IEEE ICDE2023
Anaheim, CA April 3 – 7, 2023

Route Planning in Warehouses Made Efficient: A Strip-based Framework

Dingyuan Shi[1], Nan Zhou[1], Yongxin Tong[1],

Zimu Zhou[2], Yi Xu[1], Ke Xu[1]

1 Beihang University
2 City University of Hong Kong

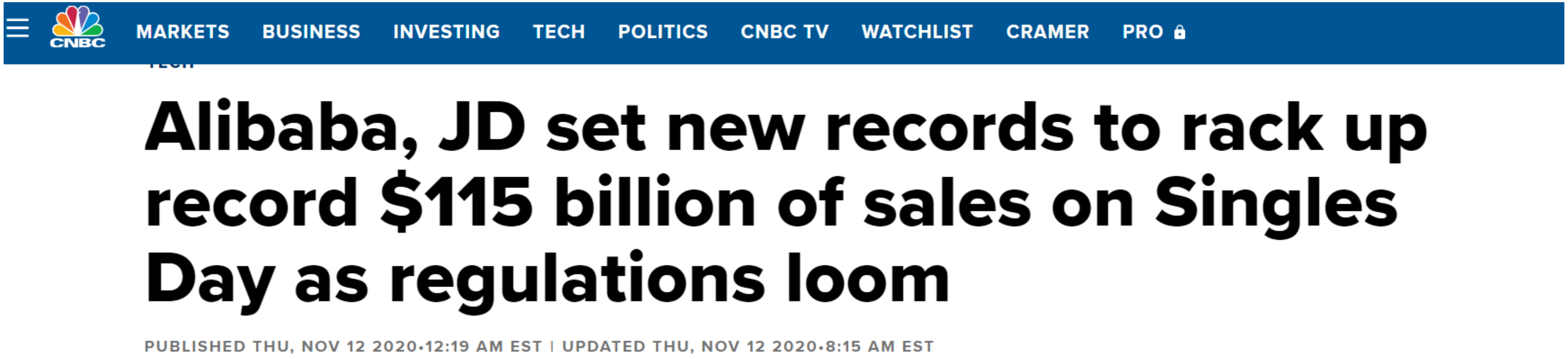
Outline

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

Outline

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

Background & Motivation



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

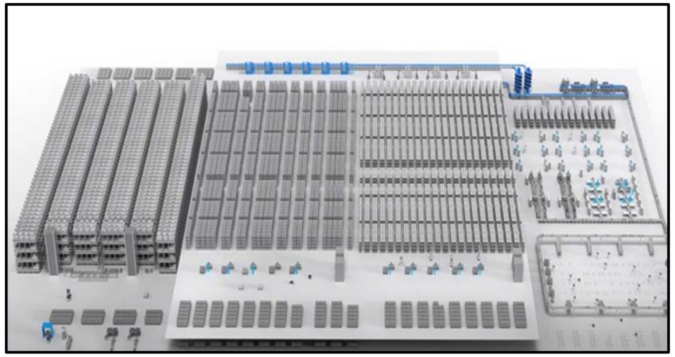
The boom of e-commerce has stimulated enormous logistic demands

Background & Motivation

- Some companies and their products/services

amazon

CAINIAO 菜鸟



Geek+

GreyOrange
Fulfilling the Future in Real Time™

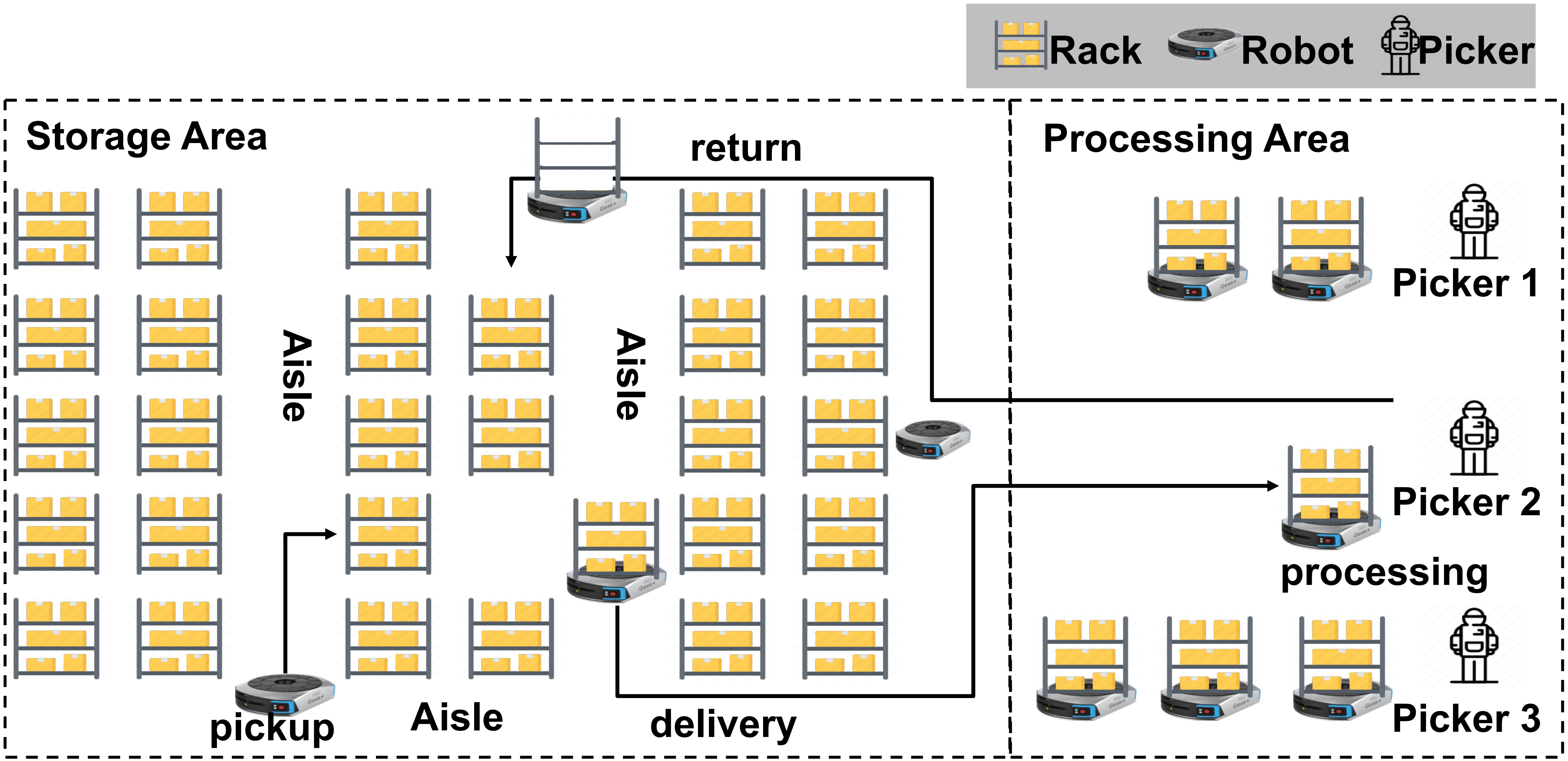
HITACHI
Inspire the Next

JDL 京东物流

Robotized warehouses are expected to improve the performance

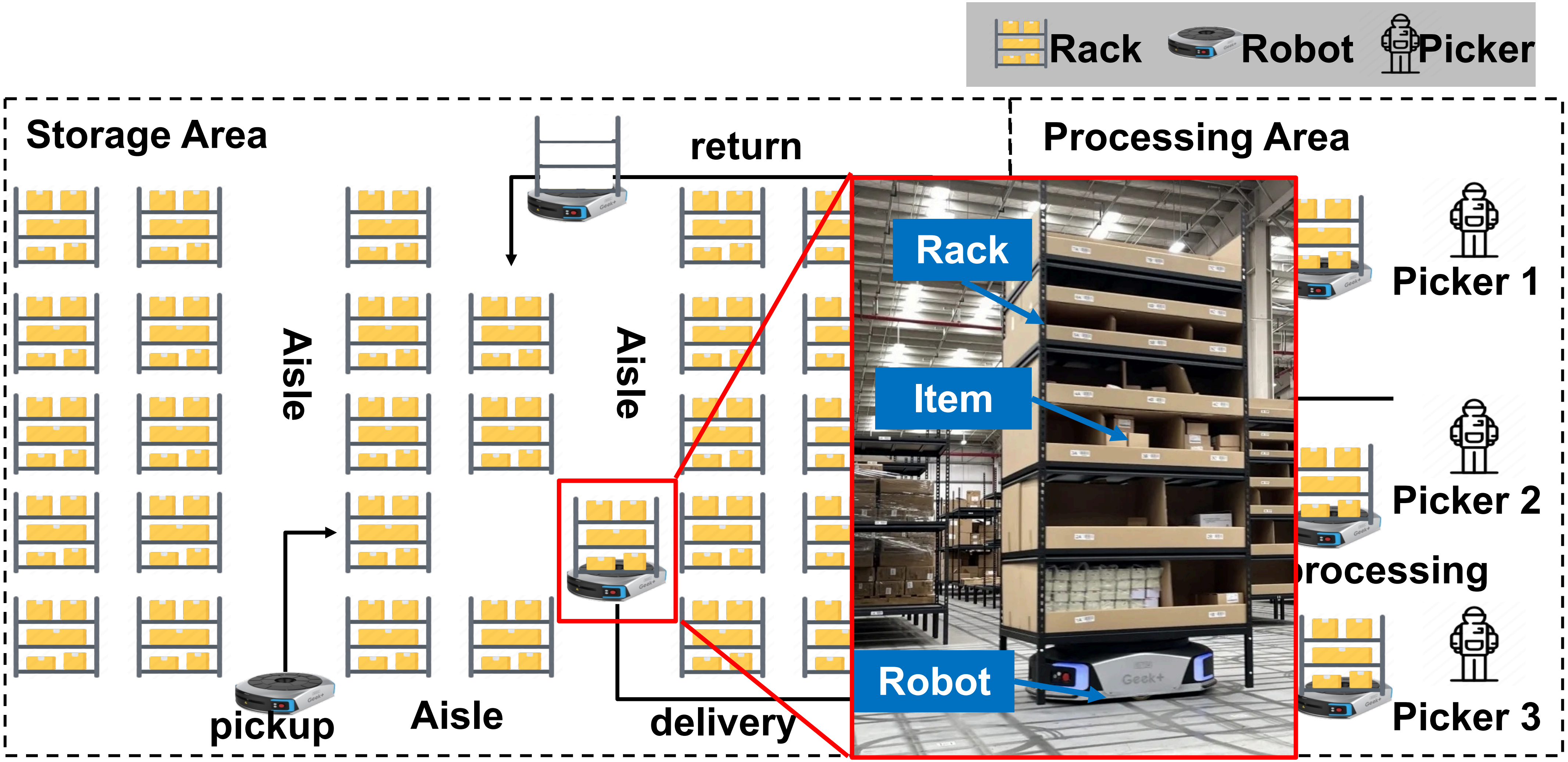
Background & Motivation

- **A typical robotized warehouse**



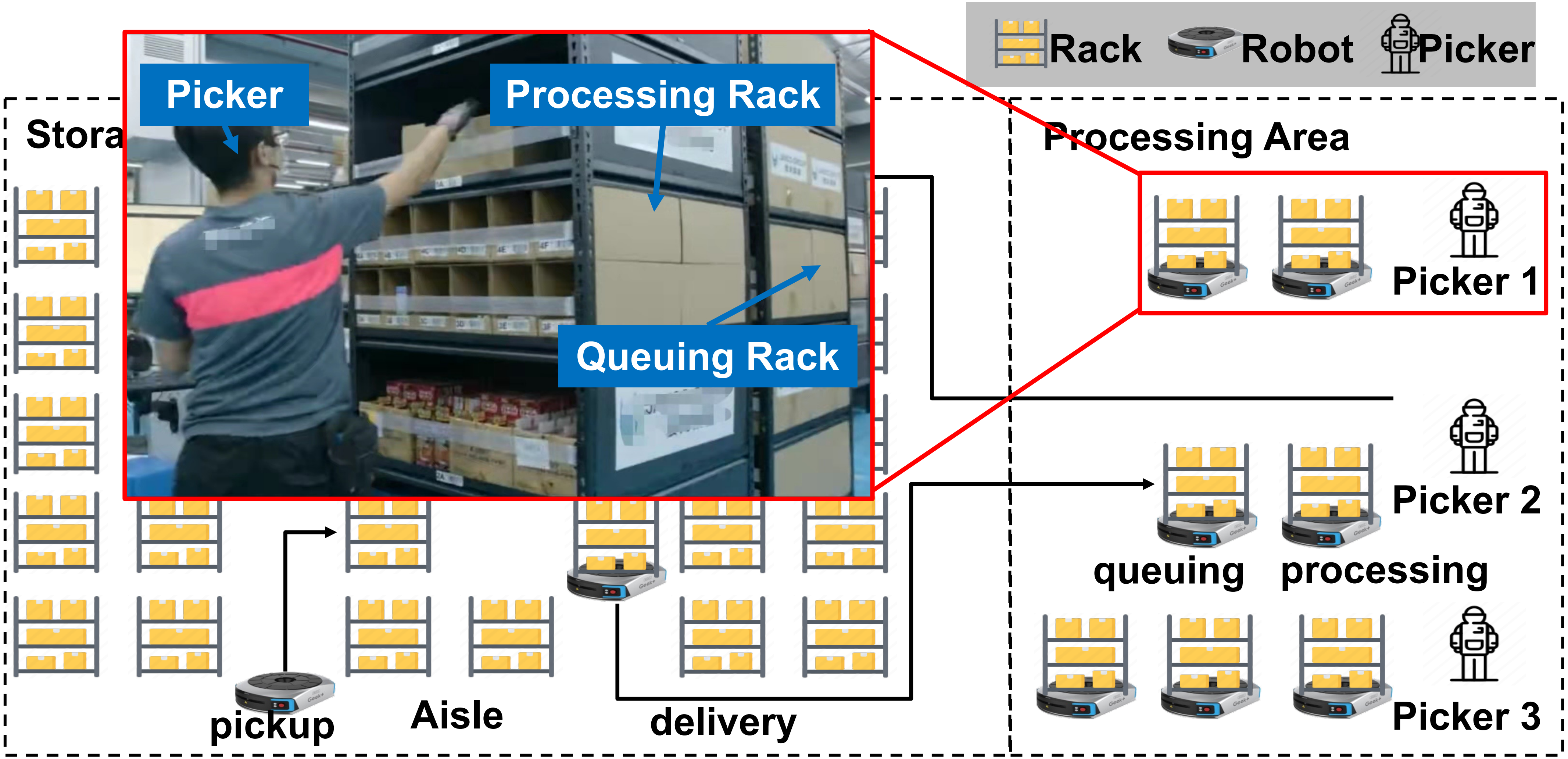
Background & Motivation

- **A typical robotized warehouse**



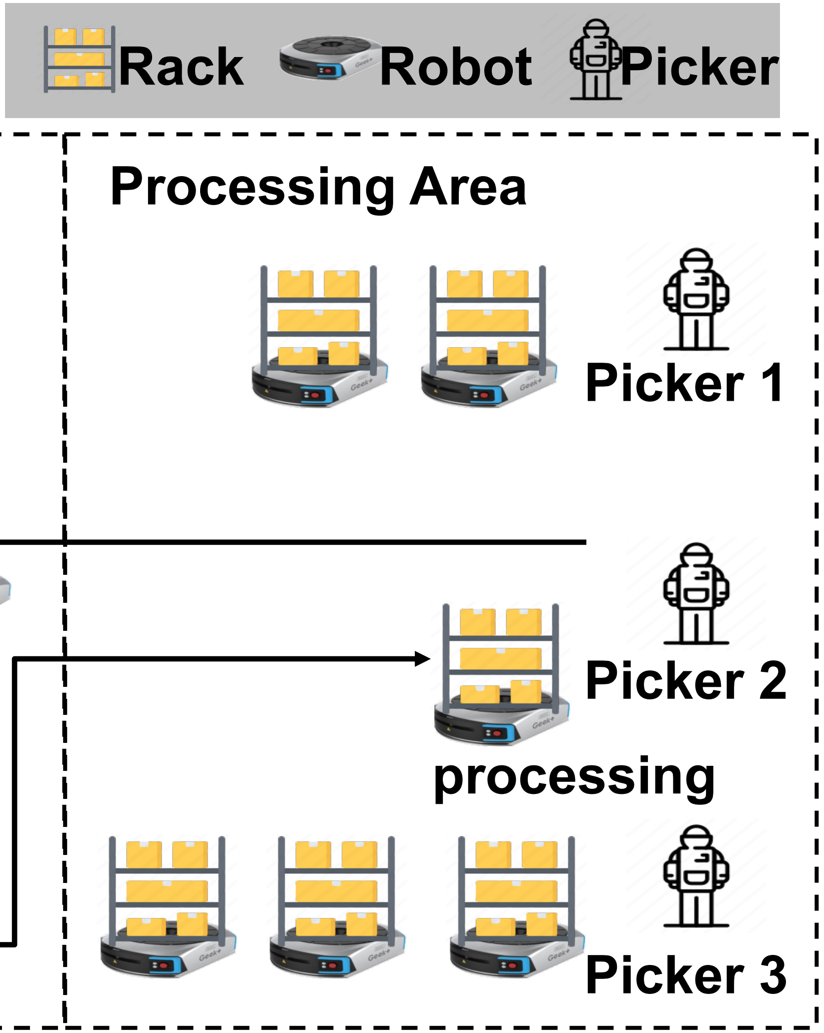
Background & Motivation

- A typical robotized warehouse



Background & Motivation

- **Transportation steps can be viewed as route planning problem**

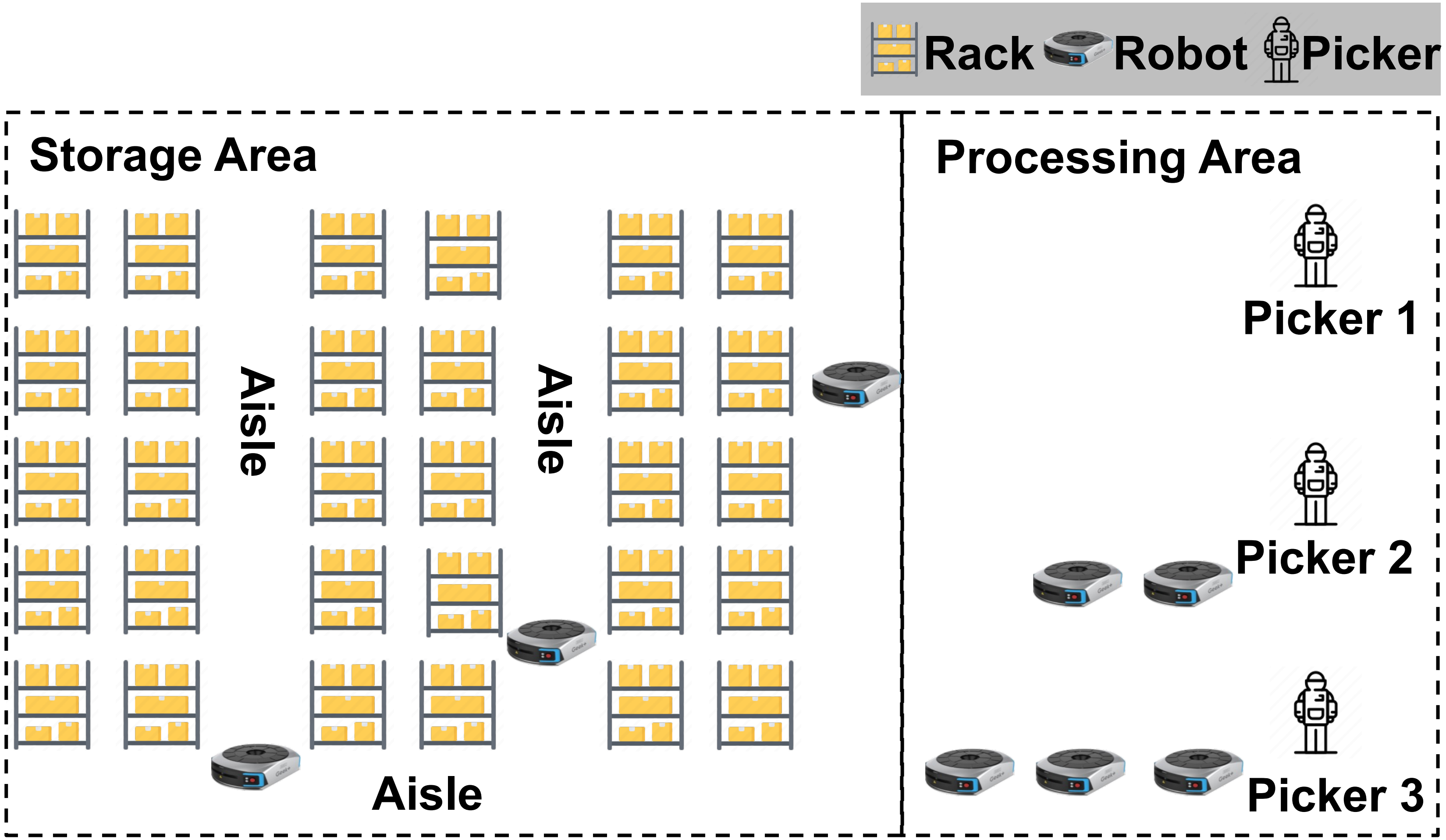


Outline

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

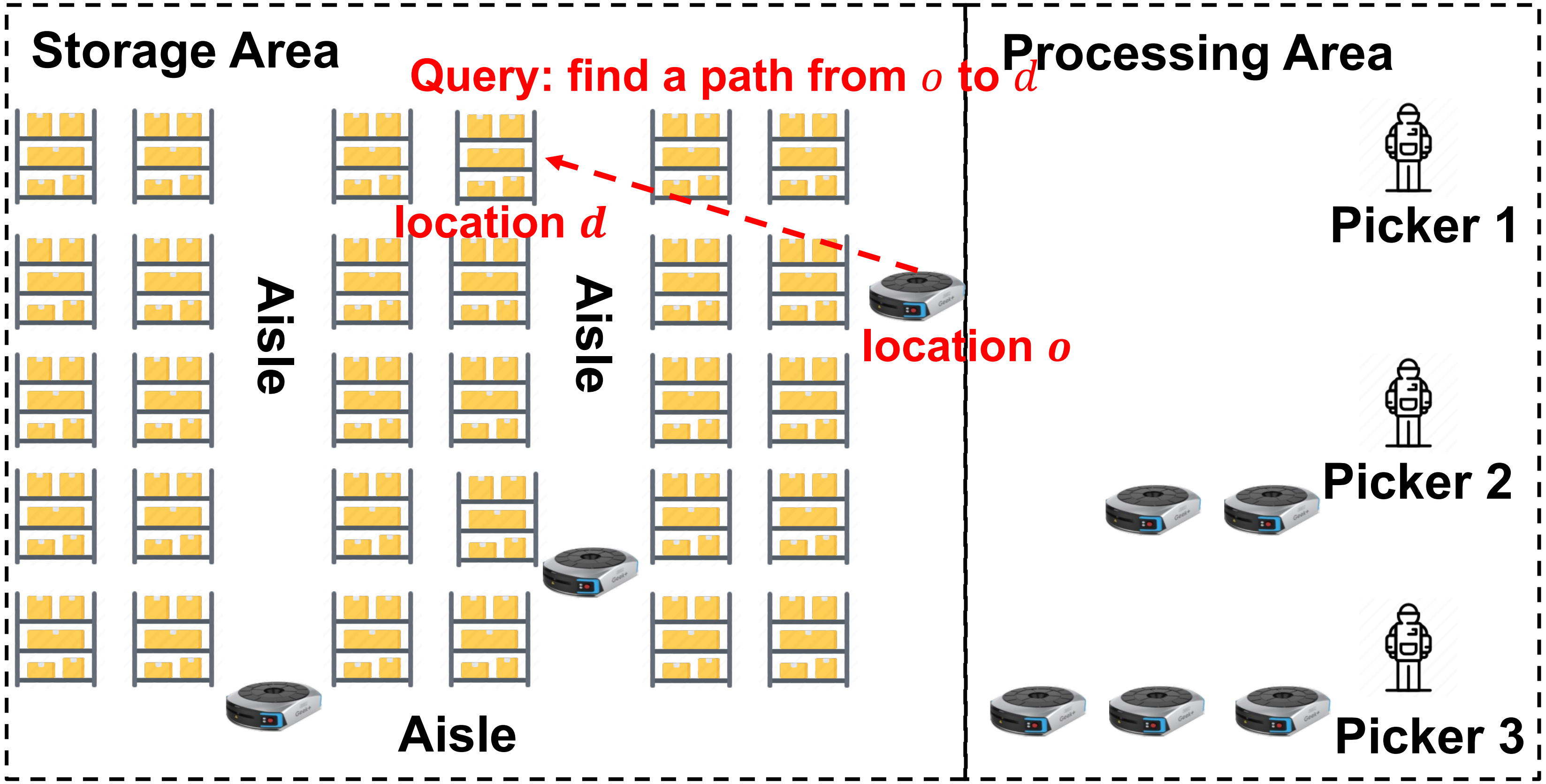
Problem Statement

- **Given: A warehouse layout and a set of robots**



Problem Statement

- **Given: A warehouse layout and a set of robots**
- **Given: Queries in an online manner**
  - **A query$< o, d >$ contains origin o and destination d**



Problem Statement

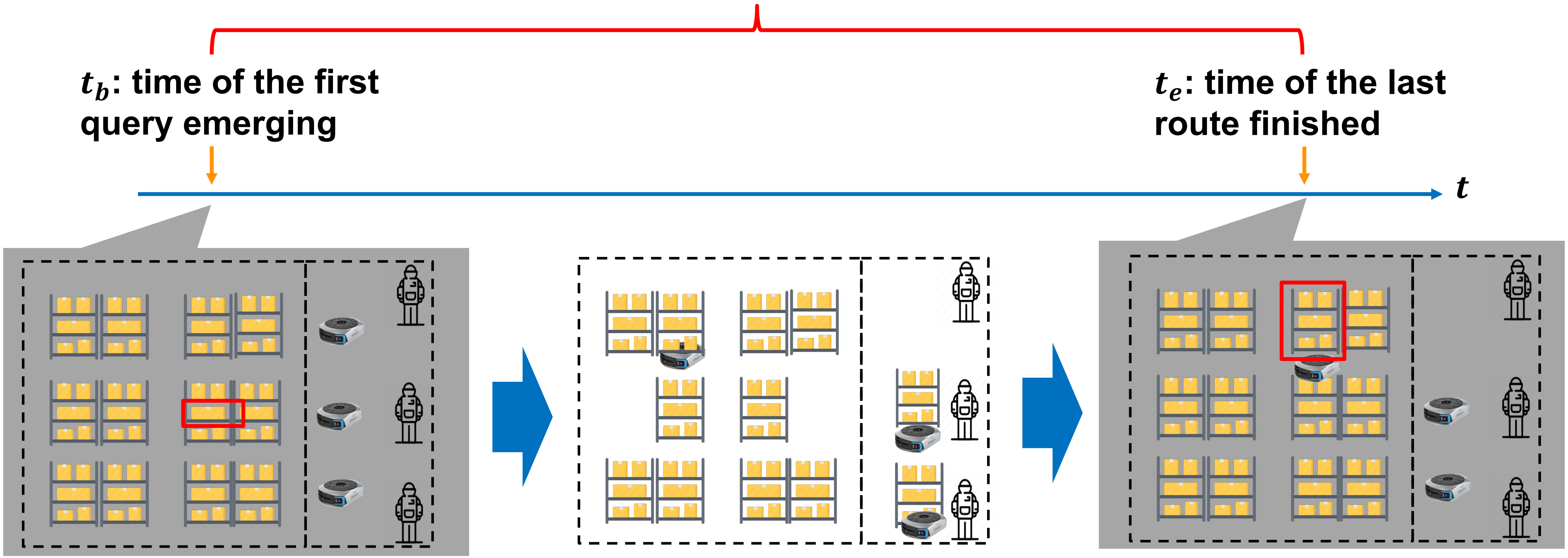
- Given: A warehouse layout and a set of robots
- Given: Queries in an online manner
- Output: Collision-free routes for queries that holistically minimize the total makespan

Constraints

Optimization Goal

Problem Statement

- **Optimization Goal: Minimizing the *Makespan***



Makespan: $M = t_e - t_b$

t_b: time of the first query emerging

t_e: time of the last route finished

t

An query emerges. A rack requires delivery

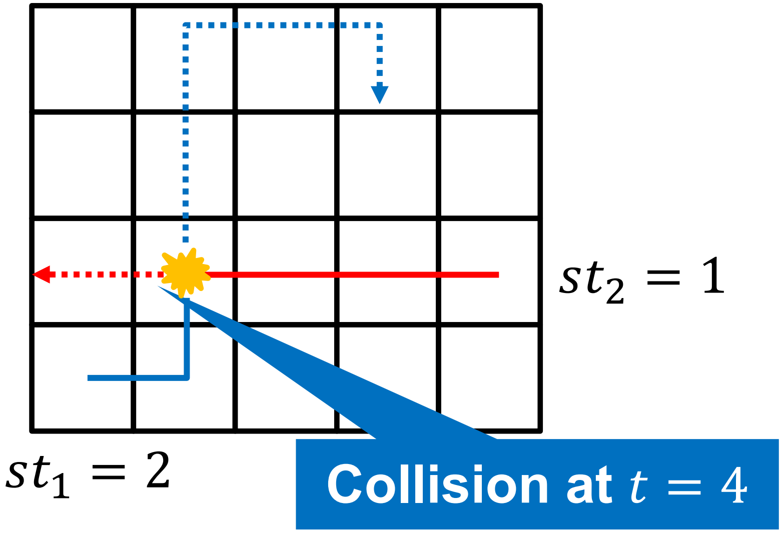
Robots deliver racks back and forth.

The last rack has been returned.

Problem Statement

- Constraints: All routes for robots should be *collision-free*

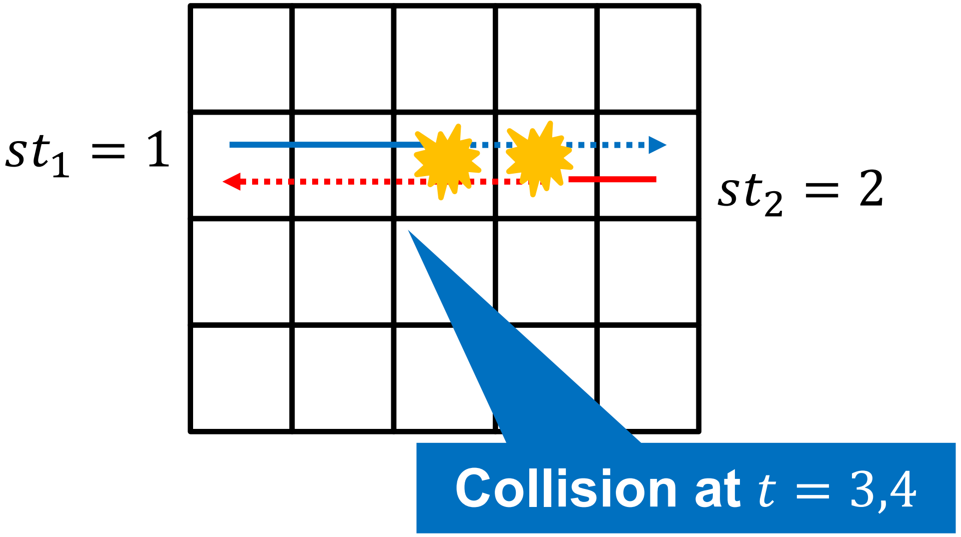
- Two types of collision
  - Static: Collisions against static racks
  - Dynamic: Collisions against other robots in motion



$st_1 = 2$

$st_2 = 1$

Collision at $t = 4$



$st_1 = 1$

$st_2 = 2$

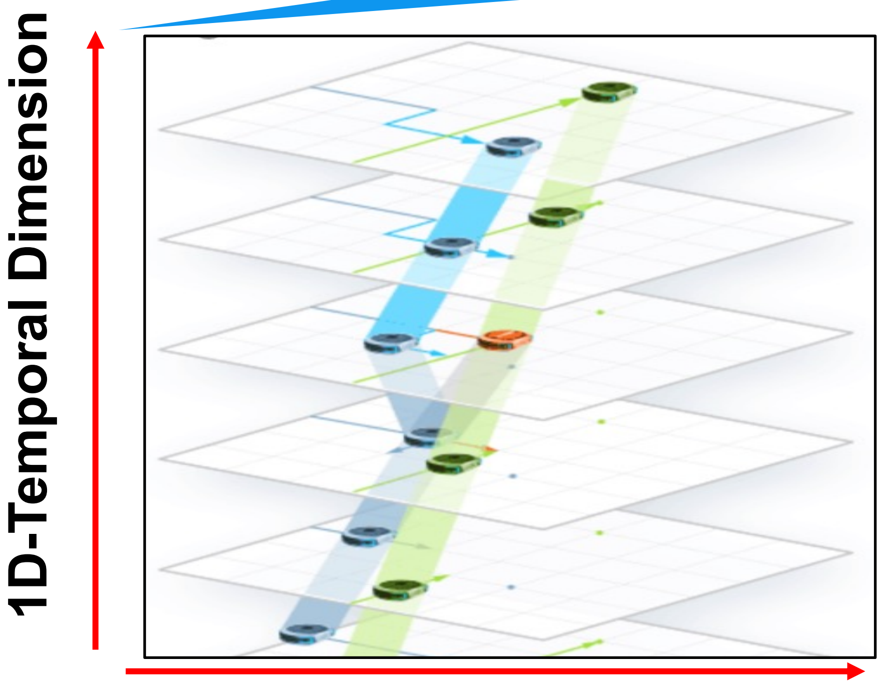
Collision at $t = 3{,}4$

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

Problem Statement

- **Challenges**
  - **Traditional methods[1] directly search on the grid-based space**

Constraints of dynamic collisions forces us to consider temporal dimension



A direct search on grids leads to a 3D search space.

2D Spatial Dimension

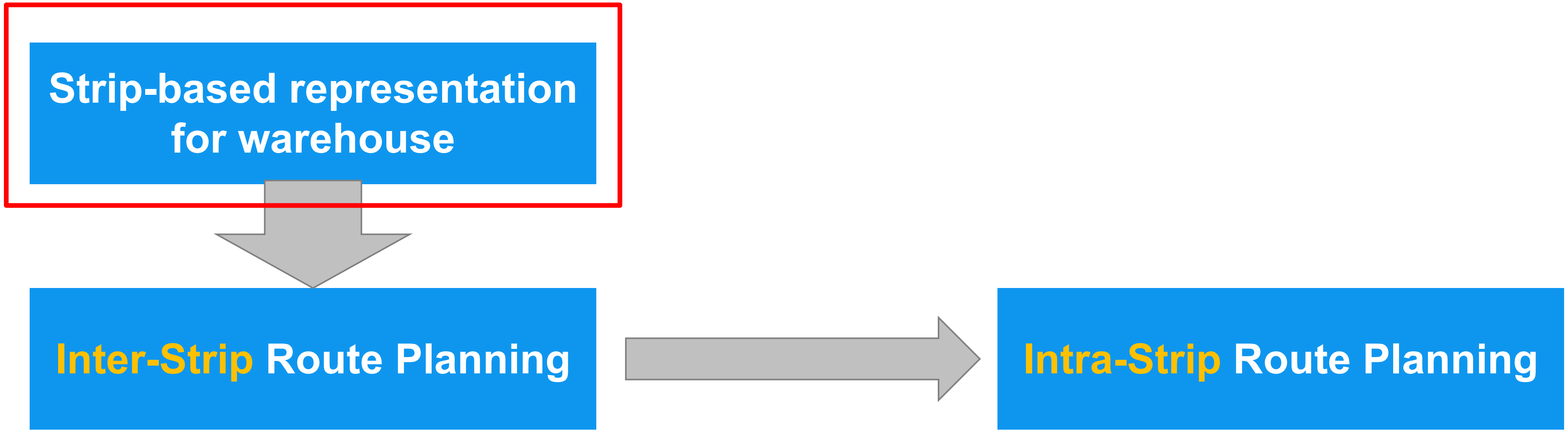
[1] G. Sharon, Conflict-based search for optimal multi-agent pathfinding, AI’15.

Outline

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

Our Solutions

- **Main Idea**
  - **Leverage the regular pattern of warehouses**
  - **Re-formalize the layout into “strips”**
  - **Search route in a two-level manner.**
- **Workflow**



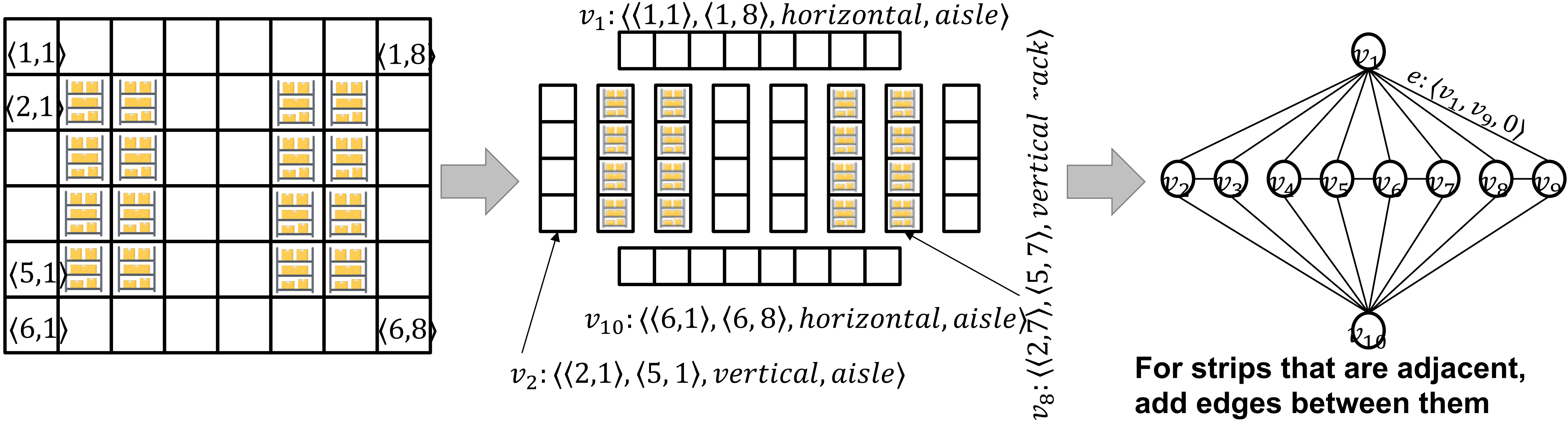
Strip-based representation for warehouse

Inter-Strip Route Planning

Intra-Strip Route Planning

Our Solutions

- **Workflow: Strip-based representation**
  - **Split warehouse aisles into “strips” (leverage regular patterns)**
  - **The width of each strip is only one**
  - **Aggregate strips as vertices and build new graph- Strip Graph**

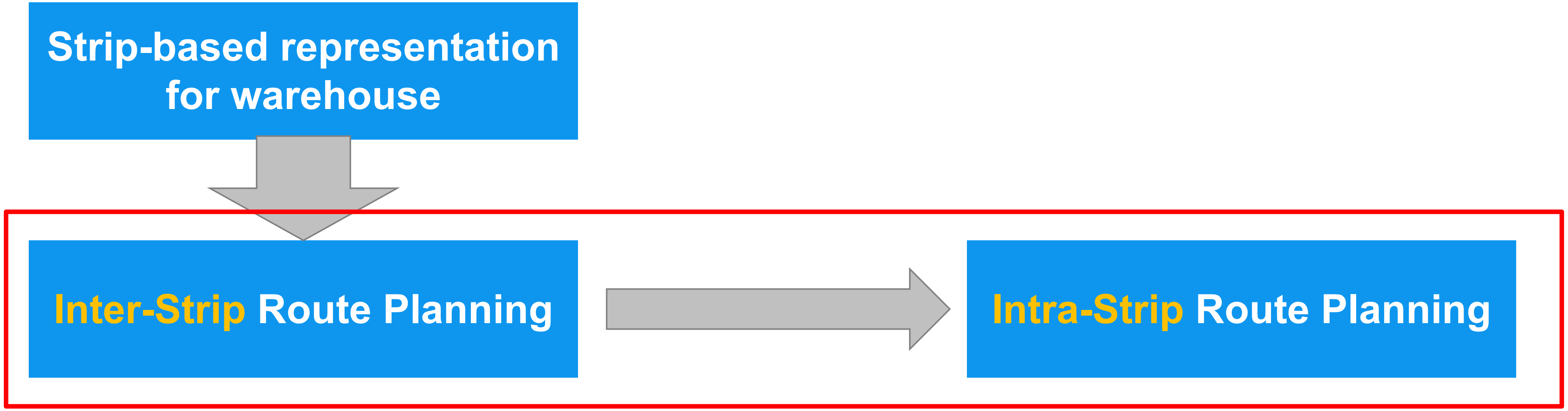

$\langle 1,1 \rangle$
$\langle 1,8 \rangle$
$\langle 2,1 \rangle$
$\langle 5,1 \rangle$
$\langle 6,1 \rangle$
$\langle 6,8 \rangle$
v_1: $\langle\langle 1,1 \rangle, \langle 1,8 \rangle, horizontal, aisle\rangle$
v_{10}: $\langle\langle 6,1 \rangle, \langle 6,8 \rangle, horizontal, aisle\rangle$
v_2: $\langle\langle 2,1 \rangle, \langle 5,1 \rangle, vertical, aisle\rangle$
v_8: $\langle\langle 2,7 \rangle, \langle 5,7 \rangle, vertical, rack\rangle$
e: $\langle v_1, v_9, 0\rangle$
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
v_{10}

For strips that are adjacent, add edges between them

Each vertex contains multiple grids, which reduces the search space

Our Solutions

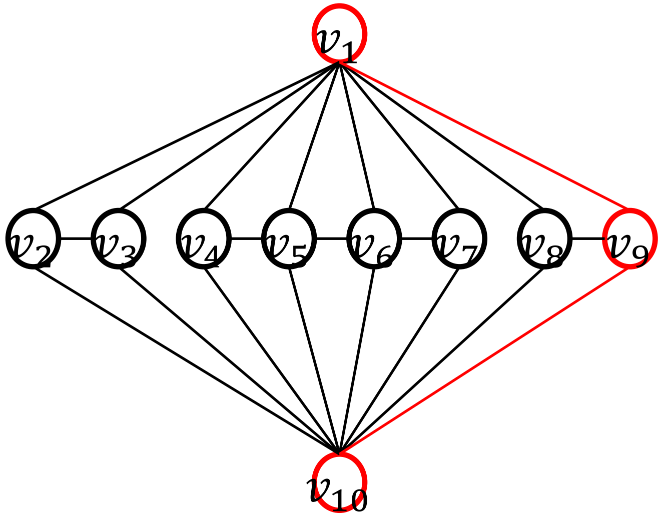
- **Main Idea**
  - **Leverage the regular pattern of warehouse**
  - **Re-formalize the layout into “strips”**
  - **Search routes in a two-level manner.**
- **Workflow**

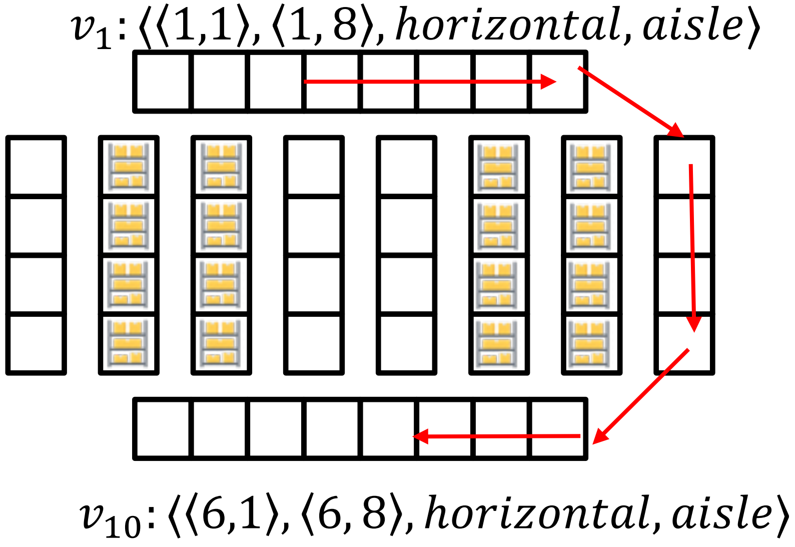

Strip-based representation for warehouse
Inter-Strip Route Planning
Intra-Strip Route Planning

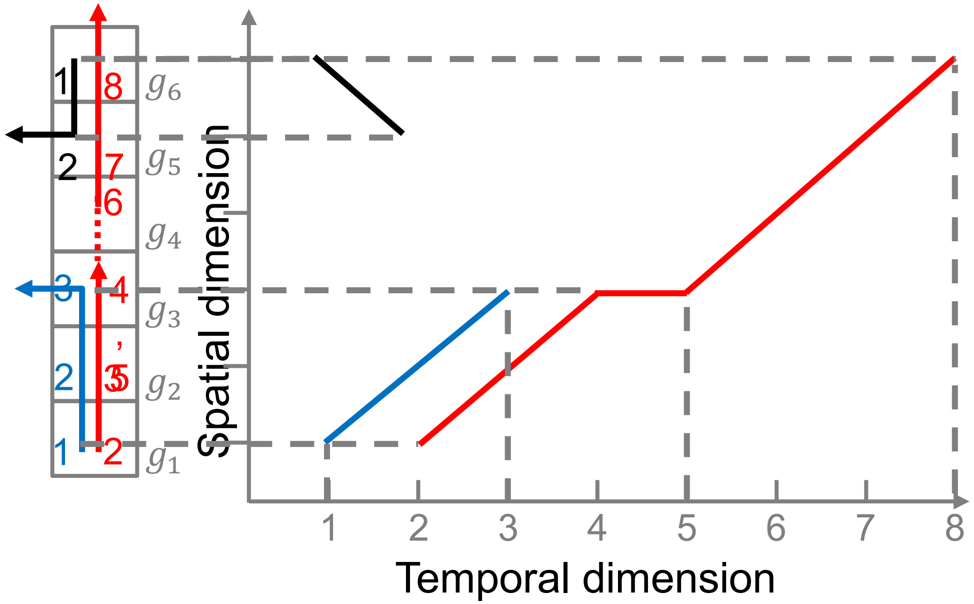
Our Solutions

- **Workflow: Planning overview**

Inter-Strip Route Planning → **Intra-Strip Route Planning**




$v_1: \langle\langle 1,1\rangle, \langle 1,8\rangle, horizontal, aisle\rangle$
$v_{10}: \langle\langle 6,1\rangle, \langle 6,8\rangle, horizontal, aisle\rangle$



First search the routes in inter-strip level without considering collision

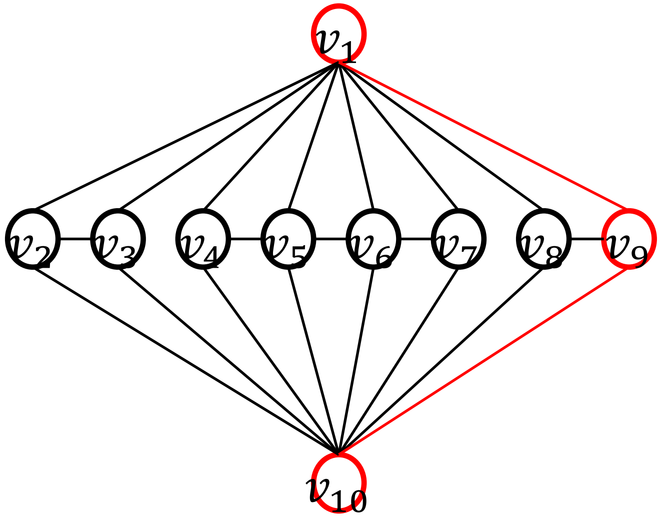
Specifies routes within a strip considering collision-avoidance

Strip representation helps confining the collision detection in a 2D space.

Our Solutions

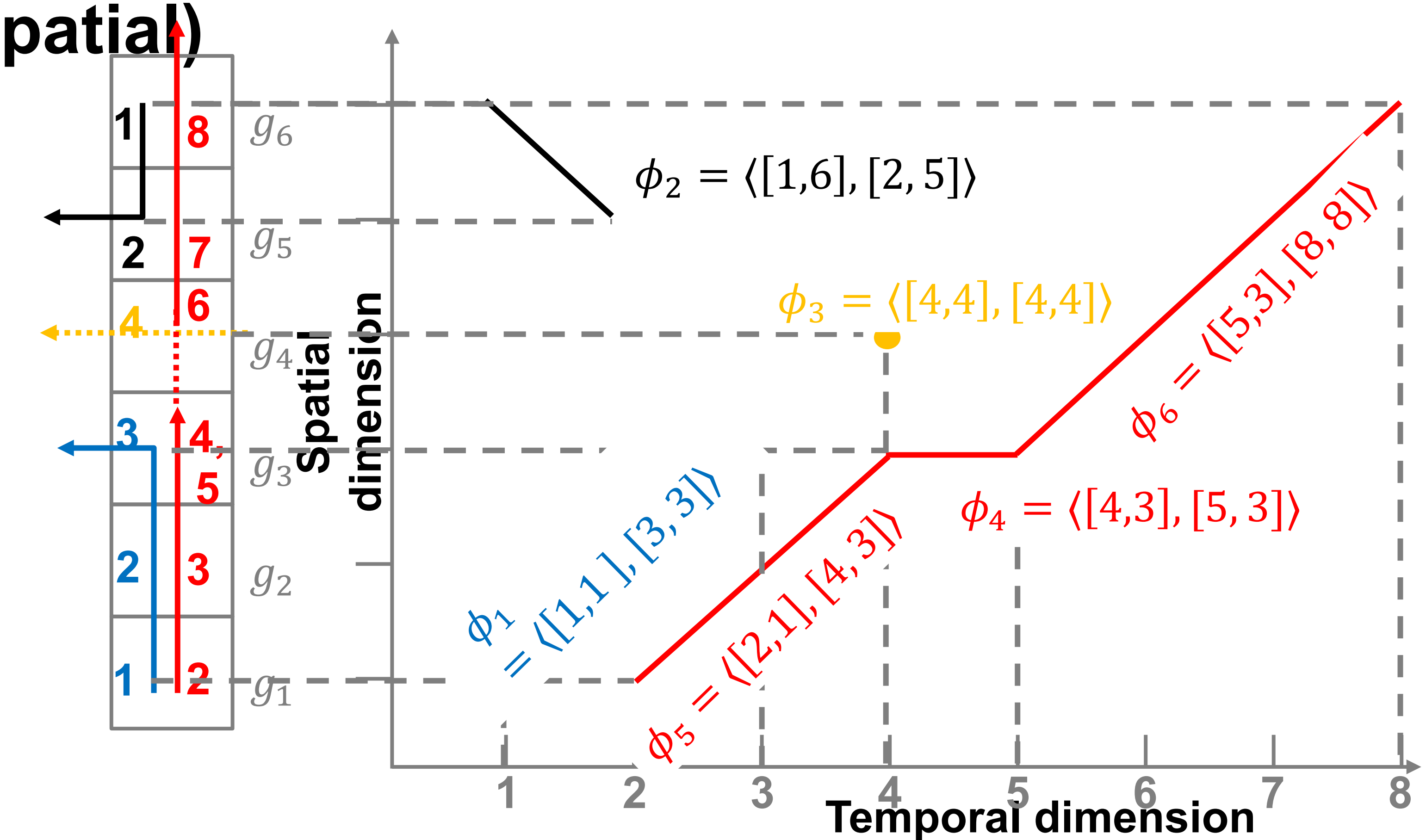
- **Workflow: Inter-strip Route Planning**
  - **Find the vertices that contain o, d**
  - **Perform shortest path on strip graph**

  - **Note: it specifies the edge weight by calling intra-strip planning when searching for a vertices.**



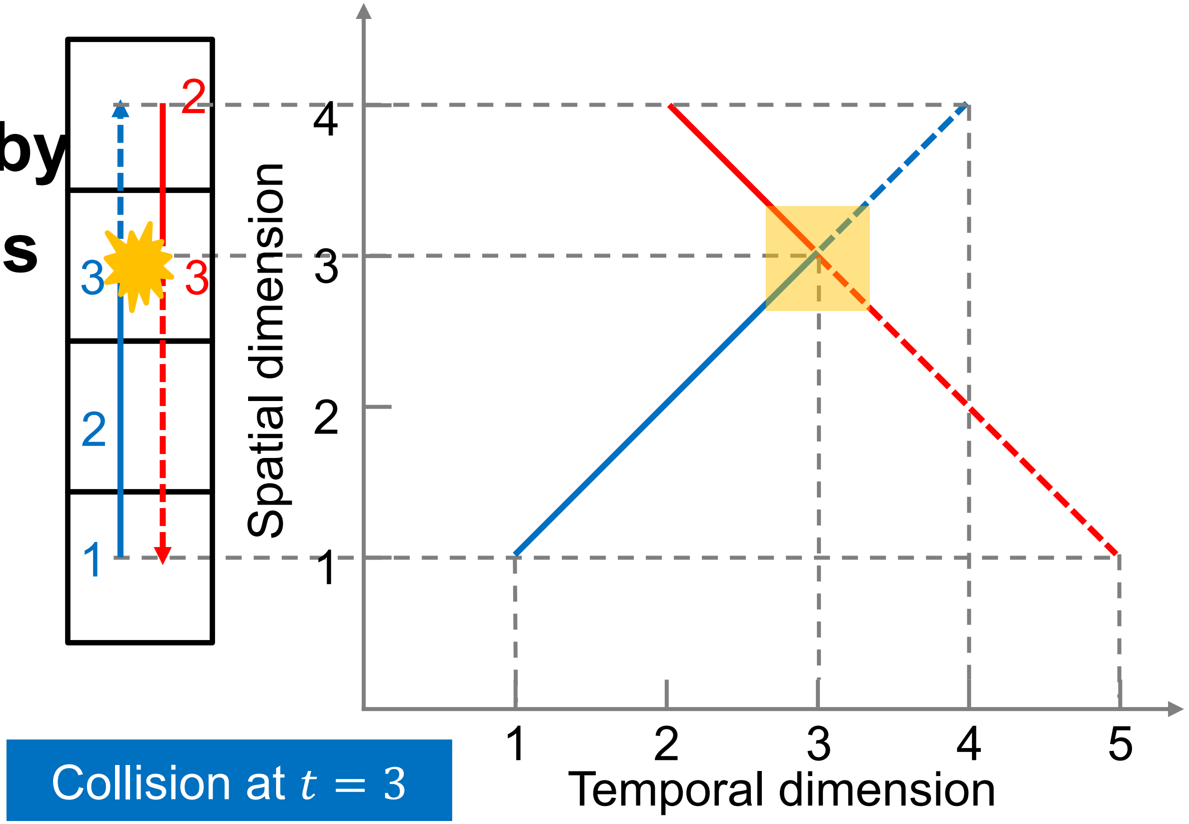
Our Solutions

- **Workflow: Intra-strip Route Planning**
  - **Searching a collision-free route within a 1-D spatial space**
  - **Each route form polylines in the 2-D search space (1-D temporal and 1-D spatial)**


1
8
g_6
2
7
g_5
4
6
g_4
3
4,
5
g_3
2
3
g_2
1
2
g_1
Spatial dimension
$\phi_2 = \langle[1,6],[2,5]\rangle$
$\phi_3 = \langle[4,4],[4,4]\rangle$
$\phi_6 = \langle[5,3],[8,8]\rangle$
$\phi_1 = \langle[1,1],[3,3]\rangle$
$\phi_5 = \langle[2,1],[4,3]\rangle$
$\phi_4 = \langle[4,3],[5,3]\rangle$
1
2
3
4
5
6
7
8
Temporal dimension

Our Solutions

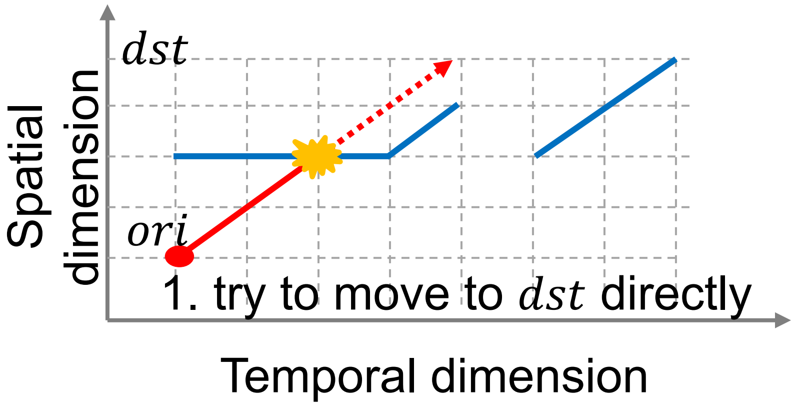
- **Workflow: Intra-strip Route Planning**
  - **Searching a collision-free route within a 1-D spatial space**
  - **Each routes form polylines in the 2-D search space (1-D temporal and 1-D spatial)**
  - **Collision can be easily detected by simply check segment intersections**

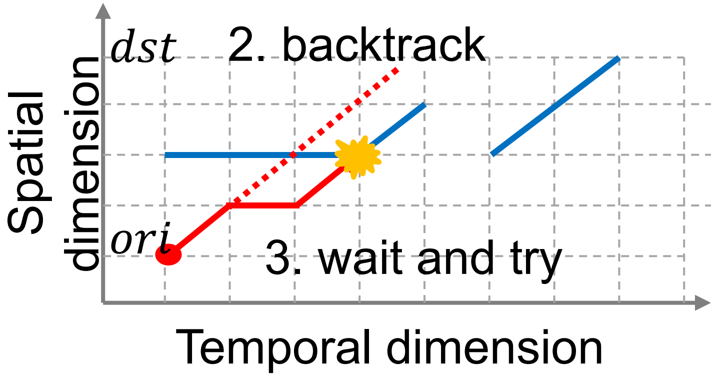


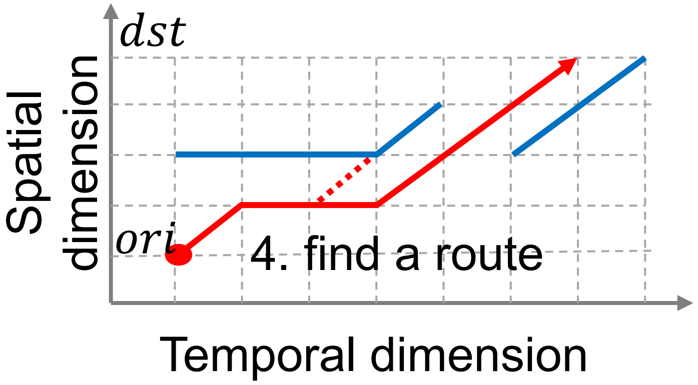
Collision at $t = 3$

Our Solutions

- **Workflow: Intra-strip Route Planning**
  - **A toy example**







Our Solutions

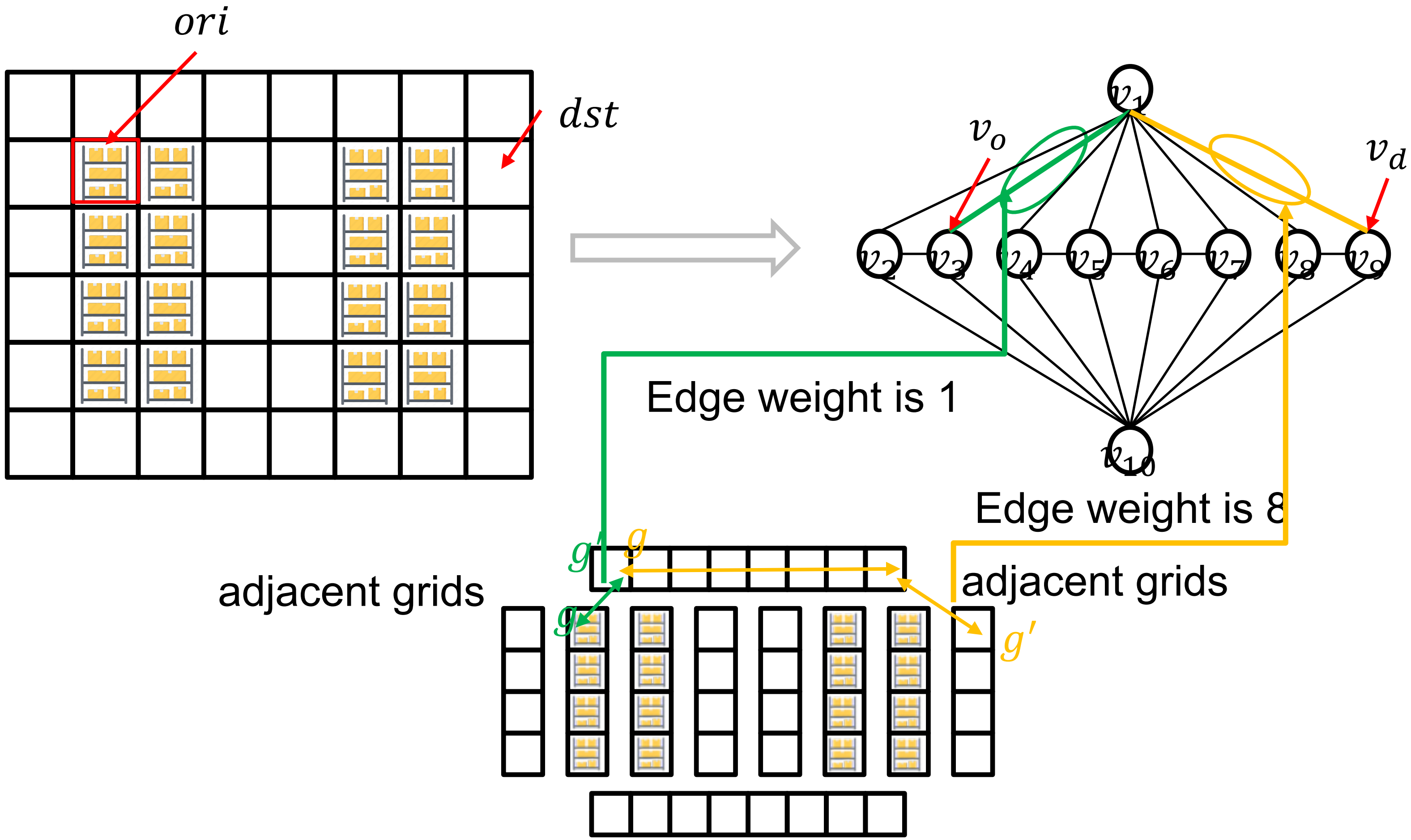
- **Workflow: Inter-strip Route Planning**
  - **Find the vertices that contain o, d**
  - **Perform shortest path finding on strip graph**

  - **Note: it specifies the edge weights by calling intra-strip planning when searching for a vertex.**

Recall

Our Solutions

- **Workflow: Inter-strip Route Planning**
  - **When finding a route within a strip, it can return a time cost for inter-strip level**



Edge weight is 1

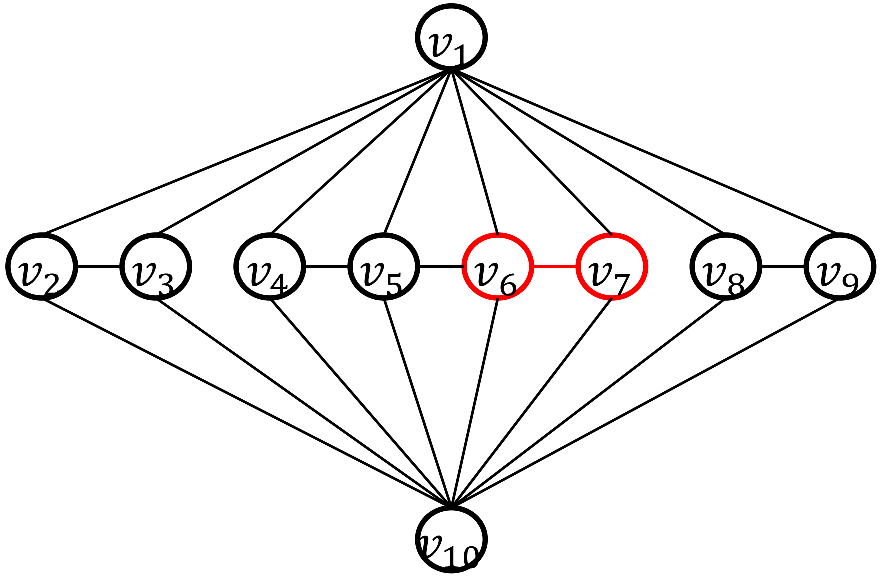
Edge weight is 8

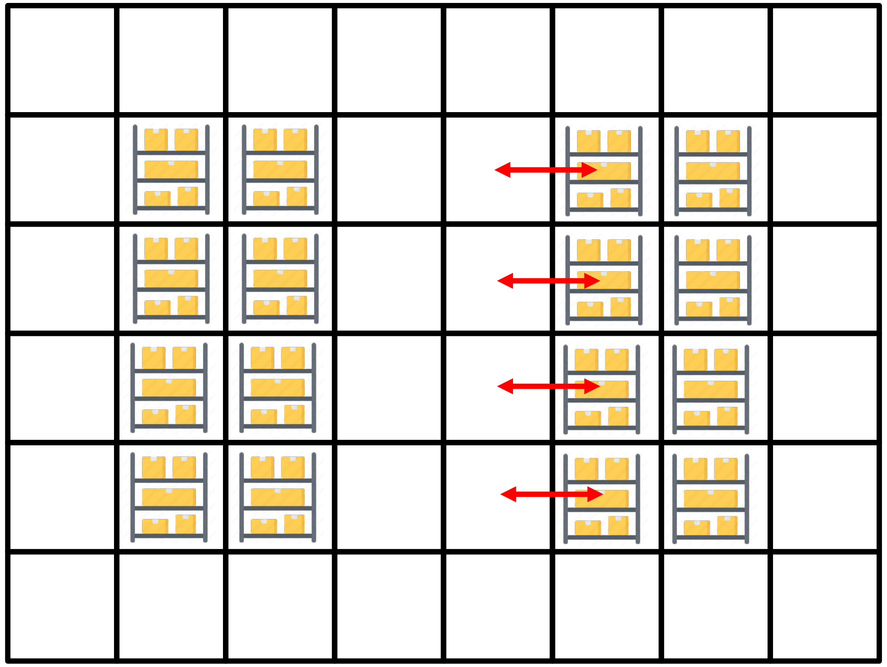
adjacent grids

adjacent grids

Our Solutions

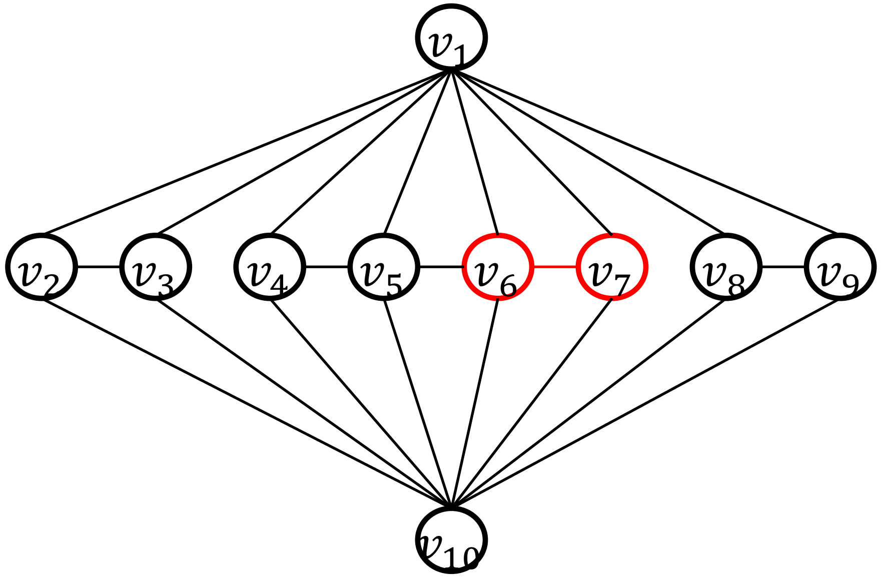
- **Other details: Multiple adjacent grids issue**
  - **Two strips may have multiple adjacent grids. However, strip graph cannot distinguish along which adjacent grid the route will transfer to other strips (only one edge).**

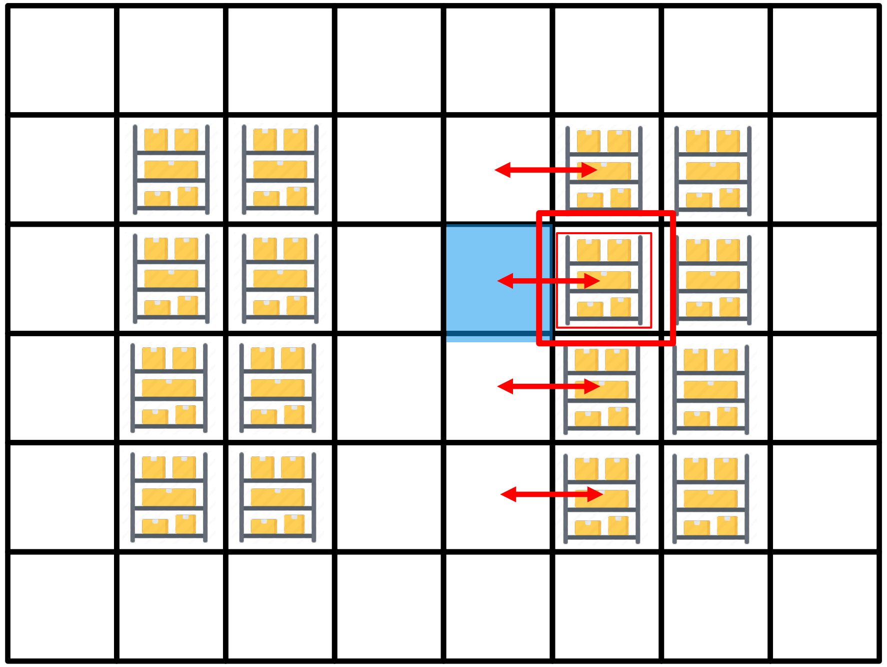




Our Solutions

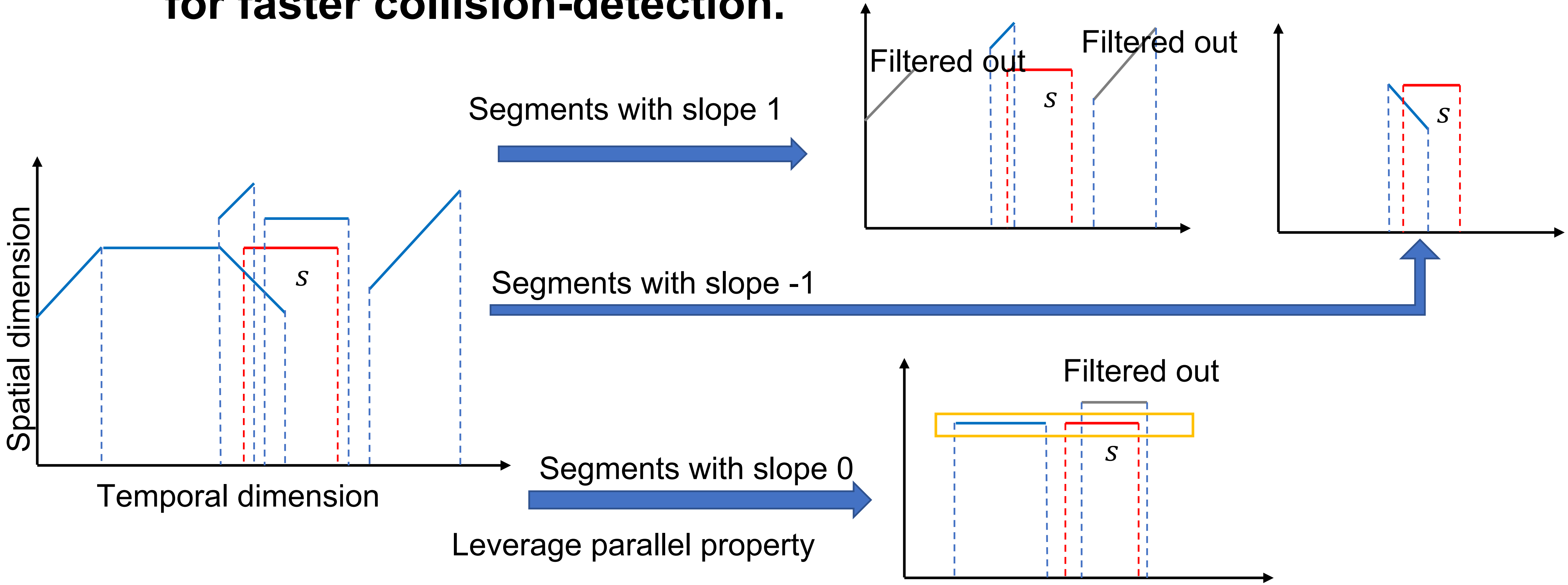
- **Other details: Multiple adjacent grids issue**
    - **Two strips may have multiple adjacent grids. However, strip graph cannot distinguish along which adjacent grid the route will transfer to other strips (only one edge).**
    - **Simply adopt a lazy change strategy: ie leave the current aisle as late as possible. (pick the red rack will change at blue grids)**





Our Solutions

- **Other details: Index for acceleration**
  - **We manage planned routes (a set of segments) into different sets by their slopes. Then build index based on the time-span for faster collision-detection.**

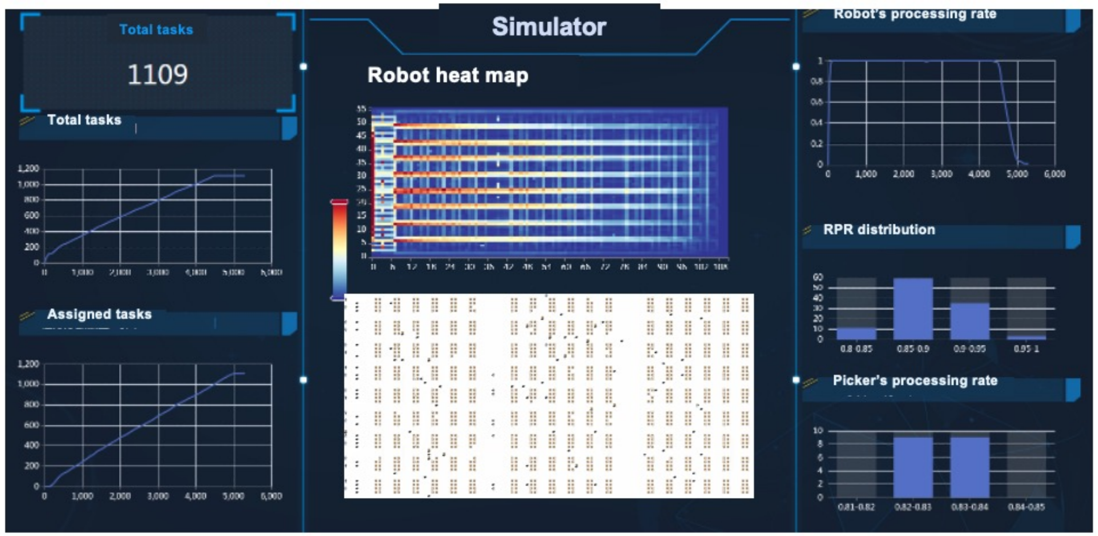


Outline

- 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.**
- **Running Information**
  - **CPU: CPU Intel(R) Xeon(R) Platinum 8269CY CPU T 3.10GHz**
  - **Memory: 20GB**



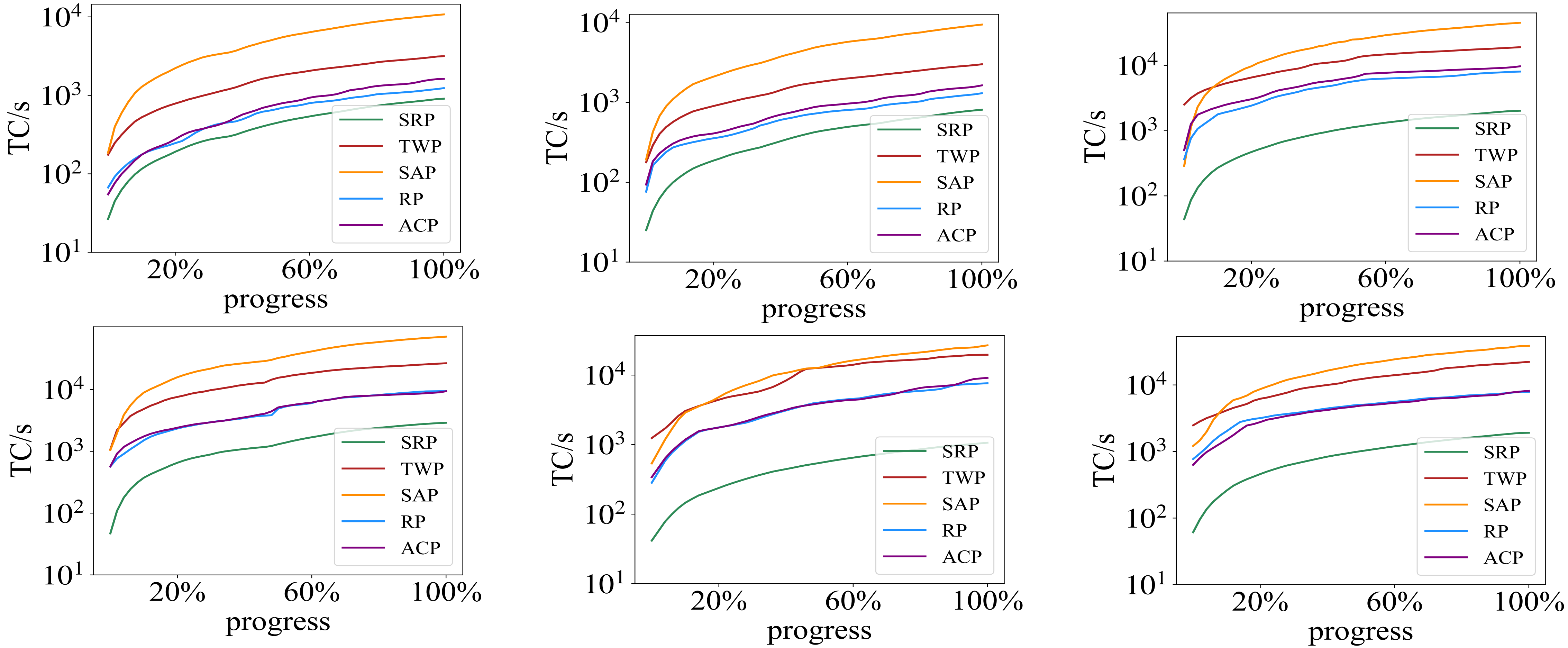
Experiments

- **Compared methods**
  - **Simple A* Planning(SAP): Search directly in a 3-D space.**
  - **Replanning(RP)[1]: Planning ignoring collisions, if collision occurs then replanning**
  - **Time Window-based Planning (TWP)[2]: Planning only in a time-window horizon**
  - **Adaptive Cached Planning (ACP)[3]: Use a cache for previous situation**
- **Evaluation metrics**
  - **Makespan: Total time duration from the first query emerges till the last query finished**
  - **Time consumption (TC): execution time cost in sections**
  - **Memory consumption (MC): execution memory cost in 100MB**

[1] J.Sˇvancara, et al. Onlinemulti- agent pathfinding. AAAI, 2019.
[2] J.Li, et al. Lifelong multi-agent path finding in large-scale warehouses. AAAI, 2021.
[3] D. Shi, et al. Adaptive taskplanning for large-scale robotized warehouses. ICDE, 2022.

Experiments

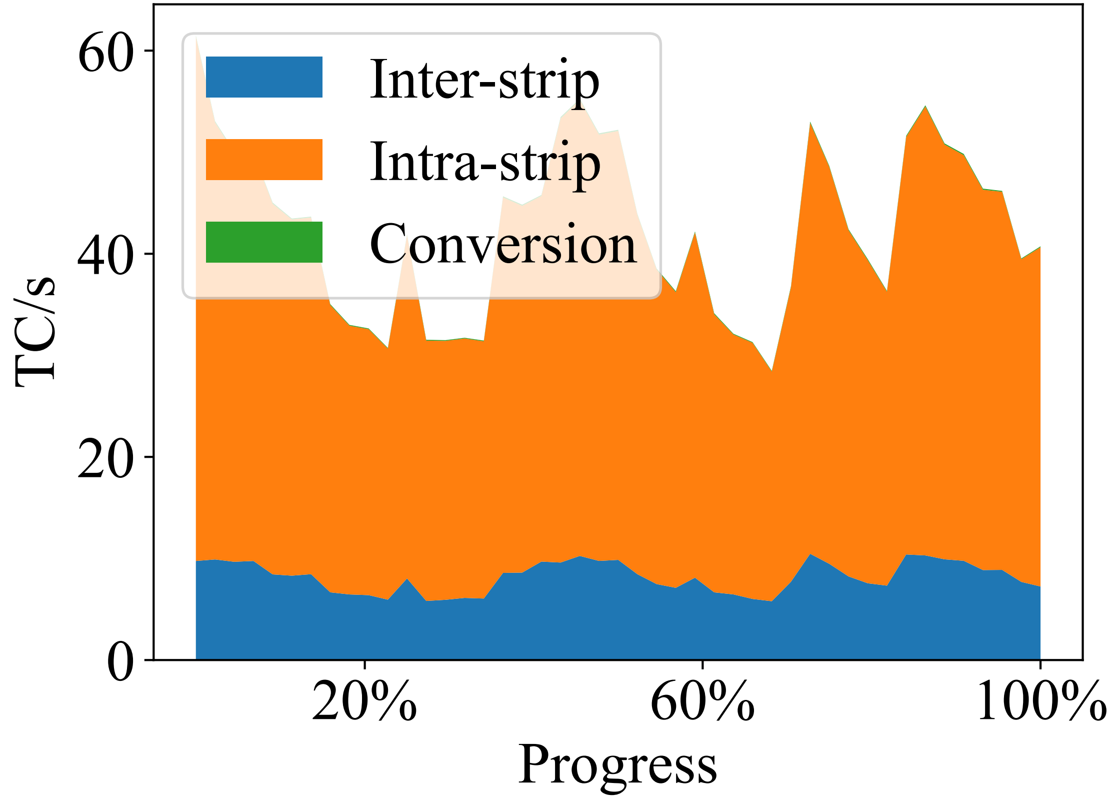
- **Time efficiency**

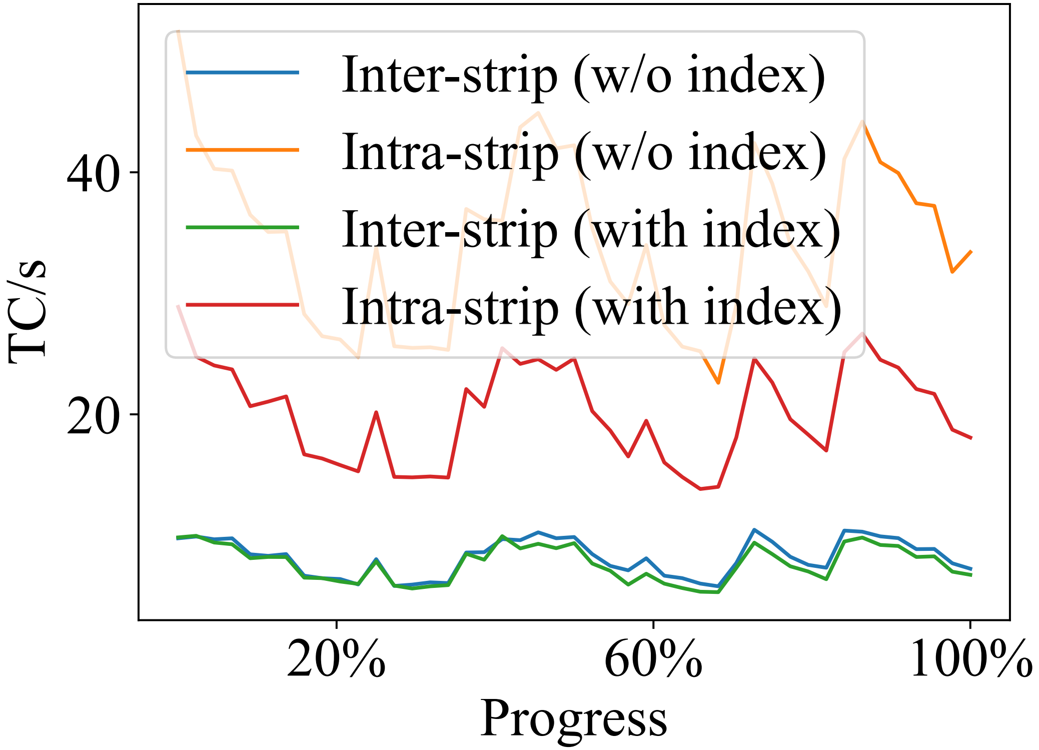


Our SRP can averagely outperform 37.3x than other methods

Experiments

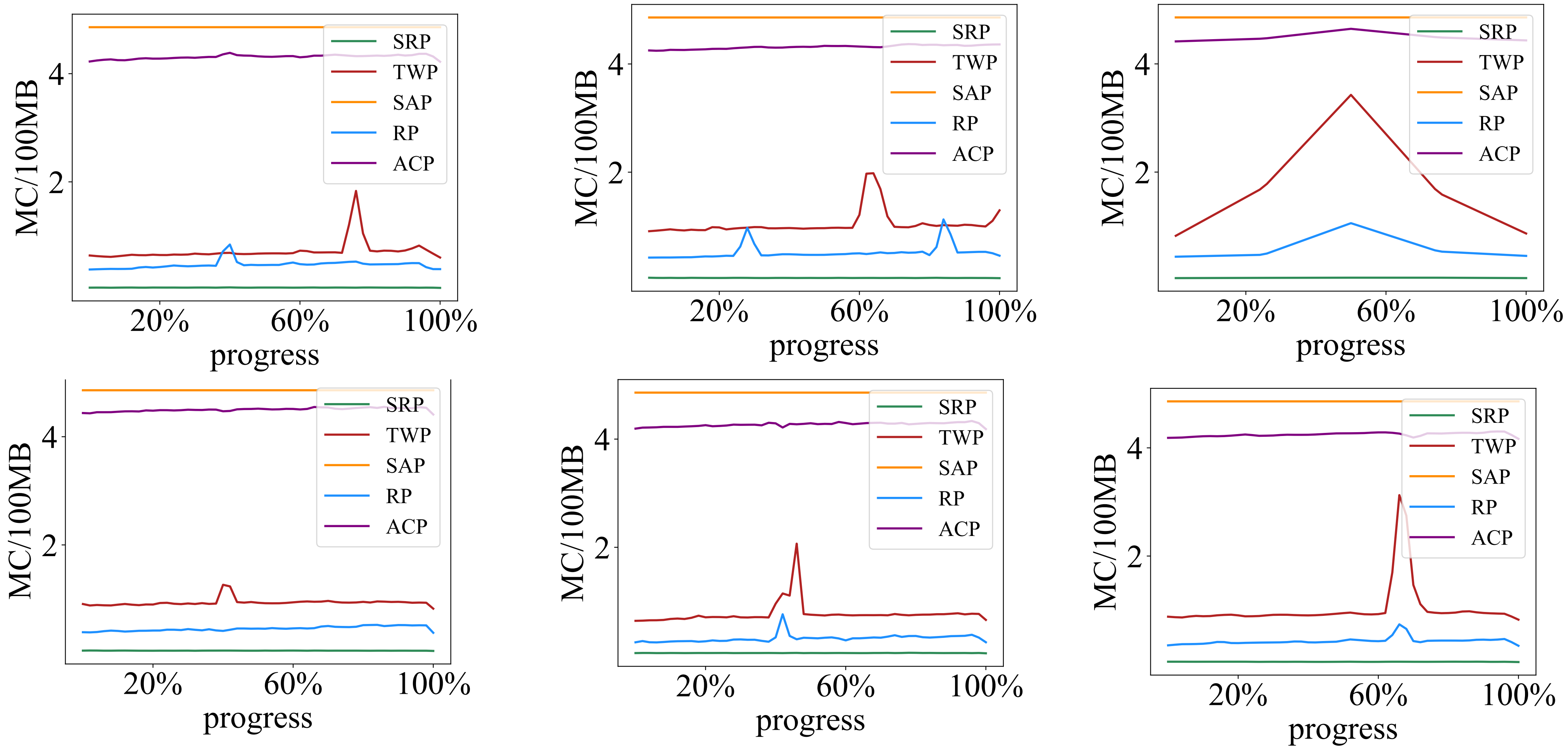
- **Time cost breakdown**
  - **Intra-strip cost takes 80%**
  - **Index helps reducing time cost by 50%**





Experiments

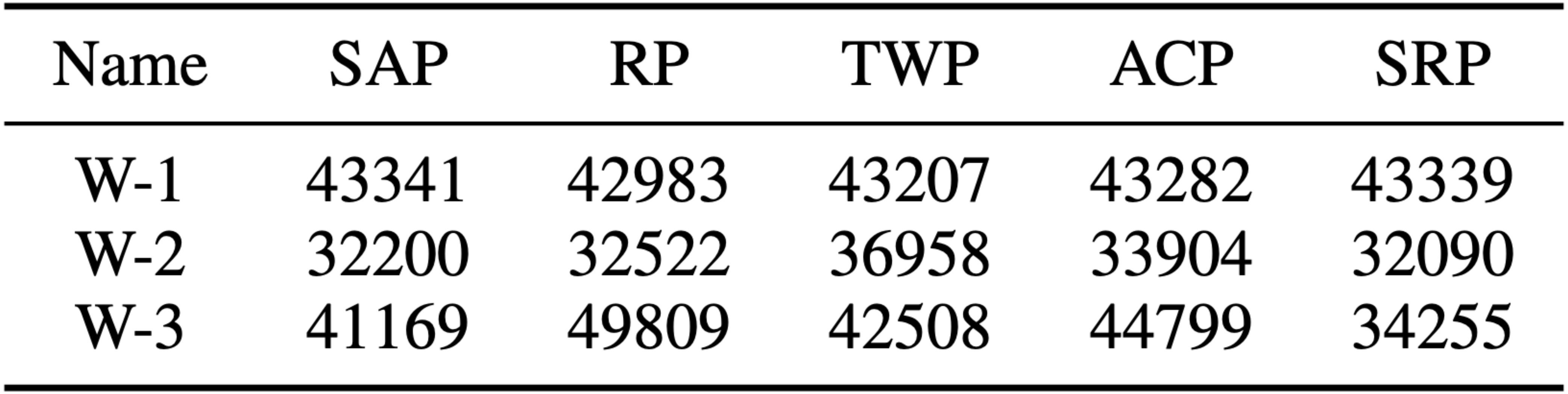
- **Memory efficiency**



Our SRP steadily beat other methods over all different datasets

Experiments

- **Effectiveness**



Name	SAP	RP	TWP	ACP	SRP
W-1	43341	42983	43207	43282	43339
W-2	32200	32522	36958	33904	32090
W-3	41169	49809	42508	44799	34255

Our SRP still maintains a competitive effectiveness.

Outline

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

Conclusion

- We propose a strip-based framework to replace the widely adopted grid-based warehouse representation.

- We devise an efficient route planning algorithm which contains inter- and intra- strip stages on new framework.

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

Q & A

Thank You