**BITS Assignment – Problem Statement Number 3**

**Artificial and Computation Intelligence**

**Group No. 35**

**Group Members:**

Akshit Gupta 2019HC04448

Amit Dahore 2019HC04214

Gaurav Pandey 2019HC04196

Priyanka Padalalu 2019HC04404

Ramakrishna 2019HC04933

1. **Explain the heuristic that can be used to solve the problem? Justify your choice.**
   * + 1. We have created different combinations of packets having maximum weight 10 kg called Sets# at the start and that will remain same throughout the algorithm. 10 kg is the maximum weight robot can take at a time.  
          *Ex- Below are few possible sets.  
          Set-1: 1 kg+ 1 kg + 1 kg+ 1 kg + 1 kg + 5 kg  
          Set-2: 2 kg + 2 kg + 6 kg*

*Set-3: 2 kg + 4 kg + 4 kg  
………………………….*

*………………………….*

*………………………….*

As creating set is random, while running the code we have noticed that 13 or 14 such sets will be created. Let’s say 13 Sets are created for this run.

1. Robot will pick any random set from above 13 sets and start its commute to rooms.

*Ex-Let’s say Robot picks set-1.*

1. Algorithm will select any Random room. If the room has capacity less than the current weight of packets with robot, we will select new Random room.

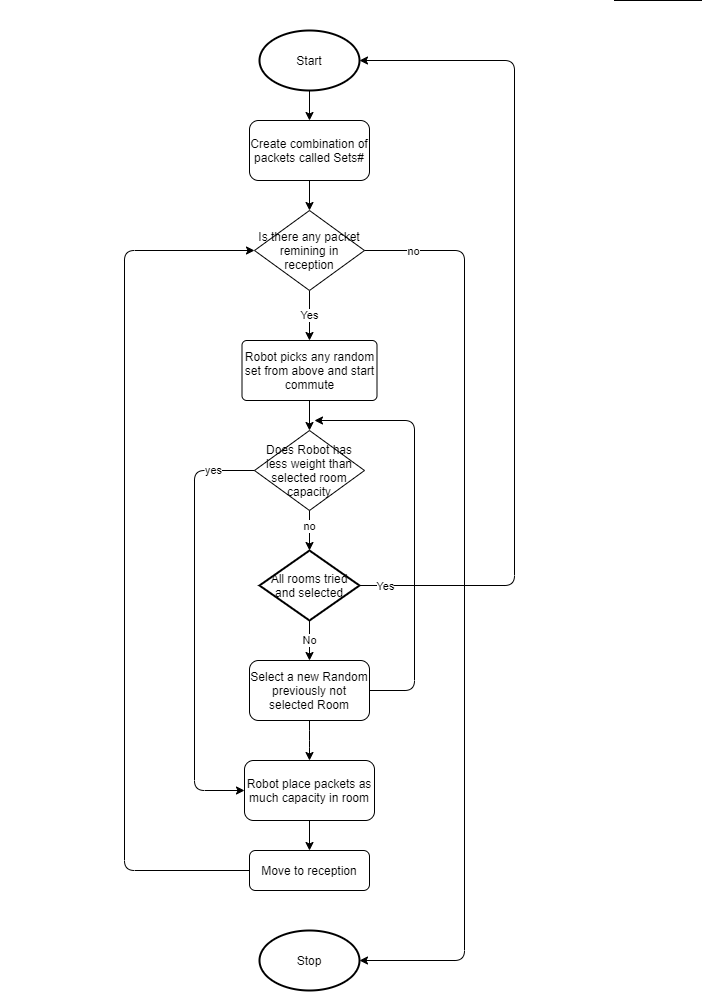
*Ex-Let’s say Random room selected was of 5 kg capacity. Since current total weight of packets carried by robot is greater than the room capacity, it will reject the room and find another room with sufficient capacity. Let’s say new Random room selected is of 20 kg*

Edge case: If there is no room available with the required capacity, we can safely assume that our initial combination is not perfect, and we should restart with step 1.

1. Robot will place the packets in the room up to the room’s capacity.

*Ex-Robot will place 10 kg packet in the selected 20 kg capacity room.*

1. Robot will come to reception and randomly selects a new set of packets.
2. In this way robot has taken all the packets.



1. **Explain the cost function associated with your search in reaching the goal.**

We have randomly grouped the packets in Sets# and total weight is maximum of 10 kg per commute. In this way we are making constant number of commutes. As number of commutes will remain same, we are ignoring it in cost function.

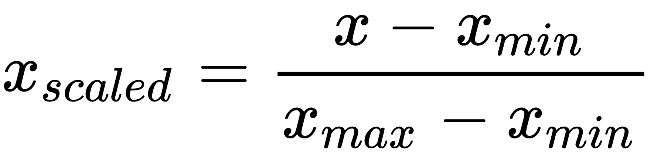
As we are filling the rooms as per the weight of the sets and once the capacity of the room is full i.e. it cannot store the new set, we will randomly select a new room for dropping payload. We have considered room utilization as our priority in the algorithm i.e. Selecting fewer rooms and Optimizing space.

Now, we will be calculating below two points:

* Space wasted: It is possible that few rooms are partially occupied and 1 kg, 2 kg fragmented space is still available.
* Unutilized Rooms: There is possibly that few rooms are fully empty.

Note: Since we are comparing the two variables with different scale of measurements i.e. One we are calculating the ration of wasted space to the total space of the room and count of unutilized rooms. Therefore, we have used Min-Max Normalization to normalize our results in evaluation function.

Formula used:



Also, for the number of rooms utilized and the space wastage, we take the summation of normalized values for the free space available in the utilized rooms (values normalized to 0 -1). Then we will take the summation of the normalized value for the total storage space of the unutilized rooms. Our cost function will be then calculated as the difference between the two.

Below is the **Objective function:**

i - t

*Where:*

*m = number of fully vacant rooms*

*n = number of utilized rooms*

*ui = normalized value of remaining storage space in room i.*

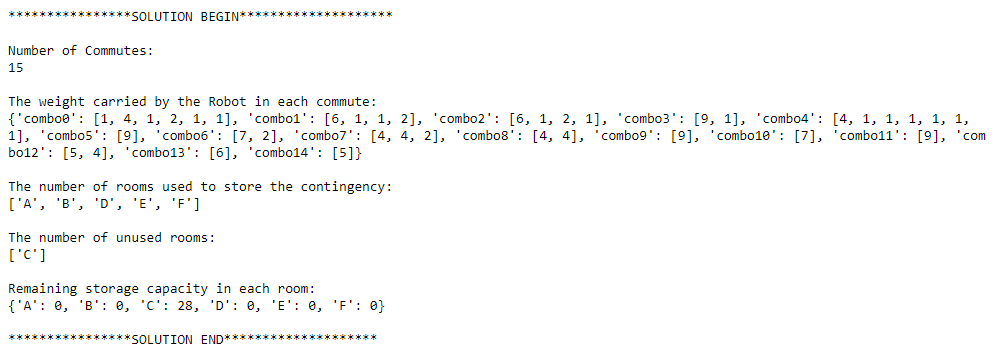
*ft = normalized value of total (unused) storage space in room t.*

1. **Choose the correct algorithm suitable for this grid search.**

We have chosen **Iterative Hill Climbing Algorithm** as it helps us to avoid the major drawback of the normal hill climbing algorithm which is its tendency to become stuck at foothills, a plateau or a ridge and gets stuck at local minima if the initial state chosen is closer to the local minima as compared to the global minima.

In our solution, to reach out to the global minima, we have used a slightly modified approach for this algorithm, wherein we start with multiple random initial state, and then proceed to find the solution to overcome getting stuck in local minima. In the end, we will find out the best solution (minimum wastage and max free rooms) from the above iterations. This increases the probability of finding the global minima as compared to the simple hill climbing algorithm.

The final solution of the algorithm looks like below:



1. **Representation of the environment, fringe and the data structures used**

We know that there are different types of agents in AI. PEAS System is used to categorize similar agents together. The PEAS system delivers the performance measure with respect to the environment, actuators and sensors of the respective agent.

PEAS stand for Performance measure, Environment, Actuator, Sensor.

Performance Measure: Performance measure is the unit to define the success of an agent. Performance varies with agents based on their different precept.

Environment: Environment is the surrounding of an agent at every instant. It keeps changing with time if the agent is set in motion.

There are 5 major types of environments:

* Fully Observable & Partially Observable
* Episodic & Static
* Static & Dynamic
* Discrete & Continuous
* Deterministic & Stochastic

Actuator: Actuator is a part of the agent that delivers the output of an action to the environment.

Sensor: Sensors are the receptive parts of an agent which takes in the input for the agent.

| **Agent** | **Performance Measure** | **Environment** | **Actuator** | **Sensor** |
| --- | --- | --- | --- | --- |
| Packet Delivery System | Space utilization,  Number of Commutes,  Time taken,  Number of unutilized rooms. | Number of Rooms,  Storage Capacity of Rooms,  Accessibility of rooms,  Robot size or weight carrying capacity,  Conveyor belt. | Algorithm,  Jointed arm,  Hardware for measuring weight. | Camera,  GPS,  Sensor,  Weighing sensor,  Joint angle sensors. |