Introduction to Artificial Intelligence
Programming Assignment 1 - Theortical Bonus

We'll show that single-agent pickup and delivery problem(SAPD) is in NP-HARD compleixity class by using a reduction from Traveling Salesman Problem which we saw in previous courses that $TSP \in$ NP-HARD , by the reduction theorem we will conclude that $SAPD \in$ NP-HARD.
First, lets show that SAPD $\in$ NP by showing the existence of a verifier V in polynomial time:
The Verifier will work as follows:
given the output to the SAPD problem - a path $\{L_1, \ldots, L_m\} \in V_G$ in the graph, our Verifier will work as follows:
1. will check that all the nodes in the Packages list $v_1, , , v_p \in P \subseteq V$, are in the path - $O(V^2)$
1.2. else: return false
2. will check that all the nodes in the Delivery list $u_1, , , u_d \in D \subseteq V$, are in the path - $O(V^2)$
2.2. else: return false
3. for each vertex $v_1, , , v_p$in the Package list:
3.1 check that its delivery location appear after it in the path. for $v_i = L_i$ check for $u_i \in L_{i+1}, \cdots, L_m$: $|V| - 1 + |V| - 2+, \ldots 1 = O(V^2)$
3.2 else return false
4. if all the packages have their delivery location after them in the output path, return True, else return false
Verifier Correctness:
Given an output to the SAPD problem - a path in graph G, our verifier will check if the agent will path through all the Packages and delivery locations given in the output, and after that will check if each Package is able to reach its Delivery location (according to the output path).
if the Verifier output is True, our output path have all the packages and delivery locations inside it, meaning our agent is visitng all those nodes. additionally if the Verifier output is True, the agent visit each package before he visit its delivery location, meaning all the packages have been delivered.
if the Verifier output is False, the agent isnt visiting all the packages\delivery locations, or visits a delivery location before he visits its package location (meaning we will have a package we didnt deliverr) so the output path isnt a valid solution to the SAPD problem.
Verifier Time complexity:

checking if all the packge\delivery location in the path (worst case we have $n = O(V)$ pickups and delivery locations (all nodes are both), so we will need to check all the verticies concluding to $2*O(V^2)$. For the last part of the Verifier we will pay $O(V^2)$ as well. concluding to the $Verifier$ being in polynomial time. (in the worst case the output path will have all the vertices).
from the Verifier Theorem we can conclude that SAPD $\in NP$.
now lets show that its in $NP - HARD$ using reduction:
Intuition:
travelling salesman problem (TSP), asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?" (wikipedia).
The TSP problem is formulated as:
Input: a graph G, set of locations $\{L_1, \ldots, L_n\} \in V_G$, distance function
Output: Hamiltonian circle between $\{L_1, \ldots, L_n\}$ - a path that visits each vertex excatly once, and returns to the origin vertex, with minimal distance.
as we see its not very far from our own problem - find the shortest path in the graph, that visitis all the Pickup, and Delivery locations in the graph.
Restateing the TSP to be more similar to SAPD porblem:
Input: A graph G, set of locations $\{L_1, \ldots, L_n\} \in V_G$ where each location has both a pickup point and a delivery point. A distance function \(d\) that gives the distance between any two locations.
Task: Find the shortest tour that visits each pickup point and its corresponding delivery point exactly once and returns to the starting location.
Reduction: Given an instance of the TSP with pickup and delivery, we construct an instance of the SAPD as follows:
1. For each pickup and delivery pair in the TSP instance, create a single location in the SAPD instance. This location represents both the pickup and delivery points.
2. Connect the locations in the SAPD instance with edges representing the distances between the corresponding pickup and delivery pairs in the TSP instance. (similar to the mst creation in our code)
3. Assign zero costs to all edges in SAPD instance.
The resulting SAPD instance essentially preserves the structure of the TSP with pickup and delivery.
Solving the SAPD instance optimally is equivalent to solving the TSP with pickup and delivery optimally, as the zero costs ensure that visiting each location is the same as visiting the corresponding pickup and delivery points in the TSP instance.
Since the TSP with pickup and delivery is NP-hard, and solving it allows us to solve the SAPD optimally, we can conclude that the SAPD is also NP-hard.
Correctness:
We want to show that output in the Package-Delivery TSP instance is True iff the output to the SAPD problem is True. by True meaning the path is correct according to each problem defnition.

if the output of the Package-Delivery TSP is True, meaning the output path visits all the package\delivery vertices, making it a correct output to the SAPD problem.
if the output of the Package-Delivery TSP is False, meaning the output path not visits all the package\delivery vertices, making it not sutiable solution to the SAPD problem, because there must be a package we didnt collect or didnt deliver.
This reduction demonstrates that the SAPD is at least as hard as the NP-hard TSP with pickup and delivery, establishing its NP-hardness.