**AGU IE213 PROJECT FINAL REPORT**

**SUBMITTED TO**

**THE DEPARTMENT OF INDUSTRIAL ENGINEERING OF**

**ABDULLAH GUL UNIVERSITY**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS**

**FOR THE DEGREE OF BACHELOR OF SCIENCE**

****

**By**

**Ahmet Caner Sağır**

**Taha Arif Tekin**

**Ahmed Demirezen**

**25.01.2022**

**Kayseri / TURKEY**

**TABLE OF CONTENTS**

[**TABLE OF CONTENTS** 1](#_Toc93945568)

[**LIST OF FIGURES** 2](#_Toc93945569)

[**1. INTRODUCTION** 3](#_Toc93945570)

[**1.1 Description of the Company** 3](#_Toc93945571)

[**1.2 Description of the Truck Loading Problem** 3](#_Toc93945572)

[**1.3 Current System of the Company** 4](#_Toc93945573)

[**2. PROBLEM DEFINITION** 5](#_Toc93945574)

[**3. OBJECTIVE OF THE PROJECT** 5](#_Toc93945575)

[**4. LITERATURE REVIEW** 6](#_Toc93945576)

[**5. METHODOLOGY** 7](#_Toc93945577)

[**5.1. Problem Analysis** 7](#_Toc93945578)

[**5.2. Python Model** 12](#_Toc93945579)

[**6. RESULTS** 15](#_Toc93945580)

[**8. CONCLUSION** 17](#_Toc93945582)

[**9. REFERENCES** 18](#_Toc93945583)

**LIST OF FIGURES**

[**Figure 1. Current Truck Loading Method** 4](#_Toc93945251)

[**Figure 2. Heuristically Placed Case** 5](#_Toc93945252)

[**Figure 3. Sample Application (Yurtay, Akcetin and Kılıç (2014))** 6](#_Toc93945253)

[**Figure 4. Python Code Sample** 12](#_Toc93945254)

[**Figure 5. Interface of the Program 3** 13](#_Toc93945255)

[**Figure 6. Product Add Part** 13](#_Toc93945256)

[**Figure 7. Vehicle Add Part** 14](#_Toc93945257)

[**Figure 8. Sample product and Vehicle added to the interface 1** 14](#_Toc93945258)

[**Figure 9. Sample product and Vehicle added to the interface 2** 15](#_Toc93945259)

[**Figure 10. Sample loading output** 15](#_Toc93945260)

**1. INTRODUCTION**

**1.1 Description of the Company**

One of Turkey's largest manufacturers, Hasçelik Kablo San. Trade A.Ş. is one of the largest manufacturers in Europe in its field. It has moved its adventure, which started in 1989 in the Hacılar district of Kayseri, to its modern factory in the Kayseri Organized Industrial Zone, built on 130.000m2 of which 75.000m2 of which is closed area.

Today, our company, which is Turkey's 164th largest private industrial enterprise, exports to 65 countries with its 550 employees.

In the light of the principle of continuous development with high quality, Hasçelik Kablo put into service its second factory in 2014, again in Kayseri Organized Industrial Zone, on an area of ​​​​45.000m2 and a closed area of ​​​​15.000m2. In this facility, OPGW - Fiber Optic Protection Conductors, Aluminum Clad Wires and Grounding Conductors, Optical Fiber Stainless Steel Tubes manufactures Optical Fiber Aluminum Clad Stainless Steel Tubes, Class 1 Solid Aluminum Conductors and Aluminum Tube - Pipes.

Hasçelik Kablo management and employees; It has adopted the principle of keeping quality in the forefront in everything done and complying with the conditions of the quality management system, laws and regulations. By creating the quality system in this direction, it is aimed to fully participate in everyone working.

**1.2 Description of the Truck Loading Problem**

The truck loading problem is the situation in which the products to be logistics are loaded into the appropriate vehicle in the most appropriate way. The loading process, which varies from state to state, depends on direction, parallelism, weight, etc. It could be due to many factors such as the factors that make the problem difficult are the desired constraints. Despite all these constraints, a solution to the problem is sought by finding the most optimal result. An intuitive approach to such problems is another solution.

**1.3 Current System of the Company**

The company intuitively loads the products that need to be loaded into the vehicle. Since the trailer of the truck is not completely closed, there are restrictions such as horizontal placement and maximum 2 on top of each other. These restrictions arise from occupational health and safety. Since they work on order, many different types of reels can be placed on a vehicle.



**Figure 1. Current Truck Loading Method**

Employees working on placing the products on the truck, while placing them intuitively, also pay attention not to put a larger product on top of a product. This restriction is also related to OHS. Because they need to prevent the products from falling by keeping their balance. Each reel weighs approximately 500 kg. There is also a constraint on the carrying capacity of the truck, but it is unlikely that it will exceed the carrying capacity within the framework of the determined constraints.

**2. PROBLEM DEFINITION**

Our problem definition is as follows. Since the products are placed heuristically on the vehicles, they cannot reach the optimum placement result. The fact that they cannot reach the optimum arrangement increases the number of vehicles used for the logistics of the products. The increase in the number of vehicles leads to more costs. In Truck loading problems, the size of the products, their volume, their weight, etc. Many factors make it difficult to reach the optimum result. For this reason, employees must place their products heuristically.

yol, gök, açık hava, ulaşım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure 2. Heuristically Placed Case**

**3. OBJECTIVE OF THE PROJECT**

Mathematical modelling can be used to find the most appropriate solution to truck loading problems. The objective was to reduce the number of tools used. We can specify this in our model in two ways. By minimizing the number of vehicles or maximizing the number of products placed in that vehicle. With this modelling, we can minimize the number of vehicles and reduce the cost.

.

**4. LITERATURE REVIEW**

Our problem is to place the reels in the container in the most effective and optimum way. As there are different types of reels and containers, we searched the literature from several sources and found and read relevant articles.

Yurtay, Akcetin and Kılıç (2014) argue that logistics has gained a much greater importance due to the increase in international trade. This necessitates logistics operations to be carried out much faster. This speed also means optimizing the loading capacity and making this application happen quickly. It is argued that it may be appropriate to develop computer support programs to make complex cargo and loading planning more efficient. After receiving the order and shipping information of the companies, higher efficiency can be achieved by optimizing this with the help of utilities. This efficiency means cheaper cost and more profitable business.

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure 3. Sample Application (Yurtay, Akcetin and Kılıç (2014))**

According to Baverin (2021), getting the most out of loading is a challenging process for many companies. The increase in competition affects companies economically to a significant extent. In 2019, it affected almost 40% of transport and truck companies due to the increase in competition. Making the right planning can help companies by putting them forward in this regard. We can load large and small products according to the optimum usage area in accordance with the regulations and rules.

According to the research of Morabito, Morales and Widmer (2000), many analysts believe that logistics is one of the most important potential sources for increasing corporate efficiency and gaining competitive advantage. It is argued that in addition to optimizing the loading areas, it will be beneficial to choose the most suitable products in the best way. Random selection of products with different types of features can be optimized thanks to the algorithm. As a result of their experiments, they realized that they created a more efficient usage area with the program that would create an algorithm and calculate these samples.

As a result of our literature review, we realized that logistics is important for company values. We have concluded that we can achieve efficiency by combining the optimization of loading capacity, which is one of the biggest problems in logistics, with modeling appropriate algorithms.

**5. METHODOLOGY**

**5.1. Problem Analysis**

We aimed to create a mathematical model that will help determine the optimum arrangement and number of products to be loaded on trucks and similar vehicles. We also aimed to develop a desktop program to assist this optimization alongside the mathematical model. This program is aimed to take the dimensions of the trucks and similar vehicles, where the products with certain dimensions will be placed, from the user and to optimize the placement thanks to the necessary algorithms. A literature search was conducted to determine the foundations of the software to be developed. According to the research, we decided to use Python to develop a desktop program. We will obtain the necessary data for the mathematical model to be created and the program to be developed by the company. We intend to reduce the cost of the company in transactions that take place after installation. The reason for this is that each vehicle to be used to load the products causes a separate cost, and the number of vehicles to be used is minimized and this cost is reduced.

Our aim will be to perform the optimum loading process, considering the product types, number of vehicles, company budget, and additional restrictions that may be added in this process.

Firstly, we analysed our data and decided as follows:

**Indices**







**Data**

r(i) radius of the cargo I

h(i) height of the cargo I

m(i) weight of the cargo I

L(j) length of the vehicle j

H(j) height of the vehicle j

W(j) width of the vehicle j

M(j) weight of the vehicle j

**Decision Variables**

 1 if the package is in the vehicle j,0 otherwise  
 x position of the package i

 y position of the package i

 z the position of the package i

 the position of right bottom of the package i

 the position of left bottom of the package i

 the position of top of the package i

 absolute value of the distance cargo i to k

 absolute value of the distance between cargo i to k

 absolute value of the distance between cargo i to k

 1 if the length of item i is parallel to X-axis, 0 otherwise.

 1 if the length of item i is parallel to Y-axis, 0 otherwise.

 1 if the width of item i is parallel to X-axis, 0 otherwise.

 1 if the width of item i is parallel to Y-axis, 0 otherwise.

Continuous variables denoting barycentre of bin j on the x-, y- and z- axes respectively







 the package k position on the x+, x-, y+, y-, z+ and z- axes respectively













 its equal to 

 its equal to 

 its equal to 

 its equal to 

 its equal to 

 its equal to 

 its equal to 

 its equal to 

 its equal to 

 its equal to 

 1 if the cargo k right of the i

 1 if the cargo k left of the i

 1 if the cargo k front of the i

 1 if the cargo k behind of the i

 1 if the cargo k top of the i

 1 if the cargo k bottom of the i

**Formulation**

metin, kol saati içeren bir resim

Açıklama otomatik olarak oluşturuldu

**S.T.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu



























































































































​​

**5.2. Python Model**

We have coded a python program that has the interface for the optimization application. We accessed the modelling libraries for the program and inserted the modelling data we specified into our code.

For example, for that constraint:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

The python code is as follows:

    for i in \_data["i"]:

        for k in range(i):

            # 3 vertices of package k should be on a package i if k is on i.

            prob += x[k] >= x[i]

            prob += x[k] <= x\_[i] + (1-zp[i][k]) \* 1000000

            prob += y[k] >= y[i]

            prob += y[k] <= y\_[i] + (1-zp[i][k]) \* 1000000

            prob += x\_[k] >= x[i]

            prob += x\_[k] <= x\_[i] + (1-zp[i][k]) \* 1000000

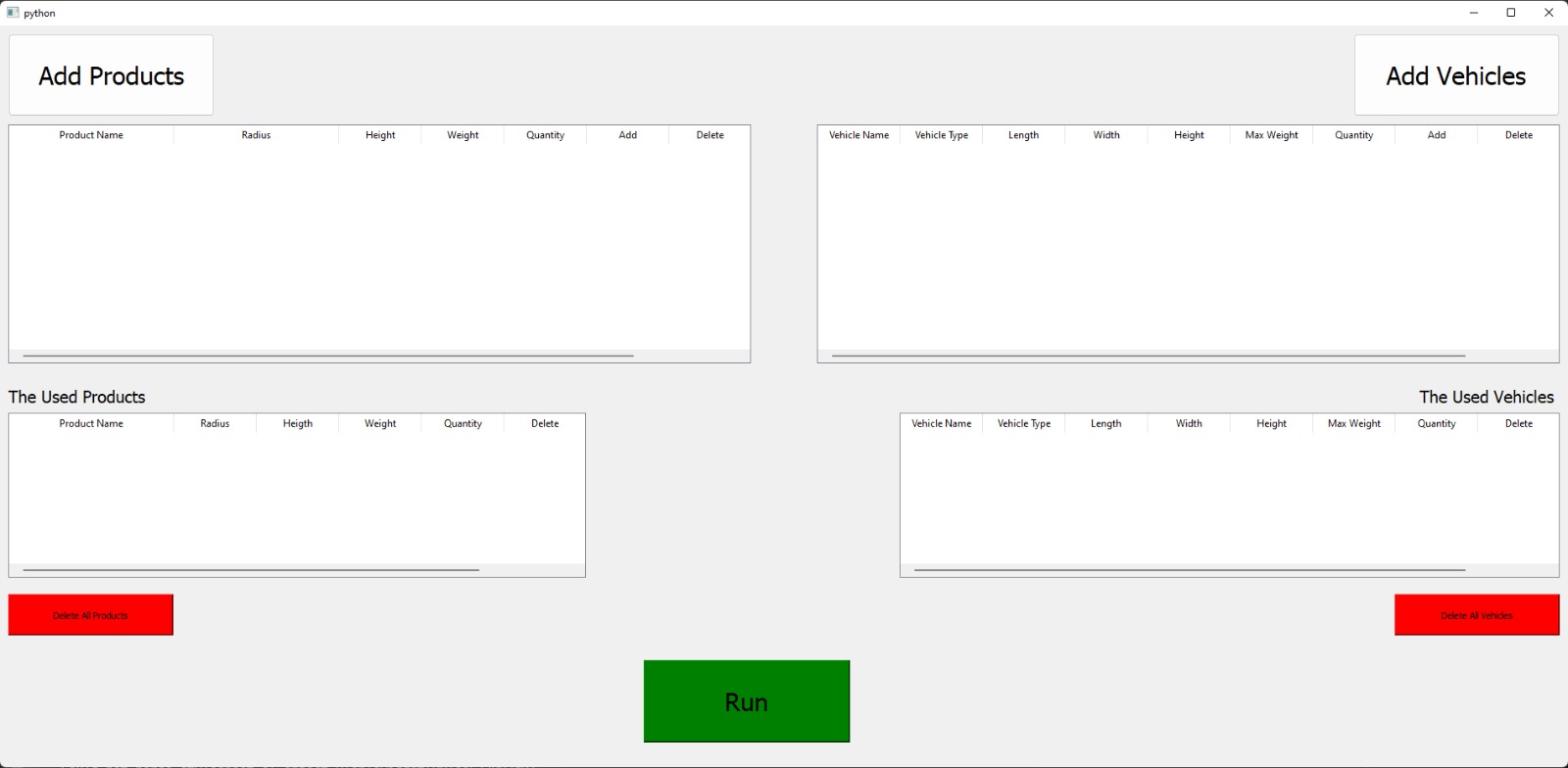
            prob += y\_[k] >= y[i]

            prob += y\_[k] <= y\_[i] + (1-zp[i][k]) \* 1000000

            prob += x\_[k] <= x\_[i] + (1-zp[i][k]) \* 1000000

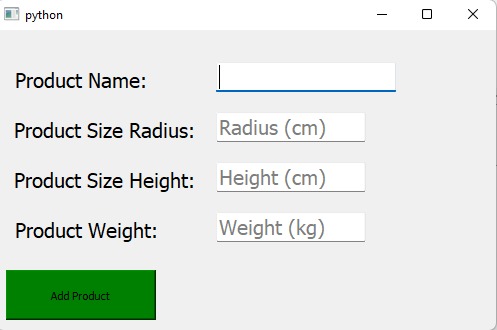
**Figure 4. Python Code Sample**

In the interface in the application, there are options to add the products they produce and the tools they have. And they have a list to see them. Additionally, there is a list for them to see. Whichever product and tool they want to use, they can send it to the list below and run the program.



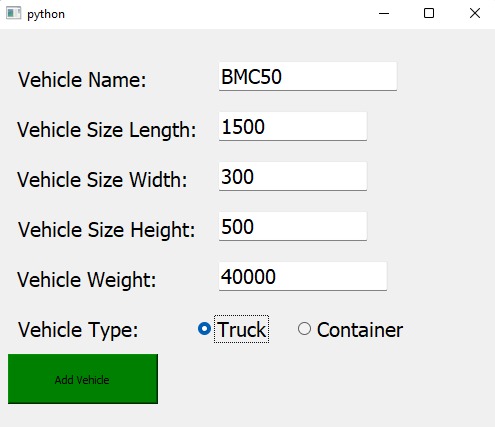
**Figure 5. Interface of the Program**

For the products, the product name, diameter, height and weight information can be added to the model.

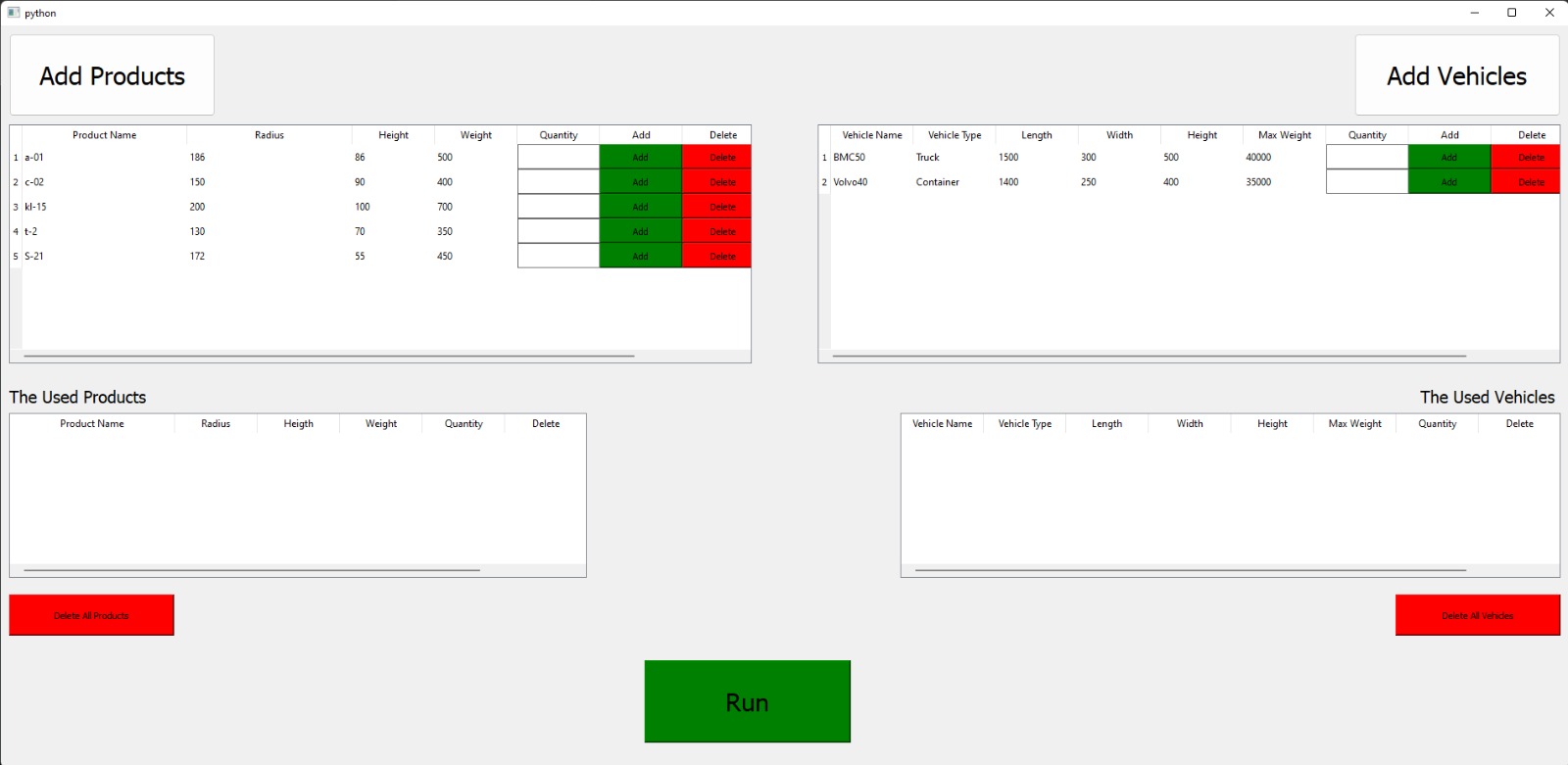


**Figure 6. Product Add Part**

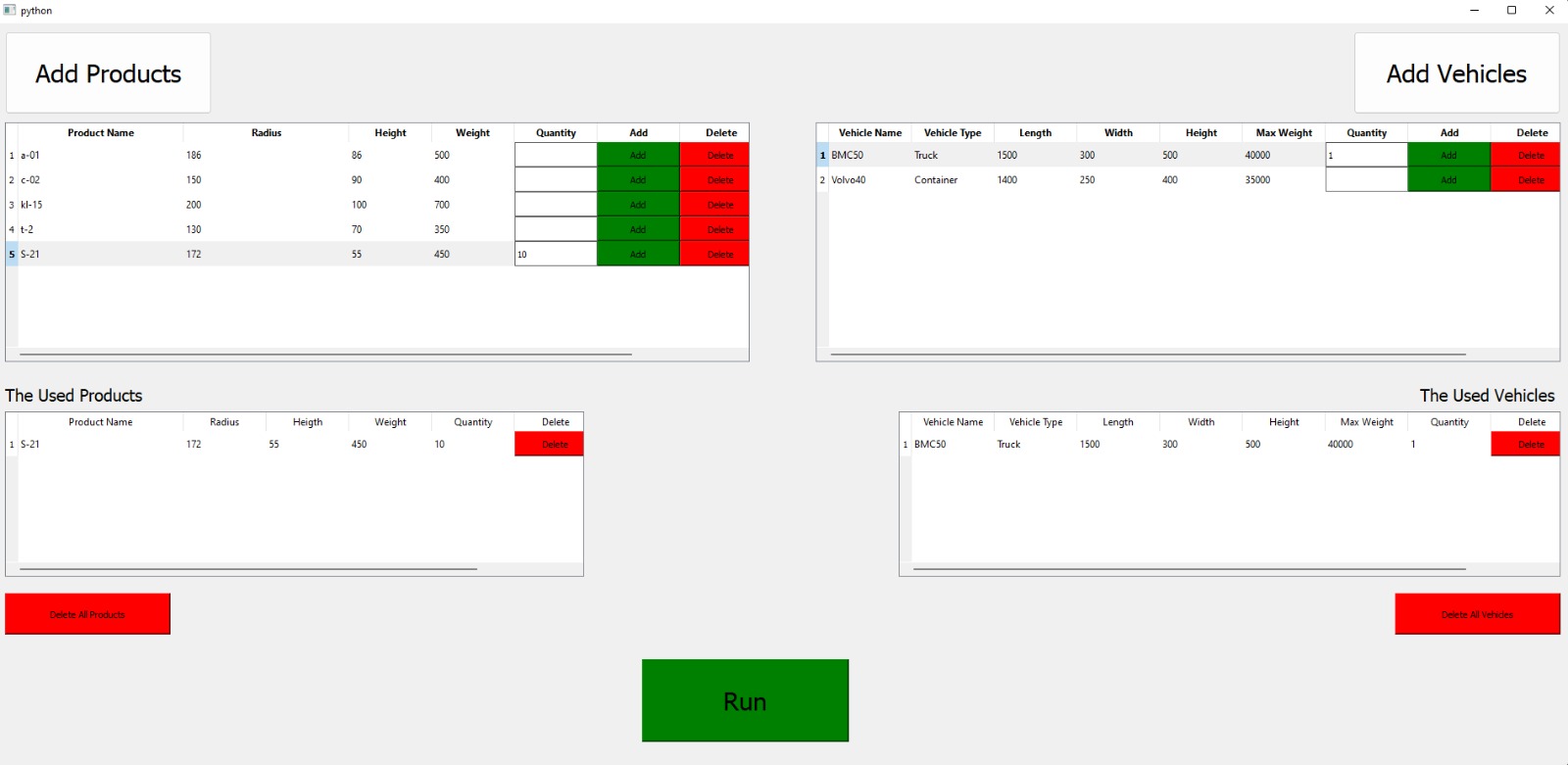
To add a vehicle, the name, length, width, height, max weight and type values can be added to the model.



**Figure 7. Vehicle Add Part**



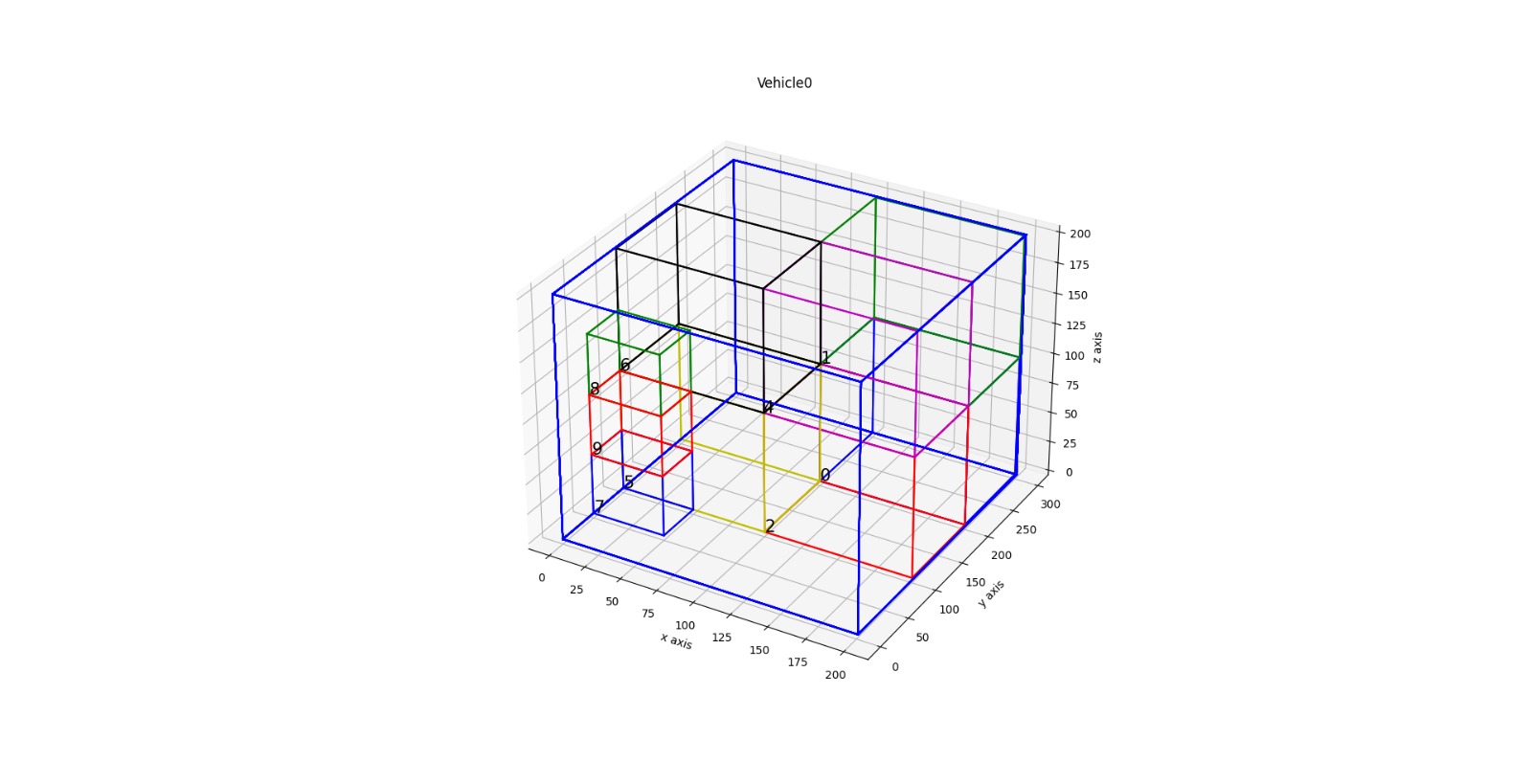
**Figure 8. Sample product and Vehicle added to the interface 1**



**Figure 9. Sample product and Vehicle added to the interface 2**

**6. RESULTS**

In the case of fitting the sample 10 products we gave as the output of the program to 1 vehicle; we obtained the following output.



**Figure 10. Sample loading output**

To get this output, we entered the following data as sample data.

data2 = {}

data2["i"] = [0,1,2,3,4,5,6,7,8,9]

data2["k"] = [0,1,2,3,4,5,6,7,8,9]

data2["j"] = [0]

data2["l"] = [0,1,2,3]

data2["pRadius"] = [100,100,100,100,100,100,100,50,50,50,50]

data2["pHeight"] = [100,100,100,100,100,100,100,50,50,50,50]

data2["pMass"] = [100,100,100,100,100,100,100,50,50,50,50]

# x

data2["vLength"] = [201,250]

# z

data2["vHeight"] = [201,250]

# y

data2["vWide"] = [300,250]

data2["vMass"] = [40000,40000]

**8. CONCLUSION**

To sum up everything that has been stated so far, after understanding the main problem in our Problem, we worked result oriented. The problem experienced by the company is to increase the number of loaded products by optimizing the arrangement of the products on the vehicle. As a result of the literature research, we have done in line with this goal, we have developed our mathematical model to solve our problem. Later, we thought of a computer program and implemented it to make it easier to use and for the end user to use it comfortably. The computer program we made was created with Python scripting language. At the same time, the mathematical model we created was included in the computer program thanks to the python Pulp library. After its creation in its interface, the program found its final form. Finally, for the convenience of the end user, the result was rendered in 3D with the Matplotlib library and printed on the screen for the user to review. After all these program developments, when we thought that our main goal here was to increase the number of loaded products, we worked as company oriented as possible in order to achieve this and worked to ensure that the end user could make the most of it.

**9. REFERENCES**

1. Beverin, A. (2021, October 8). Load Planning: How to Start Optimizing Your Deliveries. OptimoRoute. <https://optimoroute.com/load-planning/>
2. Morabito, R., Morales, S. R., & Widmer, J. A. (2000). Loading optimization of palletized products on trucks. Transportation Research Part E: Logistics and Transportation Review, 36(4), 285–296. <https://doi.org/10.1016/s1366-5545(00)00003-x>
3. YURTAY, Y., YURTAY, N., AKCETIN, E., & KILIC, A. (2014). KONTEYNERDE YÜK OPTİMİZASYONU: ÖRNEK UYGULAMA. Journal of Management and Economics Research, 23, 228. <https://doi.org/10.11611/jmer353>
4. Erbayrak, S., Özkır, V., & Mahir Yıldırım, U. (2021). Multi-objective 3D bin packing problem with load balance and product family concerns. Computers & Industrial Engineering, 159, 107518. <https://doi.org/10.1016/j.cie.2021.107518>
5. Hu, N. Z., Li, H. L., & Tsai, J. F. (2012). Solving Packing Problems by a Distributed Global Optimization Algorithm. Mathematical Problems in Engineering, 2012, 1–13. <https://doi.org/10.1155/2012/931092>