

Application of Multi-agent Reinforcement Learning in Warehouse Logistics

Nikunj Gupta
IIIT, Bangalore



Introduction to me and What I am doing here :)

Nikunj Gupta

- Final year IMTech Student @ IIIT-Bangalore
- Currently, pursuing a Master's thesis
 - Under Prof. G. Srinivasaraghavan @ IIIT-B
 - In Association with Ericsson Research
- Summer internship
 - Optimal placement of items in a warehouse

Use case scenario

Warehouse Management System

- Demand forecasting
- Placement of items in the warehouse

Overall Objective: Maximize throughput and
Reduce incurred costs

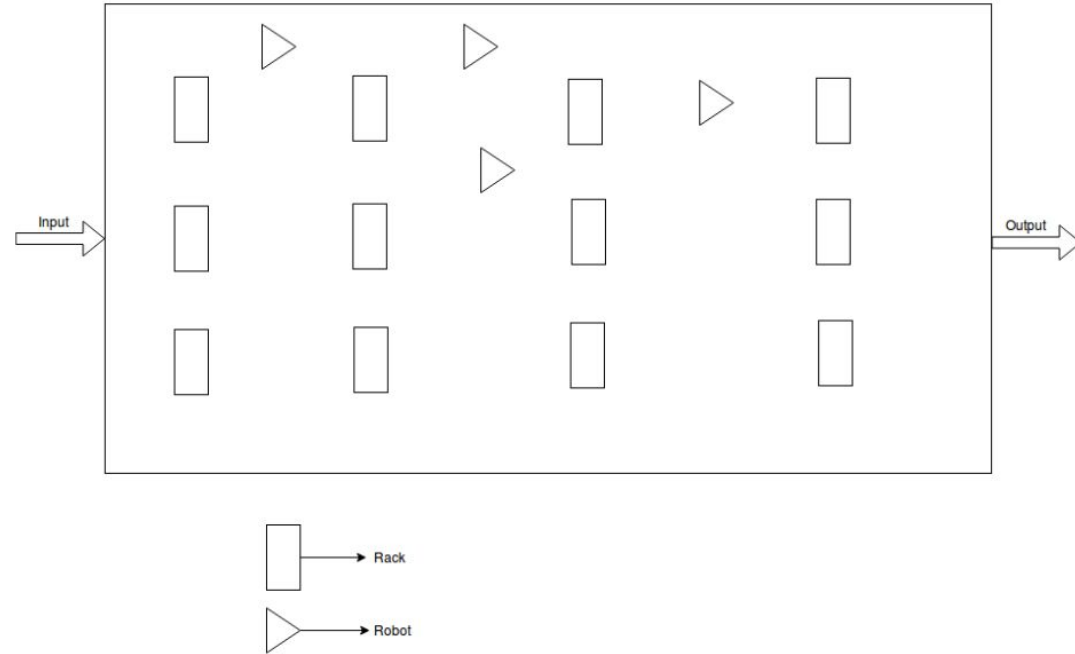


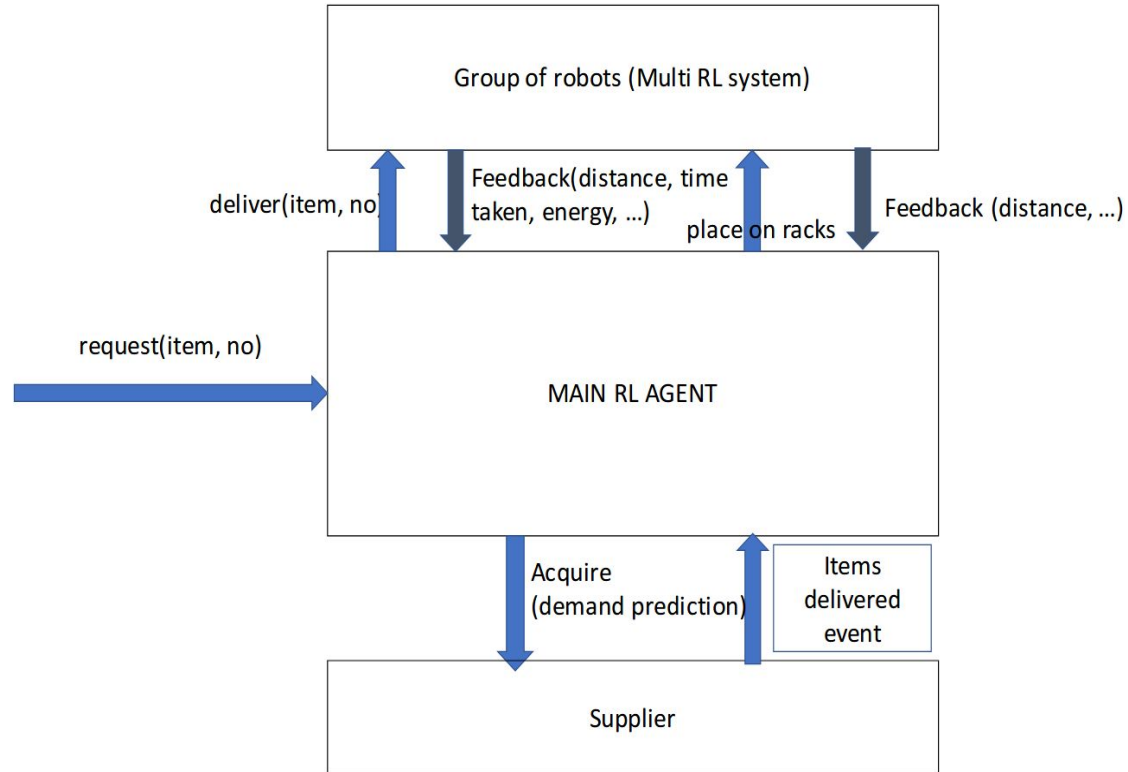
Figure 1: Warehouse (Top View)

Possible Solution Approaches and Why?

- Reinforcement Learning
 - Works very well in dynamic environments
 - Complex function approximators have shown to be effectively useful in RL too (Deep Reinforcement Learning)
 - Simulations are easy to build; Possible to extend
- Multi-Agent Systems
 - Can take care of huge scale in a distributed manner.

Solution Architecture

- One agent handling the demand and acquisition
- A set of robotic agents handling placement and supply within warehouse
- Fully Cooperative Multi-Agent Environment
 - Coordination
 - Working together to achieve a bigger goal
- Reinforcement Learning
 - Rewards
 - Penalties



States, Actions and Rewards

- Work in progress
- States
 - Items in the warehouse and their locations, robots and their locations
- Actions
 - `acquire(item, #)`, `pick(item)`, `drop(item)`
- Rewards
 - throughput, profit of sale, penalty for no-stock or overshoot of stocks

Current Work

Implementing simplistic case of only the demand and acquisition.

- A Storage of maximum 50 bags of type T, say wheat bags
- Demand: Sampled from Poisson Distribution
 - Different means for each day (to maintain a weekly pattern)
- Feedback:
 - Profit per bag
 - Penalty for ordering bags more than capacity
 - Penalty if not being able to fulfill customer's demand
- Question
 - Policy (how many bags to acquire in a day?)

State, actions...

- State
 - Tuple- (Number of bags present today N , WeekDay #)
- Action
 - Number of bags (say A) to be ordered from Supplier
- Next State: $N - M + A$
 - where M is the Number of bags sold that day

```
State: (18, 6)   Action: 9   Demand: 13   NextState: (14, 7)
```

```
State: (36, 7)   Action: 16   Demand: 19   NextState: (33, 1)
```


Model-free Q-table construction

- Q-table
 - Matrix of all possible States vs Actions
- States
 - (0-50, 1-7)
- Actions
 - 0-50
- Basically, 18207 entries

Action State	0	1	2	3	49	50
(0,1)									
(0,2)									
(0,3)									
..					...				
(25,3)									
(25,4)									
..									
..									
(50,7)									

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a')$$

Results

- Demand is always met.
- Next State is generally having close to 50 Bags.
- Agent seems to have learnt to keep the warehouse fully packed.

State: (13, 3)	Action: 29	Demand: 11	NextState: (31, 4)
State: (31, 4)	Action: 3	Demand: 9	NextState: (25, 5)
State: (25, 5)	Action: 7	Demand: 14	NextState: (18, 6)
State: (18, 6)	Action: 9	Demand: 13	NextState: (14, 7)
State: (14, 7)	Action: 46	Demand: 11	NextState: (49, 1)
State: (49, 1)	Action: 15	Demand: 12	NextState: (50, 2)
State: (50, 2)	Action: 7	Demand: 9	NextState: (48, 3)
State: (48, 3)	Action: 46	Demand: 8	NextState: (50, 4)
State: (50, 4)	Action: 2	Demand: 6	NextState: (46, 5)
State: (46, 5)	Action: 26	Demand: 11	NextState: (50, 6)
State: (50, 6)	Action: 0	Demand: 14	NextState: (36, 7)
State: (36, 7)	Action: 16	Demand: 19	NextState: (33, 1)
State: (33, 1)	Action: 46	Demand: 7	NextState: (50, 2)
State: (50, 2)	Action: 10	Demand: 7	NextState: (50, 3)
State: (50, 3)	Action: 4	Demand: 14	NextState: (40, 4)
State: (40, 4)	Action: 1	Demand: 7	NextState: (34, 5)
State: (34, 5)	Action: 9	Demand: 12	NextState: (31, 6)
State: (31, 6)	Action: 11	Demand: 16	NextState: (26, 7)
State: (26, 7)	Action: 39	Demand: 16	NextState: (49, 1)
State: (49, 1)	Action: 16	Demand: 7	NextState: (50, 2)
State: (50, 2)	Action: 2	Demand: 6	NextState: (46, 3)
State: (46, 3)	Action: 14	Demand: 14	NextState: (46, 4)
State: (46, 4)	Action: 22	Demand: 4	NextState: (50, 5)
State: (50, 5)	Action: 0	Demand: 10	NextState: (40, 6)
State: (40, 6)	Action: 45	Demand: 11	NextState: (50, 7)
State: (50, 7)	Action: 40	Demand: 18	NextState: (50, 1)
State: (50, 1)	Action: 9	Demand: 6	NextState: (50, 2)
State: (50, 2)	Action: 32	Demand: 10	NextState: (50, 3)
State: (50, 3)	Action: 30	Demand: 11	NextState: (50, 4)
State: (50, 4)	Action: 0	Demand: 8	NextState: (42, 5)

Extending it with robots

TBD

Thank You

