**Analysis of problem**

The program has many major and minor problems that needs to be addressed. Here, we have done deep analysis of problem and possible solutions:

Truck, Routes, and Paths:

The major problem with truck is how can we identify whether packages are already loaded in truck, or it is empty. Moreover, it will be quite challenge as we have to add weights specified by customers for all three different trucks. At last, finding the shortest path would be the most challenging part.

Shipment Specification:

It was mentioned that customers specify weight, box size, and destination, but it doesn't specify how this information is input into the system or how it's processed. So, how to store these data was not clearly mentioned.

Measuring Distance:

One of major problem was measuring distance using concept of Euclidean geometry for which example is provided, but we have to figure it out how to calculate these distances in the context of the algorithm and convert it into code. As visual example of distance calculations was provided but doesn't specify how these distances should be computed programmatically.

Shortest Path Algorithm:

We are provided with A\* algorithm for finding the shortest path, but it lacks a detailed description of how this algorithm should be implemented or integrated into the overall process. Implement it into logic would not be easy. Also, the text briefly introduces an potential issues with the shortest path algorithm, such as getting stuck, but doesn't provide solutions or workarounds.

Exception Handling:

There was mentioned of situations where a truck cannot find a path to the destination, but it doesn't describe how the program should handle these scenarios. To add, the text mentions storing a package in storage at the depot if no truck can take it, but it doesn't explain how to manage such storage or retrieval.

To sum up all, the provided description lacks the technical details necessary for a complete implementation. It needs a clear representation of the distance calculation methods, and detailed instructions for implementing the A\* algorithm. Additionally, exception handling, and efficiency considerations need to be addressed for a detailed functional solution.