Resource Optimization in Supply Chain

- A Case Study on Inventory Management

Lei Song

MSRA Industry Innovation Center

Joint work with (alphabetically)
Jiang Bian, Yuandong Ding, Wei Jiang, Yan Jin, Xianliang Yang, Li Zhao, Chuheng Zhang

Outline

- Optimizations in Supply Chain – An Overview
- A Case Study on Inventory Management
- More Challenges

Resource Optimizations in Supply Chain

Scenarios:

Store Mgmt.
- Space allocation
- Manpower scheduling

Operation
- Category mgmt.
- Dynamic pricing
- Markdown

Supply Mgmt.
- Inventory mgmt.
- Network optimization

Delivery
- Delivery scheduling
- Routing optimization

Marketing
- Promotion opt.
- Marketing resources allocation

Challenges

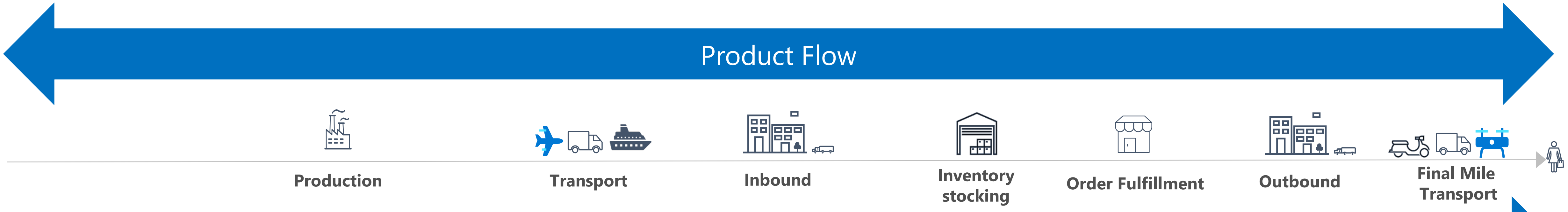
Scalability
- Problems are of large scale in practice: many SKUs, facilities, resources etc.

Coopetitive
- Co-existence of cooperative and competitive behaviors

Environment
- Dynamic environment in terms of both network connection and future supply/demand

Objective
- A trade-off between customer service level and operational cost

Case Studies – An Overview



Digital hotspots and scenarios

Application	Order Scheduling	Resource Repositioning	Inventory Management	Routing & Order Fulfillment
Problem	Scheduling orders to factories to minimize total production cost	Reposition resources efficiently to accommodate imbalanced and dynamic supply and demand.	Optimize replenishing decisions to balance supply and demand.	Optimize on-call stops routing and fulfillment strategy to minimize total routing cost.
Challenge	Fluctuant Env.	Complex Interactions	Scalable RL	Large Action Space
Technology	Multi-agent / Hierarchical / Contextualized Reinforcement Learning			
	Spatial-Temporal Forecasting			

MARO – A Platform for Supply Chain Optimization

MARO – https://github.com/microsoft/maro

Microsoft

MARO Service

For customers' domain developers (*no ML experience required*)

CIM | Inv. Mngmt. | Dynamic Pricing | Routing | ...

Data Service Interface | State Extraction Interface | Reward Shaping Interface

Model Optimization Service | Visualization Tool | ...

MARO Platform

For customers' ML developers & data scientists

HiGraSS (*Distirbuted*)

- CIM
- Bike Rebalancing
- Portfolio Management
- ...
- Customized Environments

RL Toolkit (*Distributed*)

- DQN
- A2C
- MADDPG
- ...
- Customized Algorithms

MARO Framework

For broad applied scientists & researchers

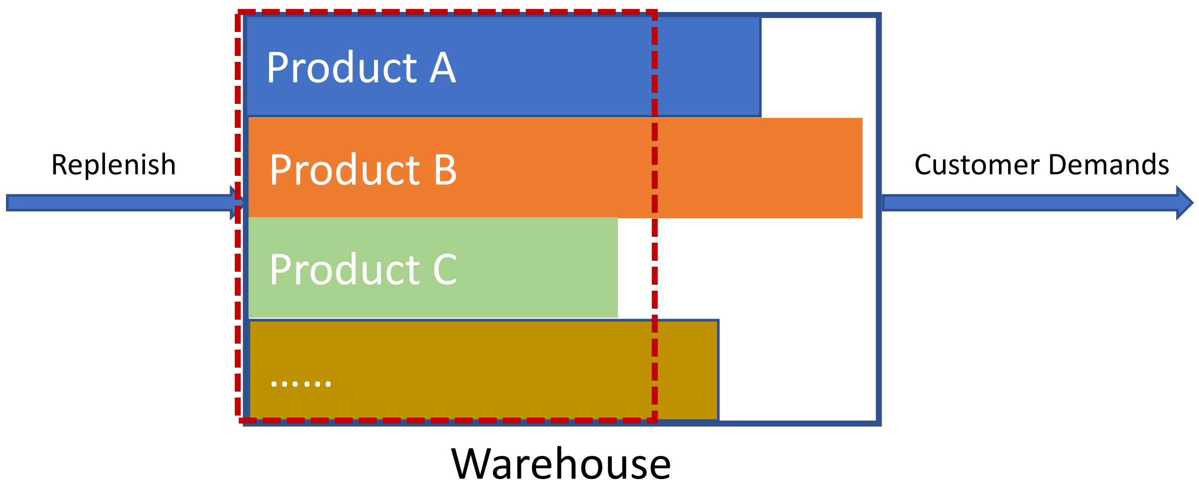
HiGraSS (*Single Instance*)

- Business Engine
- Graph Data Model
- Event Buffer
- ...

RL Toolkit (*Single Instance*)

- Agent Assembler
- Algorithm Abstraction
- Experience Pool
- ...

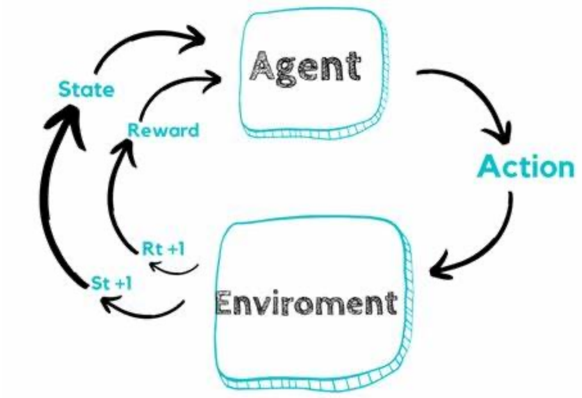
Inventory Management



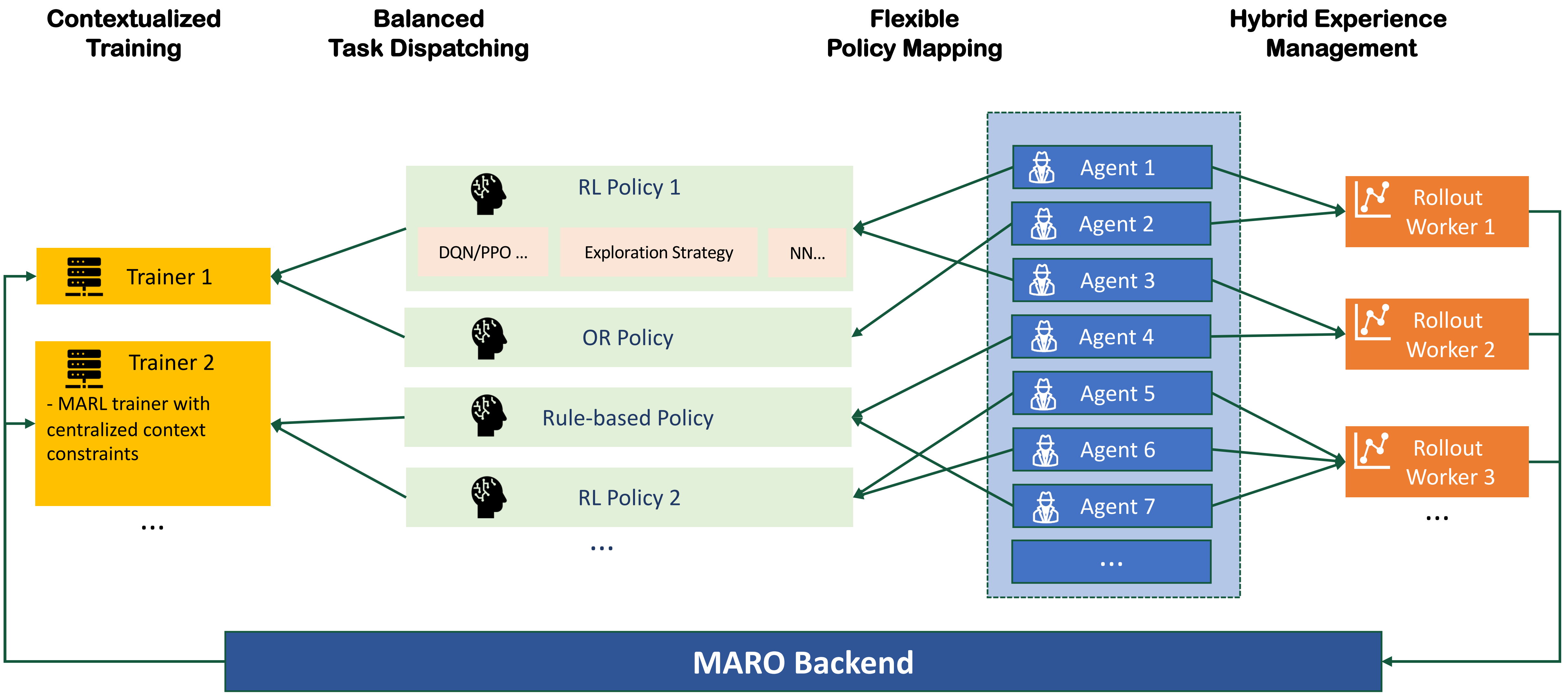
- The objective is to maintain an optimal inventory level for each product such that the **overall profit** is maximized
    - **Overstock**: holding cost is high
    - **Understock**: Loss of sales opportunities
- Dynamics
    - Customer demands
    - Supply flucturations
    - Others e.g., leading time
- Complex interaction
    - **Coorperation**: team reward
    - **Competition**: shared resource (storage capacity, budget, distrubution capacity etc.)
- Massive products
    - A normal supermarket may have more than **20K** products on shelf
    - Much more in e-commerce platforms (**millions**)

Inventory Optimization via MARL

- Why MARL
  - Flexibility – insensitive to changes of product portfolios
  - Easy to model – behaviors of each individual product is simple
  - Easy to deploy – centralized training decentralized execution (CTDE)
  - ……
- Challenges for applying MARL in practice
  - Scalability
  - Generalization
  - Non-stationary
  - Credit assignment
  - ……



Infrastrature: Distributed Training



Algorithm: Massive MARL with Limited Resource

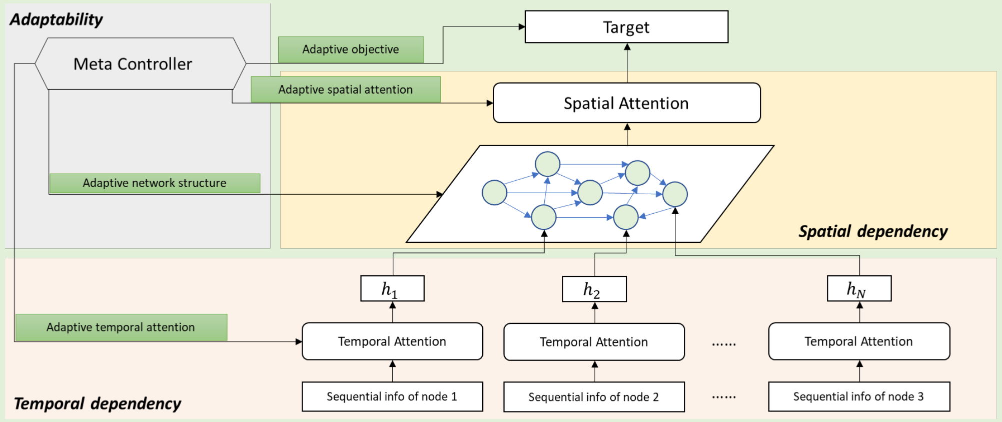
Challenges

Supply and Demand Uncertainties

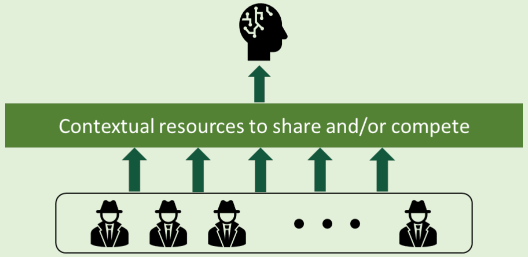
Replenishment Scheduling under Limited Warehouse Capacity

Solutions

Spatial-temporal forecasting based on graph neural network



Independent Learning via decoupling shared resource

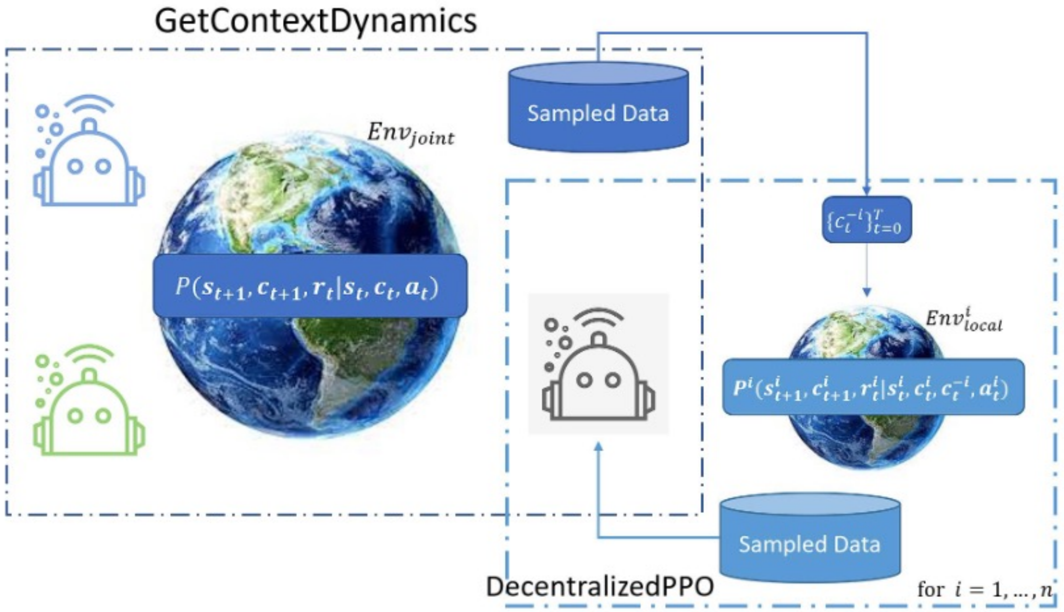


Training of i-th agent is independent from others given resource dynamics c_t^{-i}

$$max_{\pi^i}\hat{\eta}^i(\pi^i,\pi^i_{old},\pi^{-i}_{old}) = \mathbb{E}_{(c_t^{-i})\sim\mathcal{T},\pi^i_{old},\pi^{-i}_{old}}\mathbb{E}_{(s_t^i,a_t^i,c_t^i)\sim\mathcal{T}^i,\pi^i}[\sum_{t=0}^{\infty}\gamma^t r_t^i]$$

Resource dynamics learning using policies from the previous steps

Accelerate Training via Local Simulators



Assume each agent has insignificant impact on the overall environment

- Run the **joint simulator** to collect resources trajectories (contexts)
- Use the sampled contexts to initialize **local simulators** – each for an agent
- Use data collected from local simulators to train a policy

$$max_{\pi^i}\hat{\eta}^i(\pi^i, \pi^i_{old}, \pi^{-i}_{old}) = \underbrace{\mathbb{E}_{(c_t^{-i})\sim\mathcal{T},\pi^i_{old},\pi^{-i}_{old}}}_{\text{Resource dynamics learning using policies from the previous steps}} \overbrace{\mathbb{E}_{(s_t^i,a_t^i,c_t^i)\sim\mathcal{T}^i,\pi^i}[\sum_{t=0}^{\infty}\gamma^t r_t^i]}^{\text{Training of } i\text{-th agent is independent from others given resource dynamics } c_t^{-i}}$$

Inventory Management – Evaluation Results

Targeted customers: retailers / wholesalers

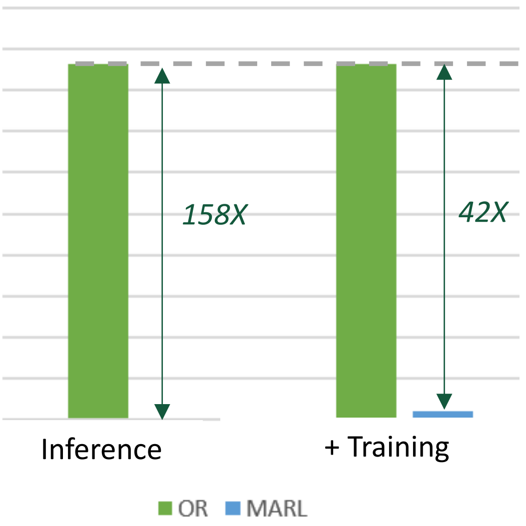
Data Requirements

- Transaction data
- Historical replenishing data
- Inventory level of all SKUs in each facility
- Nice to have: SKU price, cost, fulfillment cost, warehouse volume, SKU volume etc.

Walmart open data

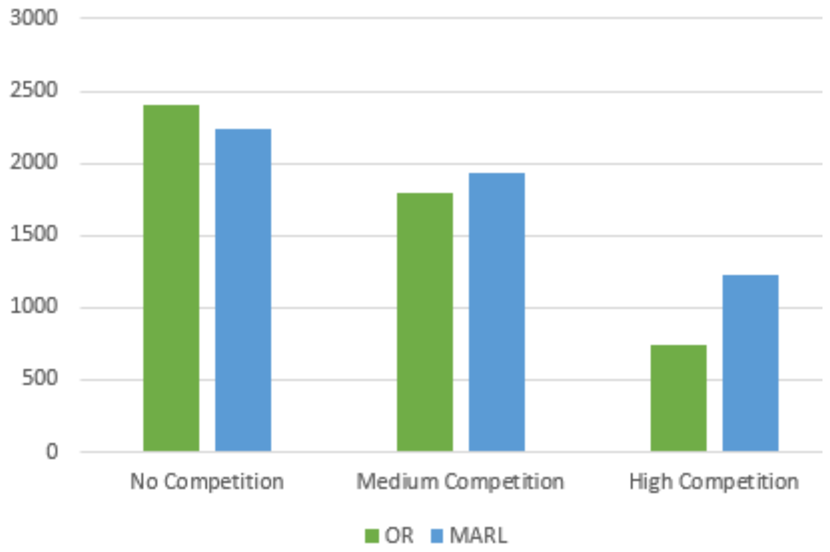
- 20-1000 SKUs
- under different sampled competition environment

Efficiency



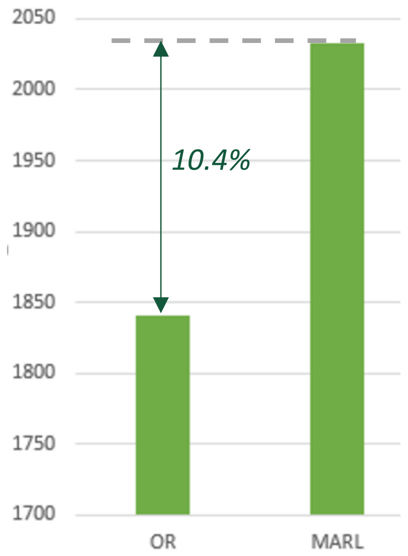
Time cost

Adaptivity



Performance under various competition conditions (warehouse capacities)

Effectiveness



Average profits under different scenarios

Experiment Settings

Using warehouse capacities as the main shared resource;
The objective is to maximize **total profits**

DC

- VLT：Real data
- Order Cost: 100¥/order
- Fulfillment Capacity: 100 orders/day
- Transportation cost: 1¥/unit/day
- Sufficient supply

Store A

- Sale, Price, Cost：Real data
- Out-of-stock loss：5%
- Holding cost：0.1¥/unit/day

Store B

- Sale, Price, Cost：Real data
- Out-of-stock loss：5%
- Holding cost：0.1¥/unit/day

Store C

- Sale, Price, Cost：Real data
- Out-of-stock loss：5%
- Holding cost：0.1¥/unit/day

Store Capacity：
SUM(sale_mean*VLT*2)

Exp. One – **Sufficient Capacity**

Store Capacity：
SUM(sale_mean*VLT*0.5)

Exp. Two – **Capacity Shortage**

Simulation Period：2021/01/01-2021/06/30

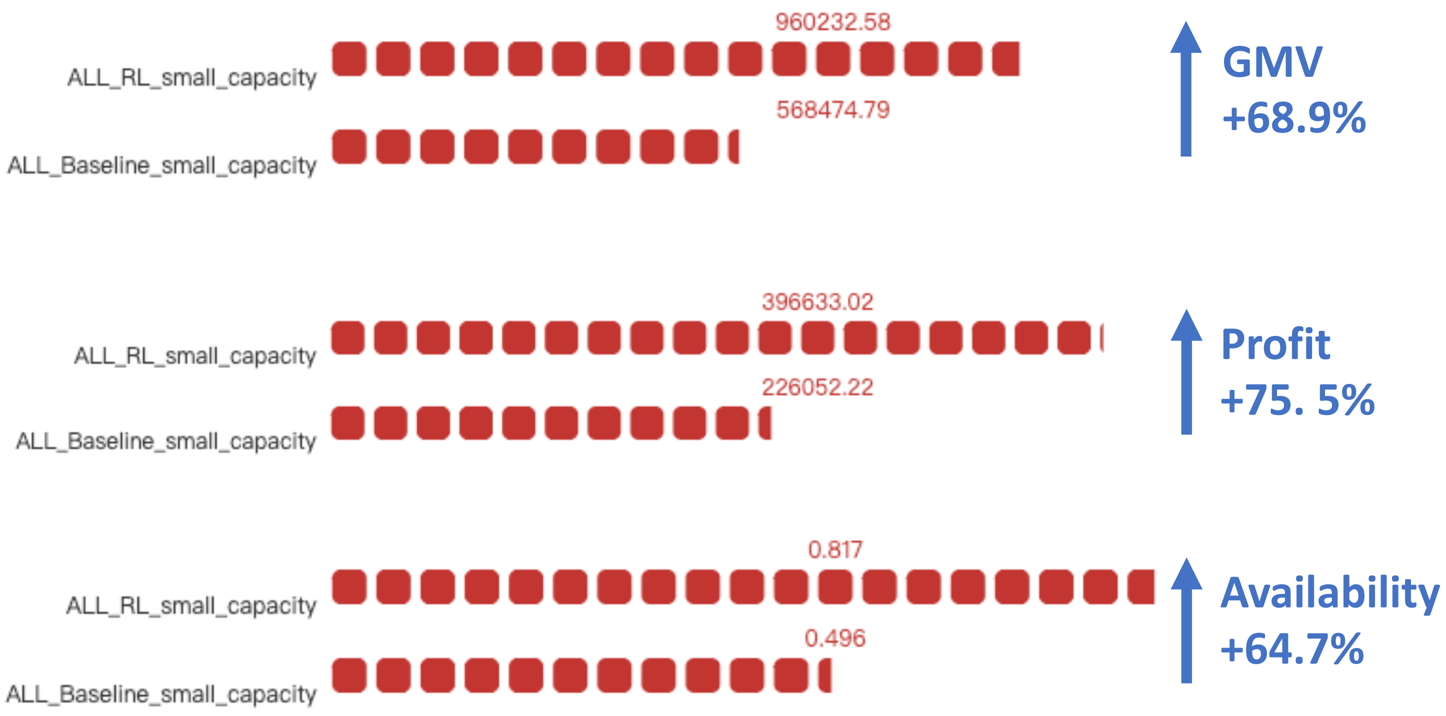
Supply Chain Simulator

Exp. One – When Capacity is Large Enough

- Comparing to baseline policy, RL policy takes all cost/profits into consideration;
- According to current setting, a better policy is to decrease the **service level** which can decrease **inventory cost** considerably;
- While maintaining a higher turnover rate, we reach a policy that sacrifices **GMV** a bit while significantly increases the **total profit**.

Exp. Two – When Capacity is Limited

- In case of capacity shortage, baseline policy does not consider **coordination** among all SKUs, hence will cause extra costs for dealing with overflows;
- In contrast, RL policy views the problem as a **global optimization** problem and can reach a policy that allocates more resources to SKUs that have higher profits.



More Challenges

- Theorectically
  - CTDE – efficiency vs. optimality
  - VDN + IGM
- Algorithmically
  - Scalability – beyonds tens of thousands of products
  - Generalization – across different products/stores
- Scenarios
  - Multi-echelon networks
  - A complex resource sharing graph
    - E.g. , multiple types of resources shared by different groups of products

Thank You

Q&A