Research Project Proposal:

Exploiting Task Distribution in Multi-Agent Pickup and Delivery

Andrea Di Pietro
andrea4.dipietro@mail.polimi.it
Computer Science and Engineering Track

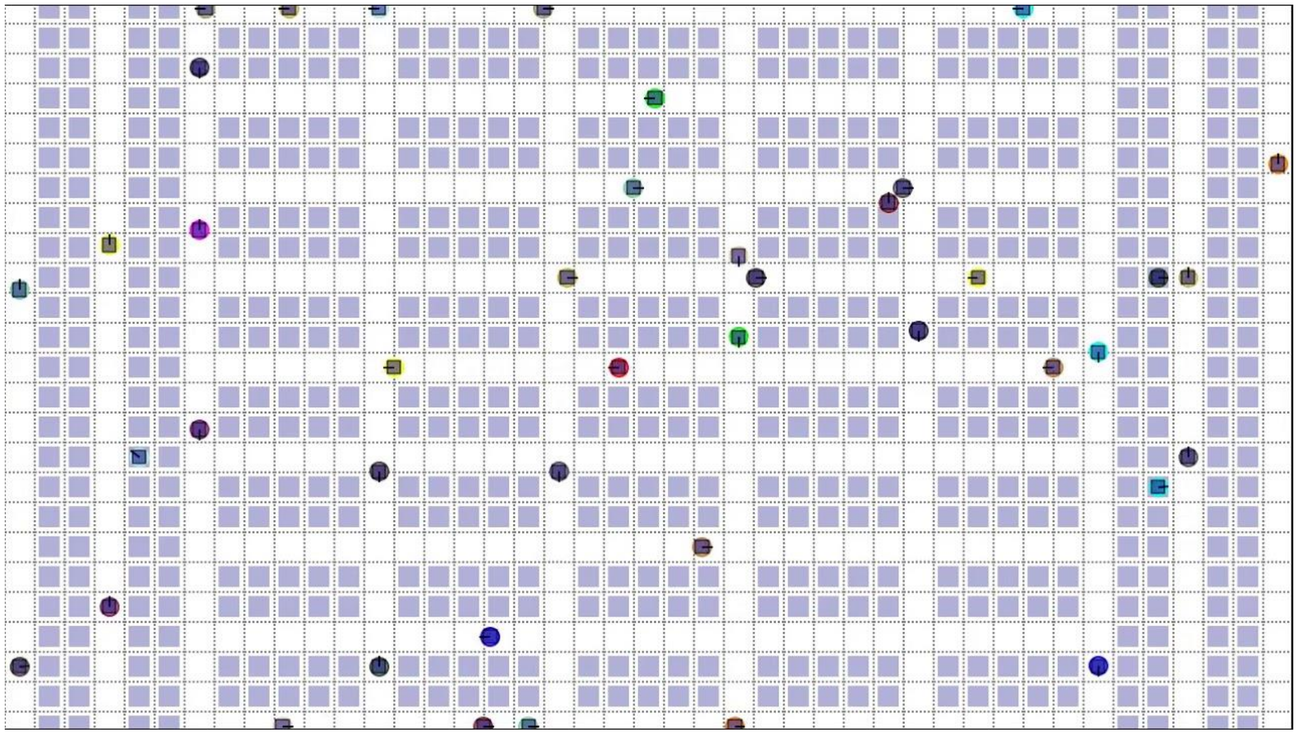
POLITECNICO
MILANO 1863

HONOURS PROGRAMME
HP-SR
in Information Technology

Introduction

- Multi-Agent Pickup and Delivery (**MAPD**) is a problem in the field of multi-robot systems in which some agents have to execute a set of tasks that is continuously updated with new tasks

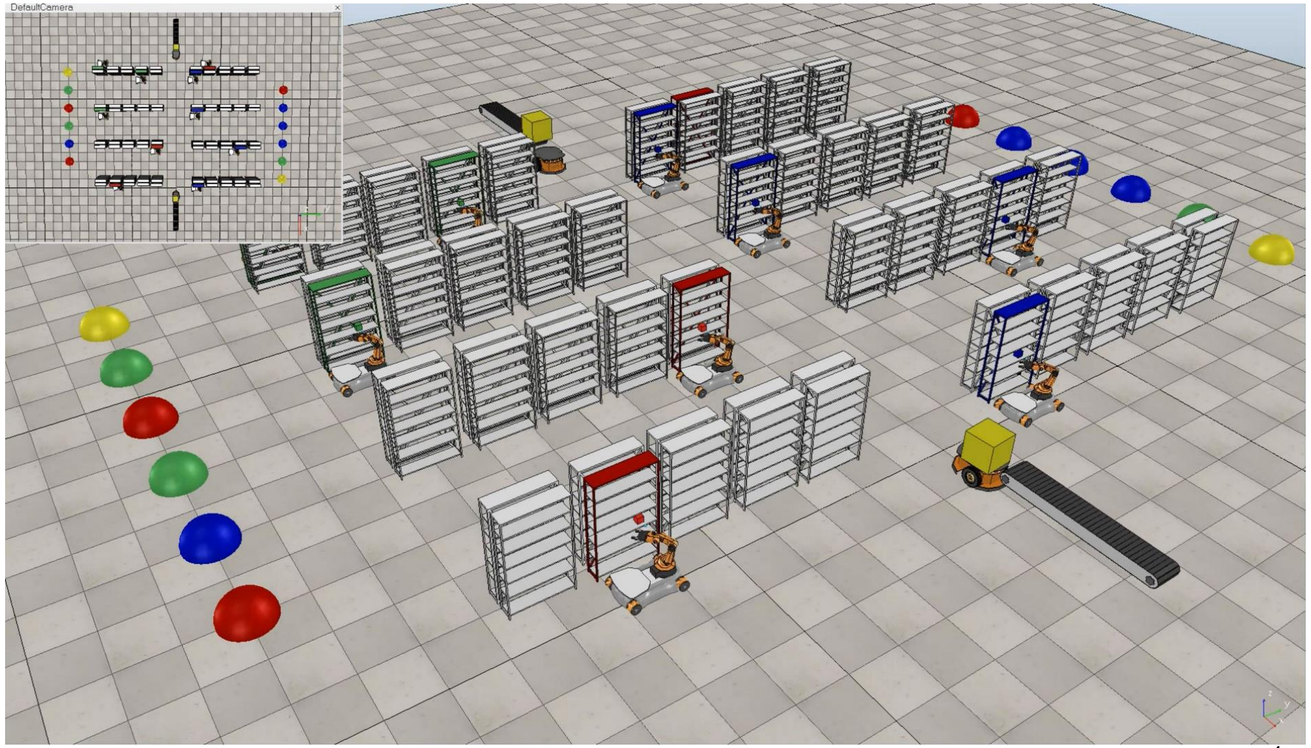
  - Each task consists in moving from a pickup location to a delivery location in a given environment



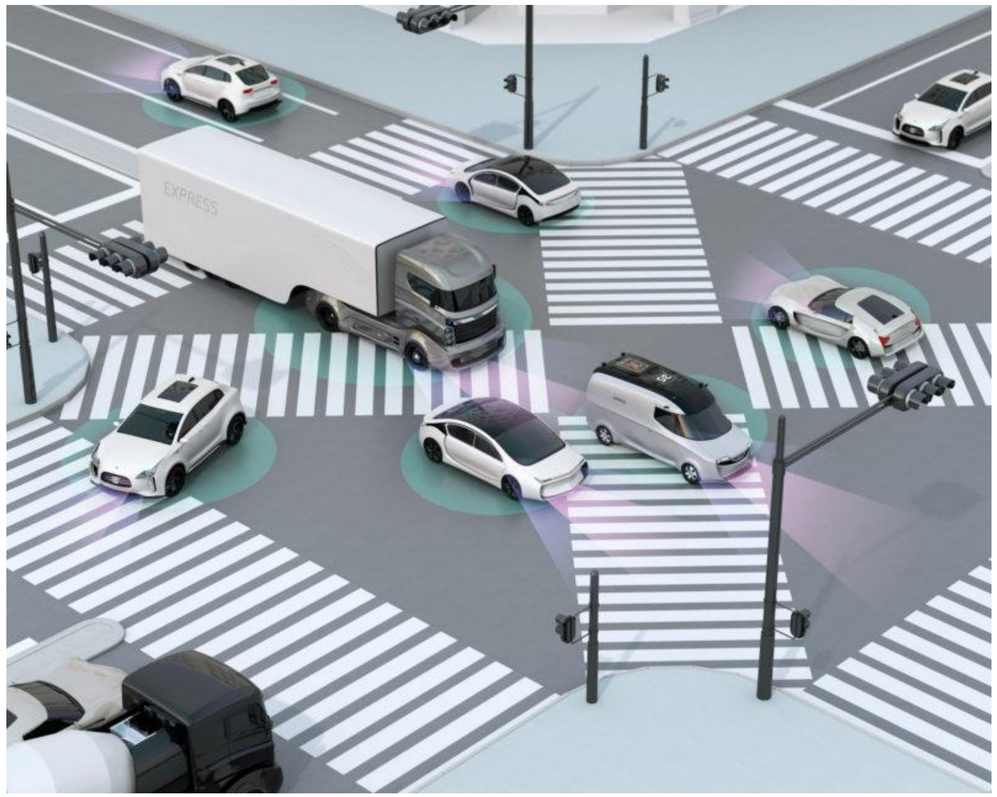
Introduction

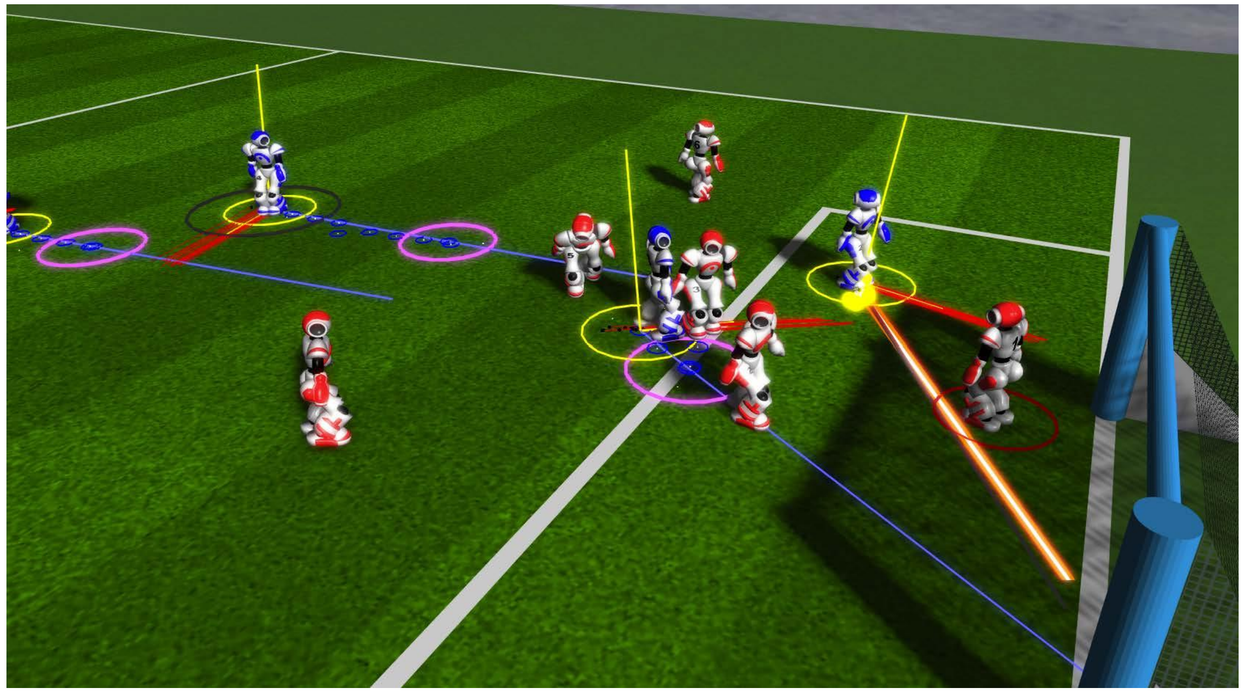
- This problem is the combination of two simpler problems:
  - Task assignment: Allocating each task to an agent
  - Multi-Agent Path Finding (**MAPF**): a problem in which each agent has to solve exactly one preassigned task [Stern et al., 2019]

Applications: Warehouses



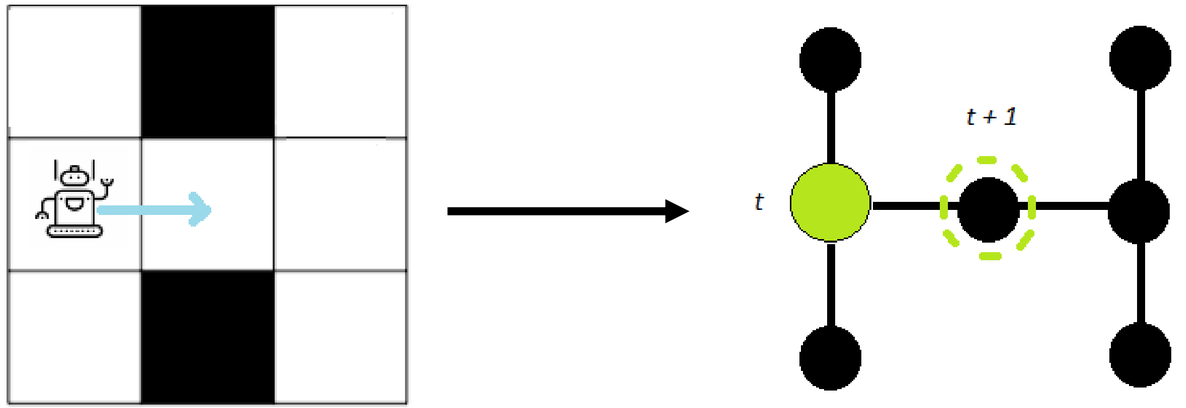
Applications: Autonomous vehicles and video games





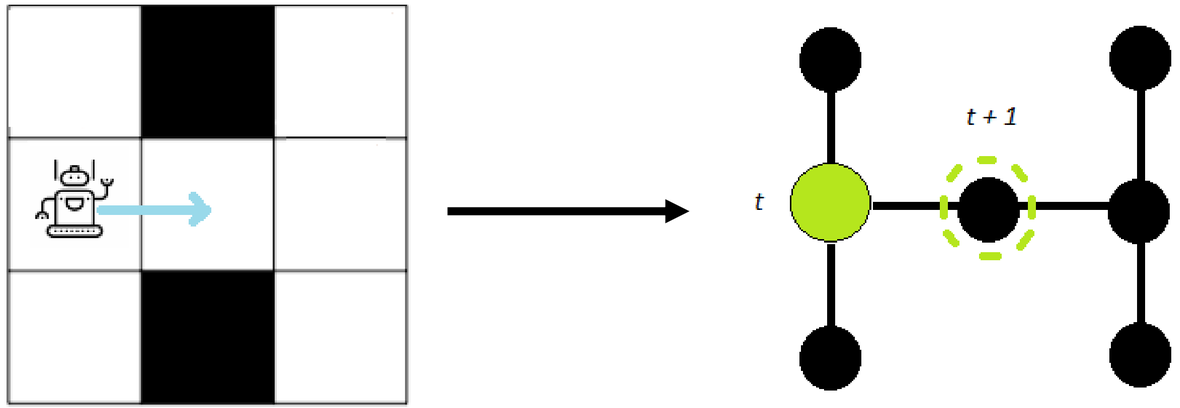
Problem formulation

- MAPF and MAPD problems are defined on an **undirected graph** *G = (V, E)*
- Time is **discrete**



Problem formulation

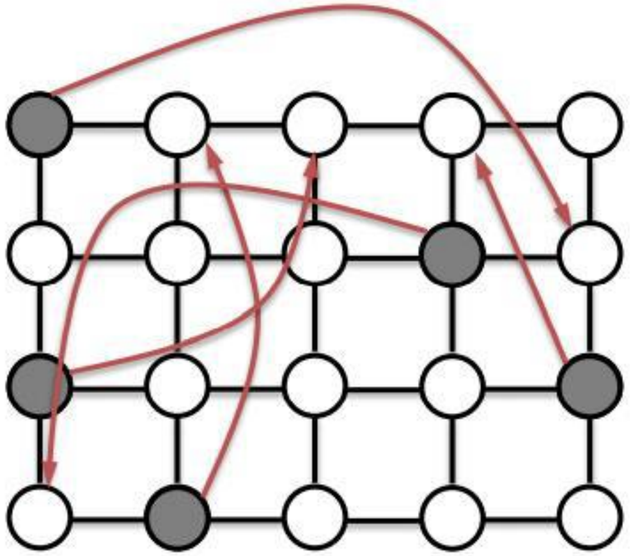
- MAPF and MAPD problems are defined on an **undirected graph** *G = (V, E)*
- Time is **discrete**



- At every time step each agent occupies a vertex of the graph and can perform two kinds of action:
  - ***Moving to an adjacent vertex*** through an edge of the graph
  - ***Waiting in the current location***

MAPF

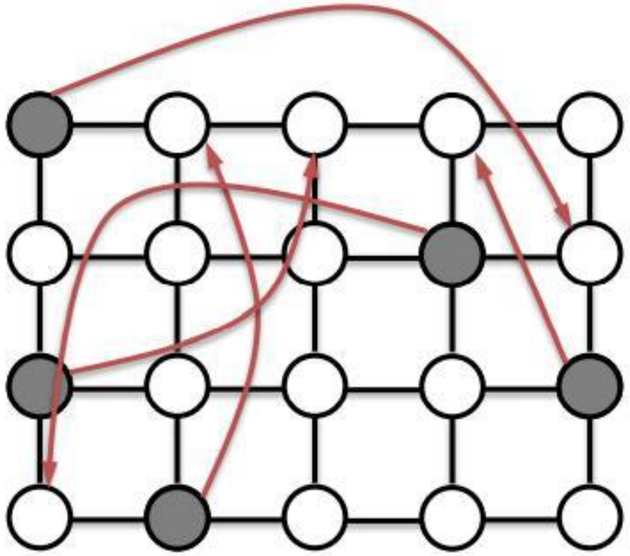
- Each agent has a source vertex s_i and a target vertex t_i



- A single-agent ***path*** for an agent is a sequence of actions that executed from the source vertex leads to the target vertex

MAPF

- Each agent has a source vertex s_i and a target vertex t_i

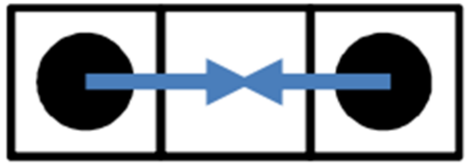


- A single-agent ***path*** for an agent is a sequence of actions that executed from the source vertex leads to the target vertex

- A solution for a MAPF problem is a ***plan*** (i.e., a set of paths for the agents) without ***collisions***

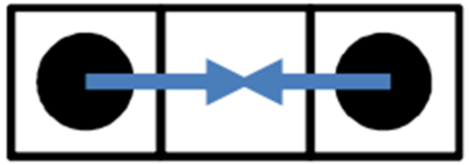
Collisions

- A ***vertex collision*** between two paths occurs if two agents occupy the same vertex at the same time step

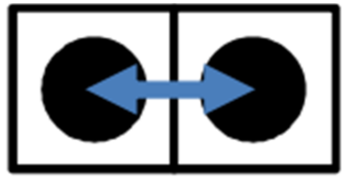


Collisions

- A ***vertex collision*** between two paths occurs if two agents occupy the same vertex at the same time step



- A ***swap collision*** between two paths occurs if two agents traverse the same edge in opposite direction and swap locations at the same time step



MAPF – Objective Functions

- ***Makespan:*** the number of steps after which all agents have reached their targets without collisions

- ***Sum of costs:*** the sum of the numbers of steps of paths planned by single agents

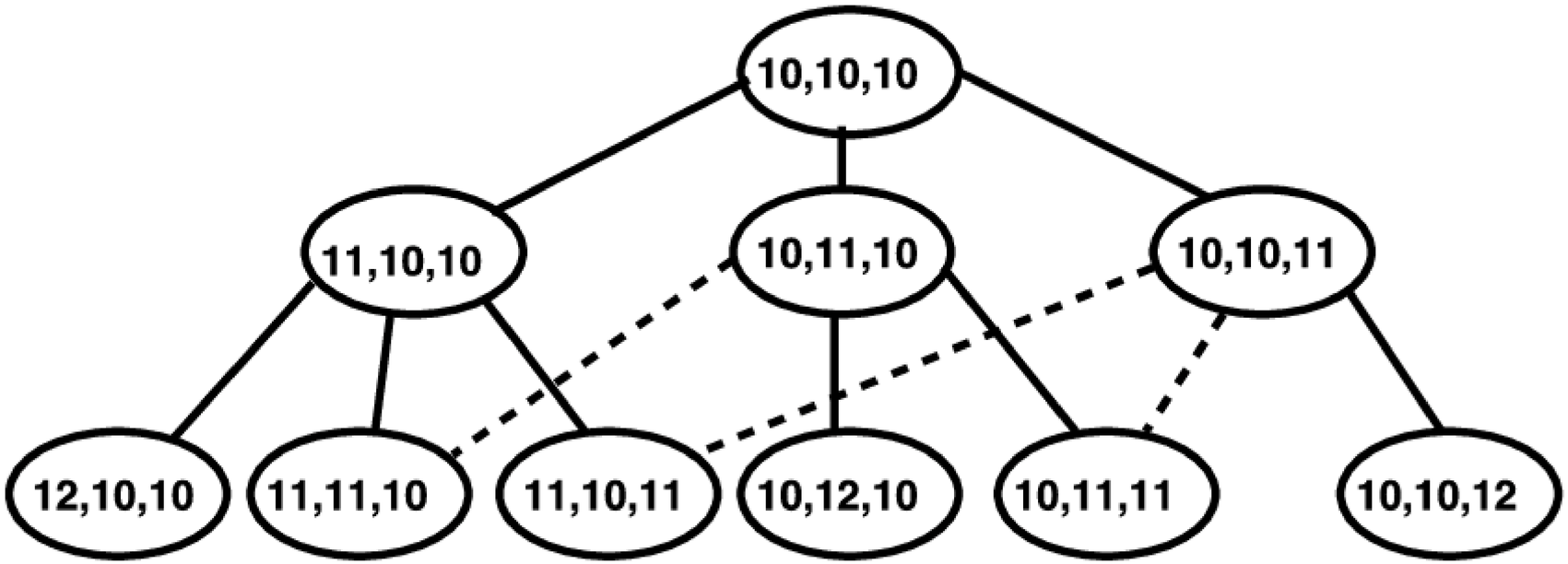
- Finding an optimal solution for a MAPF problem is ***NP-Hard*** for both of these objective functions to minimize [Yu and LaValle, 2013]

MAPF - Algorithms

- MAPF Algorithms can be classified according to different criteria:
  - ***Completeness***: An algorithm is complete if it guarantees to return a solution when a solution exists
  - ***Optimality***: An algorithm is optimal if it guarantees to provide an optimal solution for the given instance of the problem, otherwise it is suboptimal

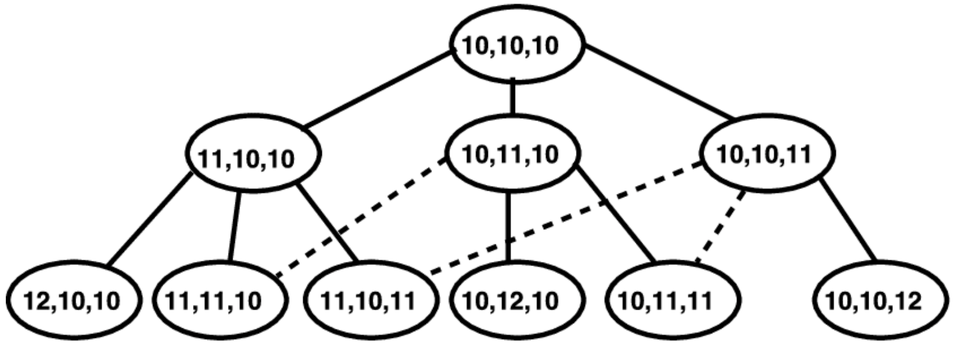
MAPF – Optimal Algorithms

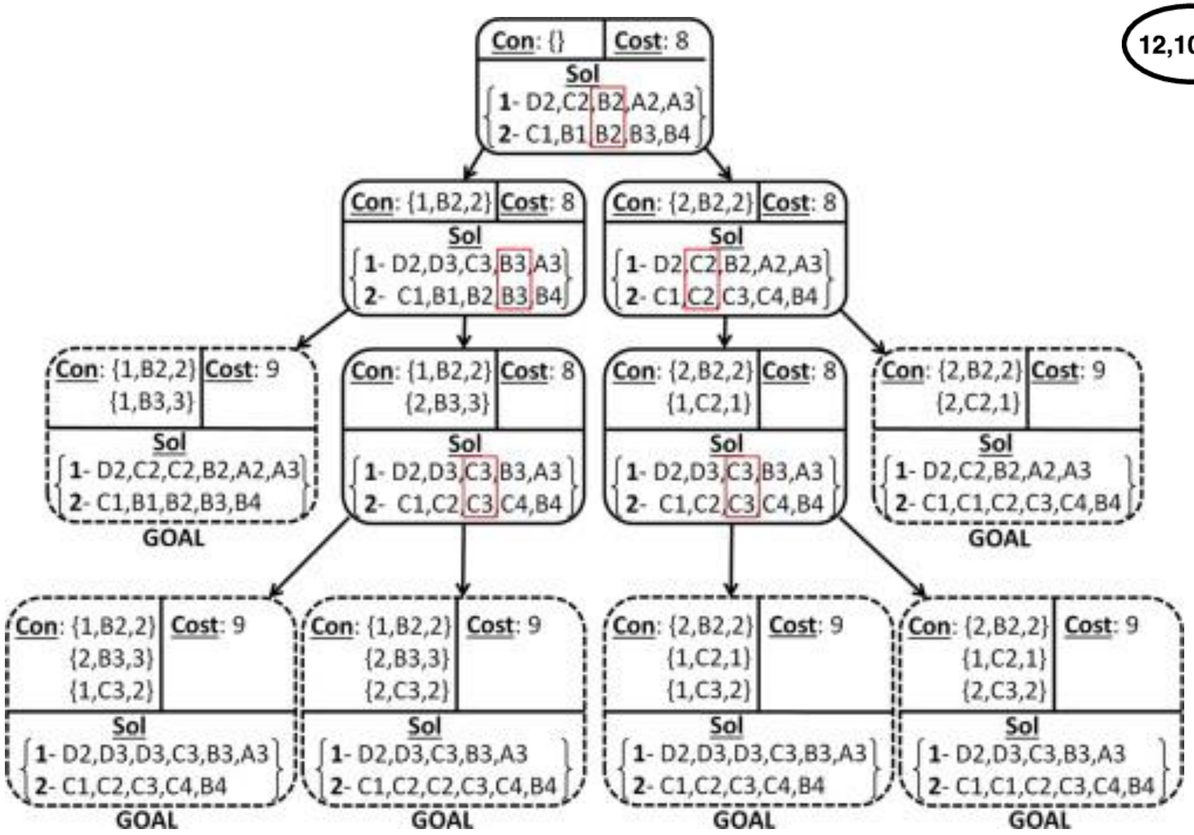
- ***Increasing Cost Tree Search (ICTS)***
  [Sharon et al., 2011]



MAPF – Optimal Algorithms

- *Increasing Cost Tree Search (ICTS)*

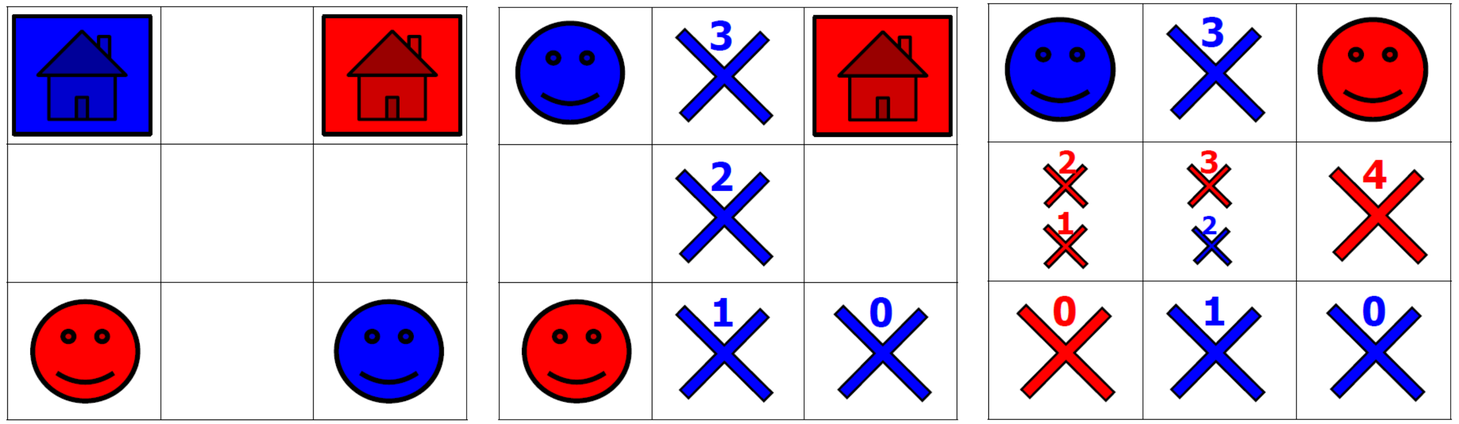




- ***Conflict-Based Search (CBS)***
  [Sharon et al., 2012]

MAPF – Suboptimal Algorithms

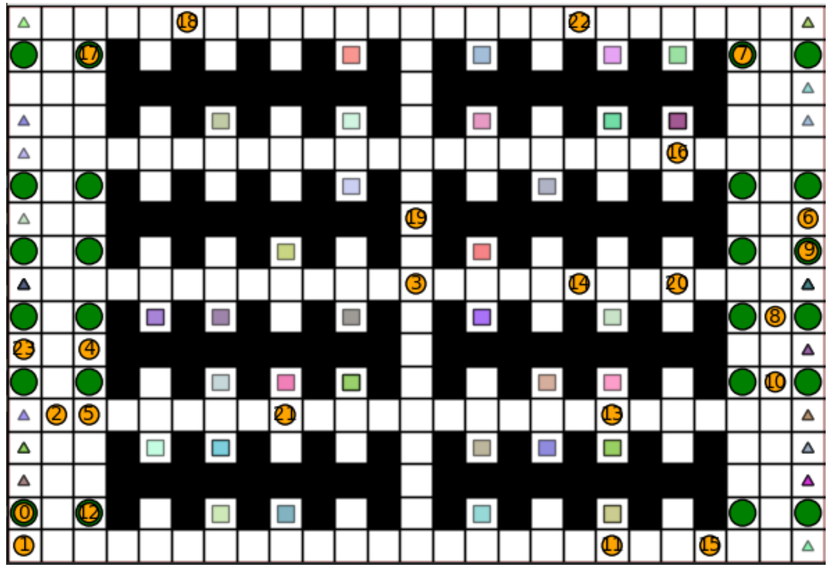
- ***Prioritized Planning***
  [Van Den Berg and Overmars, 2005]



- Prioritized planning is also ***incomplete*** because it does not guarantee to find a solution even if solutions exist for the given MAPF instance

MAPD

- *T* is the set of unexecuted ***tasks***. *T* is updated at each time step with the new incoming tasks
- Each task is associated to a pickup location s_i and a delivery location g_i
- An agent is free if and only if it is not executing any task
- A plan is a set of paths that single agents follow to execute the assigned tasks and it is continuously updated with new assignments and plannings
- A solution of a MAPD instance is a plan such that all tasks are executed in a bounded amount of time



MAPD – Objective Functions

- ***Makespan:*** the number of steps after which all the tasks in T have been completed

- ***Service time:*** average number of steps between the addition of a task to the system and the completion of the task

- Being a generalization of MAPF, finding an optimal solution for a MAPD problem is ***NP-Hard***.

MAPD Algorithms - Approaches

- ***Offline MAPD***: All tasks are known in advance and the planning is performed *a priori* before starting the execution [Liu et al., 2019]

MAPD Algorithms - Approaches

- ***Offline MAPD***: All tasks are known in advance and the planning is performed *a priori* before starting the execution [Liu et al., 2019]

- ***Online MAPD***: Information about tasks (arrival time step, pickup and delivery locations) are not known in advance [Ma et al., 2017]. There are different approaches:

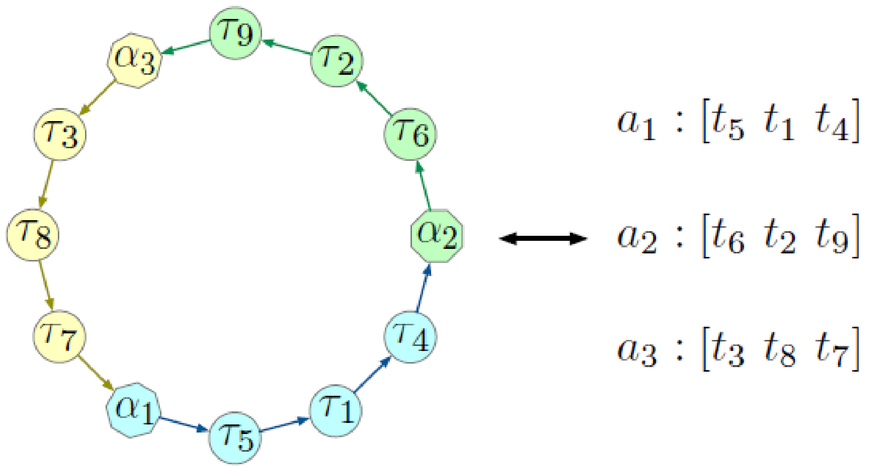
  - All paths are replanned at every time step to consider new tasks

  - Path replanning is performed only for some of the agents at each time step

Offline MAPD Algorithms

- Task Assignment and Prioritized path planning (***TA-Prioritized***):

  1. Each agent is assigned a sequence of tasks
  2. Prioritized Planning is used for agents' path planning

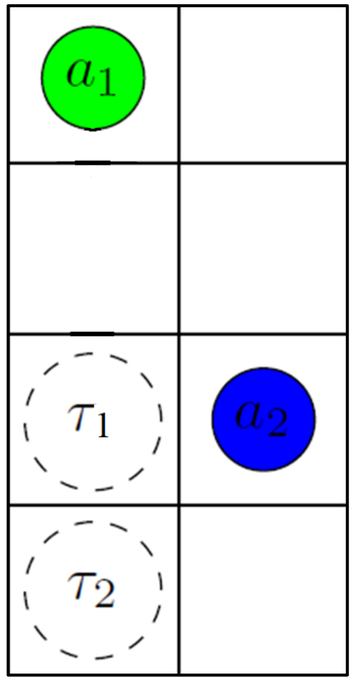


- Task Assignment and Hybrid path planning (***TA-Hybrid***):

  - A hybrid path planning strategy is used combining CBS to a min-cost max-flow algorithm

Online MAPD Algorithms: Token Passing (TP)

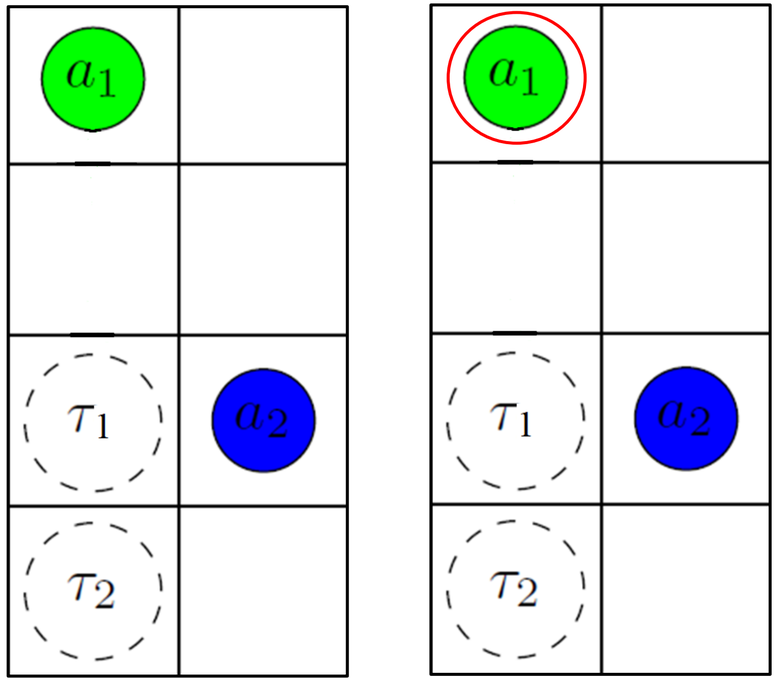
- A token is shared among the agents and contains the current paths of agents and the status of tasks



Online MAPD Algorithms: Token Passing (TP)

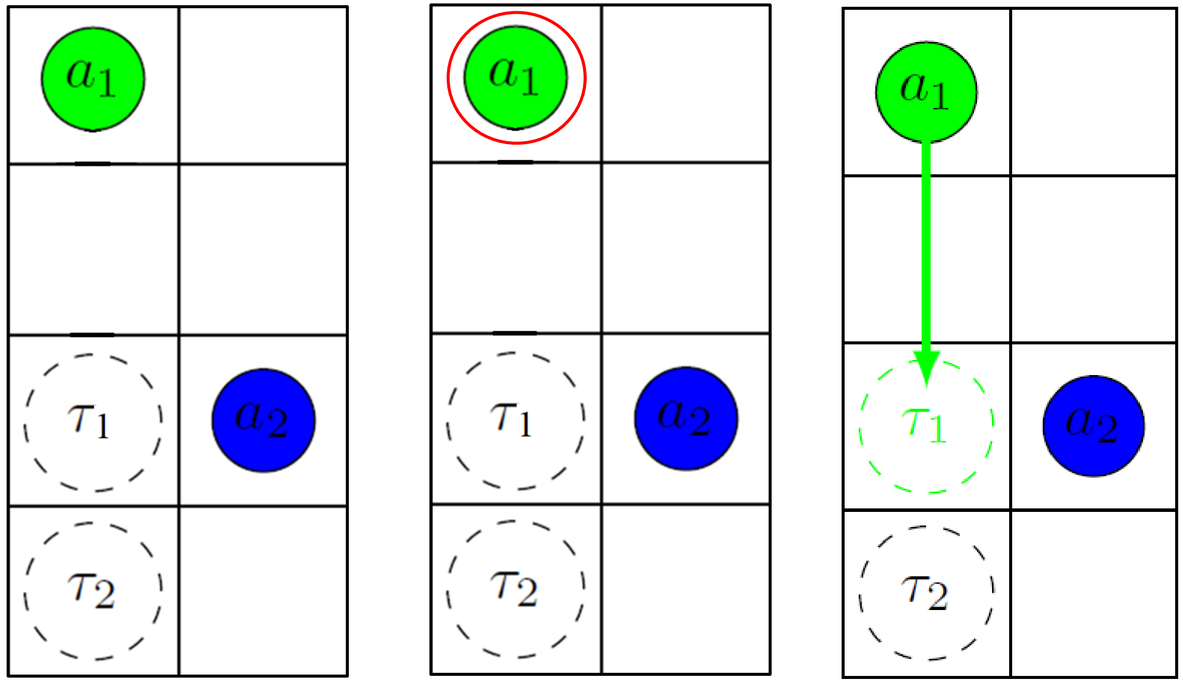
- A token is shared among the agents and contains the current paths of agents and the status of tasks

- Each free agent requests the token once per time step



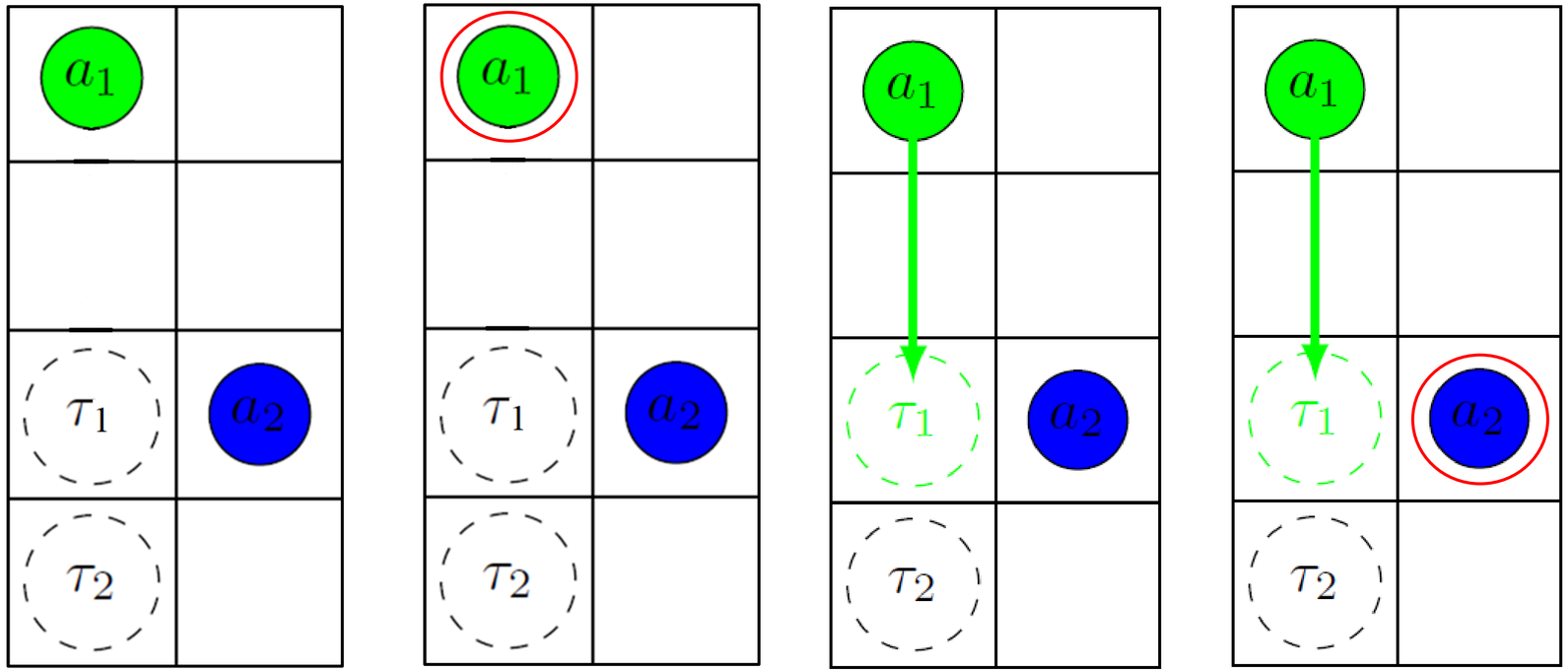
Online MAPD Algorithms: Token Passing (TP)

- A token is shared among the agents and contains the current paths of agents and the status of tasks
- Each free agent requests the token once per time step
- If there exist unassigned tasks such that no path of other agents in the token ends in the pickup or delivery location of the task, the agent chooses the task with minimum path cost and updates the token



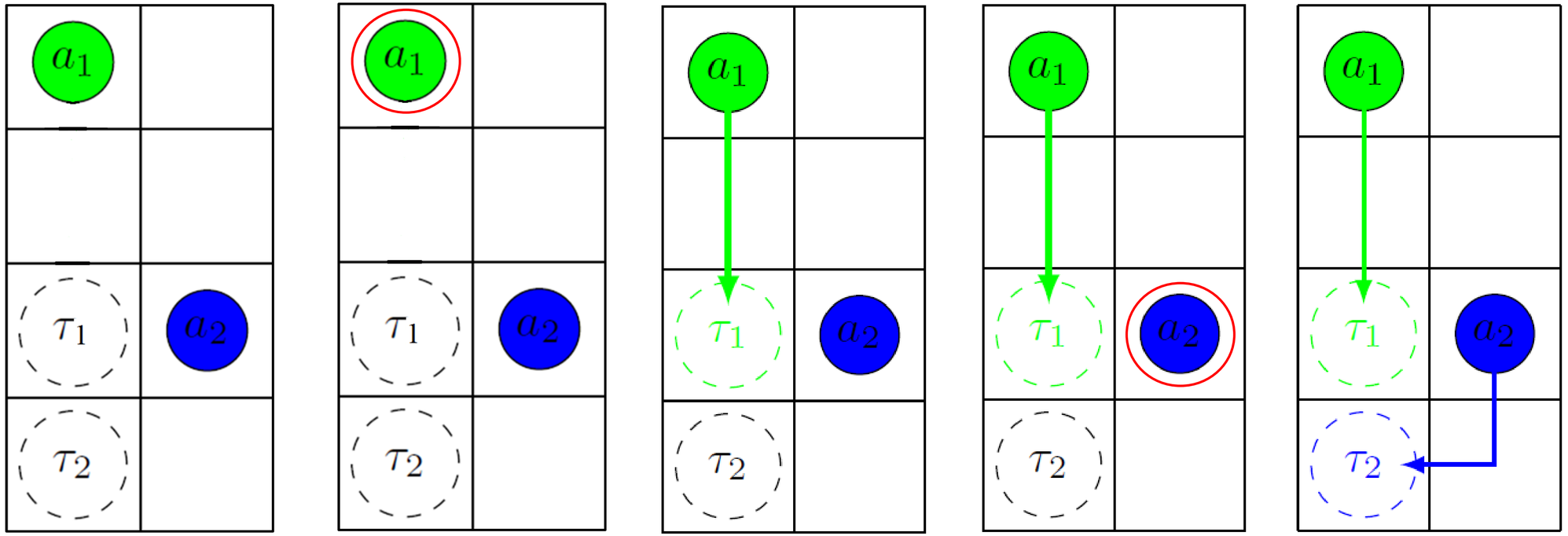
Online MAPD Algorithms: Token Passing (TP)

- A token is shared among the agents and contains the current paths of agents and the status of tasks
- Each free agent requests the token once per time step
- If there exist unassigned tasks such that no path of other agents in the token ends in the pickup or delivery location of the task, the agent chooses the task with minimum path cost and updates the token



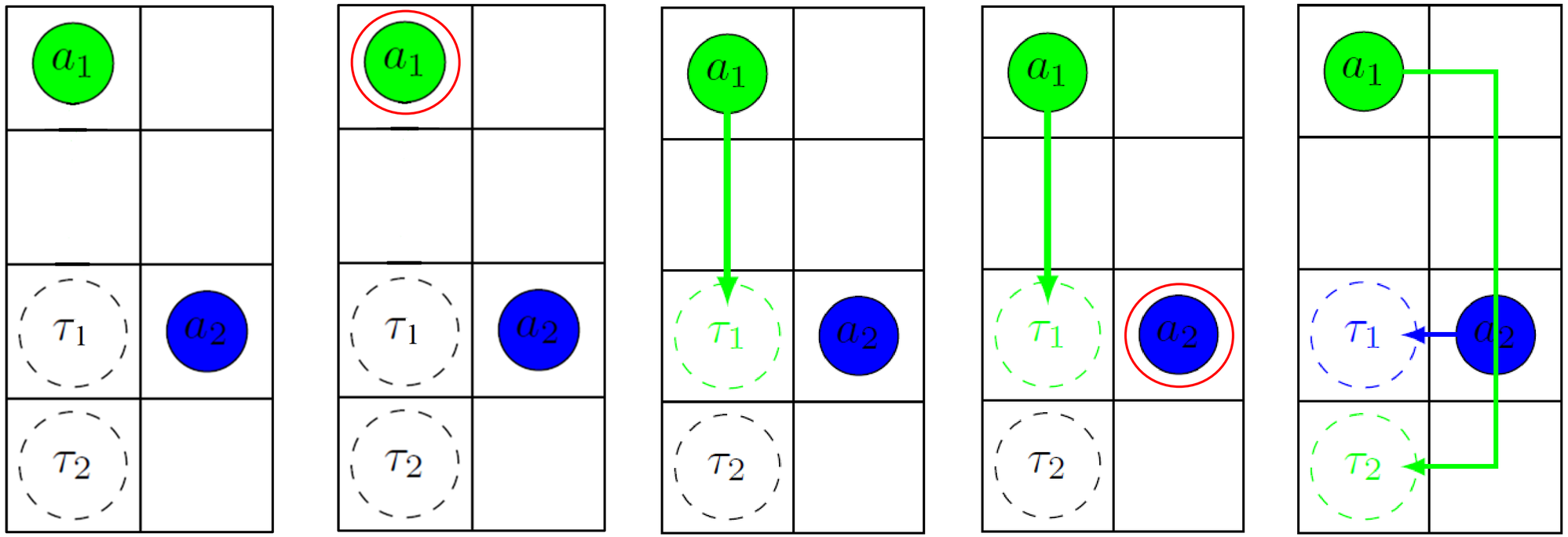
Online MAPD Algorithms: Token Passing (TP)

- A token is shared among the agents and contains the current paths of agents and the status of tasks
- Each free agent requests the token once per time step
- If there exist unassigned tasks such that no path of other agents in the token ends in the pickup or delivery location of the task, the agent chooses the task with minimum path cost and updates the token



Online MAPD Algorithms: Token Passing with Task Swaps (TPTS)

- The algorithm is similar to TP but a free agent can also choose to be assigned an already assigned task if the assigned agent has not yet reached the pickup location of the task



Online MAPD Algorithms

- *Central* [Ma et al, 2017] is a centralized algorithm that makes decisions for multiple agents at a time. It uses a centralized algorithm for tasks assignment and CBS for paths planning

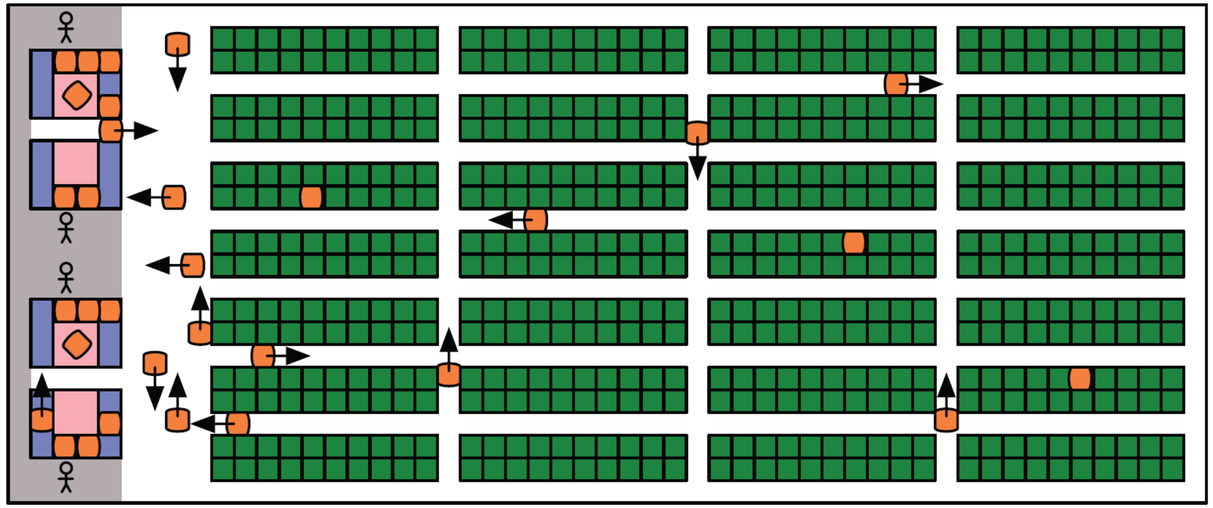
- *Rolling-Horizon Collision Resolution* [Li et al, 2021] decomposes a MAPD instance into a sequence of windowed MAPF instances and replans paths once every h time steps solving collisions only for a limited amount of time steps

Task distribution

- The time and space distribution of tasks across the environment can impact the efficiency of an algorithm

- Most of the works about MAPF and MAPD don’t run experiments about the effects of different tasks distributions on the performance of the algorithms

- Many experimental data are the result of tests on environments where the distribution of pickup or delivery locations of tasks is random and uniform, but in real environments (e.g., warehouses) pickup or delivery locations usually have more complex distributions



Task distribution

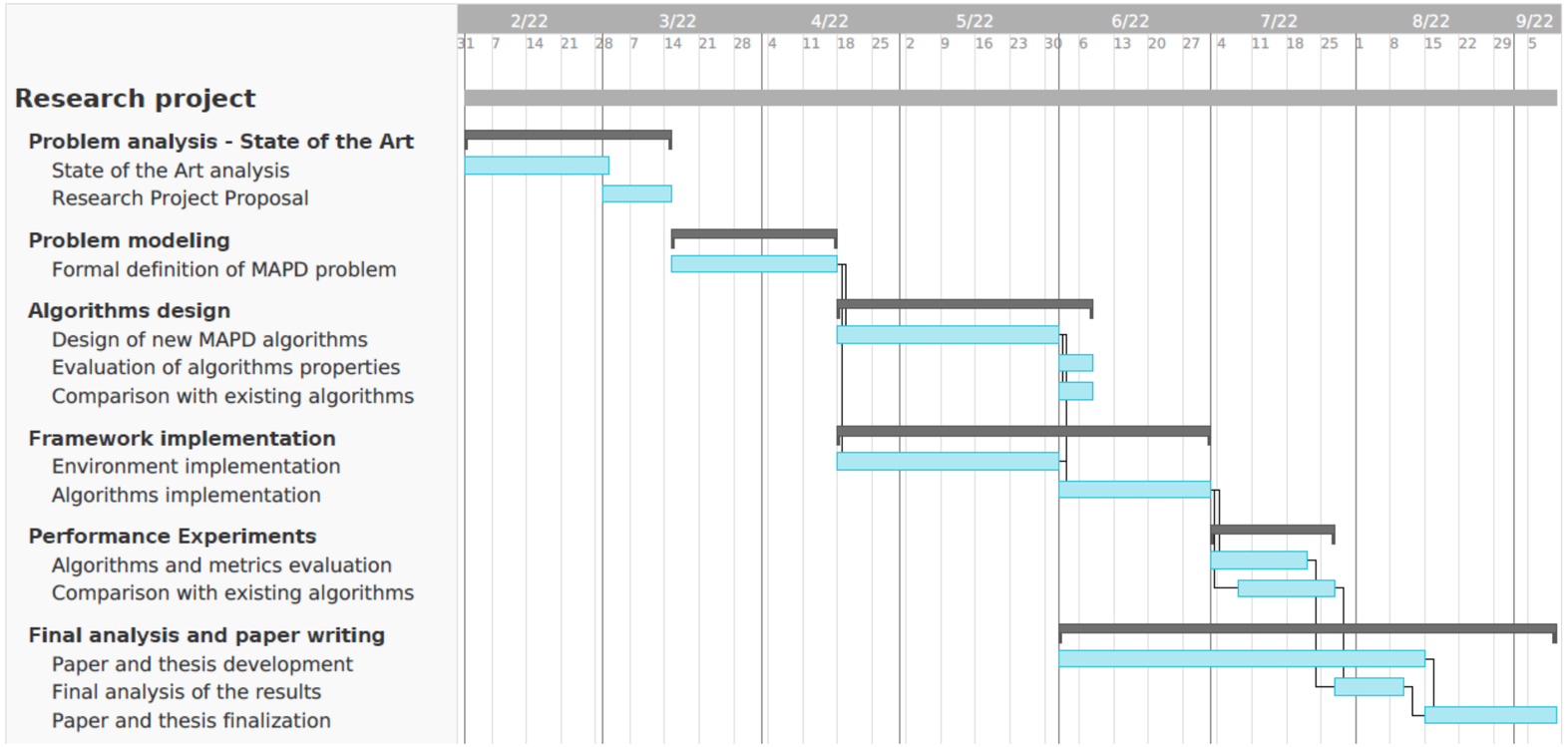
- Offline MAPD algorithms assume that the arrival time step, the pickup location and the delivery location of tasks are completely known *a priori*
- Online MAPD algorithms assume that nothing is known about tasks until they are added to the system
- In some real applications data about pickup and delivery locations and arrival frequency of tasks are actually known or can be estimated in advance
- This allows to define $P(v_p, v_d, t)$ as the probability of a task with some given pickup and delivery locations to be added to the system at time step t

Research Project

- The goals of the research are:
  - Defining a new model of the MAPD problem which includes information about the time and space distribution of tasks
  - Designing new MAPD algorithms that can use the ***probability distribution of the task arrival time and of the task locations*** across the environment to perform tasks assignment and path planning
  - Evaluating how the use of information about tasks distribution impacts on the performance of the provided solutions compared to existing MAPD algorithms, with a focus on models of real environments and realistic tasks distribution such as the ones of warehouse environments

Research Project Plan

1. Analysis of the State of the Art and of existing MAPD algorithms
2. Definition of a new MAPD model which includes the probability distribution of tasks
3. Design of new MAPD algorithm
4. Implementation of a framework for testing and comparison of MAPD algorithms
5. Evaluation of the performance metrics of the new algorithms and comparison with existent algorithms



Evaluation metrics

- Standard MAPD objective functions: ***makespan*** and ***service time***
- *Quality of Service* (e.g., the *throughput*)
- *Time* and *space computational complexity*

Thank you for your attention!