

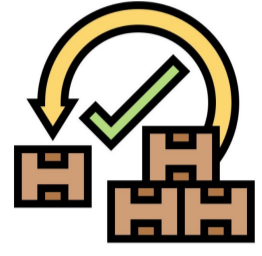
## PROBLEM

PIL's current cargo allocation process is manually calculated by their trade managers. It is **time consuming, error prone and non-optimal** which affects the company's profit.

## OBJECTIVE

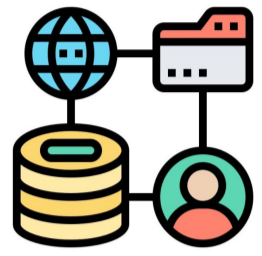
To create a program to **recommend an optimal allocation** for 20'/40' cargo containers to yield the **highest contribution margin** for a specific described shipping line.

## METHODOLOGY



### Requirements

Understand constraints and pain points faced by the trade managers



### System Analysis

Study the specifications and conceptualise suitable optimisation methods



### Design

Formulate mathematical algorithm and test using Excel



### Implementation

Produce the actual code required for the model using Python



### Testing

Perform stress test and gather feedbacks for improvement



### Integration

Full integration of Python into PIL's system for all users

## PROTOTYPE

**STEP 1:**  
Decided to test on West Coast Central and America 2

**STEP 2:**  
Formulated objective function and constraints

In WS2, the mathematical formulation is shown below:

$$\text{Max } \sum_{i \in I} \sum_{j \in J} C_{ij} x_{ij}$$

s.t.

**Physical Constraints:**

$$\sum_{i \in I} \sum_{j \in J} x_{ij} \leq U$$

$$\sum_{i \in I} \sum_{j \in J} x_{ij} \leq Z$$

$$\sum_{i \in I} \sum_{j \in J} x_{ij} \leq 311 \text{ Moves}$$

$$\sum_{i \in I} \sum_{j \in J} x_{ij} \leq 11224 \text{ Tons}$$

$$\sum_{i \in I} \sum_{j \in J} x_{ij} \leq 5946 \text{ Tons}$$

**Business Constraints:**

$$x_{ij} \leq x_{ij}^u \quad \forall i \in I, j \in J, k \in E$$

$$\sum_{i \in I} \sum_{j \in J} x_{ij} + 2 \sum_{i \in I} x_{ij} \geq T_j \quad \forall i \in I, j \in J$$

$$\sum_{i \in I} \sum_{j \in J} x_{ij} + 2 \sum_{i \in I} x_{ij} \geq T_j \quad \forall j \in J, D \in \{2,3\}$$

$$x_{ij} \geq 0 \text{ and integer} \quad \forall i \in I, j \in J, k \in E$$

Maximise the objective function: **Sum of total contribution margin** from each container in the optimum allocation.

Objective function subject to **physical constraints** such as **draft restrictions, weight limit and capacity limit.**

Objective function subject to **business constraints** such as **minimum and maximum no. of containers** from certain ports.

**STEP 3:**  
Created a **prototype model** for testing

The spreadsheet shows an objective function cell at the top right, a grid of constraints for various routes and vessels, and a 'Cargo Allocation' section at the bottom right with columns for different vessel types and their respective container counts.

**STEP 4:**  
Compared results from model with historical allocation of WS2,

<b>+\$110,348.50</b> Increase in Profit	<b>\$736,448.25</b> Profit from Model	<b>16%</b> Percentage of Change in Profit
--	--	--



It is one of the **most complex** shipping routes



It has one of the **most port-pairs**



It has **move count and draft restriction** as constraints

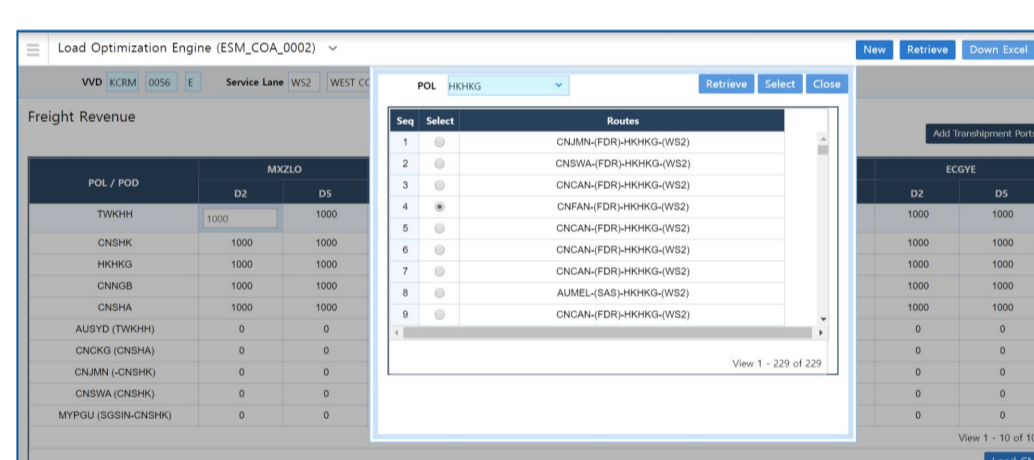
## IMPLEMENTATION

```
max_constr_02_list = [FormulationPILContainerPort_02_MBL, ...]
max_constr_04_list = [FormulationPILContainerPort_04_MBL, ...]
max_constr_06_list = [FormulationPILContainerPort_06_MBL, ...]
max_constr_08_list = [FormulationPILContainerPort_08_MBL, ...]
max_constr_10_list = [FormulationPILContainerPort_10_MBL, ...]
max_constr_12_list = [FormulationPILContainerPort_12_MBL, ...]
max_constr_14_list = [FormulationPILContainerPort_14_MBL, ...]
max_constr_16_list = [FormulationPILContainerPort_16_MBL, ...]
max_constr_18_list = [FormulationPILContainerPort_18_MBL, ...]
max_constr_20_list = [FormulationPILContainerPort_20_MBL, ...]
max_constr_22_list = [FormulationPILContainerPort_22_MBL, ...]
max_constr_24_list = [FormulationPILContainerPort_24_MBL, ...]
max_constr_26_list = [FormulationPILContainerPort_26_MBL, ...]
max_constr_28_list = [FormulationPILContainerPort_28_MBL, ...]
max_constr_30_list = [FormulationPILContainerPort_30_MBL, ...]
max_constr_32_list = [FormulationPILContainerPort_32_MBL, ...]
max_constr_34_list = [FormulationPILContainerPort_34_MBL, ...]
max_constr_36_list = [FormulationPILContainerPort_36_MBL, ...]
max_constr_38_list = [FormulationPILContainerPort_38_MBL, ...]
max_constr_40_list = [FormulationPILContainerPort_40_MBL, ...]
max_constr_42_list = [FormulationPILContainerPort_42_MBL, ...]
max_constr_44_list = [FormulationPILContainerPort_44_MBL, ...]
max_constr_46_list = [FormulationPILContainerPort_46_MBL, ...]
max_constr_48_list = [FormulationPILContainerPort_48_MBL, ...]
max_constr_50_list = [FormulationPILContainerPort_50_MBL, ...]
max_constr_52_list = [FormulationPILContainerPort_52_MBL, ...]
max_constr_54_list = [FormulationPILContainerPort_54_MBL, ...]
max_constr_56_list = [FormulationPILContainerPort_56_MBL, ...]
max_constr_58_list = [FormulationPILContainerPort_58_MBL, ...]
max_constr_60_list = [FormulationPILContainerPort_60_MBL, ...]
max_constr_62_list = [FormulationPILContainerPort_62_MBL, ...]
max_constr_64_list = [FormulationPILContainerPort_64_MBL, ...]
max_constr_66_list = [FormulationPILContainerPort_66_MBL, ...]
max_constr_68_list = [FormulationPILContainerPort_68_MBL, ...]
max_constr_70_list = [FormulationPILContainerPort_70_MBL, ...]
max_constr_72_list = [FormulationPILContainerPort_72_MBL, ...]
max_constr_74_list = [FormulationPILContainerPort_74_MBL, ...]
max_constr_76_list = [FormulationPILContainerPort_76_MBL, ...]
max_constr_78_list = [FormulationPILContainerPort_78_MBL, ...]
max_constr_80_list = [FormulationPILContainerPort_80_MBL, ...]
max_constr_82_list = [FormulationPILContainerPort_82_MBL, ...]
max_constr_84_list = [FormulationPILContainerPort_84_MBL, ...]
max_constr_86_list = [FormulationPILContainerPort_86_MBL, ...]
max_constr_88_list = [FormulationPILContainerPort_88_MBL, ...]
max_constr_90_list = [FormulationPILContainerPort_90_MBL, ...]
max_constr_92_list = [FormulationPILContainerPort_92_MBL, ...]
max_constr_94_list = [FormulationPILContainerPort_94_MBL, ...]
max_constr_96_list = [FormulationPILContainerPort_96_MBL, ...]
max_constr_98_list = [FormulationPILContainerPort_98_MBL, ...]
max_constr_100_list = [FormulationPILContainerPort_100_MBL, ...]
model = pulp.LpProblem('Load Liner Optimization', pulp.LpMaximize)
```

### Coding the Application

PuLP package is used to **formulate and solve the integer program in Python**. PuLP uses the **Coin-or Branch and Cut (CBC) solver**, which uses a branch-bound and cut method to solve IP problems.

The frontend interface, hosted on PIL's LMS platform, and the optimisation engine communicate via **JSON outputs**.



### Designing User Interface

After evaluating requirements from end users, the team designed an interface that is simple and user friendly. It will **automatically fill** in key data of a chosen vessel and shipping route.

It also **allows users to update important details** and constraints if required.

### Final Integration

The team **successfully created a model** that gives the optimum allocation for different vessels and all shipping routes.

The application was **successfully integrated into PIL's Liner Management System**. All trade managers can use this application.

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
\$K\$3	TEU Limits Simulated	1293	316.532172	1293	82.1540102	65
\$K\$4	Weight Limits Simulated	14223	147.249815	14223	959.48	962.845
\$K\$5	Move Count Simulated	311	552.950282	311	15	40
\$K\$6	Draft PECL Simulated	8017.08	0	31244	1E+30	3226.92
\$K\$7	Draft ECVG Simulated	4667.61	0	5946	1E+30	1278.39
\$K\$8	Allocation for KM Simulate	64	-549.424574	64	85.7444147	13
\$K\$9	Allocation for SHK Simulate	388	0	272	116	1E+30
\$K\$10	Allocation for MGB Simulate	414	-738.842958	414	116	13
\$K\$11	Allocation for SMA Simulate	427	-872.321534	427	90	13
\$K\$12	Minimum for AK Simulate	232	-9.0212795	232	13	90
\$K\$13	Minimum for GT Simulate	349	-64.685063	349	13	90
\$K\$14	Minimum for PE Simulate	311	0	271	40	1E+30
\$K\$15	Minimum for EC Simulate	401	0	356	65	1E+30

### Sensitivity Analysis

The team also **performed sensitivity analysis** to identify the effect of a small change in a constraint on the objective function.

Users will be able to **identify the impact** that a constraint has on the contribution margin using its **shadow price**.

## ACHIEVEMENTS & BENEFITS

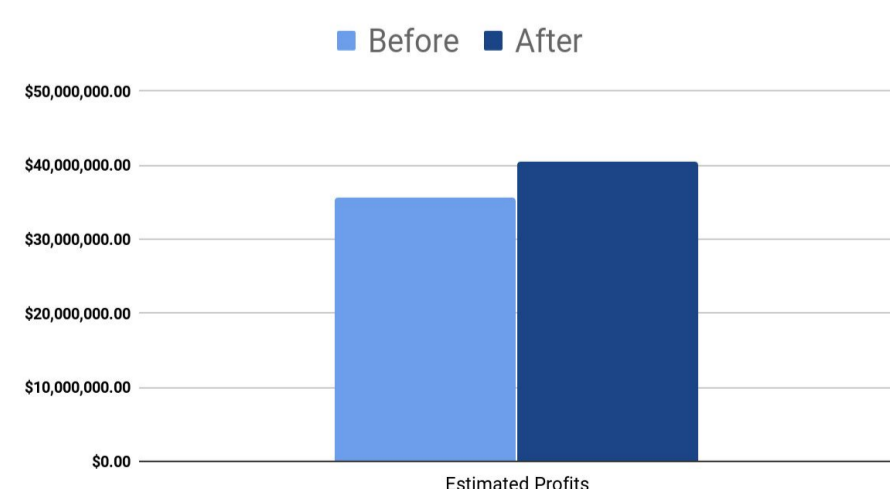
### Impacts

Instantly gives optimum allocation, **saving time**

**Reduced errors** in allocation

**Increase revenue** for all shipping trade routes

### Model Outcome Projected Profits Across All Routes Using The Model



Increase in Profits

**\$4,981,060.00**

Percentage Increase in Profit

**+10%**

### Benefits of Model

**Ease of Use**

**Scalable** for future service lines

**Highly accessible**

## FUTURE DIRECTION

**Calculating profit** in addition to contribution margin by fixing cost

**Incorporating simulation models** to reduce waiting time for ports

**Interconnecting different services** for more comprehensive solution

**Employing Machine Learning** to forecast demand and supply for portpair constraints.

## KEY SKILLS

**Soft Skills**  
+ **Project Management** Used Gantt Chart and Waterfall methodology  
+ **Consulting** Understood user pain points and challenges

**Technical Skills**  
+ **Programming** Coded program in Python language  
+ **Operation Research** Applied Integer Linear Programming concepts

+ **Process Improvement** Automated complex calculation and reduced human error  
+ **Data Analysis & Visualisation** Performed sensitivity analysis, Created interface to display results