Following is a warehouse-logistics problem. Some generalizations are possible but I have tried to pin down to some concrete settings. If you want you can generalize but the concrete setting will serve as the reference point.

There is a warehouse with grid markers to specify locations.

1. one input bay at location (Ix, Iy)
2. one output bay at location (Ox, Oy)
3. one charging bay at location (Cx, Cy)
4. 12 racks in a 4x3 rectangle (with locations such that there are corridors between two racks)
5. Each rack has one shelf of capacity of 5 objects

There are 4 robots, each one is capable of

1. Moving (from a point A to a point B)
2. Picking an object from a rack/bay
3. Dropping an object on a rack/bay
4. *Getting itself recharged by going to the charging station and requesting for recharge //for the time being consider that the robots do not need recharging.

There is a sequence of external requests at times t1 < t2 < t3 < … Each request is of type

1. Store o1:T1, o2:T2,.. //object o1 which is of type T1, etc
2. Get n1:T1, n2:T2,.. //n1 objects of type T1, ….

When there is a store request, one or more robots are allocated to take the objects from the input bay to the free racks. When there is a get request, again one or more robots are allocated to pick up the required objects from the racks and drop them on the output bays.

All the operations have some costs e.g. cost of moving an object from A to B depends upon the distance and the type of object. This determines the cost for Store and Get.

Problem: Given the warehouse description above and a cost C, design an RL based system to ensure the cost of Get requests are less than C.

Notes:

1. When there are more Get requests of certain type of objects, these objects could be arranged to be near the output bay. If there are no free racks near the output bay, then objects can be swapped to make space. Such swapping can be done by the robots in the spare time. This intuition should be validated by RL.
2. Swapping also incurs cost. There should be a generalized notion of cost to capture the “efficiency” of the warehouse. For any finite sequence S_k of requests, let the no. of objects to be accessed (Get) is N_k and the total cost incurred by the sequence is C_k. One definition of efficiency could be C_k/N_k: the lesser the value, the more efficient the warehouse is.