

Accio Robotics Screening Round

HANDS-ON EVALUATION TASK

Role: Software Development Engineer

Code Documentation:

The attached python code will generate random orders in a warehouse scenario and simulates two picking modes for the generated orders. Below is a brief documentation for this code:

1. Parameters Used:

- 'N': Number of robots in the warehouse.
- 'L' and 'W': Grid size of the warehouse.
- 'P': Number of packing stations in the warehouse.
- 'mean_interval': Mean interval between order generations (in seconds).
- 'variance': Variance for order generation time.
- 'mean_items_per_order': Mean number of items per order.
- 'variance_items_per_order': Variance for item count distribution.
- 'packing_time_per_item': Time required to pack one item (in seconds).
- 'robot_speed': Speed of the robots in meters per second.
- 'simulation_time': Total simulation time in seconds (1 hour in this case).

2. Initialization:

- Initialize variables and data structures and create an empty list to store orders.
- Set the initial packing station, order number and current time.

3. Order Generation (Part 1):

- Generate random orders within the specified simulation time.
- For each order, record the generated time, order number, a list of randomly selected items, and the assigned packing station.
- Update variables for the next order generation.

4. Functions:

- 'generate_item_list()': Generates a random item list for an order.
- 'manhattan_distance()': Calculates the Manhattan distance between two nodes in the grid.

Manhattan Distance:

In this problem, the Manhattan distance is used to calculate the distance between two nodes present in the grid pattern of warehouse. It is used to estimate the travel distance for robots to move between nodes in the grid.

➤ Calculating Robot Travel Distance:

When a robot is picking items and delivering them to the designated packing station, Manhattan distance is used to calculate the distance travelled by the robot. The code considers the grid-based structure of the warehouse and calculates the distance as the sum of the absolute differences of the x-coordinates and y-coordinates between the current location and the destination node.

For example, if a robot is moving from node (2, 3) to node (4, 6) in the grid, the Manhattan distance would be calculated as:

$$\text{Manhattan Distance} = |4 - 2| + |6 - 3| = 2 + 3 = 5$$

This distance is used to estimate the travel time for the robot based on its speed.

➤ Determining Packing Station Distance:

Manhattan distance is also used to calculate the distance between the current location of the robot and the designated packing station. This is done to account for the time it takes for the robot to move from the last pick node to the packing station.

The Manhattan distance is a simple and effective way to estimate travel distance within a grid-based warehouse environment, allowing the code to calculate travel times for the robots in the simulation.

5. Simulation of Picking Modes (Part 2):

- Loop through each order and its item list.
- Simulate both picking modes (whole order and split order) for each item.
- Calculate the time taken for each picking mode, considering robot availability and packing time.
- Store the completion times for both modes in separate lists.

6. Results (Data Storage and CSV Output):

- Store the completion times for both modes along with order numbers and pick nodes in separate lists.
- Create data dictionaries for each mode's results.
- Create a Pandas DataFrame to organize the data.
- Write the results to a CSV file named "report.csv".

1	Order Num	Pick_node	completed_time_mode_1	completed_time_mode_2
2	1	30	00:00:25	00:00:25
3	2	13	00:00:51	00:00:51
4	3	1	00:27:32	00:03:37
5	3	14	00:27:32	00:06:57
6	3	15	00:27:32	00:10:17
7	3	18	00:27:32	00:13:37
8	3	31	00:27:32	00:16:57
9	3	32	00:27:32	00:20:17
10	3	35	00:27:32	00:23:37
11	3	54	00:27:32	00:26:57
12	4	2	01:27:30	00:31:35

The CSV file has the details of **Order Num**, **Pick_node**, **completed_time_mode_1** (whole order picked by single robot) and **completed_time_mode_2** (split the order by robots) and saves the results into a file.

Order Num 3, has 8 Pick nodes which are in the increasing order (1,14,15,18,31,32,35,54) and for mode_1 as single robot pick all the items completion time for each item is same which is (00:27:32) and for mode_2 as items were picked by multiple robot's and completion time for each item will be different and also second item will be picked only if it already picks the first item.

Below is the documentation for the common libraries and dependencies used in the code:

The code utilizes the following Python libraries and dependencies and make sure to install these libraries before running the code.

1. Spider IDE: Created the script and generated the results into csv file using Spider's integrated development environment. The csv file will be generated in the same path where the script file is copied and executed.
2. NumPy: It is used for numerical operations and array handling. It can be installed by using pip (pip install numpy).
3. Pandas: It is used for data manipulation and Data Frame management. It can be installed by using pip (pip install pandas).
4. Time and DateTime: These libraries are used for working with time formats. We used this to convert HH:MM:SS formats to only seconds (i.e. 01:00:00 to 3600 seconds for example) and vice versa. These libraries are part of Python's standard library and do not require separate installation.
5. Random: This library is used for generating random numbers and also to make random selections. It is also part of Python's standard library and does not require separate installation.

Steps to run the script:

1. Open Spider IDE or make sure the above-mentioned libraries were installed into your environment.
2. Copy the **project.py** file and execute it using Spider or any other python compiler to which the specified libraries were installed.
3. It will automatically generate the results into **report.csv** file. Since we are using random functions, if the script is executed multiple times the results in csv file will be varied.
4. Before executing the project.py file multiple times, please ensure that report.csv file if it is already closed. Else it says (PermissionError: [Errno 13] Permission denied: 'report.csv').
5. As specified in the problem statement, results were generated considering LxW grid as 9x6 Grid. These values can always be changed to required values using L, W parameters in the code. In fact all the variable can be changed to any random value.

Future Scope:

- As we used only time format for this project, once the completion time in any of the mode reaches to 24hrs (i.e 23:59:59) it starts from (00:00:00). In this case if we add date (DD:MM:YYYY) to this project then the results will be more precise. We can add this as a future scope.
- Enable real-time monitoring and control of these packing systems. It helps in predictive maintenance, reducing downtime and optimizing the overall performance of these systems.
- It helps in avoiding the repetition of same order getting packed multiple times which lead to loss for the firm or warehouse.