Adaptive Task Planning for Large-Scale Robotized Warehouses

Dingyuan Shi[1], Yongxin Tong[1], Zimu Zhou[2],

Ke Xu[1], Wenzhe Tan[3], Hongbo Li[3]

1 Beihang University

2 Singapore Management University

3 Geekplus Technology Co., Ltd.

Outline

- **Background & Motivation**
- **Problem Statement**
- **Our Solutions**
- **Experiments**
- **Conclusion**

Outline

- Background & Motivation
- Problem Statement
- Our Solutions
- Experiments
- Conclusion

Background & Motivation

CNBC | MARKETS | BUSINESS | INVESTING | TECH | POLITICS | CNBC TV | WATCHLIST | CRAMER | PRO

Alibaba, JD set new records to rack up record $115 billion of sales on Singles Day as regulations loom

PUBLISHED THU, NOV 12 2020•12:19 AM EST | UPDATED THU, NOV 12 2020•8:15 AM EST

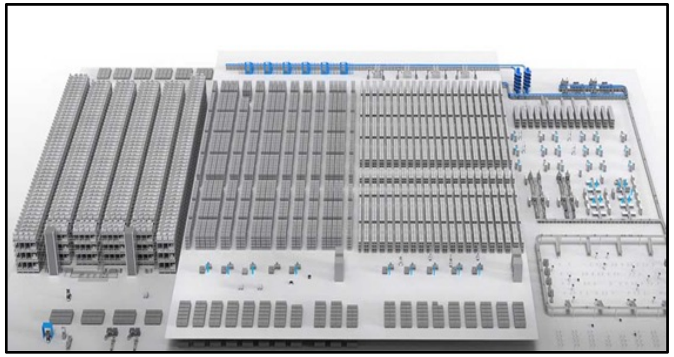
The boom of e-commerce has stimulated enormous logistic demands

Background & Motivation

- **Some companies and their products/services**

amazon

CAINIAO 菜鸟



Geek+

GreyOrange
Fulfilling the Future in Real Time™

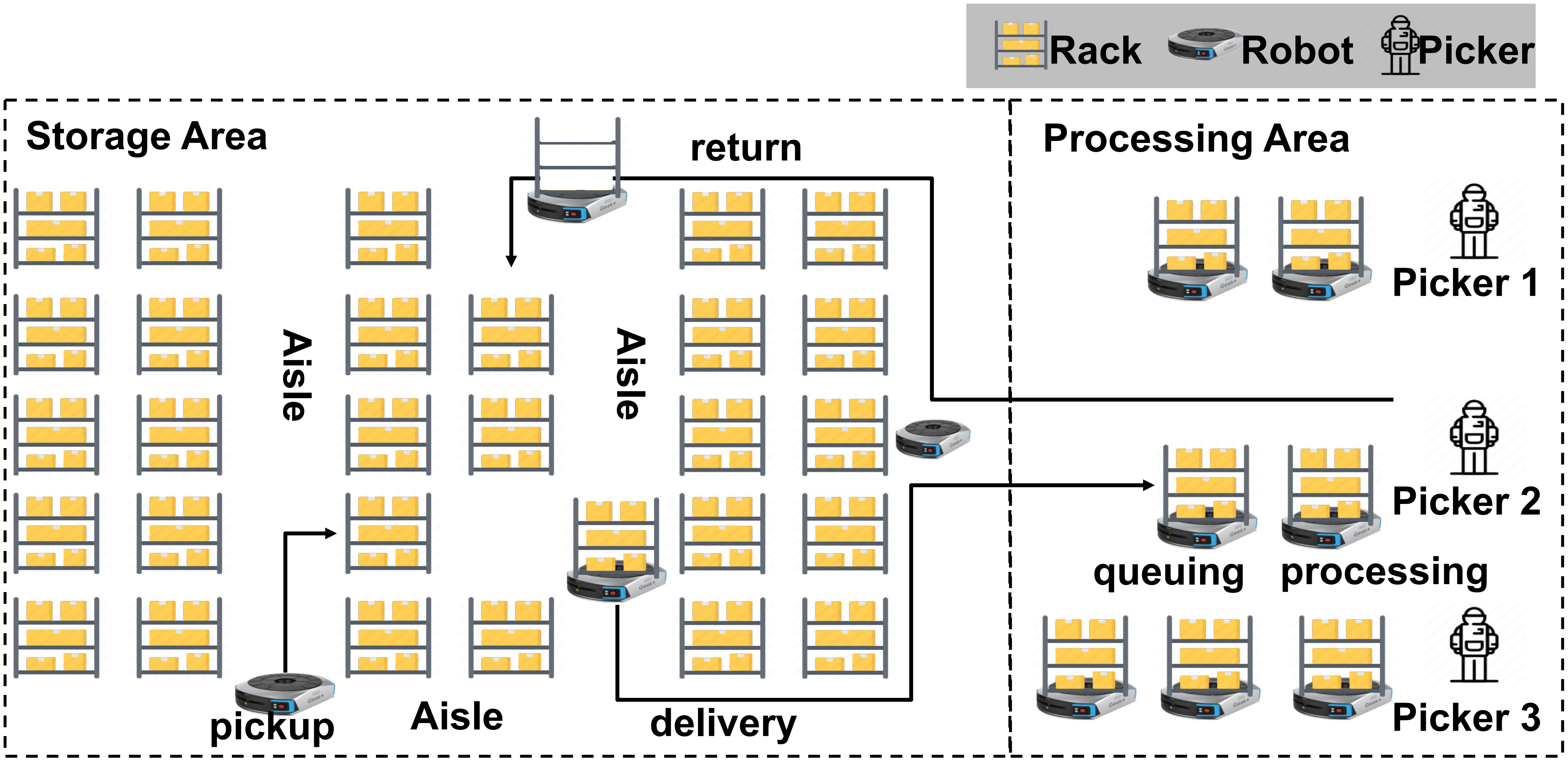
HITACHI
Inspire the Next

JDL 京东物流

Robotized warehouses are expected to improve the performance

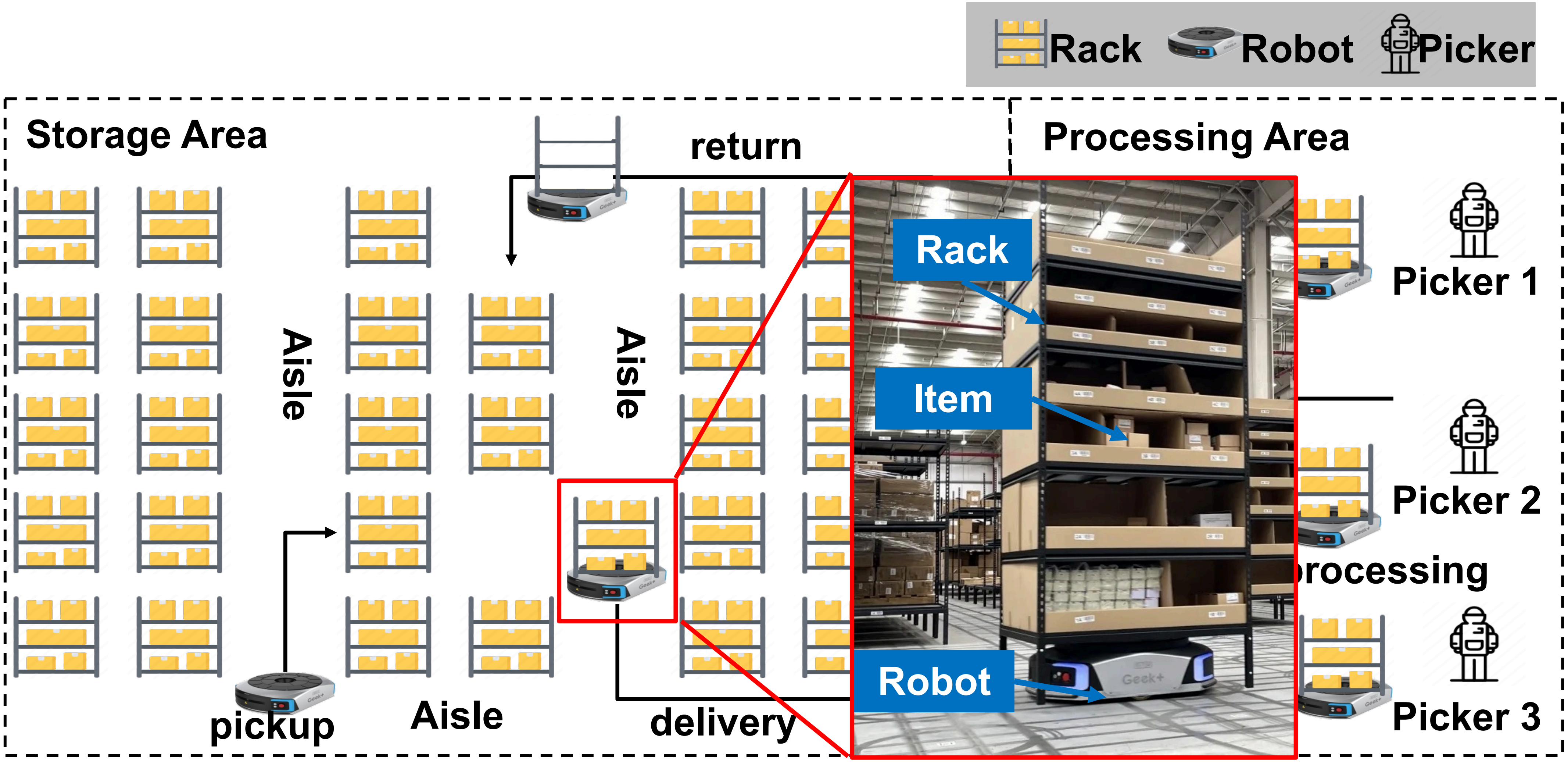
Background & Motivation

- A typical robotized warehouse



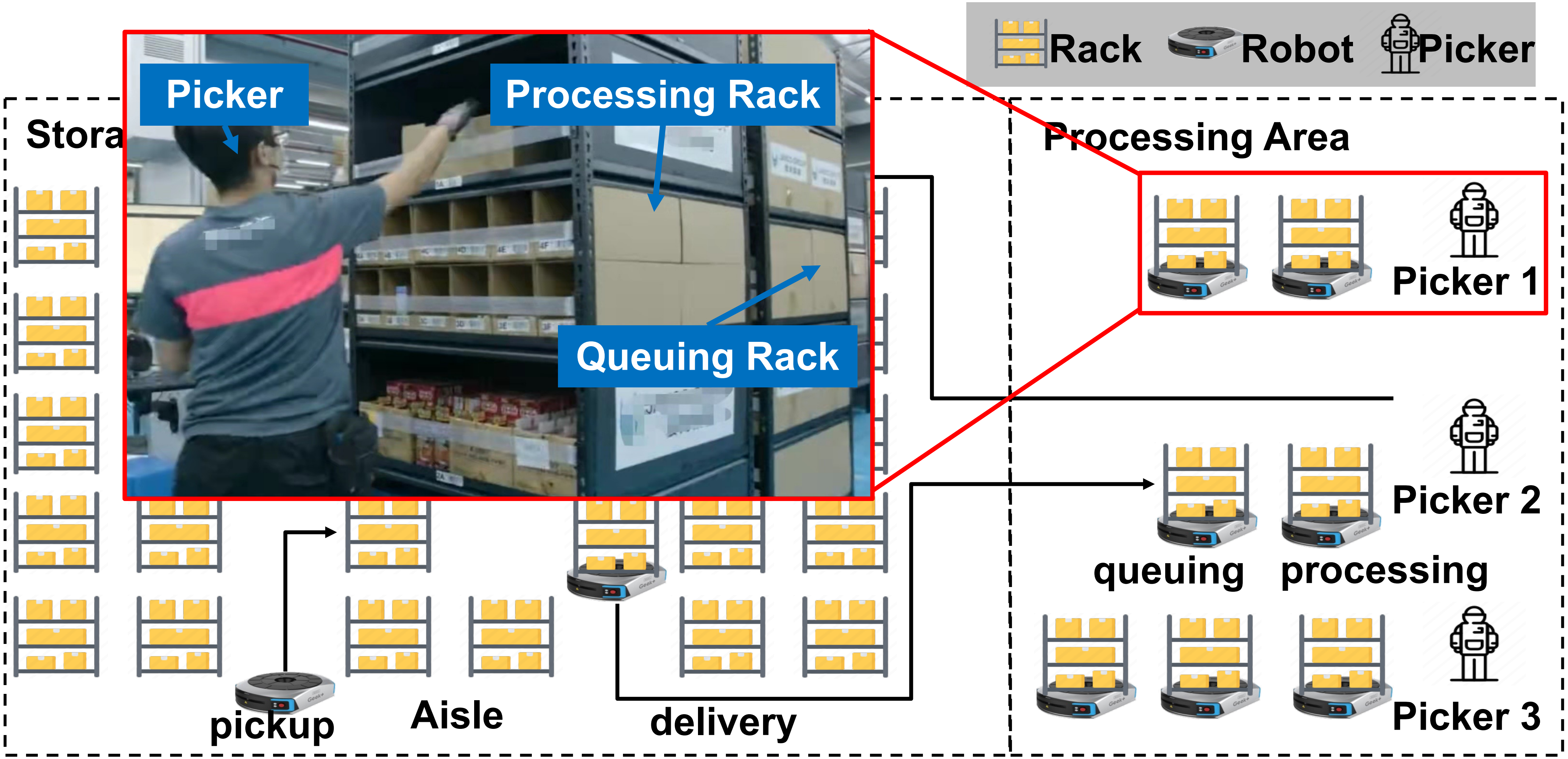
Background & Motivation

- A typical robotized warehouse



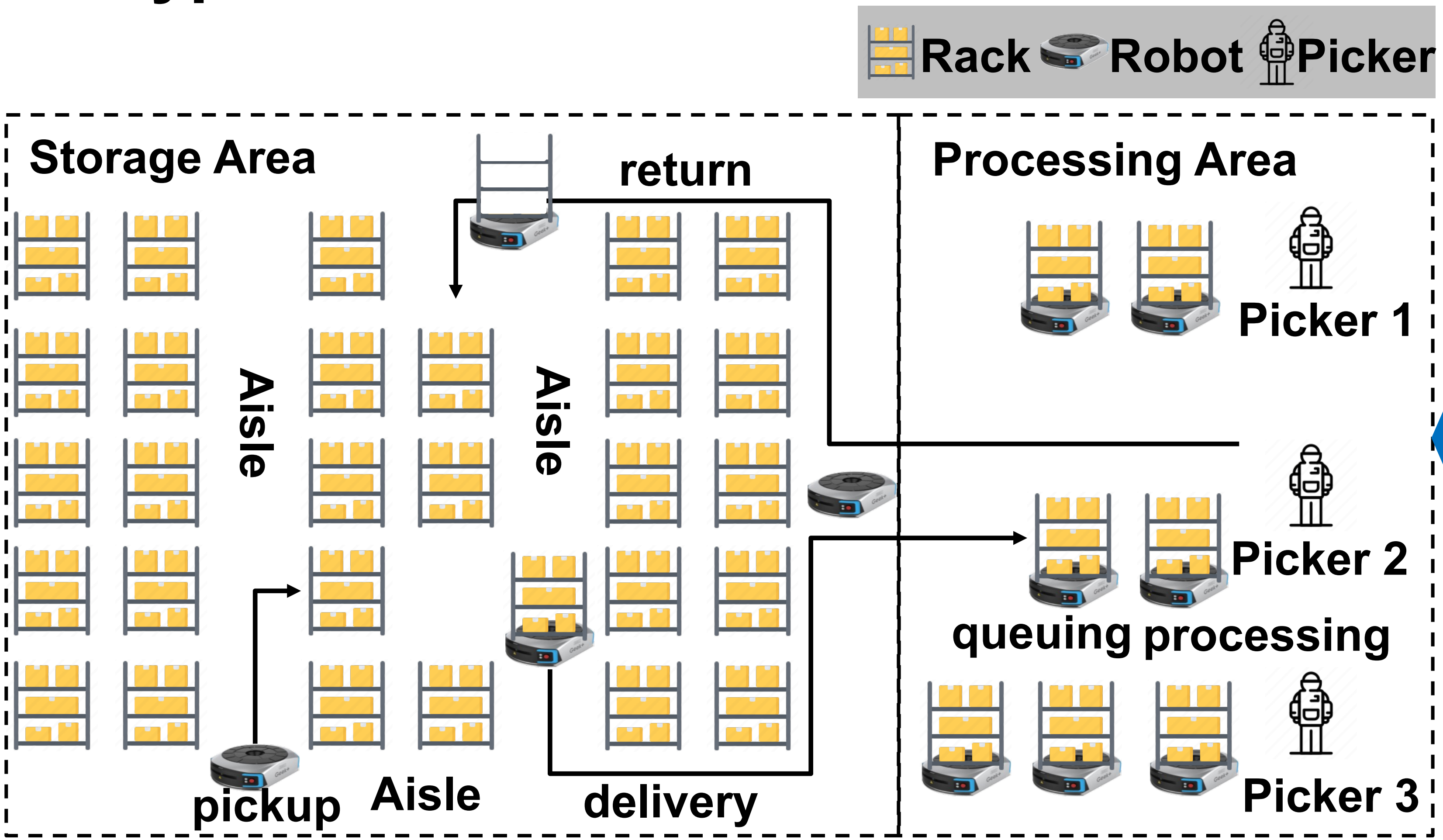
Background & Motivation

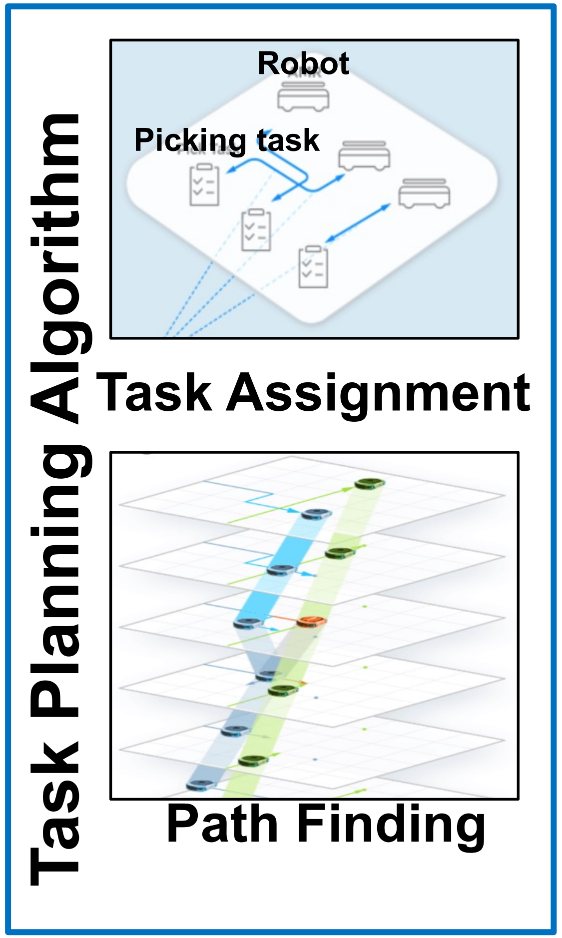
- A typical robotized warehouse



Background & Motivation

- **A typical robotized warehouse**





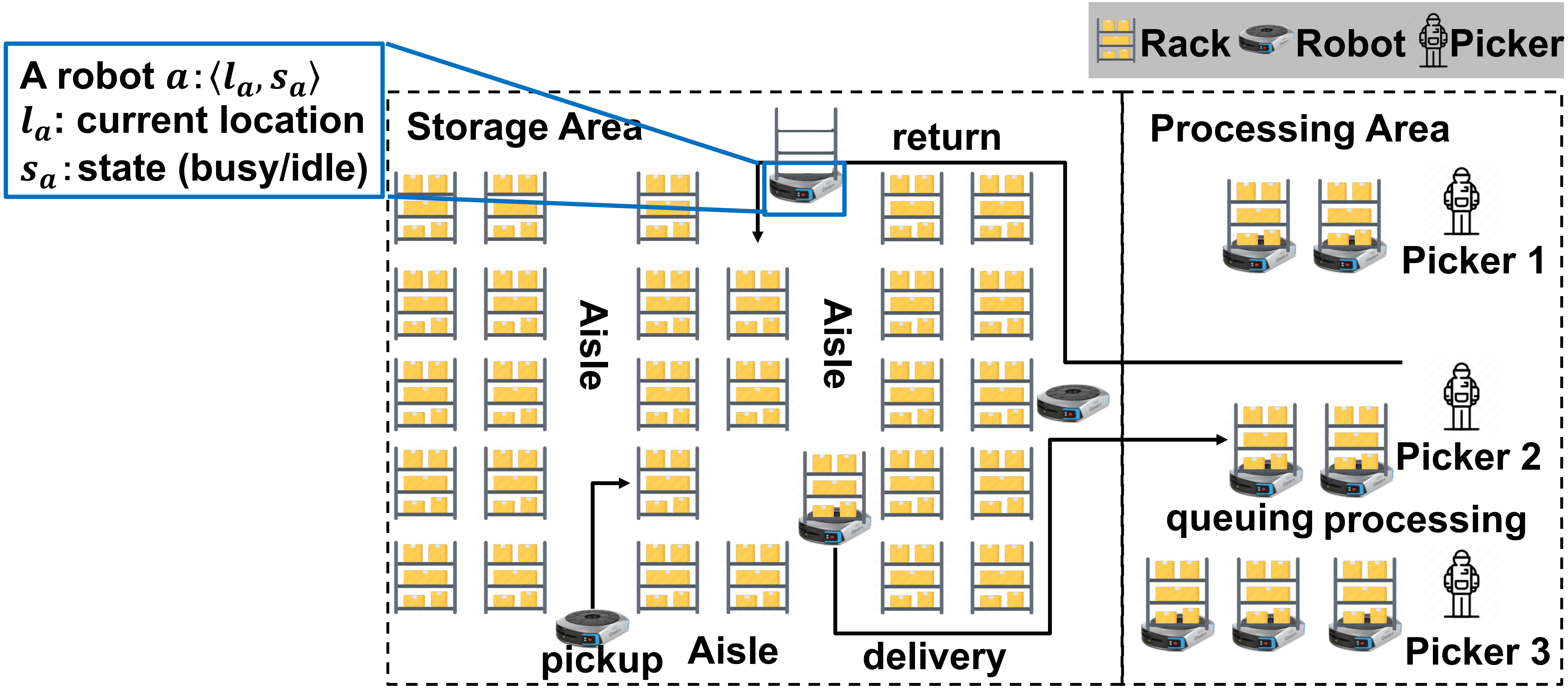
Task planning algorithms are the key to improve the efficiency

Outline

- Background & Motivation
- Problem Statement
- Our Solutions
- Experiments
- Conclusion

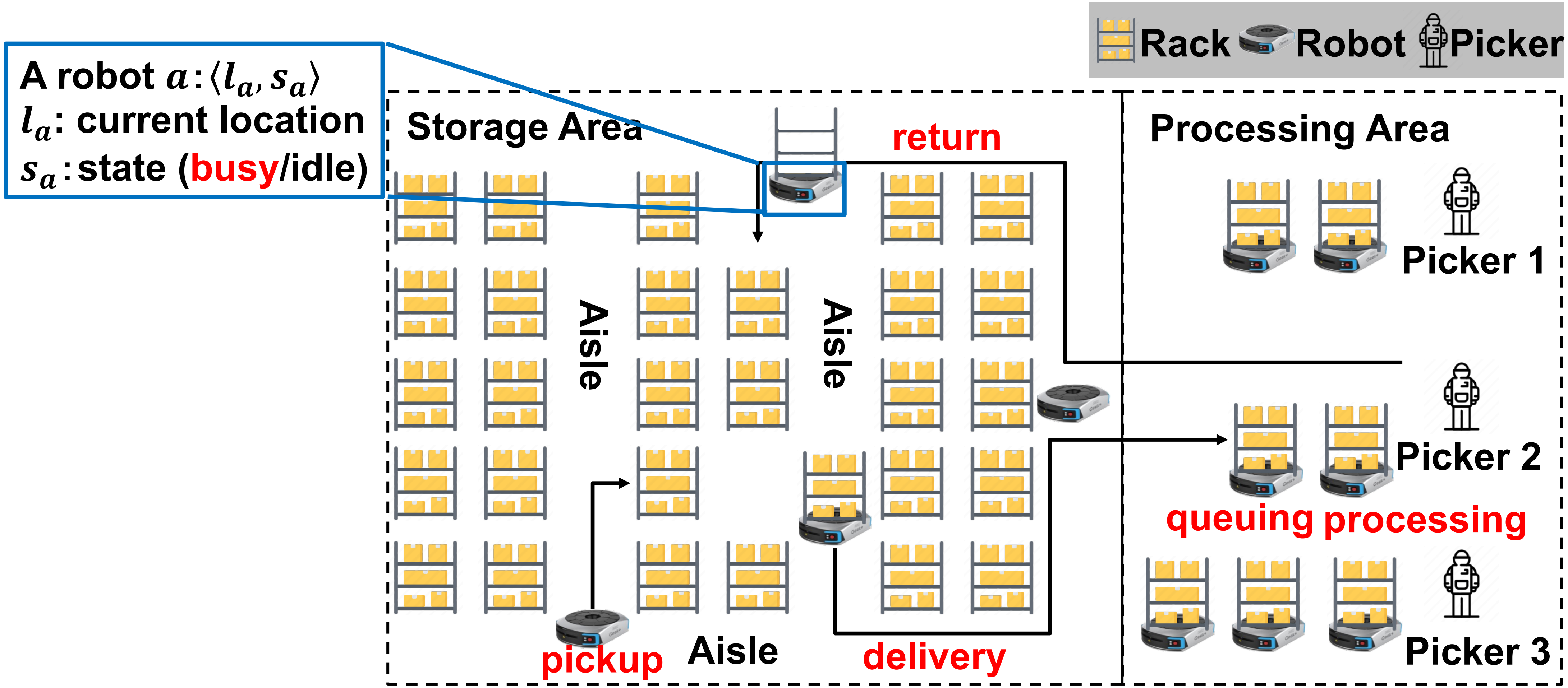
Problem Statement

- **Robotized Warehouse Scenario**



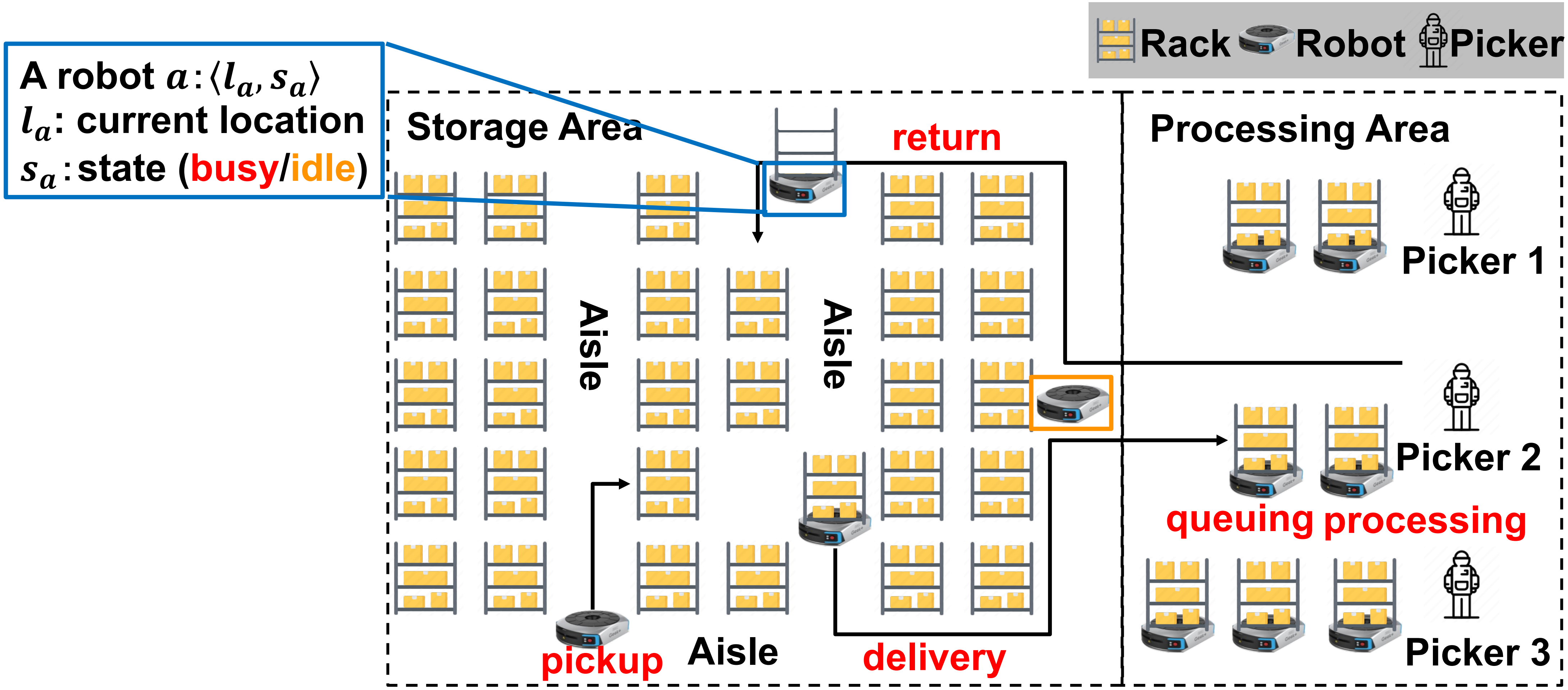
Problem Statement

- **Robotized Warehouse Scenario**



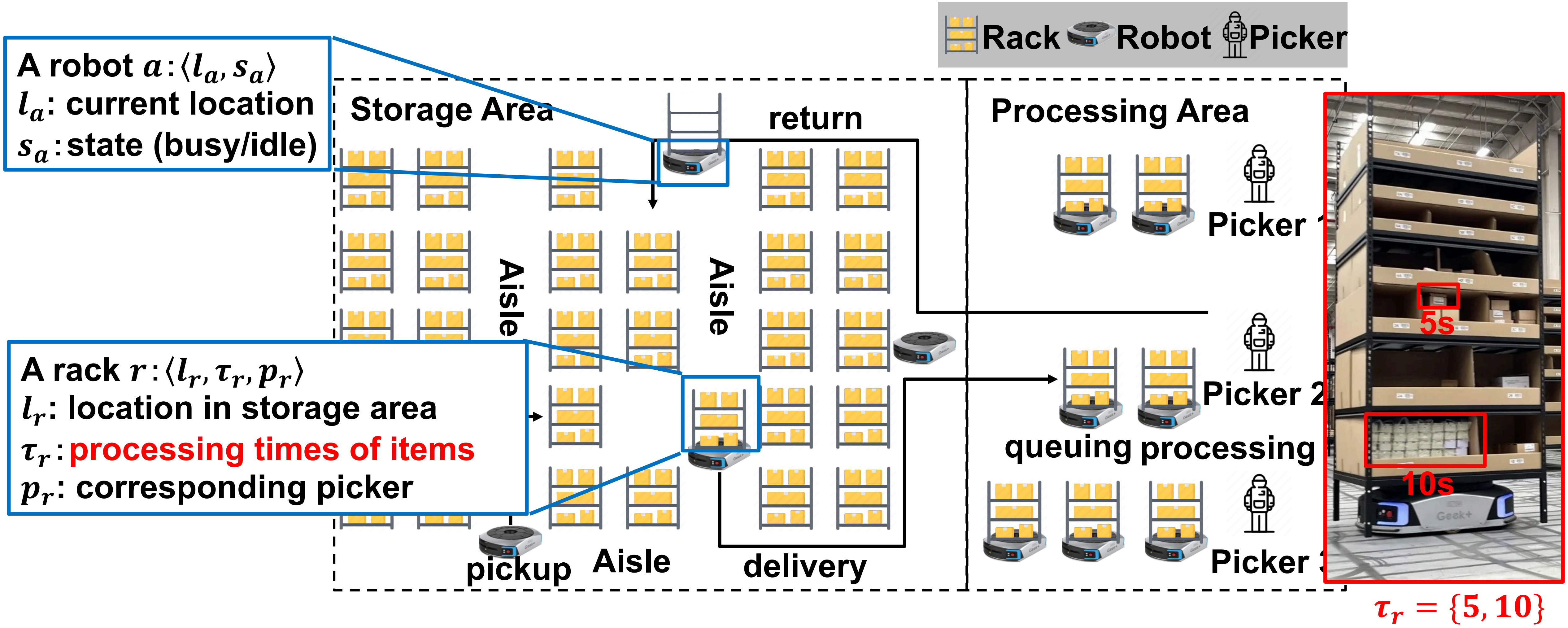
Problem Statement

- **Robotized Warehouse Scenario**



Problem Statement

- **Robotized Warehouse Scenario**

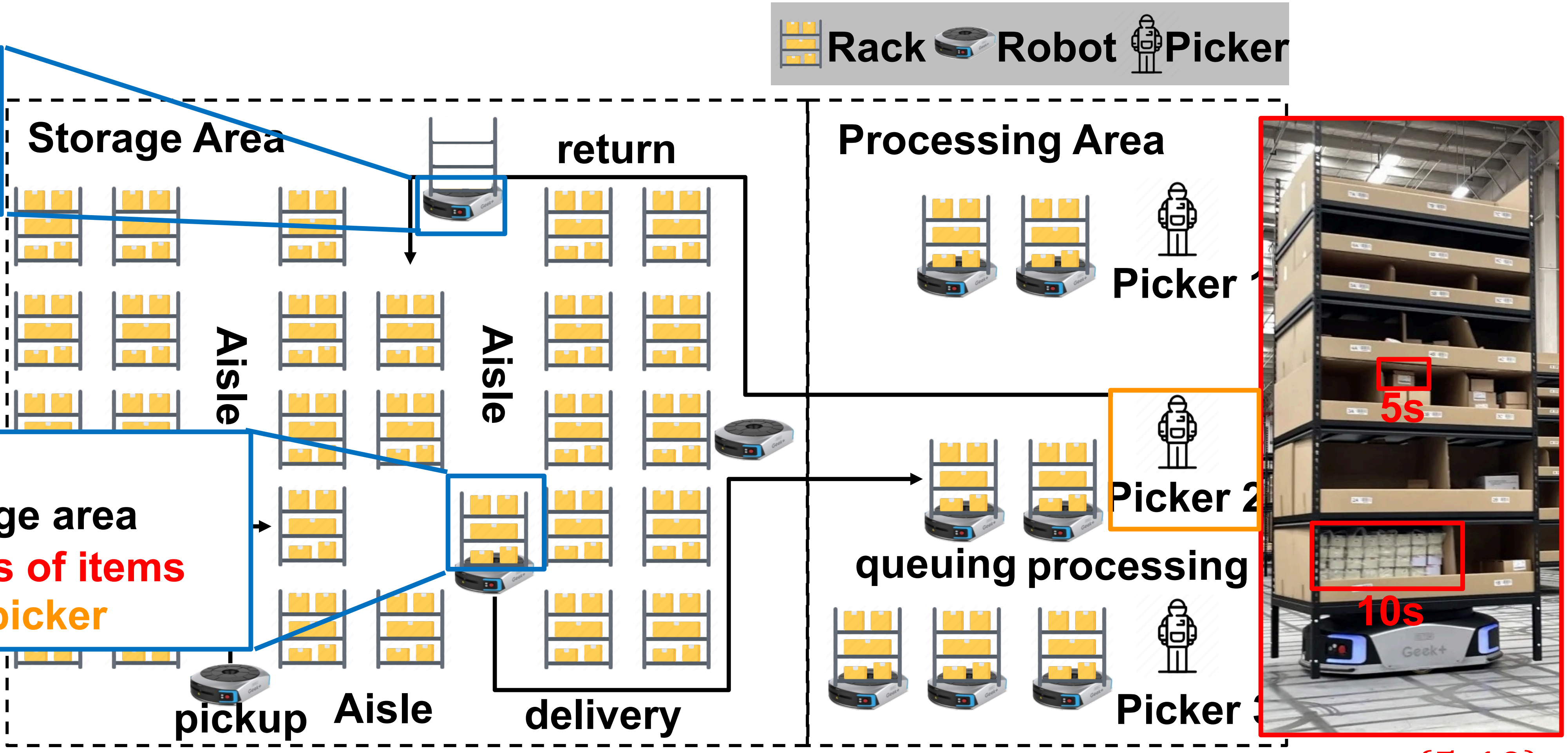


Note: Items emerge dynamically.

Problem Statement

- **Robotized Warehouse Scenario**

A robot a: $\langle l_a, s_a \rangle$
l_a: current location
s_a: state (busy/idle)

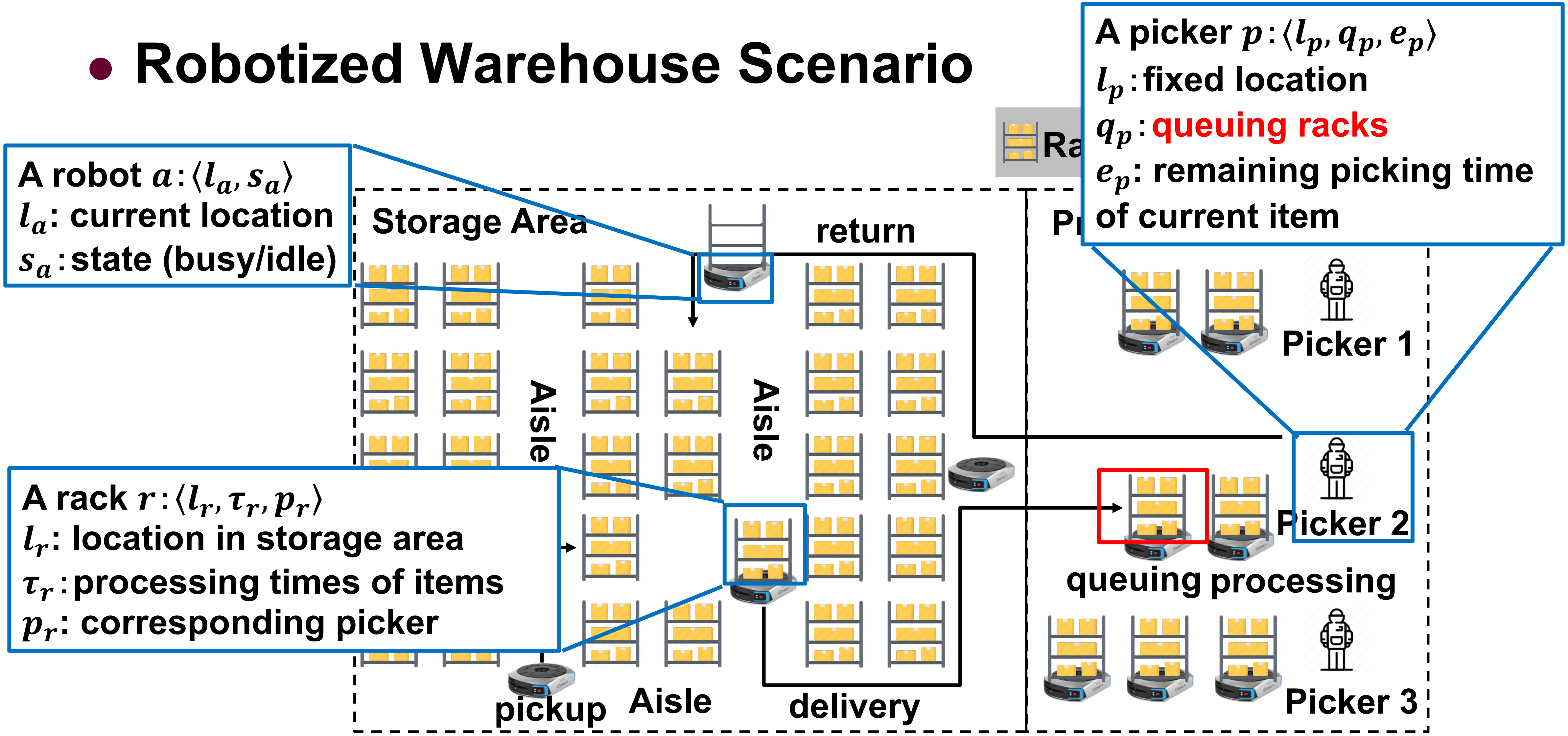


A rack r: $\langle l_r, \tau_r, p_r \rangle$
l_r: location in storage area
τ_r: processing times of items
p_r: corresponding picker

$\tau_r = \{5, 10\}$

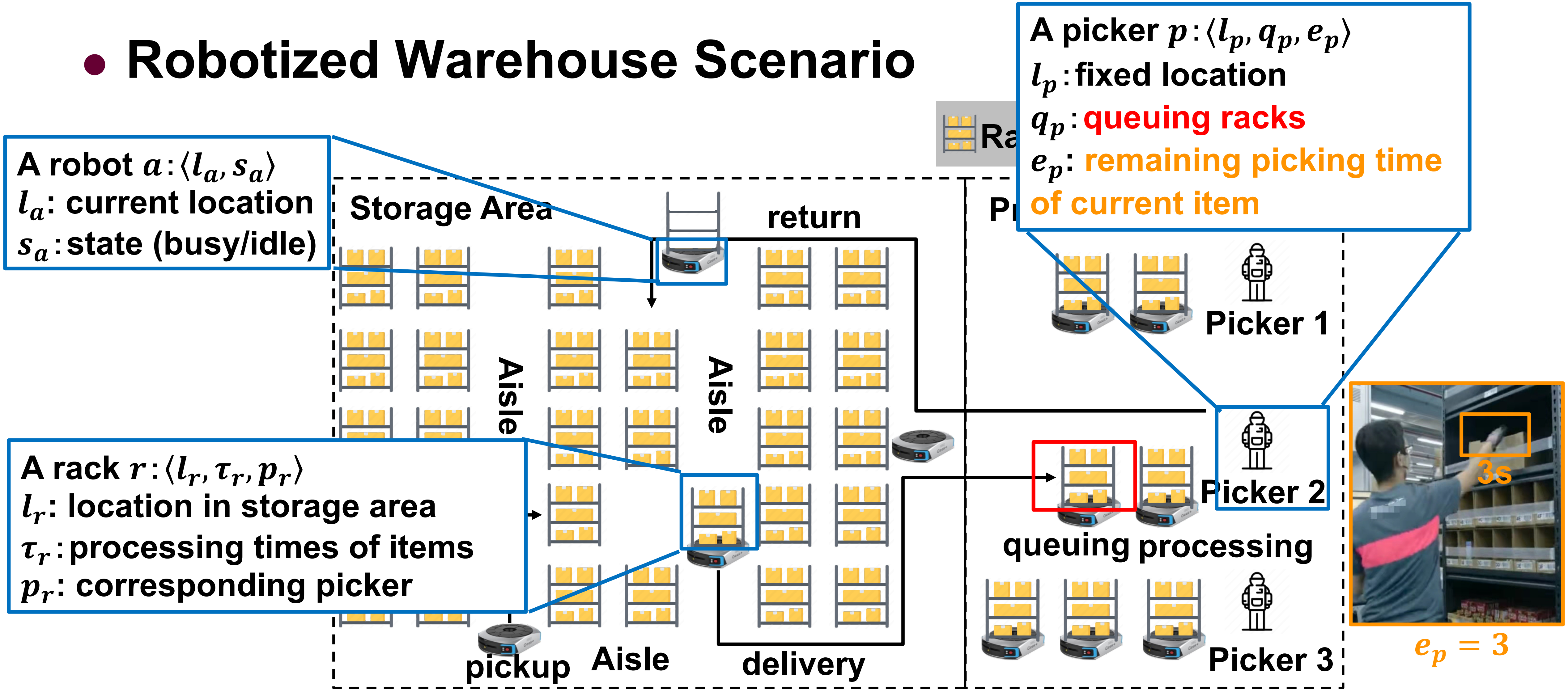
Problem Statement

- **Robotized Warehouse Scenario**



Problem Statement

- **Robotized Warehouse Scenario**



Problem Statement

- **Robotized Warehouse Scenario**

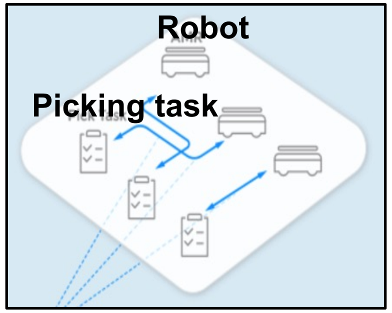
Calls at every timestep

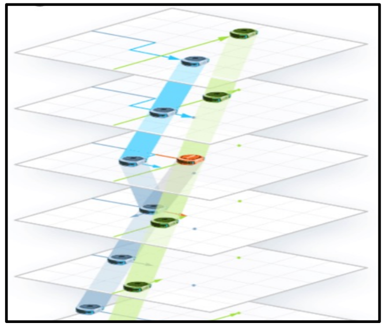
Rack
$\langle l_r, \tau_r, p_r \rangle$

Robot
$\langle l_a, s_a \rangle$

Picker
$\langle l_p, q_p, e_p \rangle$

Task Planning Algorithm





Input
Information of racks, robots, pickers

Output
Selection scheme：set of racks to be delivered
Paths: instructs robots to pickup deliver and return

Problem Statement

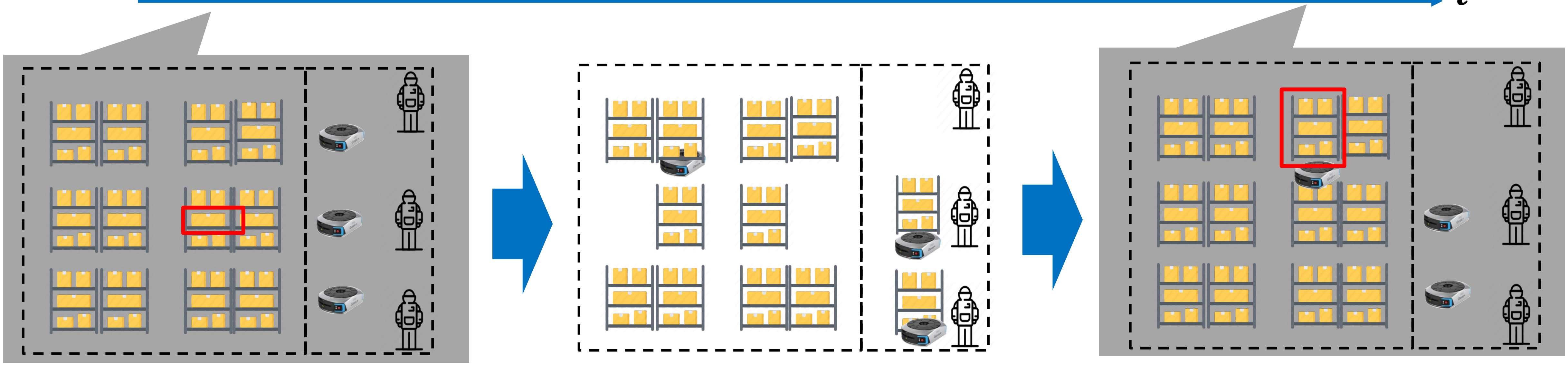
- **Optimization Goals: Minimizing the *Makespan***

Makespan: $M = t_e - t_b$

t_b: time of the first picking item emerges

t_e: time of the last rack returned

t



An item emerges, the rack requires delivering.

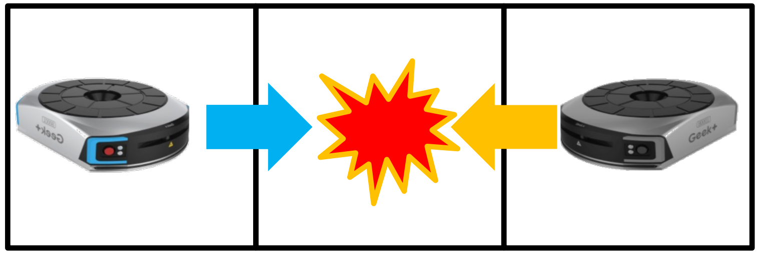
Processing by pickers and robots.

The last rack has been returned.

Problem Statement

- **Constraints: Conflict-Free**
  - **All paths for robots should be conflict-free**

- **Two types of conflict**
  - **Single-Grid Conflict**
    - **Two robots try to visit one grid at the same time, causing single-grid conflict.**

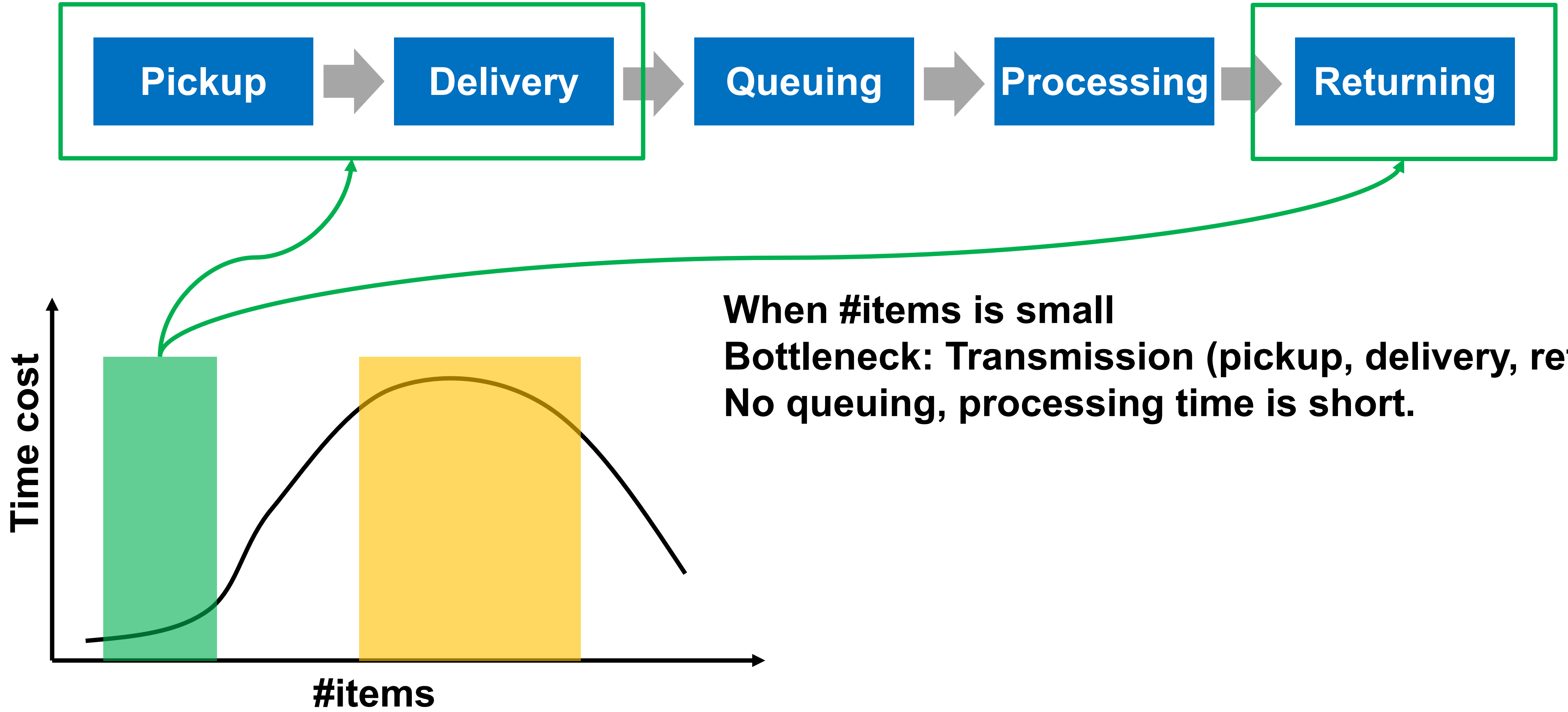


  - **Inter-Grid Conflict**
    - **Two robots try passing over each other, causing inter-grid conflict.**

Note: We model the space in a grid-based manner.

Problem Statement

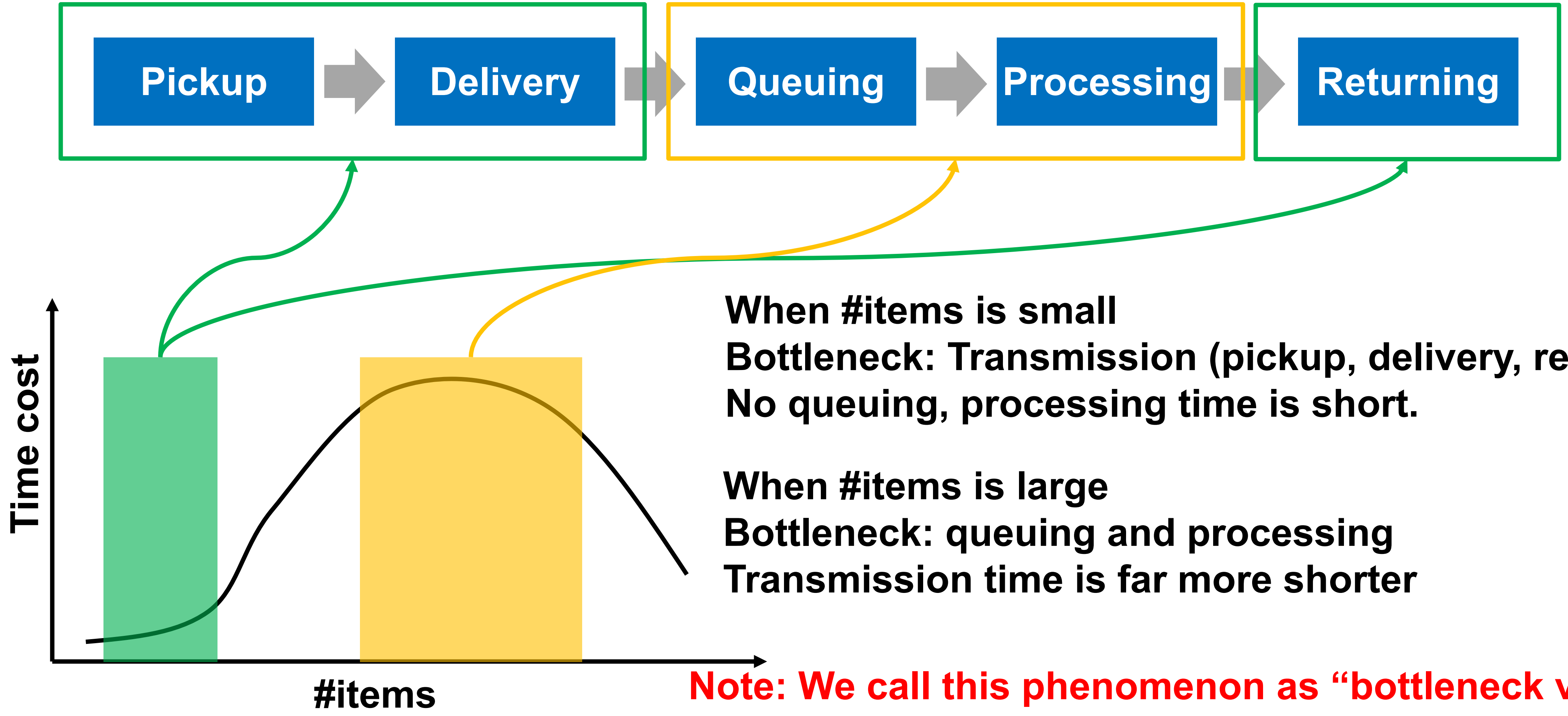
- **Challenges**
  - **Inflexible planning to time-varying item arrival**



When #items is small
Bottleneck: Transmission (pickup, delivery, returning)
No queuing, processing time is short.

Problem Statement

- **Challenges**
  - **Inflexible planning to time-varying item arrival**



When #items is small
Bottleneck: Transmission (pickup, delivery, returning)
No queuing, processing time is short.

When #items is large
Bottleneck: queuing and processing
Transmission time is far more shorter

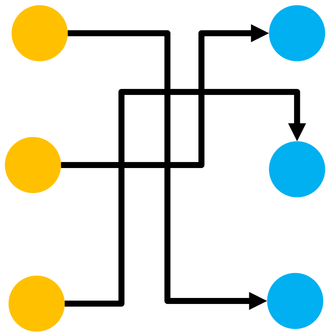
Note: We call this phenomenon as "bottleneck variation"

Problem Statement

- Challenges
  - Inefficient planning for massive robots and items

Calls for efficient design

Multi-Agent Path Finding [1]



Goal
Minimize path length

Constrain
Conflict-free

Problem
Time consuming

Existing research
Proved to be NP-Hard [2]
Branching & Bound Search [3]

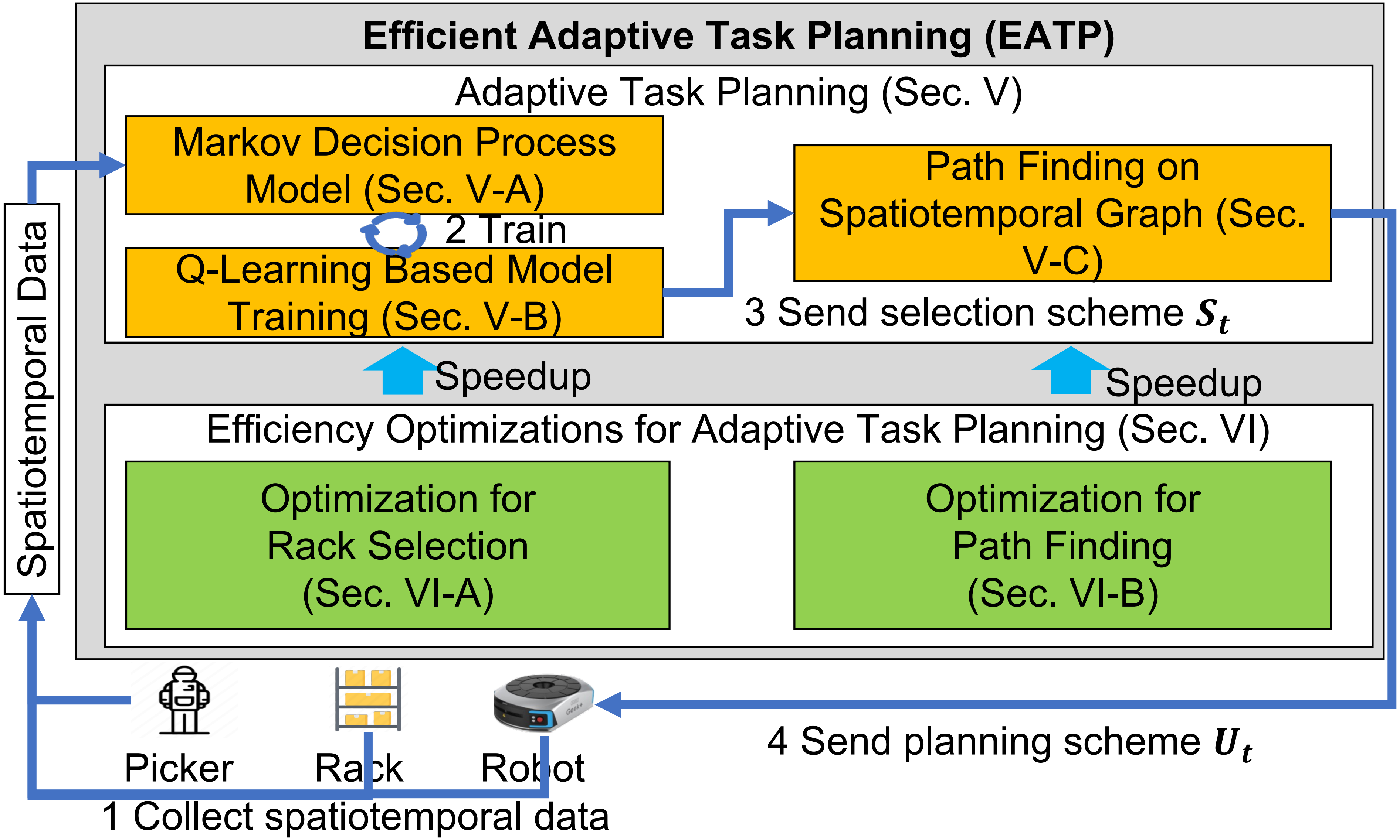
[1] R. Stern, et al, Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks, SoCS'19.
[2] P. Surynek, An optimization variant of multi-robot path planning is intractable, AAAI'10.
[3] G. Sharon, Conflict-based search for optimal multi-agent pathfinding, AI'15.

Outline

- Background & Motivation
- Problem Statement
- Our Solutions
- Experiments
- Conclusion

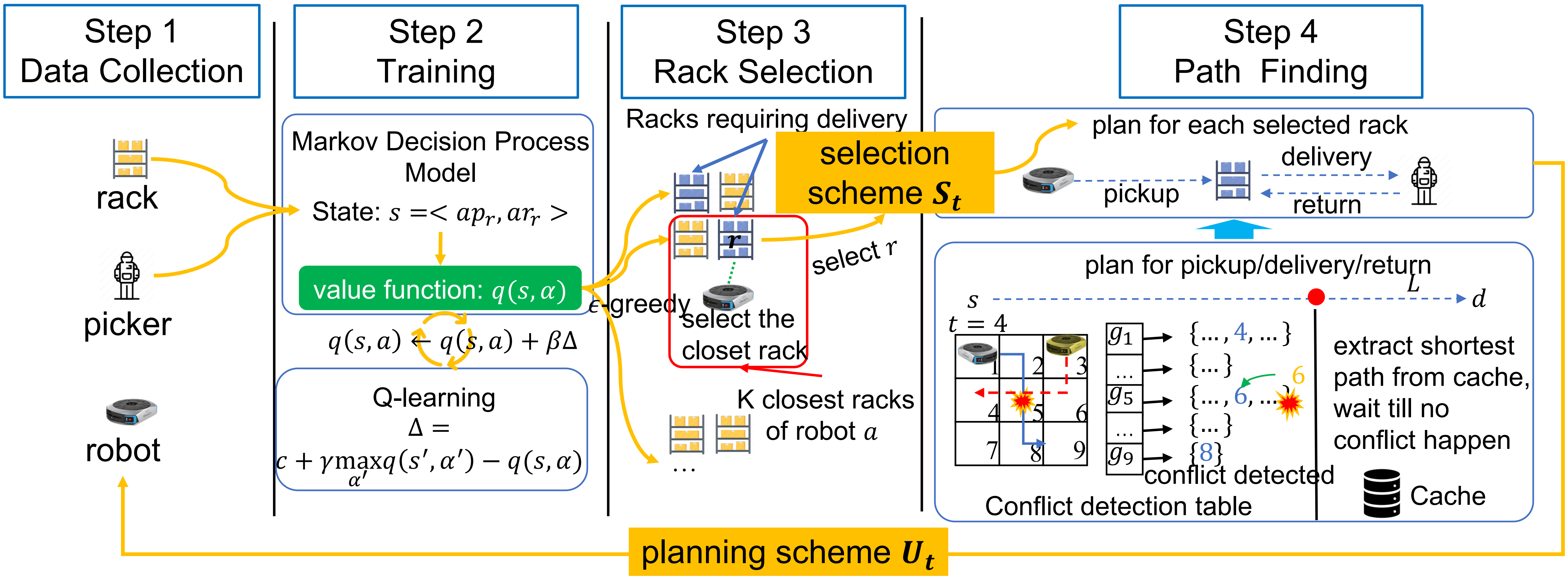
Our Solutions

- **System overview**



Our Solutions

- **Workflow**



Our Solutions

- **Workflow: 1 Data Collection**

rack

A rack r : $\langle l_r, \tau_r, p_r \rangle$
l_r: location in storage area
τ_r : processing times of items
p_r: corresponding picker

picker

A picker p : $\langle l_p, q_p, e_p \rangle$
l_p : fixed location
q_p : queuing racks
e_p: remaining picking time of current item

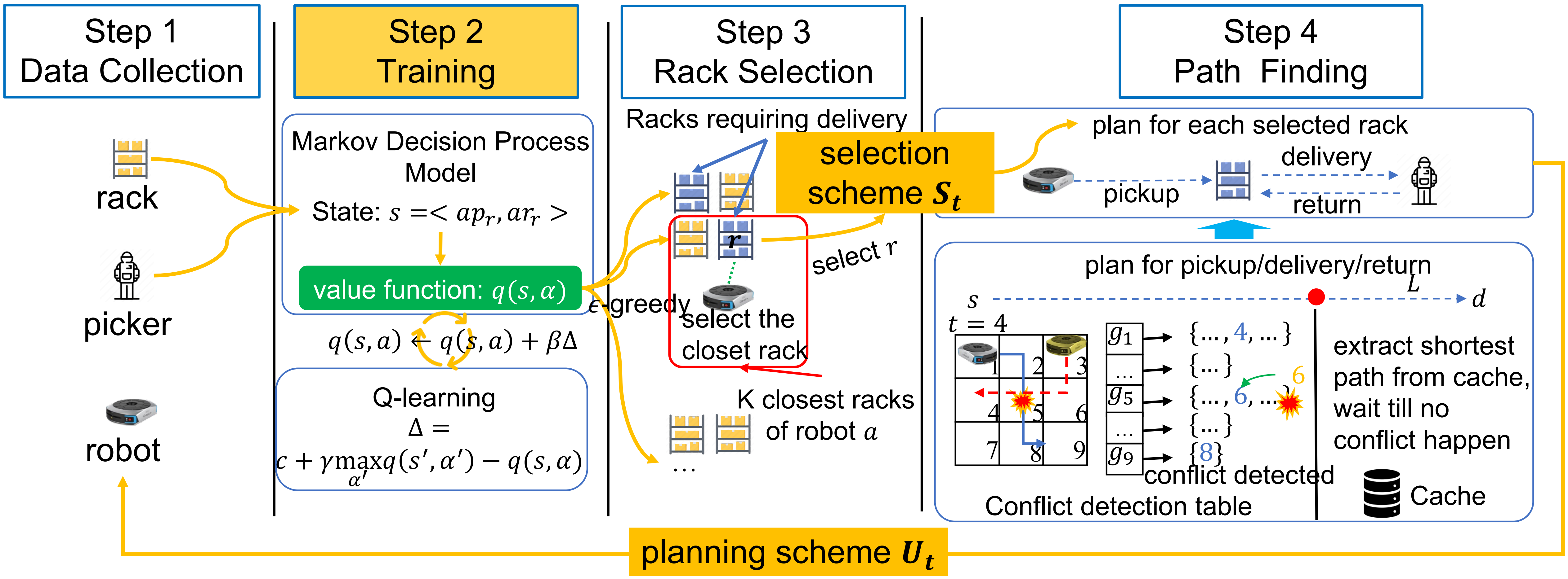
robot

A robot a : $\langle l_a, s_a \rangle$
l_a: current location
s_a : state (busy/idle)

Task Planning Algorithm

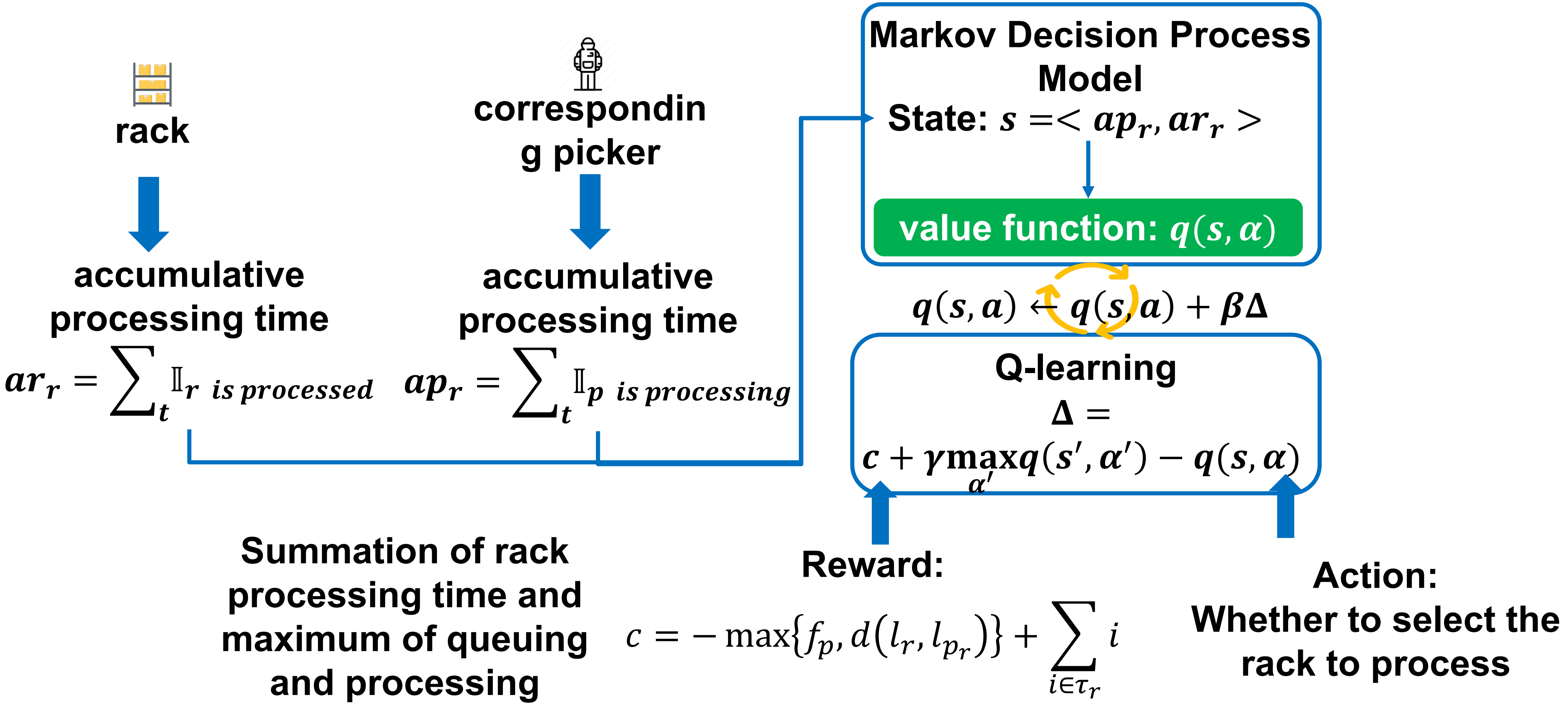
Our Solutions

- **Workflow**



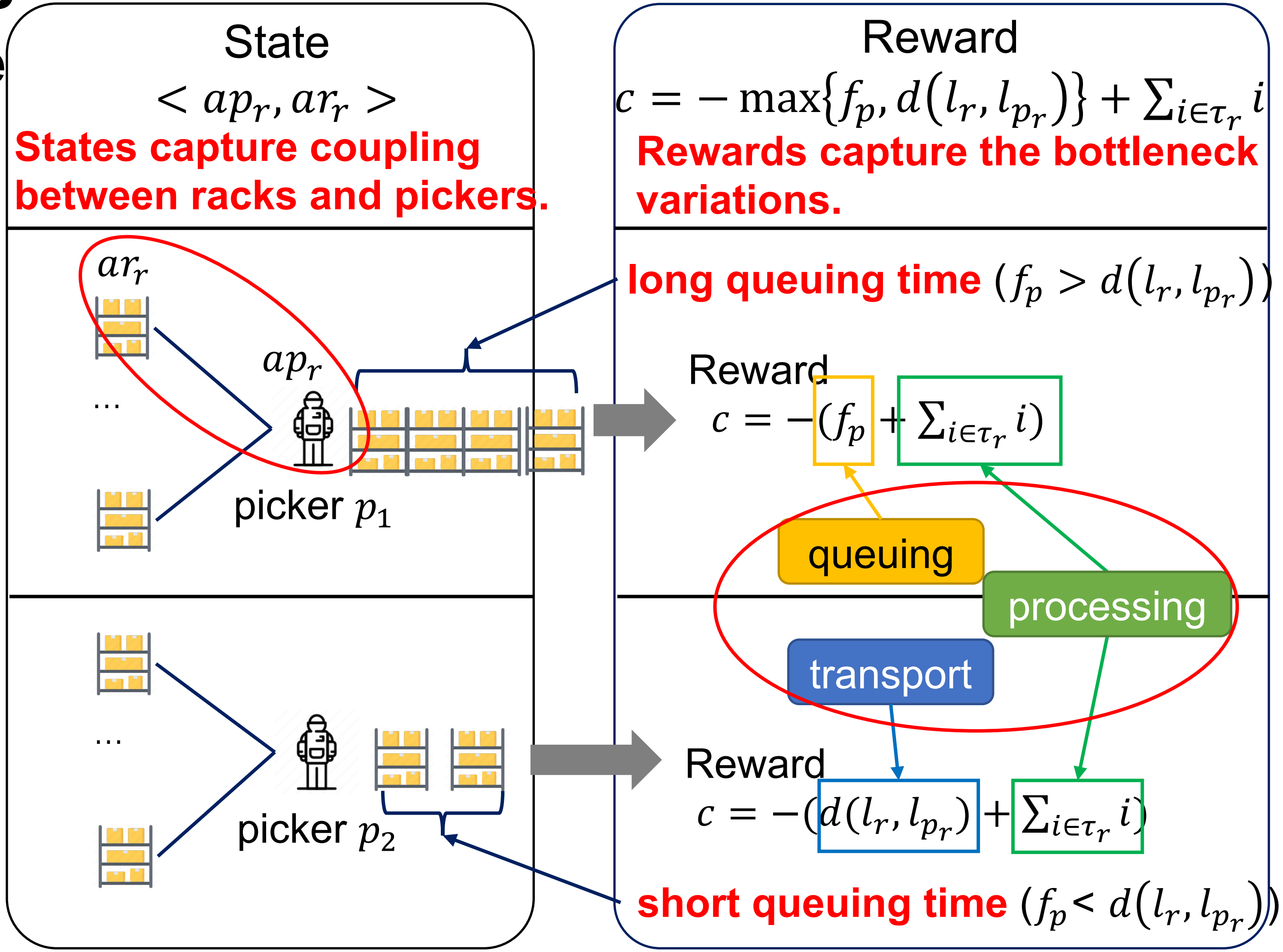
Our Solutions

- **Workflow：2 Training**



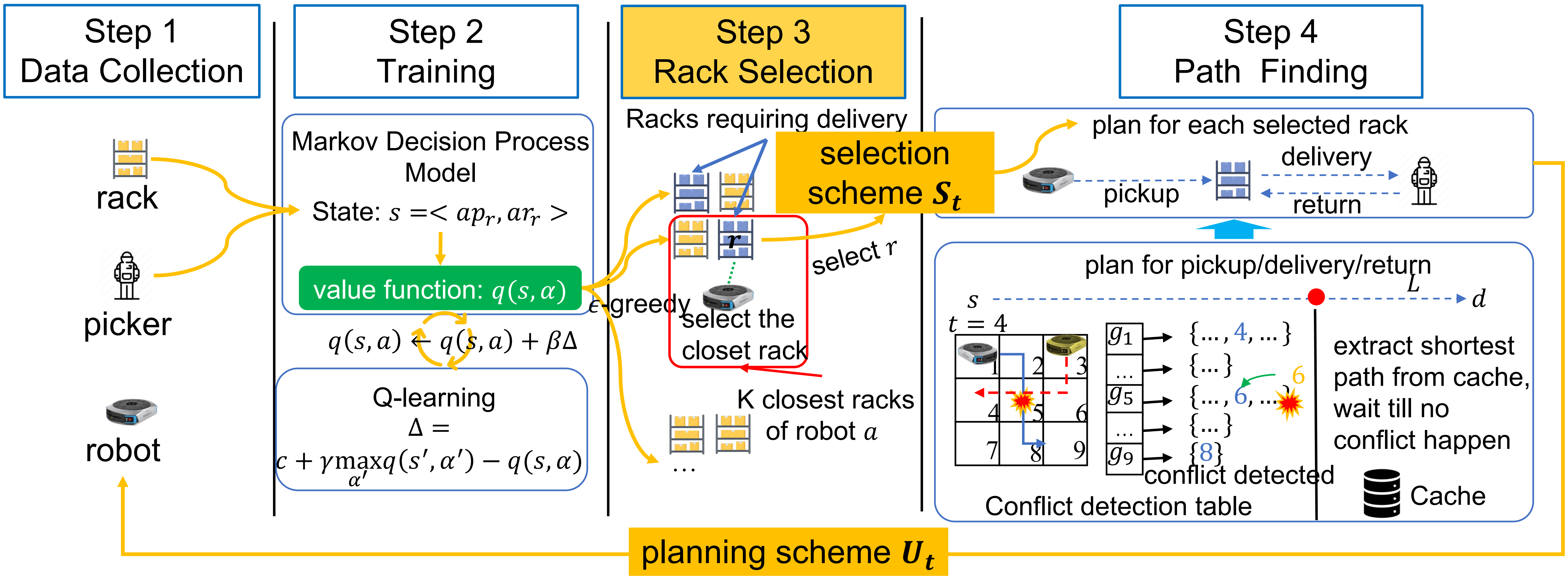
Our Solutions

- **Workflow：2 Training**
  - **Why this design can be adaptive?**



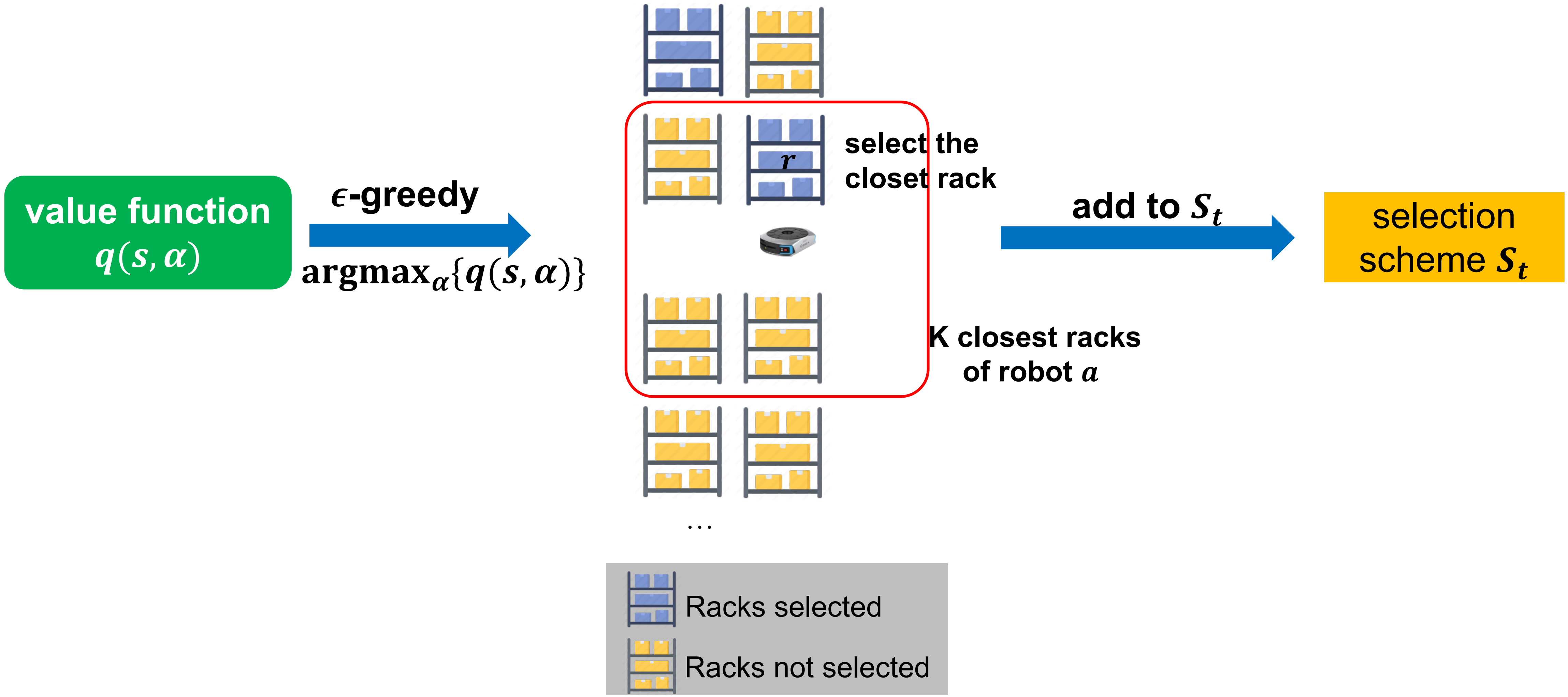
Our Solutions

- **Workflow**



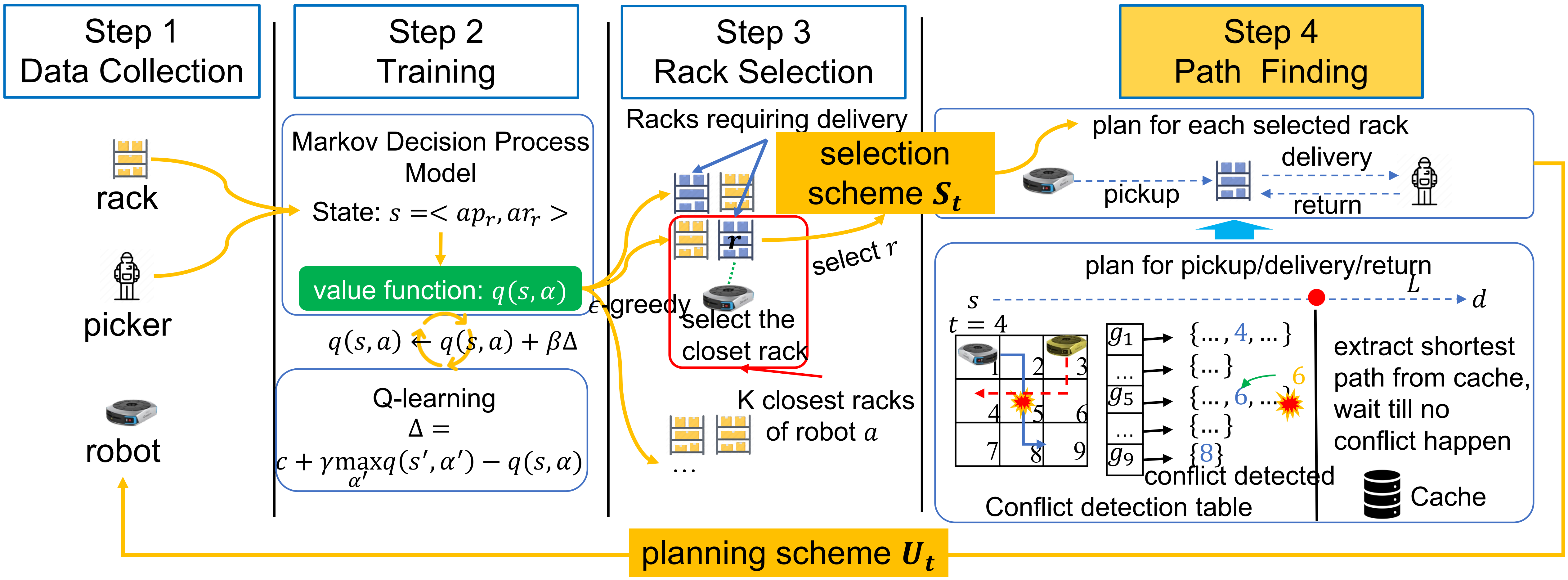
Our Solutions

- **Workflow: 3 Rack Selection**



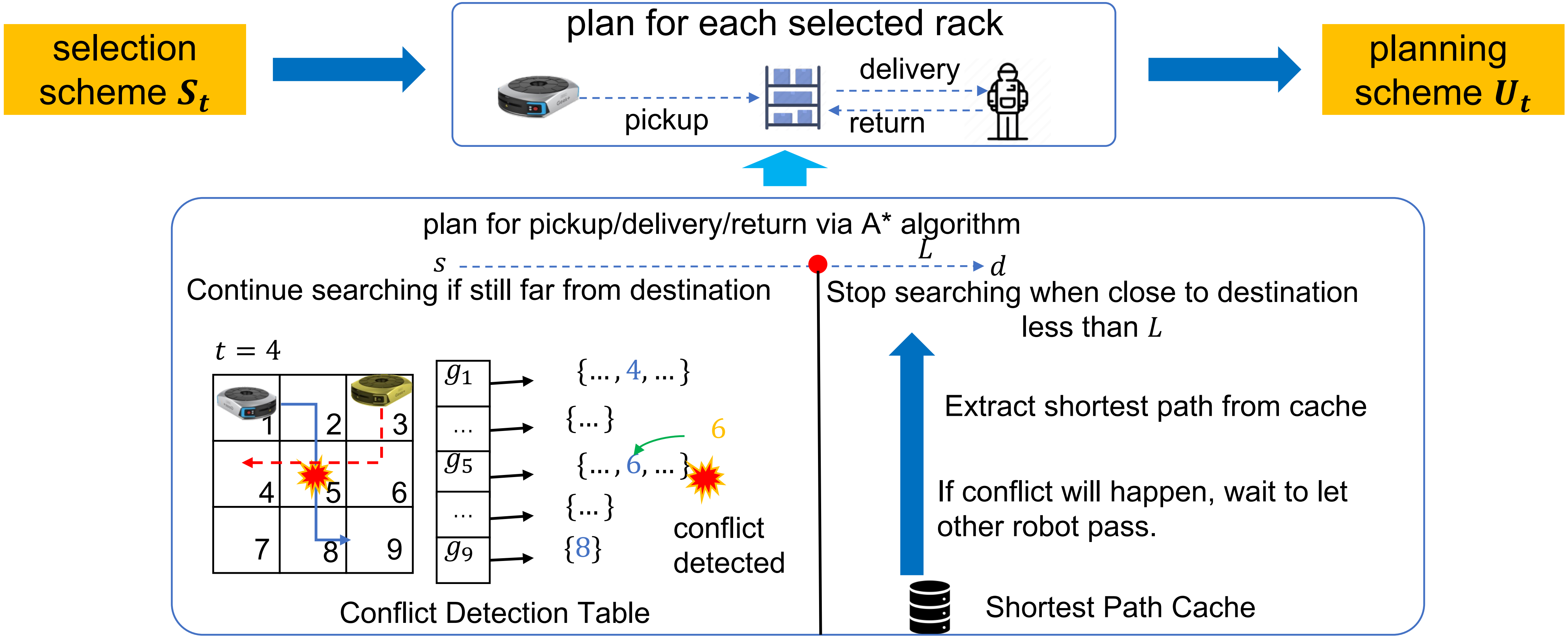
Our Solutions

- **Workflow**



Our Solutions

- **Workflow: 4 Path Finding**

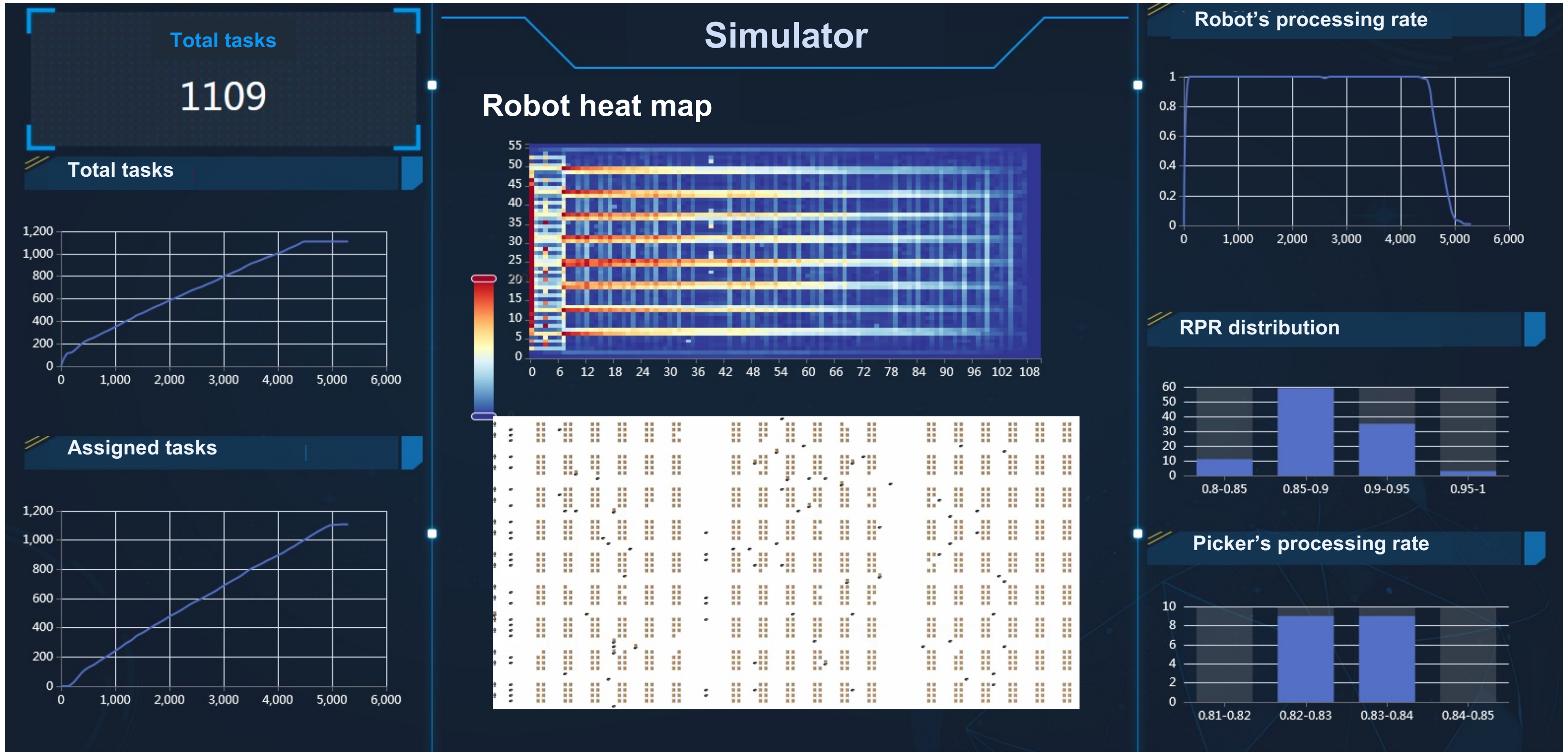


Outline

- Background & Motivation
- Problem Statement
- Our Solutions
- Experiments
- Conclusion

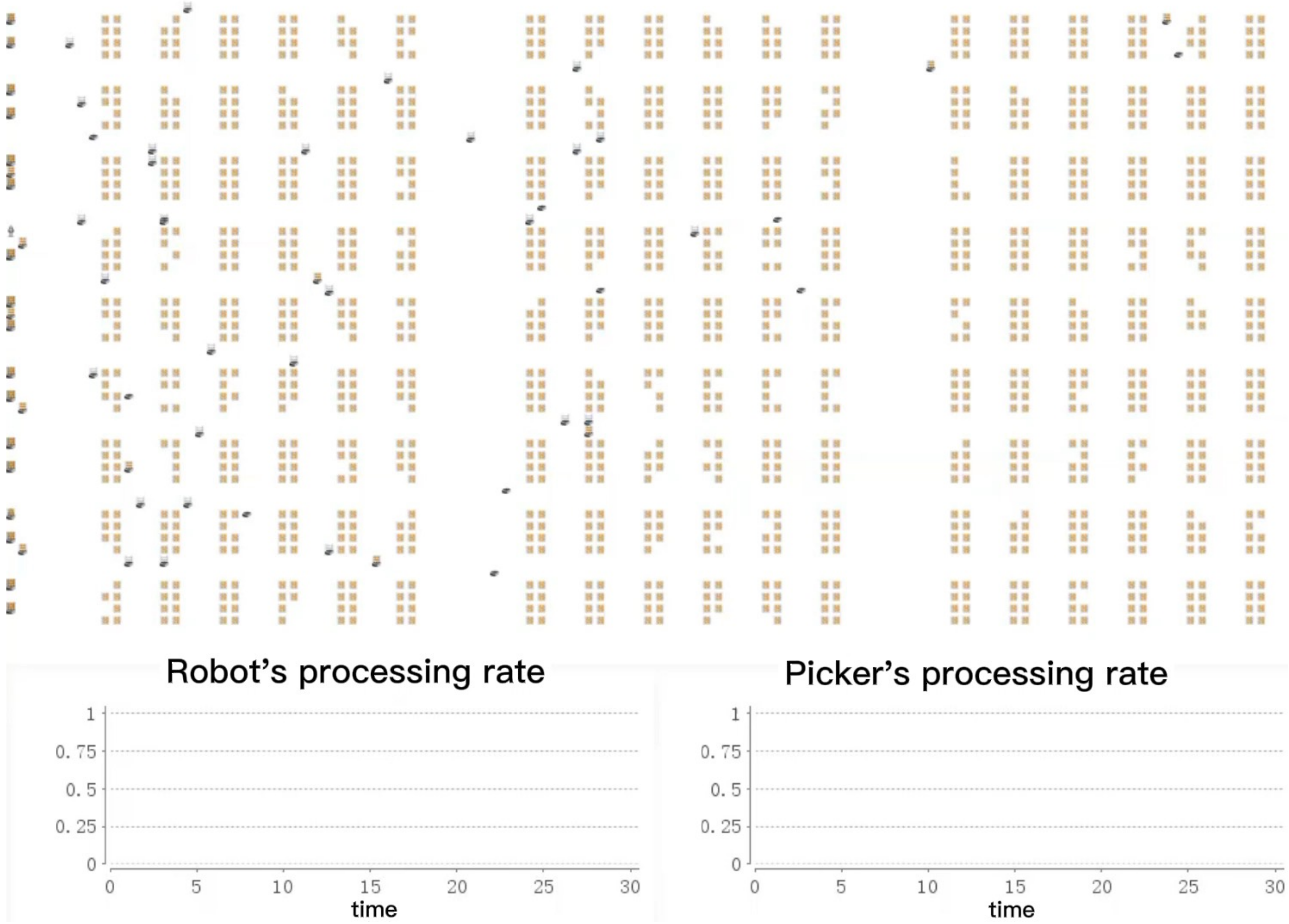
Experiments

- **Validation Environment**
  - **Dataset**
    - **Synthesized and real data from Geekplus Technology Co., Ltd.**
  - **Simulator**
    - **Collects information of robots, racks and pickers, executes task planning algorithm.**



Experiments

- **Validation Environment**



Experiments

- **Validation Environment**
  - **Dataset**
    - **Synthesized and real data from Geekplus Technology Co., Ltd.**
  - **Simulator**
    - **Collects information of robots, racks and pickers, executes task planning algorithm.**
- **Running Information**
  - **CPU: CPU Intel(R) Xeon(R) Platinum 8269CY CPU T 3.10GHz**
  - **Memory: 20GB**
- **Parameter Setting**
  - ϵ **−greedy:** $\epsilon = 0.1$
  - **Learning rate:** $\beta = 0.1$

Experiments

- **Comparing methods**
  - **NTP[1]: selects racks whose corresponding picker finishes picking earliest**
  - **LEF[2]: selects racks whose items are emerged earliest**
  - **ILP[3]: integer linear programming based solution**

- **Evaluation metrics**
  - **Makespan**
  - **Picker's Processing Rate (PPR): ratio of picker's processing time to total time**
  - **Robot's Working Rate (RWR): ratio of robot's working time to total time**
  - **Selection Time Consumption (STC): time consumption of selection procedure**
  - **Planning Time Consumption (PTC): time consumption of path planning procedure**

[1] H. Ma, et al, Lifelong multi-agent path finding for online pickup and delivery tasks," AAMAS'17.
[2] D. Deng, et al, Task selection in spatial crowdsourcing from worker's perspective, GeoInformatica'16.
[3] N. Boysen, et al, Parts-to-picker based order processing in a rack-moving mobile robots environment, EJOR'17.

Experiments

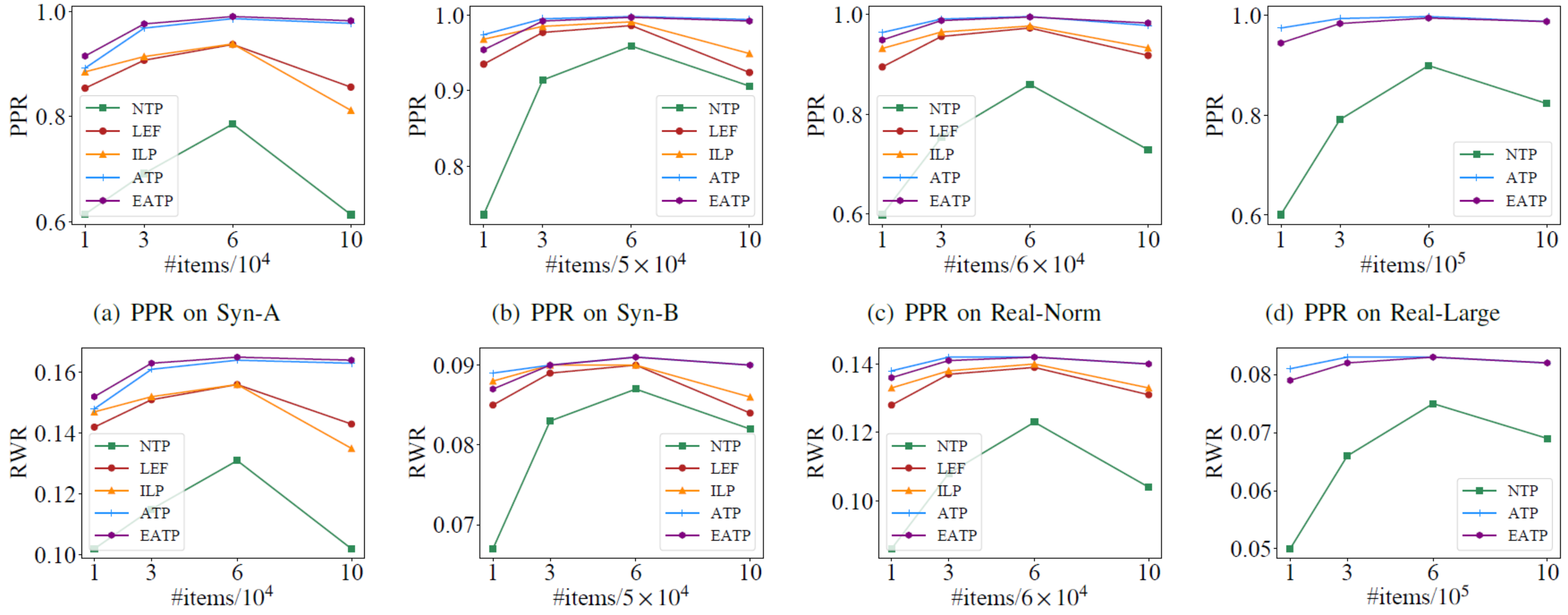
- **Makespan Comparison**
  - **Lower values mean better performances**

	Syn-A	Syn-B	Real-Norm	Real-Large
NTP	95,713	229,865	222,044	264,139
LEF	68,736	225,484	176,317	-
ILP	72,423	219,555	173,446	-
ATP (Ours)	**60,193**	**209,531**	**165,438**	**220,257**
EATP (Ours)	**60,753**	**209,866**	**164,628**	**220,263**

Our ATP/EATP performs best among all other methods over all datasets

Experiments

- **PPR/RWR Comparison**
  - **Higher values mean better performances**

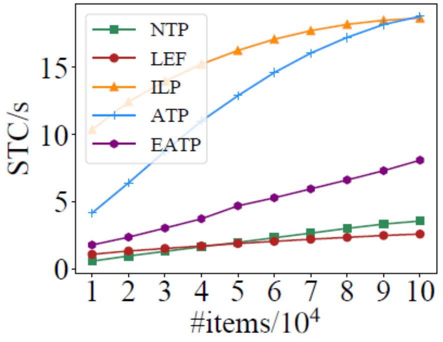


(a) PPR on Syn-A
(b) PPR on Syn-B
(c) PPR on Real-Norm
(d) PPR on Real-Large
(e) RWR on Syn-A
(f) RWR on Syn-B
(g) RWR on Real-Norm
(h) RWR on Real-Large

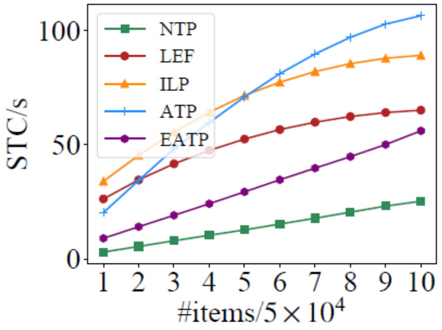
Our ATP/EATP schedules robots and pickers more sufficiently

Experiments

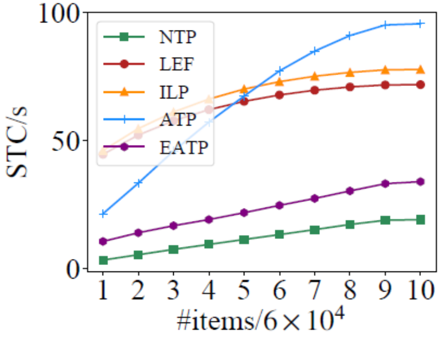
- ## Efficiency Comparison
  - ### Lower values mean better performances



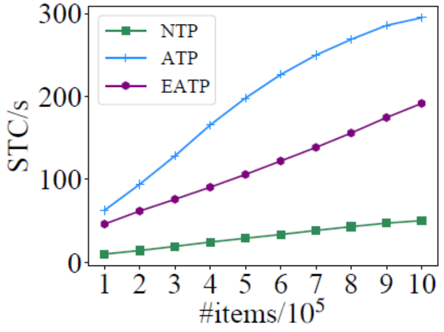
(a) STC on Syn-A



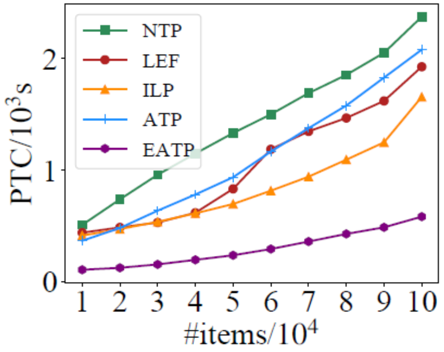
(b) STC on Syn-B



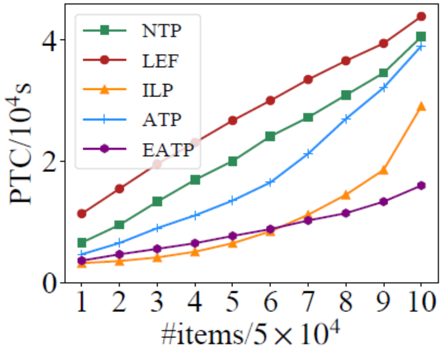
(c) STC on Real-Norm



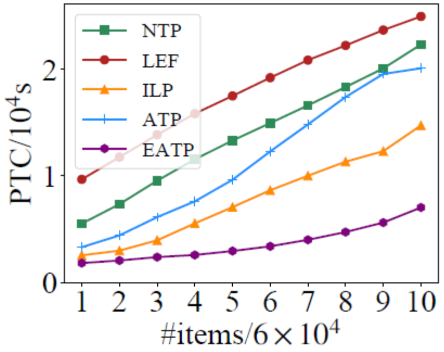
(d) STC on Real-Large



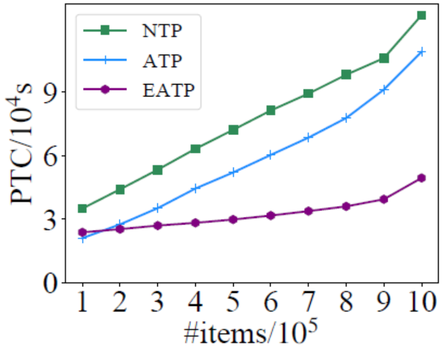
(e) PTC on Syn-A



(f) PTC on Syn-B



(g) PTC on Real-Norm



(h) PTC on Real-Large

Our EATP is more efficient than other methods

Experiments

- **Case Study**

Selected at $t = 392$

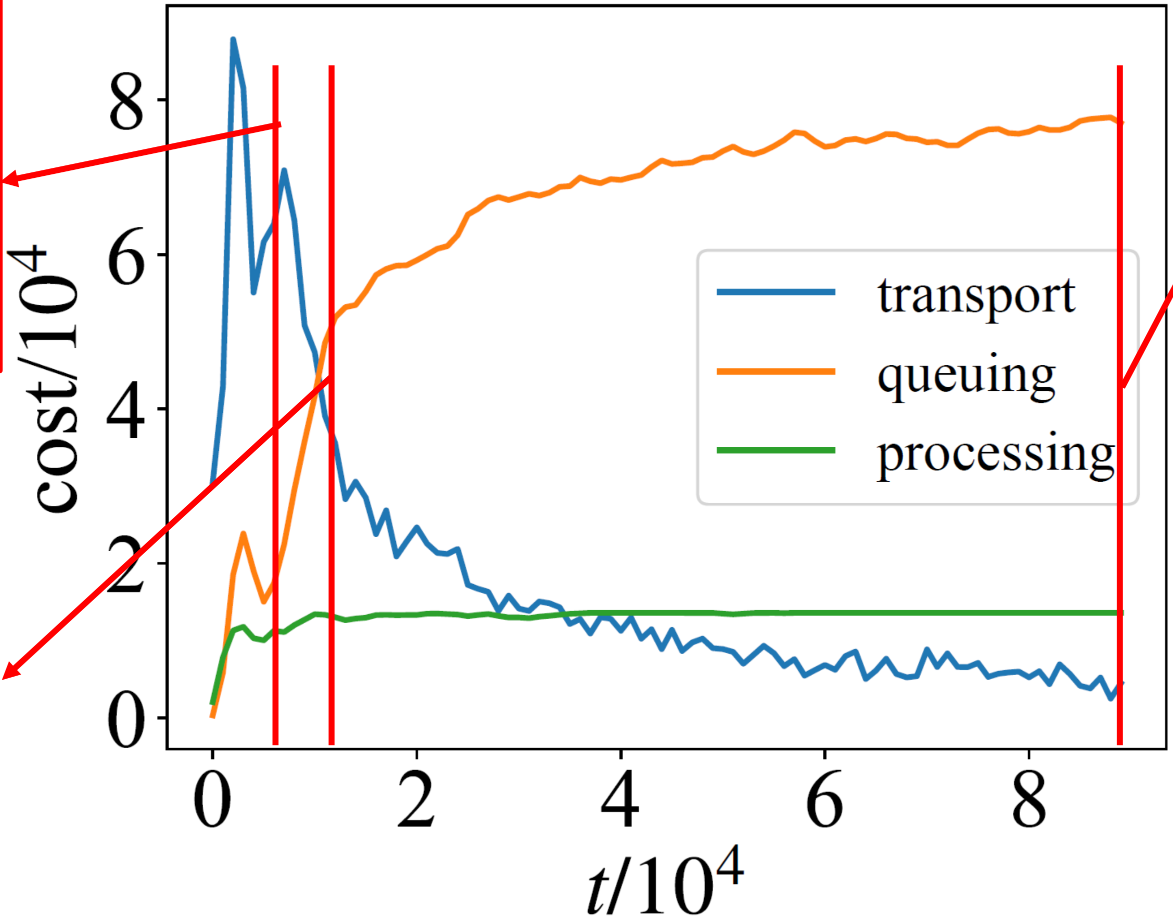
Transmit rack with only 2 items to pick

bottleneck: transport

Selected at $t = 2508$

Transmit rack with 10 items to pick.

bottleneck: transport queuing



Selected at $t = 9974$

Transmit rack with 47 items to pick.

bottleneck: queuing

Variations of number of items cause the bottleneck variations.

Outline

- Background & Motivation
- Problem Statement
- Our Solutions
- Experiments
- Conclusion

Conclusion

- **We propose a task planning algorithm for robotized warehouses, aiming to sense bottleneck variations and adaptively make decisions.**

- **We devise an efficient path finding algorithm which approximately searches for conflict-free paths.**

- **Experiments on real history data validate the performances on effectiveness and efficiency.**

Q & A

Thank You