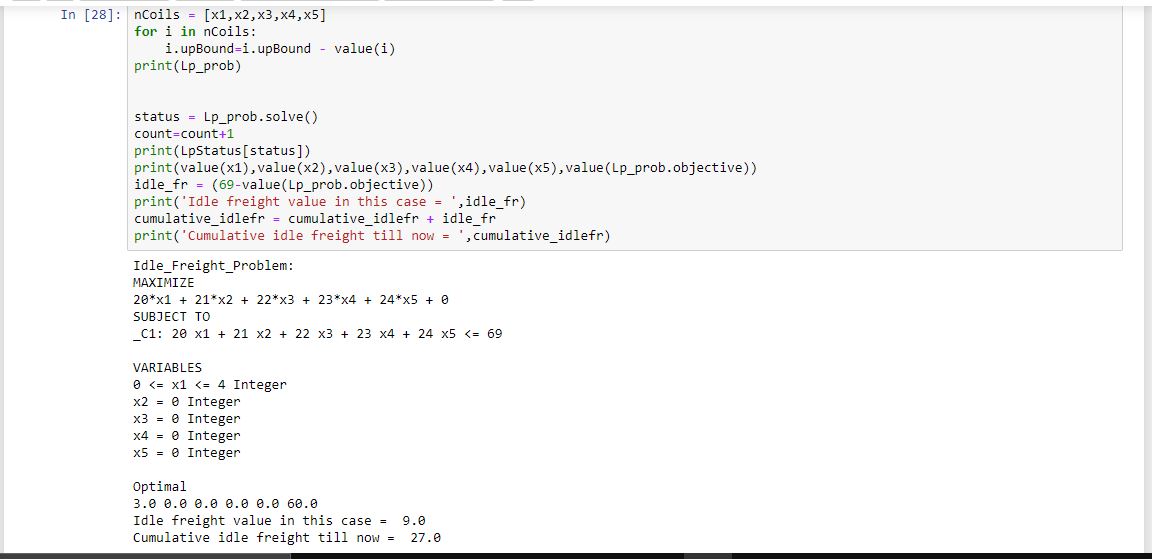


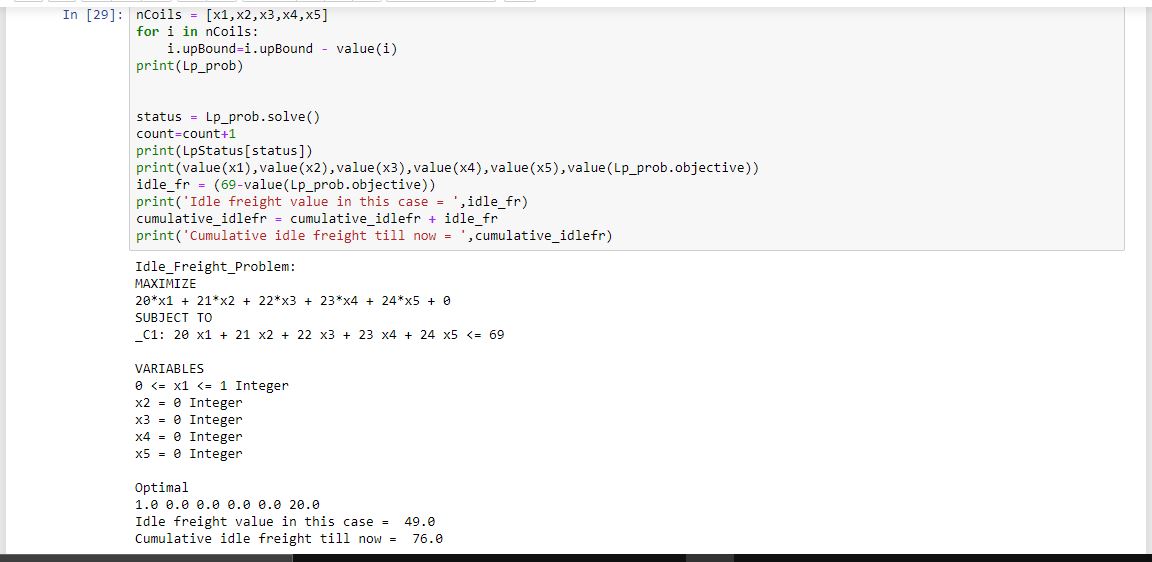


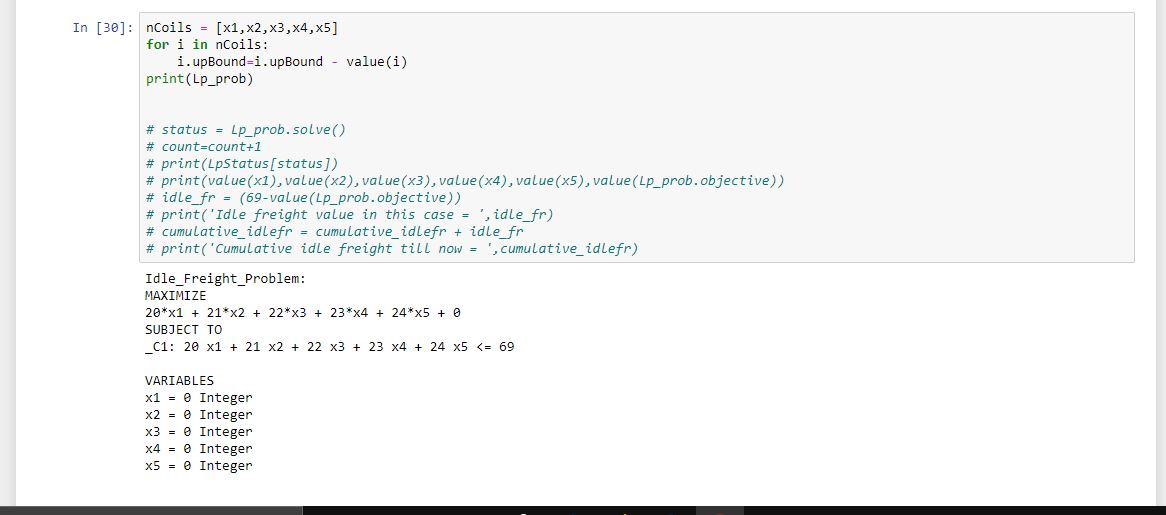


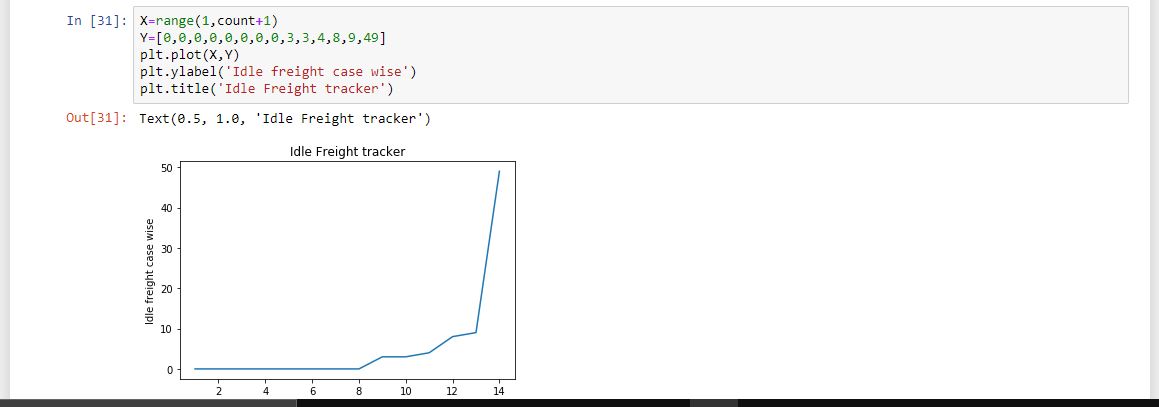


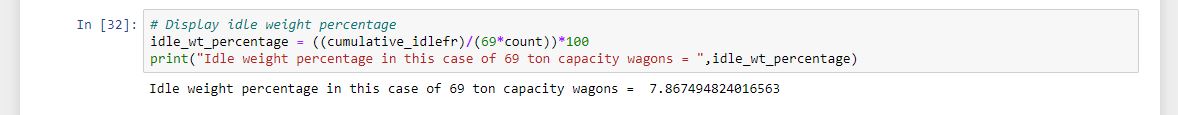












**Different Case Results (refer “Assumptions”):**

Idle weight percentage for “Idle Freight Subcase 4\_2” =7.867%

Idle weight percentage for “Idle Freight Problem” =13.690%

Idle weight percentage for “Idle Freight Subcase 2” =9.937%

Idle weight percentage for “Idle Freight Subcase 3” =9.740%

Idle weight percentage for “Idle Freight Subcase 4” =21.516%

Idle weight percentage for “Idle Freight Subcase 4\_3” =10.101%

Hence, we can infer from the above case results that a wagon friendly mix of coil sizes fit better, and are much more effective in reducing idle weight percentage than random non wagon friendly coil sizes. A careful analysis must be made before deciding on the loading procedure for a given wagon.

**Conclusion:**

This project provided me with an insight into technical areas like Optimization Modelling and Interactive Python programming. Apart from this, my project also helped me get a glimpse into the inner workings of a reputed MNC like Tata Steel.

This project has helped me gain a lot of knowledge/skills in technical areas as well as topics like project/time management.

Optimization & Efficiency are everyday necessities in the current world of business. This was my first hands-on experience in this area, and I look forward to getting opportunities of working on such projects in the future.

Thank you,

Rahul Vishal Mishra.

**References:**

Most of the research done for this project have been done through the following sources: -

* [https://towardsdatascience.com](https://towardsdatascience.com/)
* <https://pypi.org/project/PuLP/>
* <https://www.geeksforgeeks.org/python-linear-programming-in-pulp/>
* <https://benalexkeen.com/linear-programming-with-python-and-pulp/>
* Introduction to Operations Research – Frederick S Hillier and Gerald J. Lieberman.