Robustness in Multi-Agent Pickup and Delivery

Giacomo Lodigiani
giacomo.lodigiani@mail.polimi.it
T2A Track

HONOURS PROGRAMME HP-SR in Information Technology

POLITECNICO MILANO 1863

Introduction

MAPF

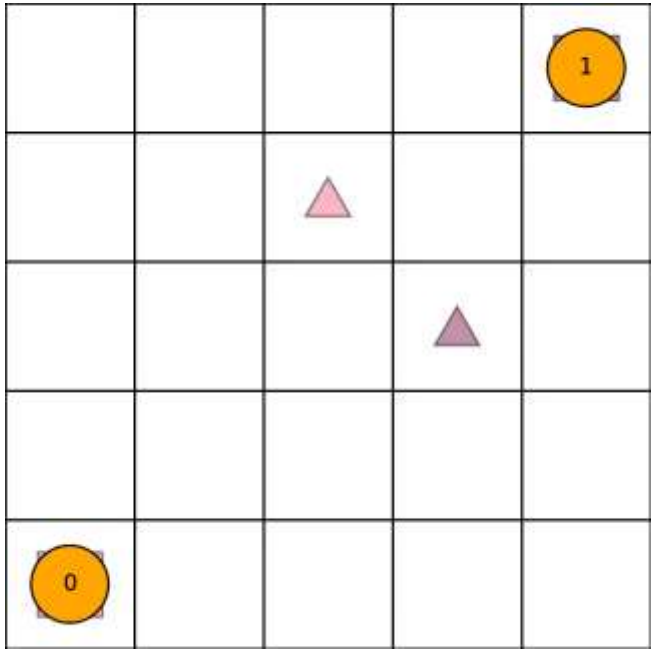
Multi-Agent Path Finding (**MAPF**) [Stern et al., 2019] is a problem in the broader field of Multi-Robot Systems in which multiple agents must plan paths to preassigned targets, avoiding collisions and optimizing some cost function.

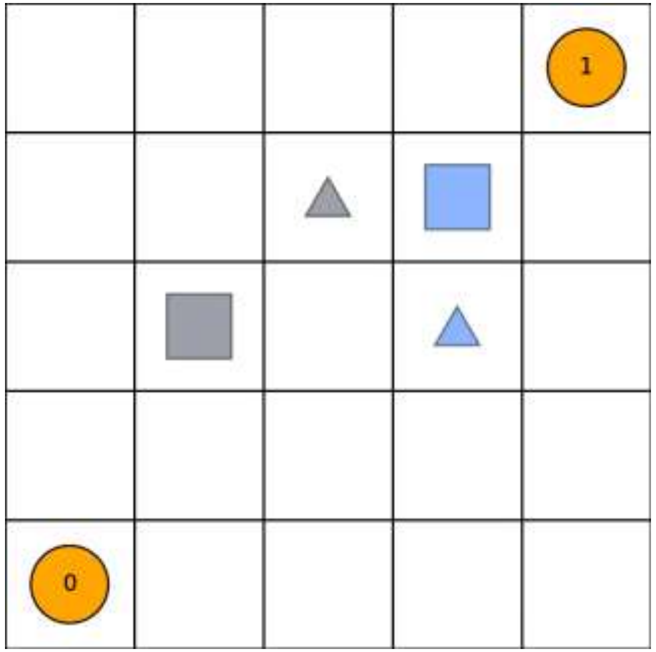
MAPD

Multi-Agent Pickup and Delivery (**MAPD**) [Ma, 2020] is an extension of MAPF in which agents have also the freedom to assign themselves to targets which, unlike in the MAPF problem, are not a fixed set but may **change** at any time step.

POLITECNICO
MILANO 1863

Example



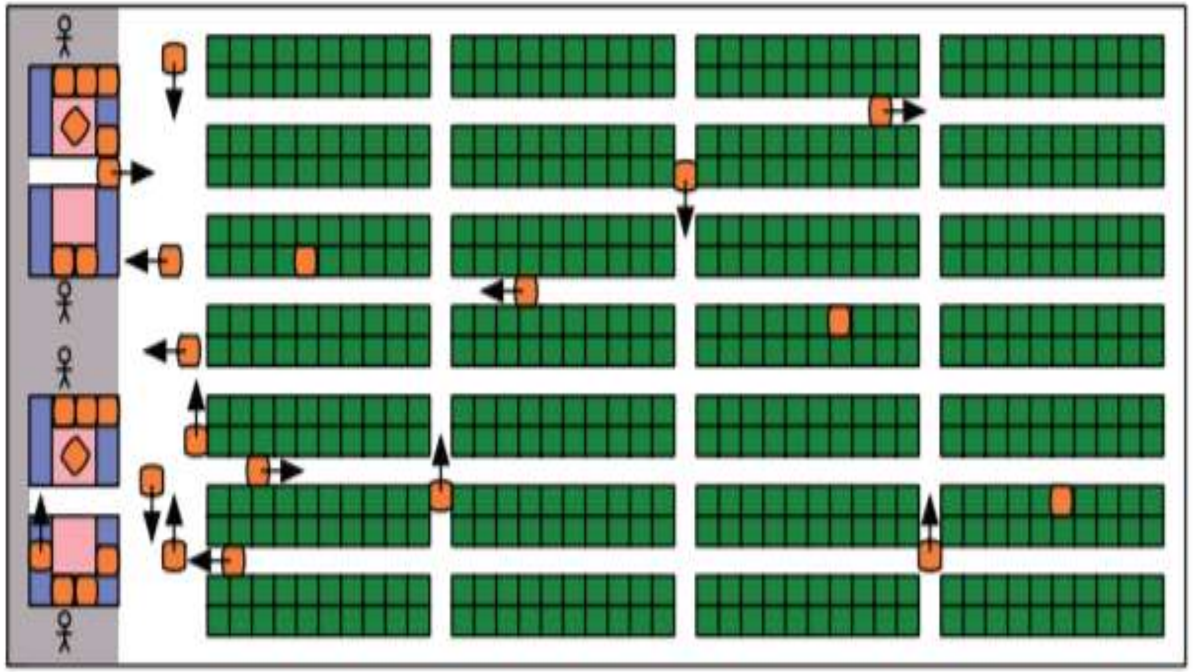


MAPF

MAPD

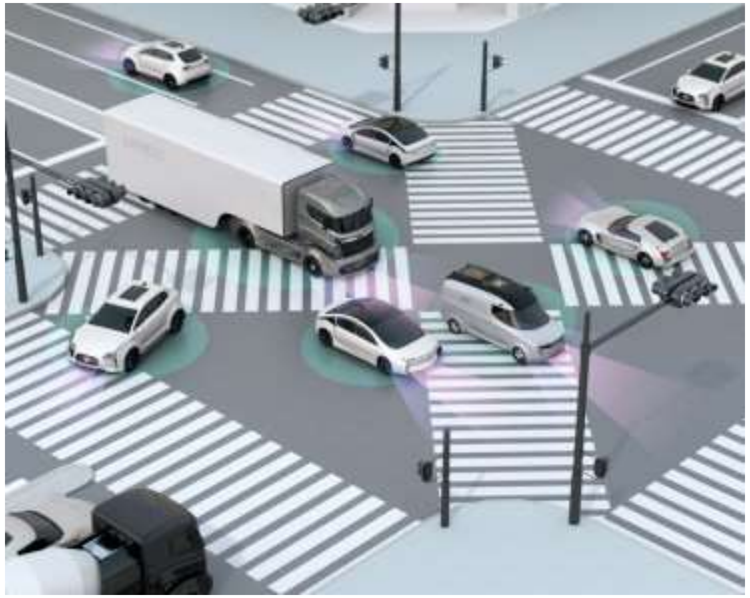
POLITECNICO
MILANO 1863

Applications



Logistics

Applications



Autonomous vehicles

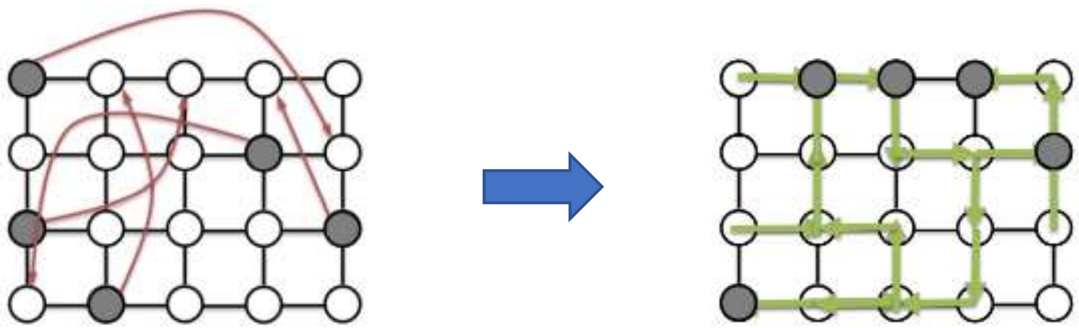
Aircraft towing

Videogames

MAPF

MAPF

- MAPF problem instance: undirected graph $G = (V, E)$, set of l agents $\{a_i | i \in [l]\}$.
- Each agent a_i has a start vertex $a_i \in V$ and a goal vertex $g_i \in V$.
- A MAPF *plan* consists of a *path* (sequence of actions leading an agent from its start location to its goal) π_i for each agent a_i. A MAPF solution is a MAPF plan whose paths are **collision-free**. The problem of MAPF is to find a solution minimizing a given cost measure, like *makespan* or *flowtime.*



POLITECNICO
MILANO 1863

MAPD

MAPD

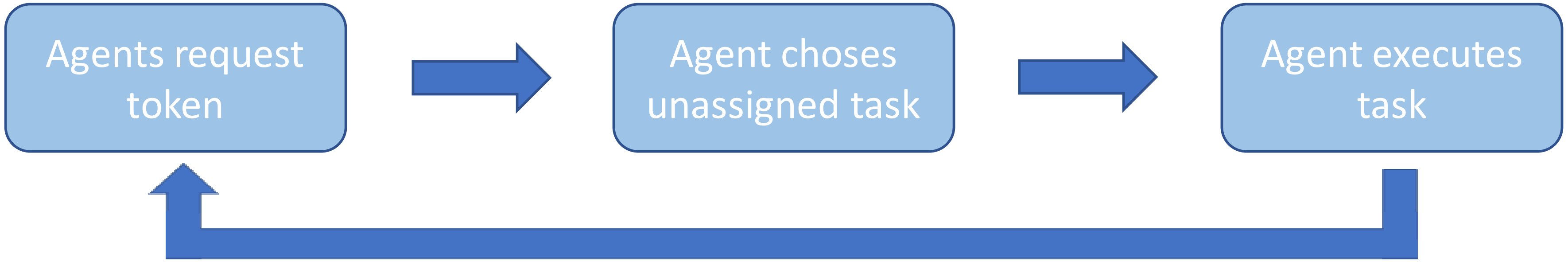
- MAPD problem instance: undirected graph $G = (V, E)$, set of l agents $\{a_i | i \in [l]\}$, **task set** T.
- Tasks are not preassigned to agents.
- **Task set** T changes **dynamically** as, at each time step, new tasks can be added to the system.
- A task can be assigned to one free agent at a time.
- The problem of MAPD is to find **collision-free** paths for the agents to finish executing all tasks, minimizing a cost measure like *makespan* or *service time.*

POLITECNICO MILANO 1863

MAPD algorithms

Token Passing [Ma et al., 2017]

Token Passing (TP) is a **decoupled** algorithm based on a token, a synchronized shared block of memory that contains the current paths of all agents, the task set, and the task assignments that record which task is currently assigned to which agent.



POLITECNICO
MILANO 1863

Complexity

MAPF complexity

MAPF is **NP-hard** to solve optimally on general graphs for both makespan minimization and flowtime minimization [Yu and LaValle, 2013].

MAPD complexity

Being a generalization of MAPF, also MAPD is **NP-hard** to solve optimally [Ma, 2020].

POLITECNICO
MILANO 1863

Robustness

Robustness

Property of a MAPF or MAPD **algorithm** of being able to **complete all** the **tasks** even in case some unexpected event forces some deviation of the execution from the original plan. For MAPF:

- k-robustness.
- p-robustness.

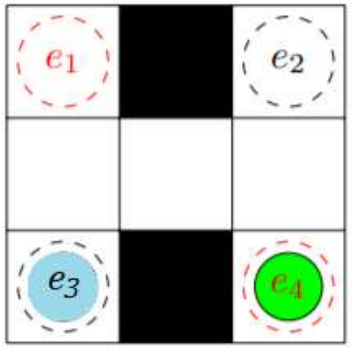
Long-term robustness [Ma, 2020]

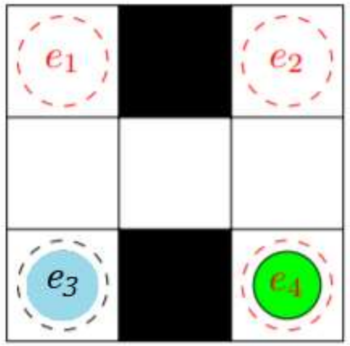
Feasibility condition of a MAPD problem instance (also called well-formedness):

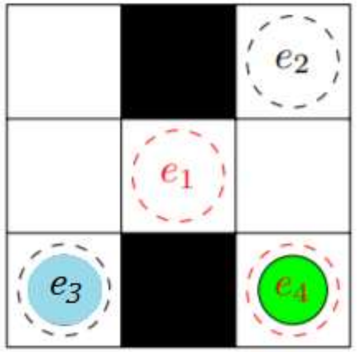
1. The number of tasks is finite.
2. There are no fewer non-task endpoints (designated vertices that allow the agents to rest) than the number of agents.
3. For any two endpoints, there exists a path between them that traverses no other endpoints.

POLITECNICO
MILANO 1863

Long-term robustness examples







Well-formed

Non well-formed

Non well-formed

Purpose of the research

Theory vs. applications

- Robustness helps to overcome some idealistic assumptions made by the models: in reality delays and other issues can hinder some properties (e.g., the absence of collisions) of a solution.
- Robustness in the time-extended setting of MAPD has not been yet consistently studied. The goal of this research is to contribute to bridge this gap by studying the impact of delays in the MAPD problem and by proposing solutions useful in real applications.

POLITECNICO
MILANO 1863

Contributions of the research

Theoretical contribution

Robustness of MAPD to the occurrence of **delays** is studied by defining a variant of the problem called **MAPD-d** and analyzing some of its properties.

Algorithmic contribution

Two new algorithms are proposed, **k-TP** and **p-TP**, which adopt the approach of **robust planning**, computing paths that limit the risk of collisions caused by potential delays.

Experimental contribution

Algorithms are compared by running experiments in simulated environments and the **trade-offs** offered by different levels of robustness are evaluated.

POLITECNICO
MILANO 1863

MAPD with delays

MAPD-d

- MAPD with delays (MAPD-d) is a MAPD problem where the execution of the computed paths can be affected, at any time step t, by delays.
- The actual sequence of vertices traversed by an agent is called **execution trace**.
- Delays are represented by a **time-varying set** $D(t)$, which specifies the subset of agents that delay the execution of their paths during time step t.
- The temporal realization of $D(t)$ is unknown, so a solution is computed accounting for robustness to delays that might happen.

POLITECNICO
MILANO 1863

MAPD with delays

Long-term robustness of MAPD-d

- Delays only affect execution and do not harm long-term robustness, with the only exception being when an agent is delayed indefinitely.
- A fourth condition is added to guarantee long-term robustness:

4. There is not any agent that belongs to $D(t)$ for all $t \geq T$.

POLITECNICO MILANO 1863

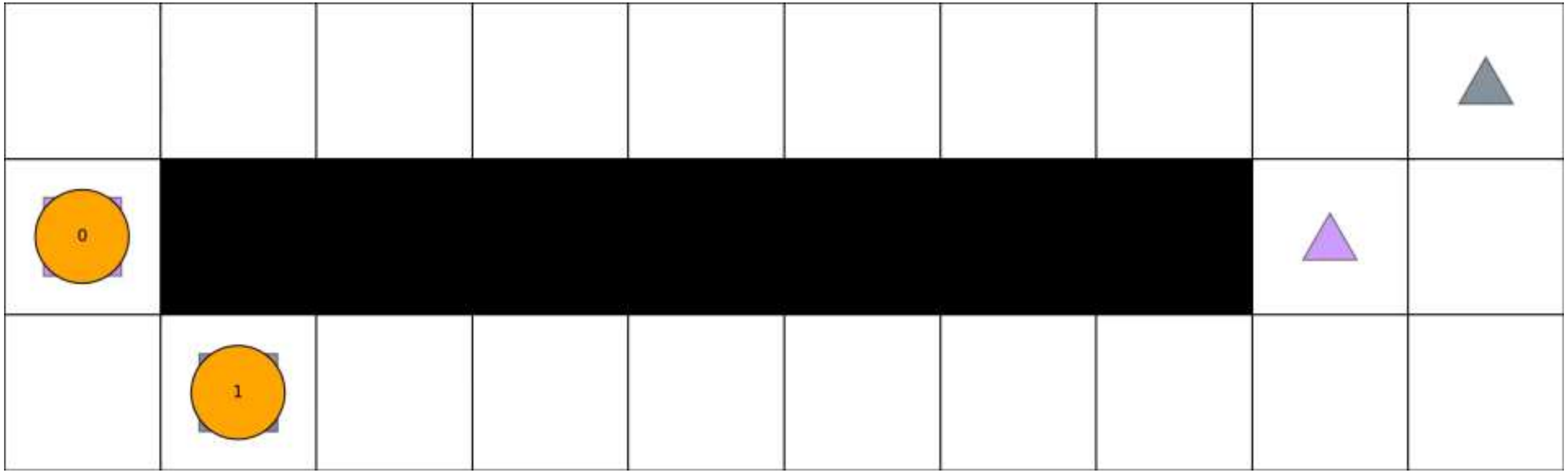
MAPD-d algorithm: TP with recovery routines

TP with recovery routines

- Algorithms able to solve well-formed MAPD problems, like Token Passing (TP), are in principle able to solve well-formed MAPD-d problems.
- Problem: delays cause planned paths to possibly collide.
- Solution: add a simple recovery routine to the TP algorithm such that, when a collision is detected, it assigns the token to one of the colliding agents to allow replanning.
- Used as **baseline** algorithm.

POLITECNICO
MILANO 1863

Example

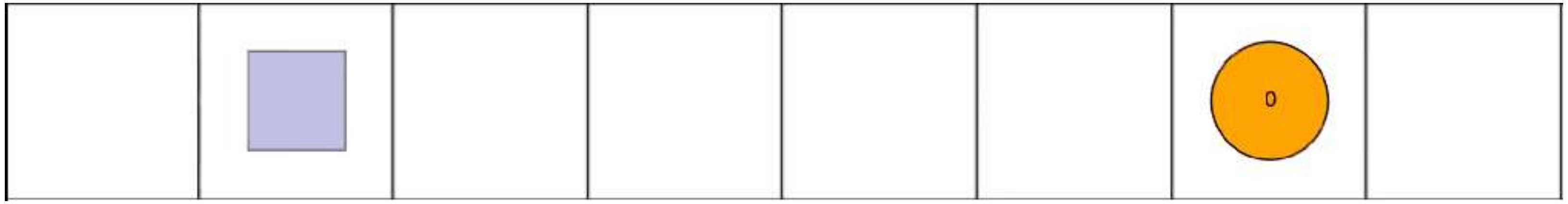


MAPD-d algorithm: k-TP

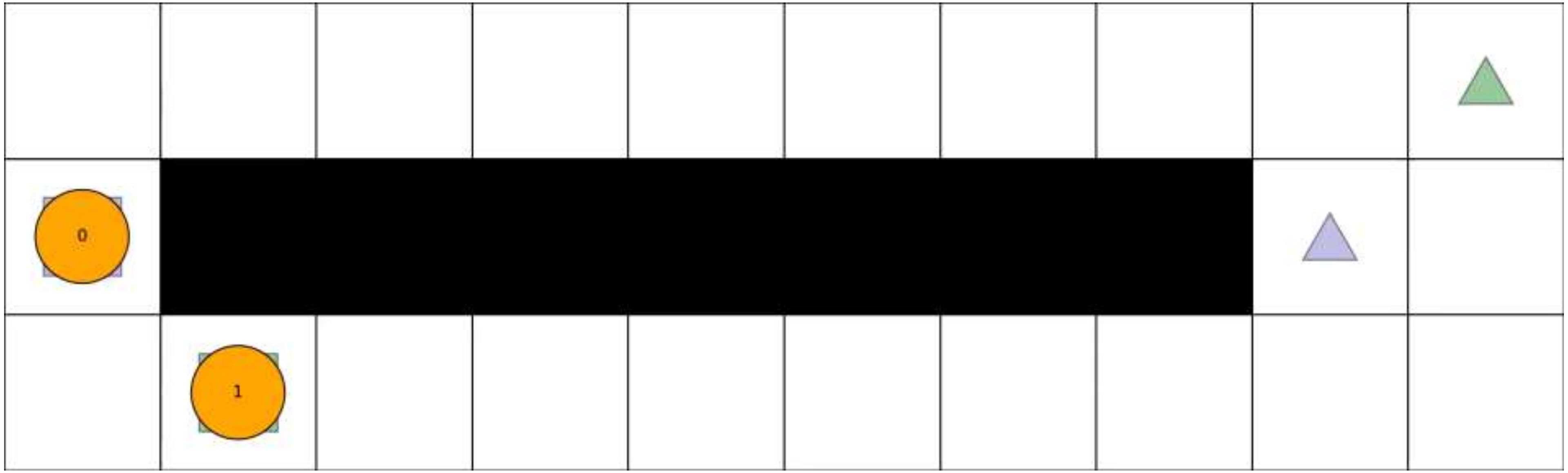
k-TP

- Instead of reacting to delays, plan considering that delays may occur, reducing the need of replanning.
- Adds the concept of k-robustness to TP.
- Collisions due to delays should be avoided not only considering the paths already planned, but also those planned in the future.
- New constraints are added when planning, the **k-extension** of the path. These constraints prevent a vertex from being occupied again for k time steps before and after it has been occupied by agent.

k = 2



Example



MAPD-d algorithm: p-TP

p-TP

- Instead of reacting to delays, plan considering that delays may occur, reducing the need of replanning.
- Adds the concept of p-robustness to TP.
- Suppose an agent has a probability p_d of being delayed at any time step.
- Use **Markov chains** to calculate the probability of an agent a_i of being in a vertex of its planned path π_i at a certain time step. Hence, for any vertex traversed by the path π_i we calculate its collision probability. All the probabilities of the steps along the path are summed to obtain the collision probability for the path π_i.
- Finally, set a maximum **threshold** for collision probability and add a new path to the token only if its collision probability is no greater than the threshold.

Algorithms comparison

k-TP vs. p-TP

Differently from TP with recovery routines, both k-TP and p-TP offer robustness guarantees, limiting the impact of potential delays.

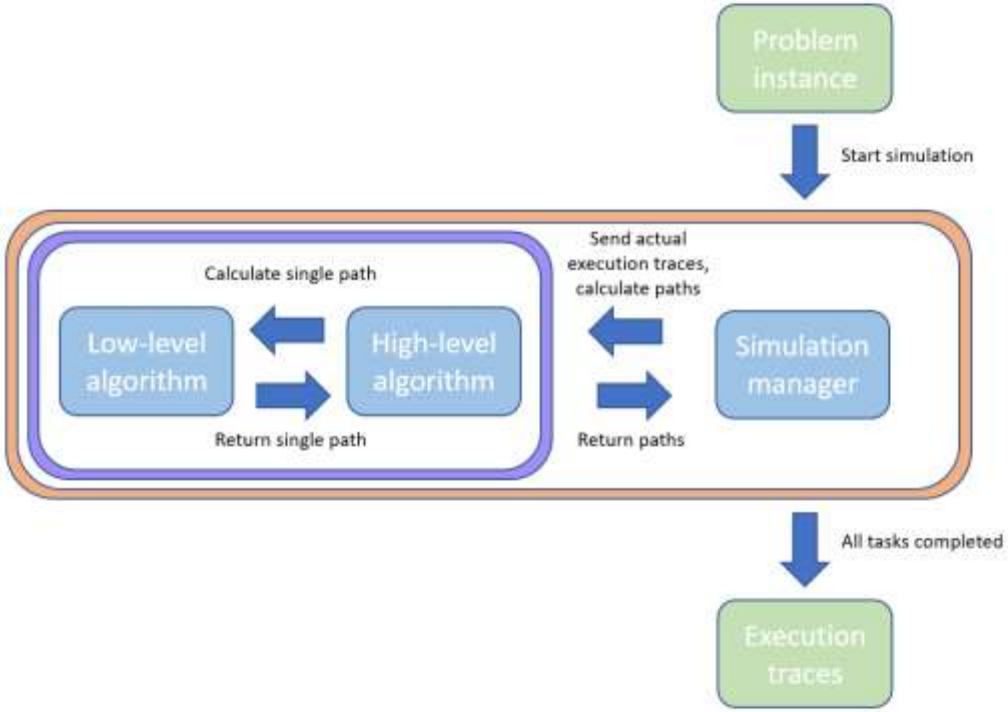
- k-TP gives a deterministic guarantee: even if each agent is delayed up to k times, no collision occurs.
- p-TP gives a probabilistic guarantee: new paths have a collision probability no grater than a given threshold.

POLITECNICO
MILANO 1863

Experimental setting

Simulation pipeline

Given a MAPD-d problem instance, allows to select a robust algorithm to solve it and outputs the execution traces of all the agents.



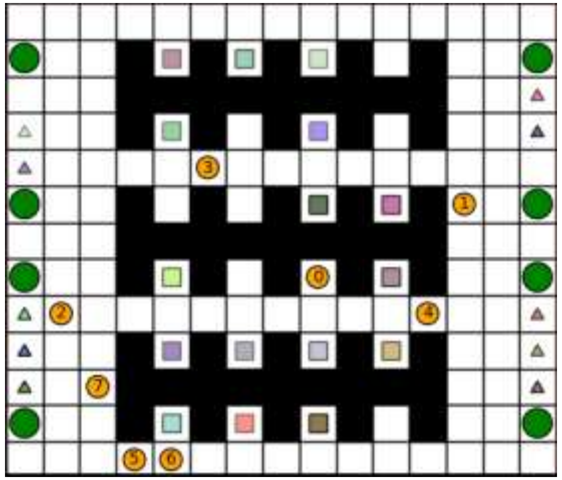
POLITECNICO
MILANO 1863

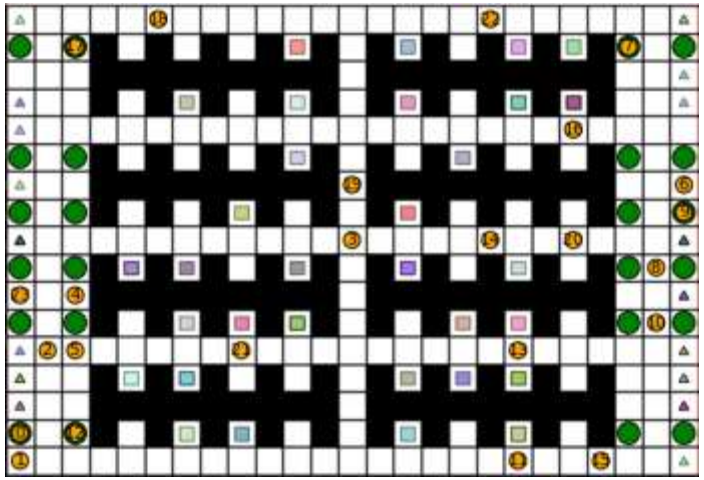
Experimental setting

Settings

- Experiments have been run in several simulated warehouse environments that are standard in the literature [Wurman et al., 2007] [Ma et al., 2017].
- Algorithms have been compared with different task frequencies, number of delays, number of agents.
- Metrics: total **cost of a solution**, **number of replans** during execution, **runtime**.

Small warehouse

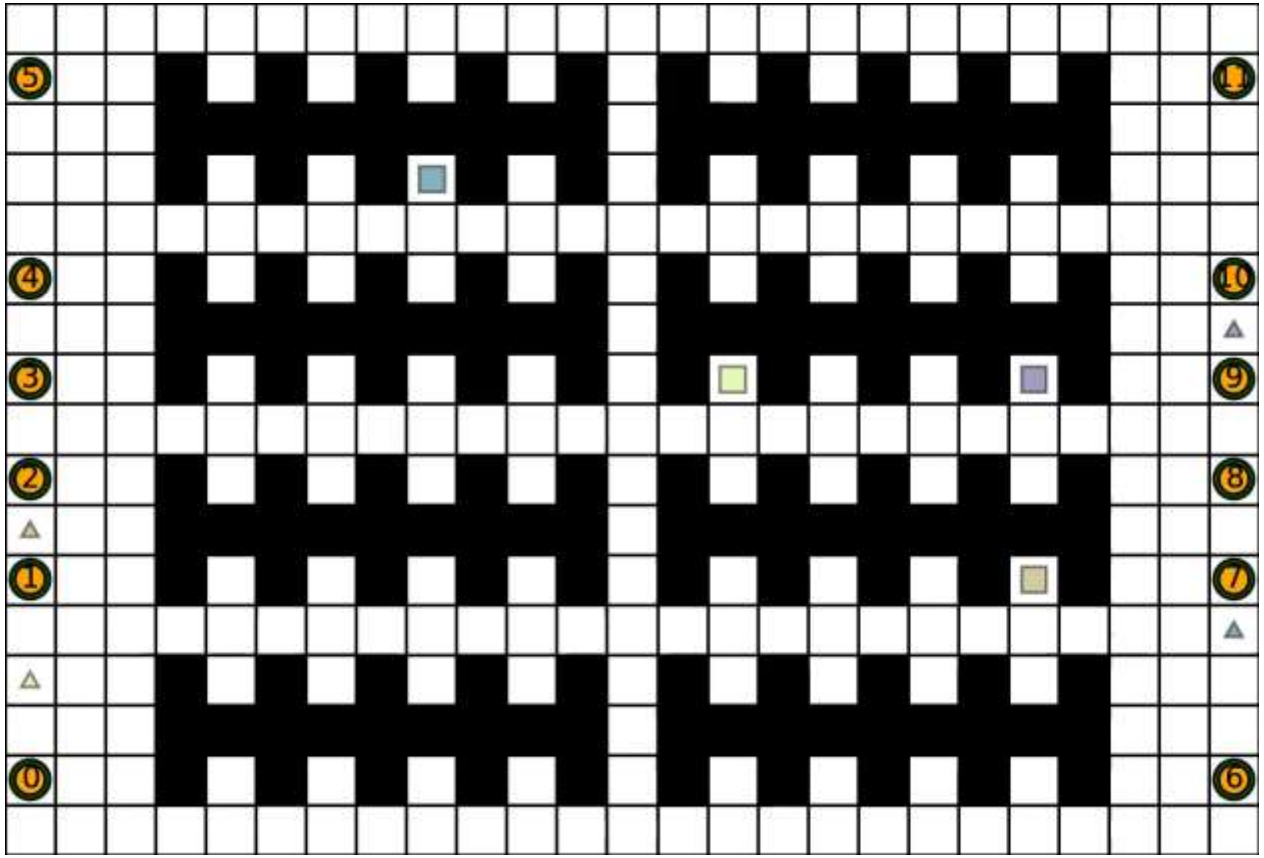




Large warehouse

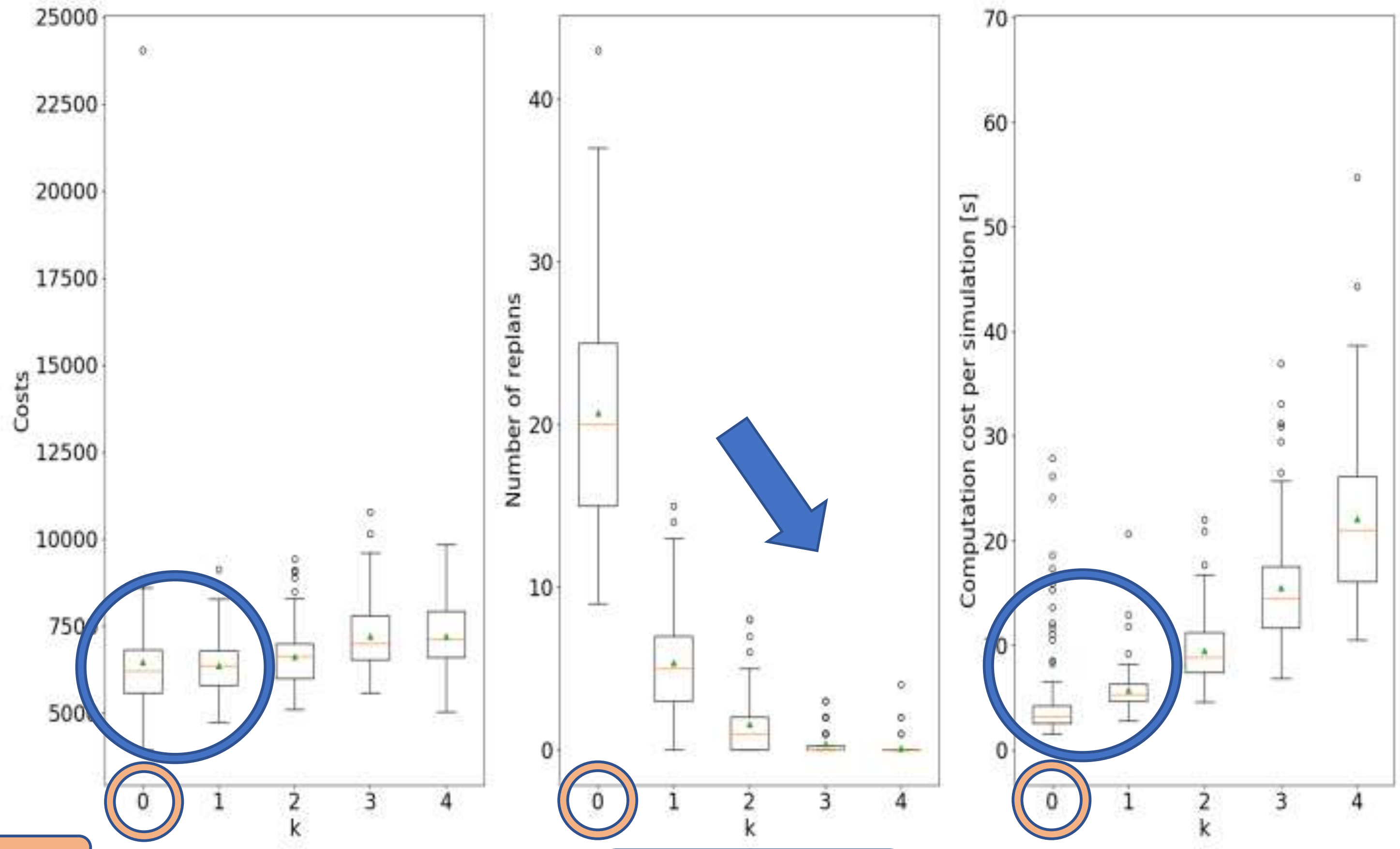
POLITECNICO MILANO 1863

Example



Experimental results

- Large warehouse
- 24 agents
- 10 delays per agent
- 50 tasks



Baseline

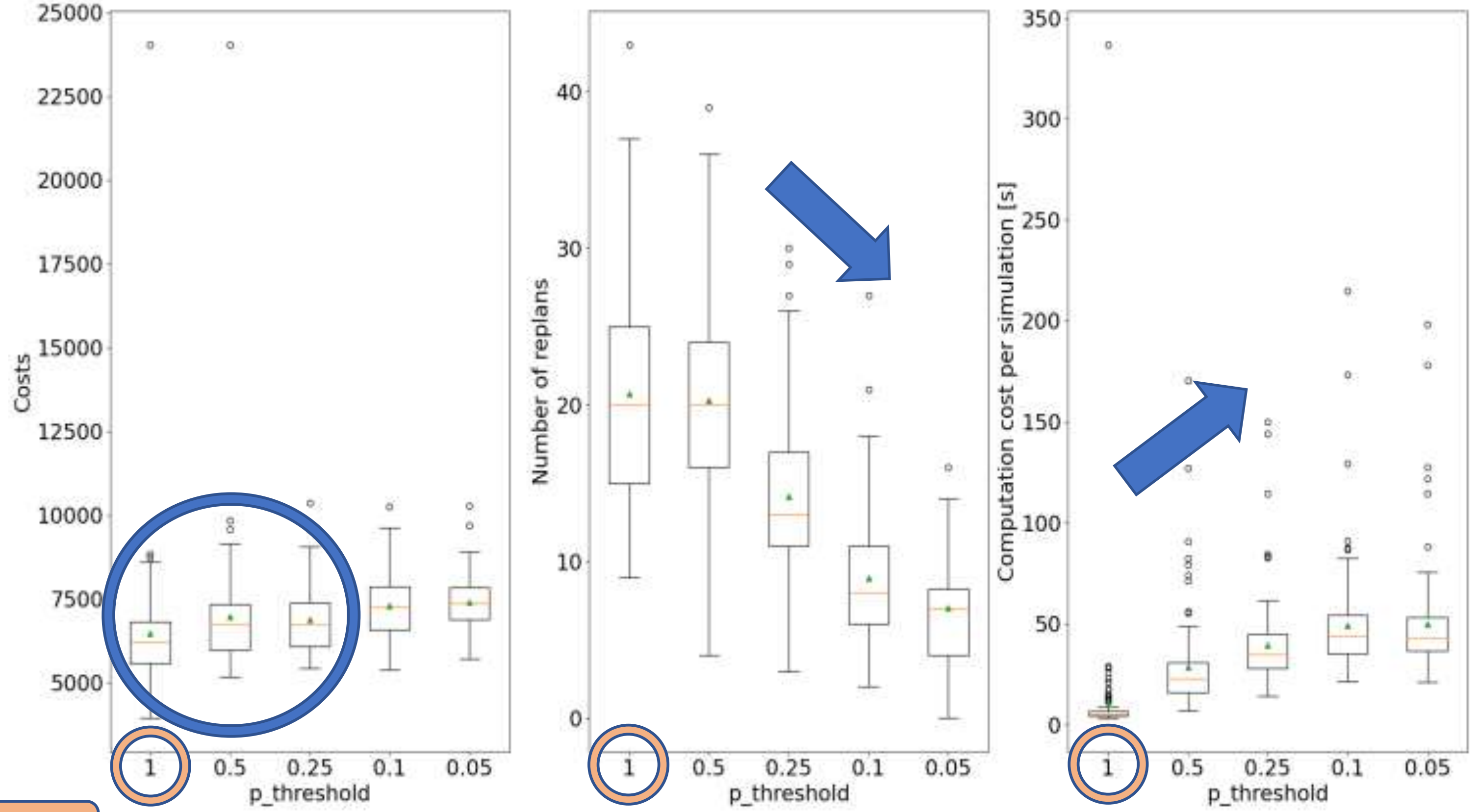
k-TP

- Replans greatly reduced
- Small or no increase in cost
- Small or no increase in runtime (for low k)

POLITECNICO MILANO 1863

Experimental results

- Large warehouse
- 24 agents
- 10 delays per agent
- 50 tasks



Baseline

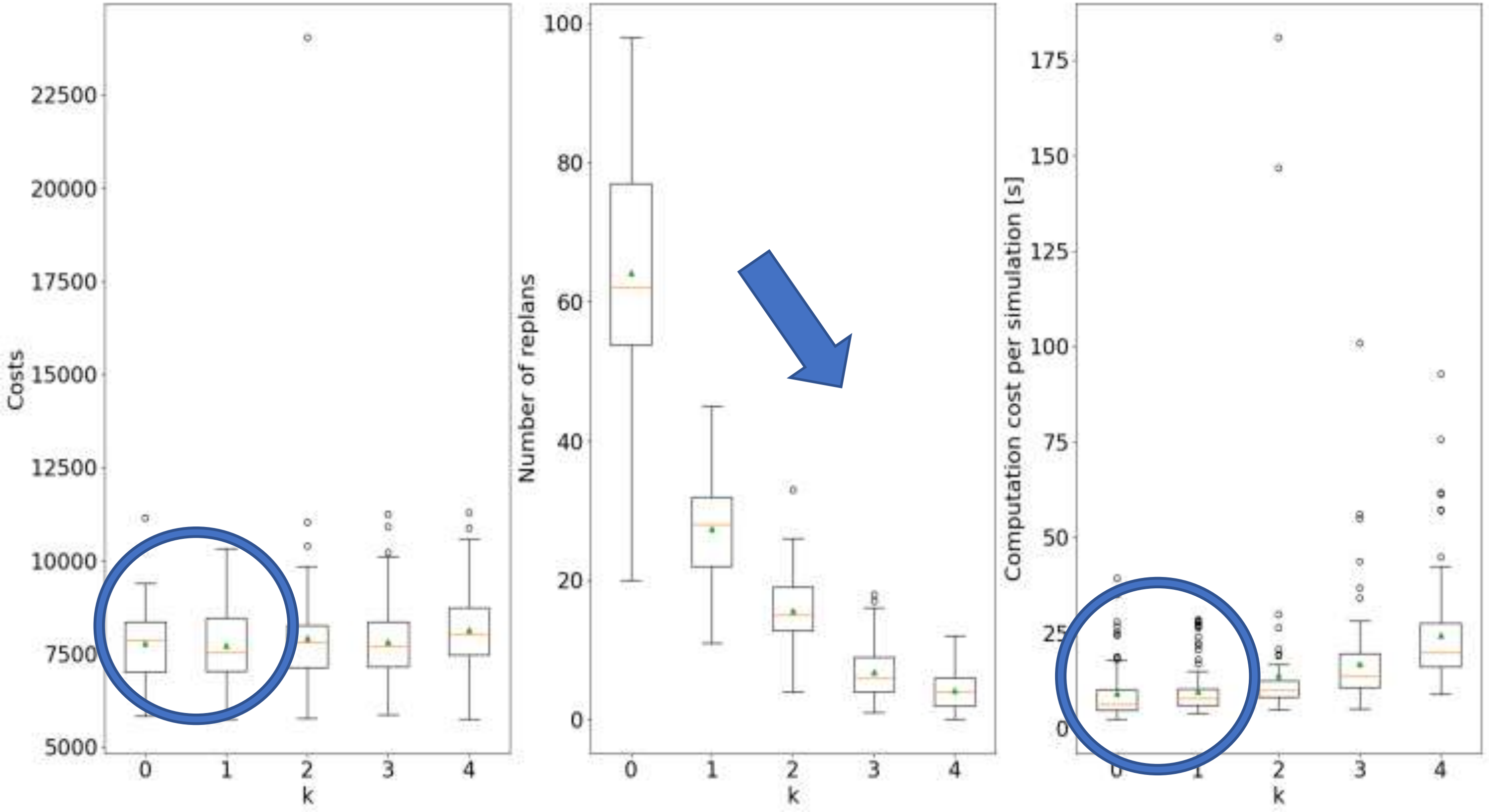
p-TP, p_d = 0.02

- Replans reduced
- Small or no increase in cost
- Increase in runtime

POLITECNICO MILANO 1863

Experimental results: more delays

- Large warehouse
- 24 agents
- 50 delays per agent
- 50 tasks



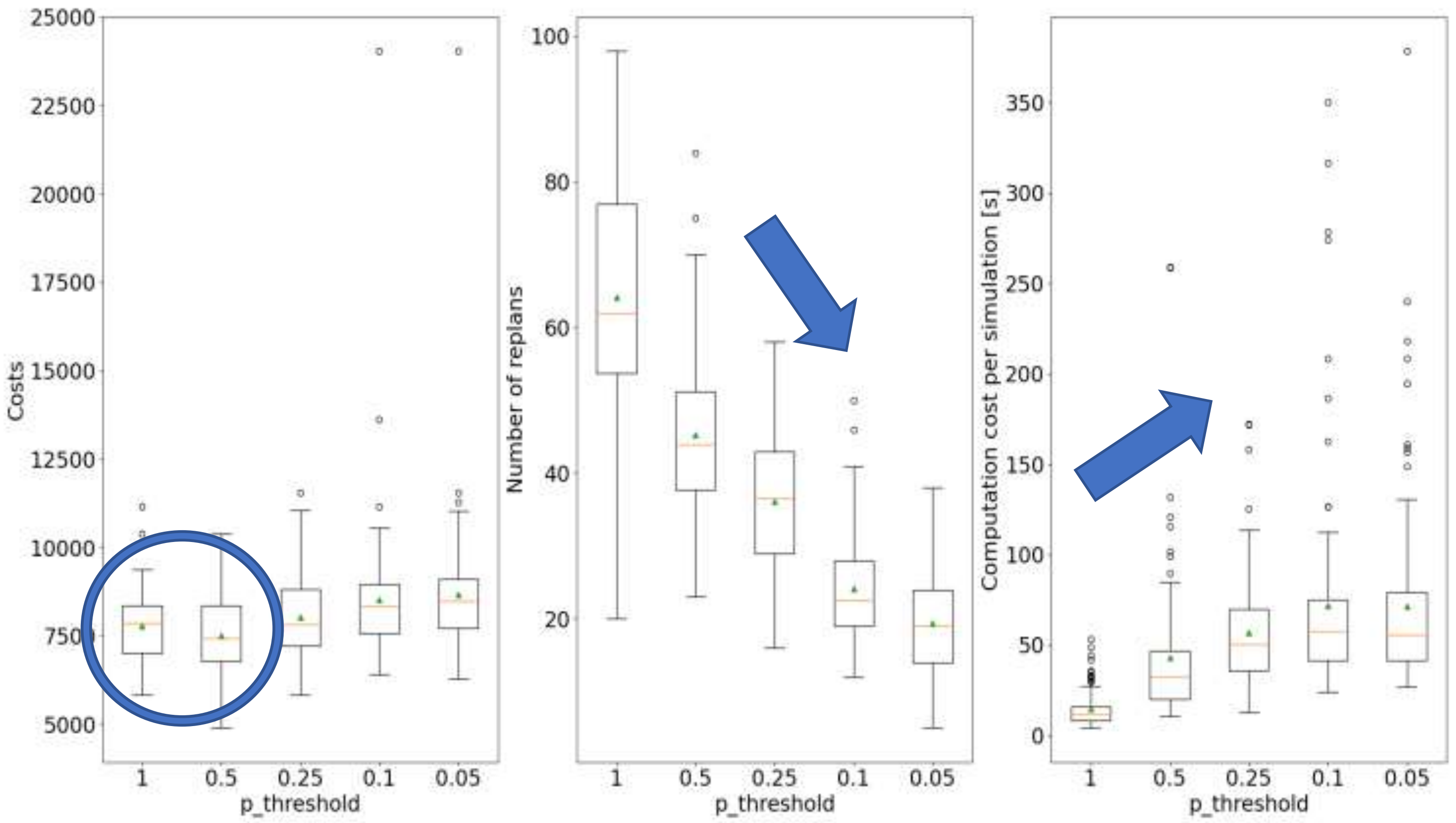
- Lots of replans with recoveries
- Replans greatly reduced
- Small or no increase in cost
- Small or no increase in runtime (for low k)

k-TP

POLITECNICO
MILANO 1863

Experimental results: more delays

- Large warehouse
- 24 agents
- 50 delays per agent
- 50 tasks



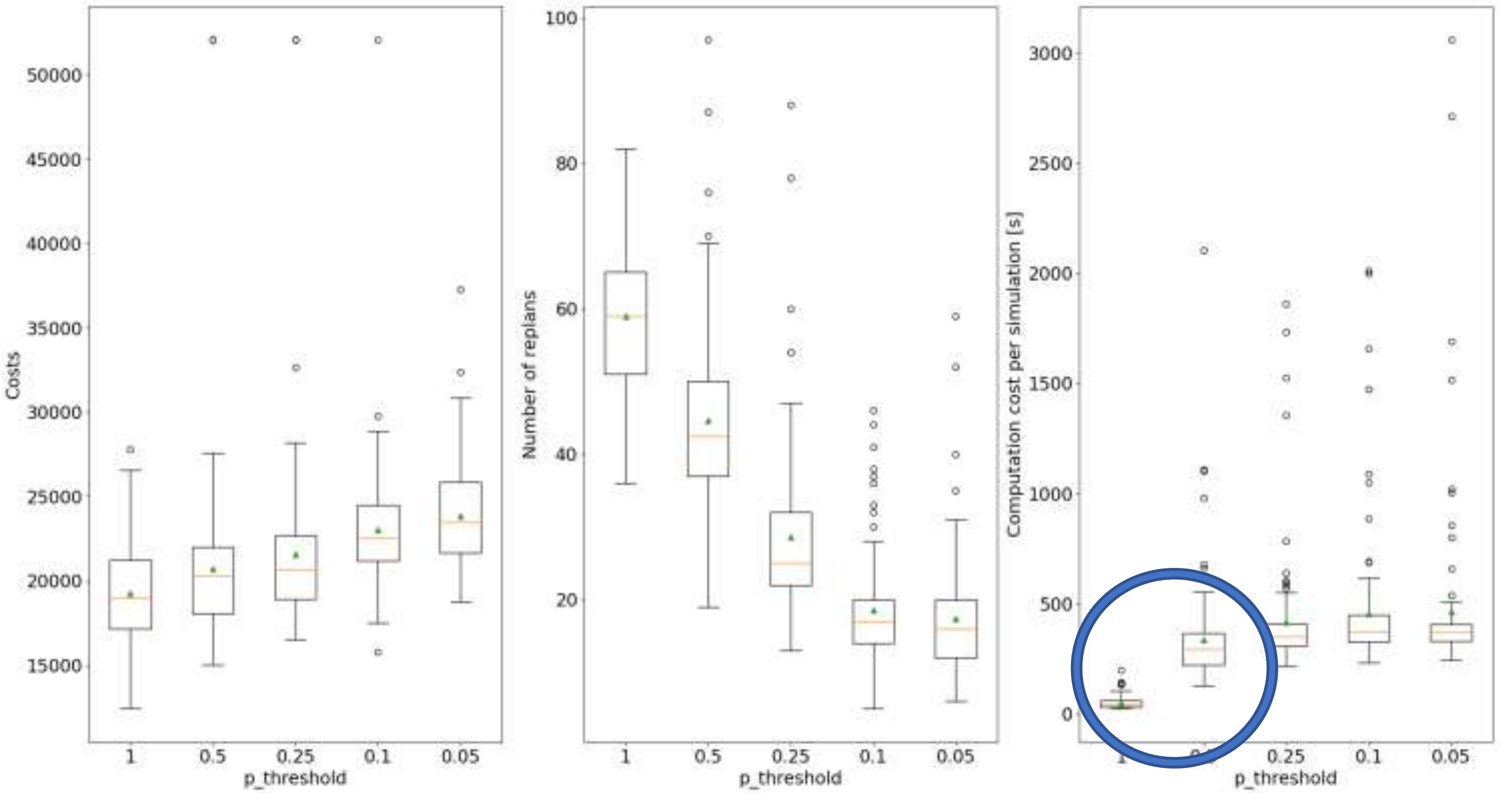
- Lots of replans with recoveries
- Replans greatly reduced
- Small or no increase in cost
- Increase in runtime

p-TP, p_d = 0.02

POLITECNICO
MILANO 1863

Experimental results: larger environment

- Larger warehouse
- 52 agents
- 10 delays per agent
- 100 tasks



p-TP, p_d = 0.02

- Similar results as before for k-TP and p-TP
- p-TP runtime grows to hundreds of seconds

POLITECNICO MILANO 1863

Conclusions and future work

Conclusions

This work provided a theoretical, algorithmic and experimental contribution to Multi-Agent Path Planning:

- A variation of the MAPD problem, MAPD-d, has been introduced to deal with an important practical issue encountered in real applications: delays.
- Two algorithms have been presented, k-TP and p-TP, to solve well-formed MAPD-d problem instances with deterministic and probabilistic robustness guarantees.
- These algorithms have been compared against a baseline algorithm: k-TP showed the best results in terms of robustness-cost trade-off, but p-TP still offers great opportunities for future improvements.

Future work

Future research will address the enhancement of p-TP and the experimental testing of the proposed algorithms in more real-world settings.

POLITECNICO
MILANO 1863

AAMAS 2022

AAMAS 2022

This work has been submitted to AAMAS 2022.

Acceptance notification: 20^{th} December.

Thanks for your attention!

POLITECNICO
MILANO 1863