# Multi-Agent Pickup and Delivery with Task Probability Distribution

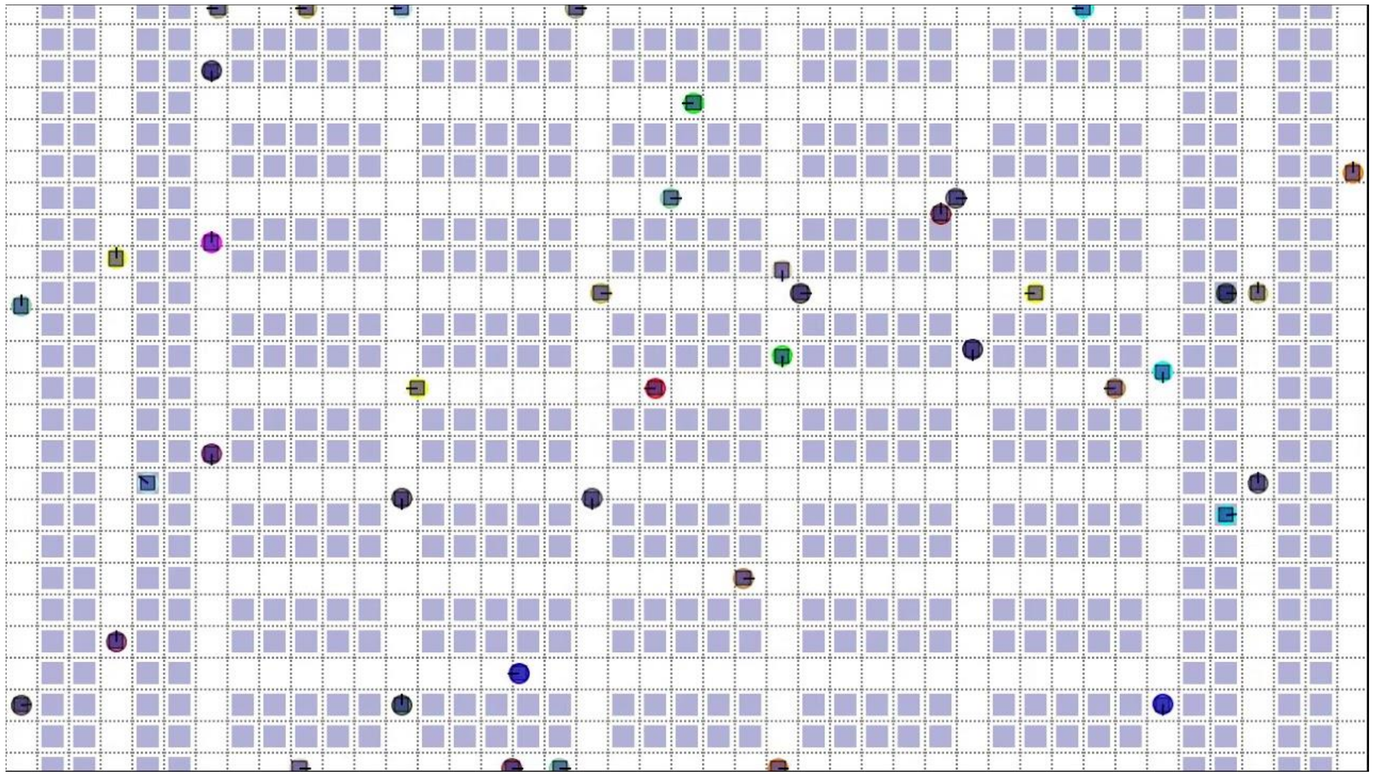
Andrea Di Pietro
andrea4.dipietro@mail.polimi.it
Computer Science and Engineering Track

POLITECNICO
MILANO 1863

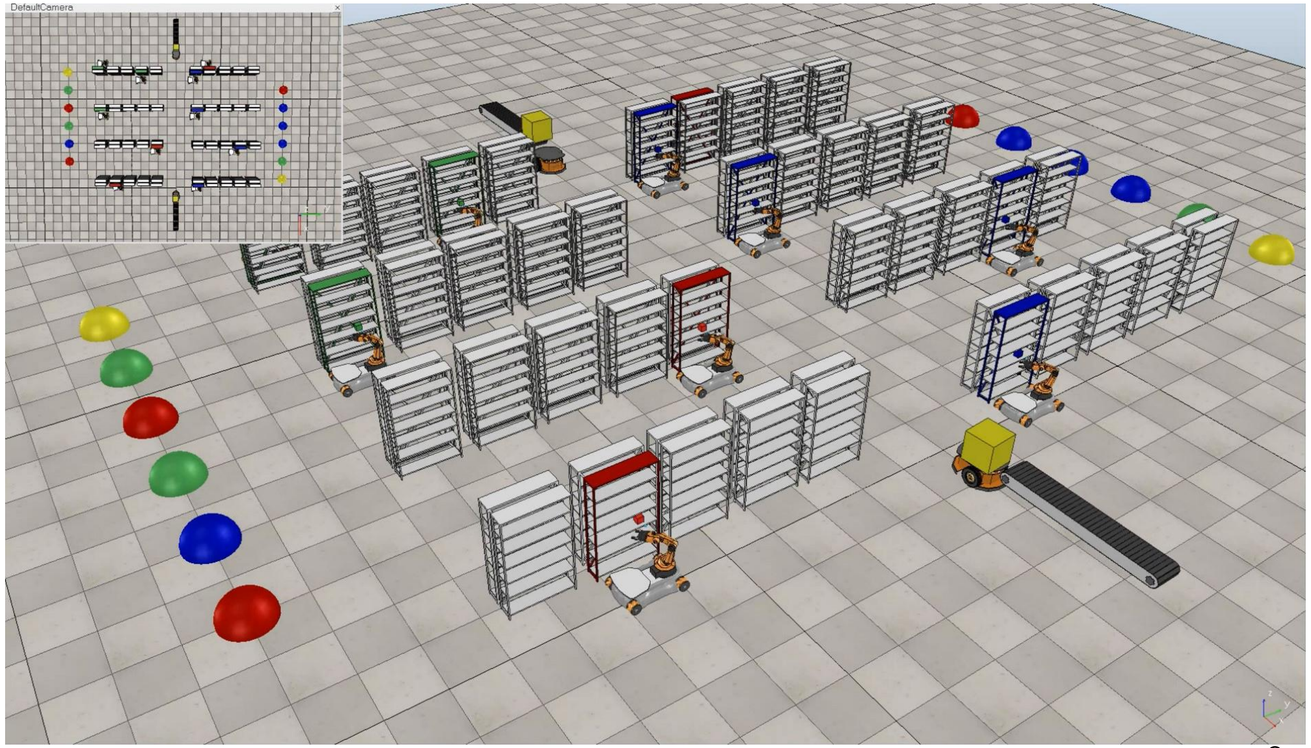
HONOURS PROGRAMME
HP-SR
in Information Technology

# MAPD

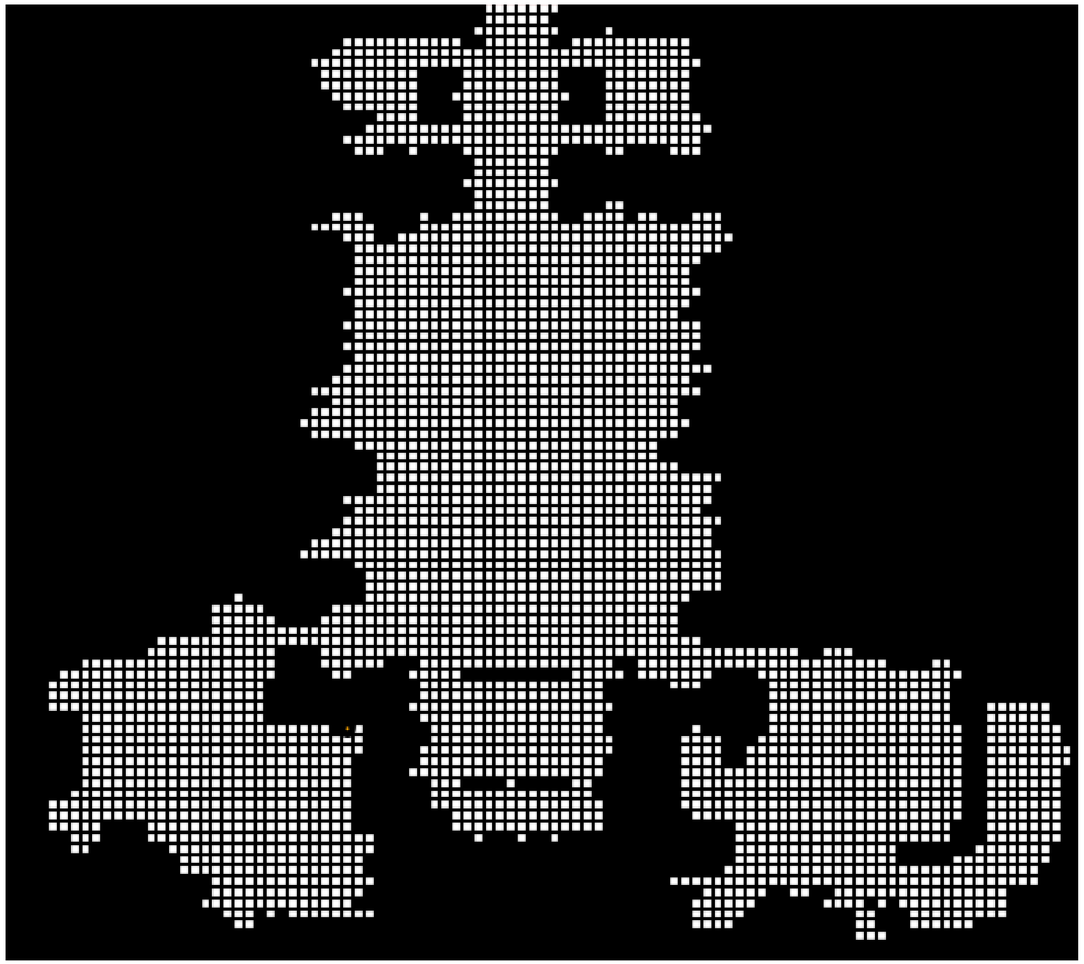
- Multi-Agent Pickup and Delivery [Ma, 2020]
- Multi-robot systems
- A continuously updated set of tasks

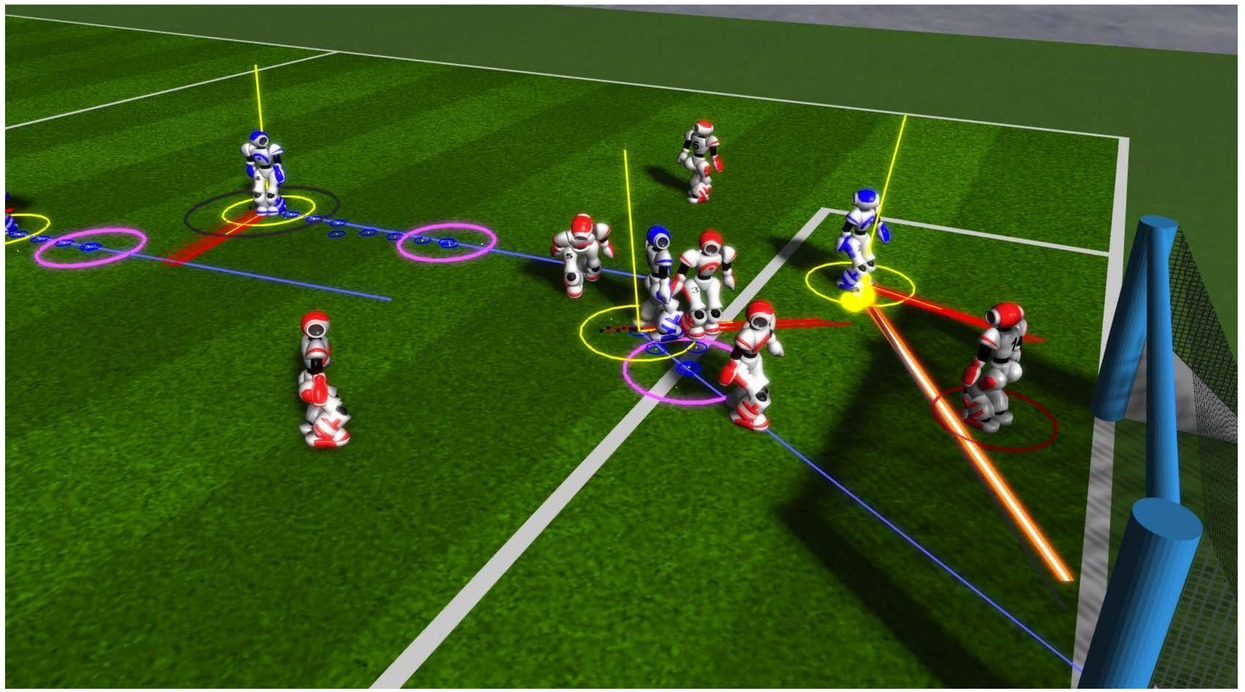


# Applications: Warehouses

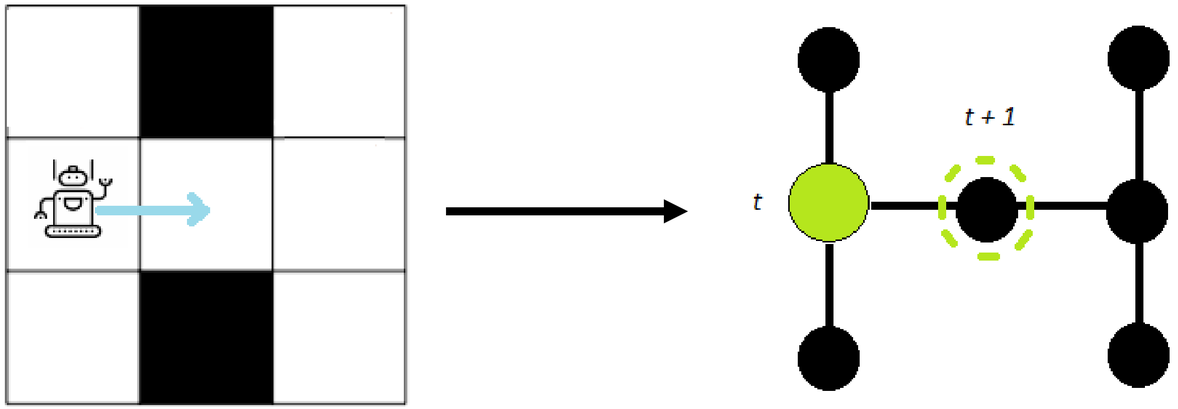
# Applications: Videogames





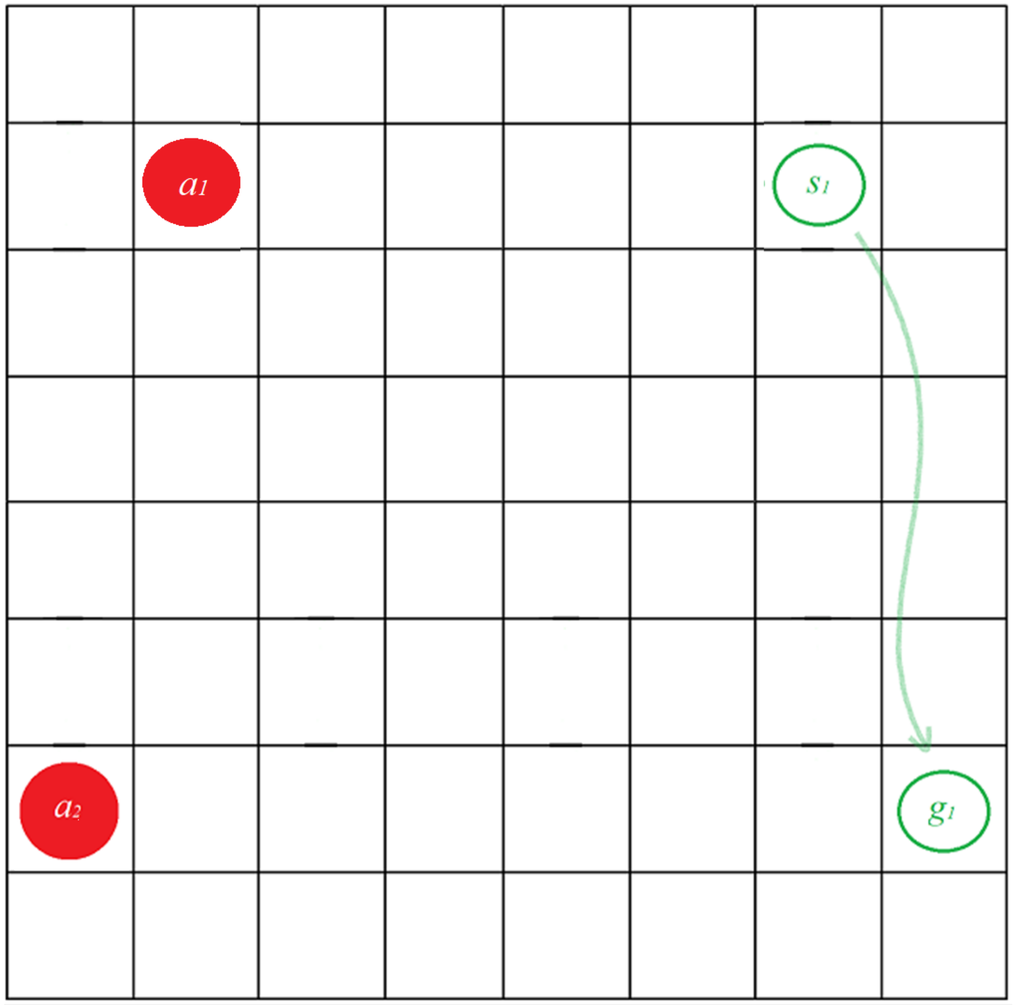
# MAPD

- Combination of two simpler problems:
  - Task assignment
  - **MAPF**: Multi-Agent Path Finding [Stern et al., 2019]
- An **undirected graph** *G = (V, E)*
- Time is **discrete**
- Agents can stay in the current location or move to an adjacent vertex



# MAPD

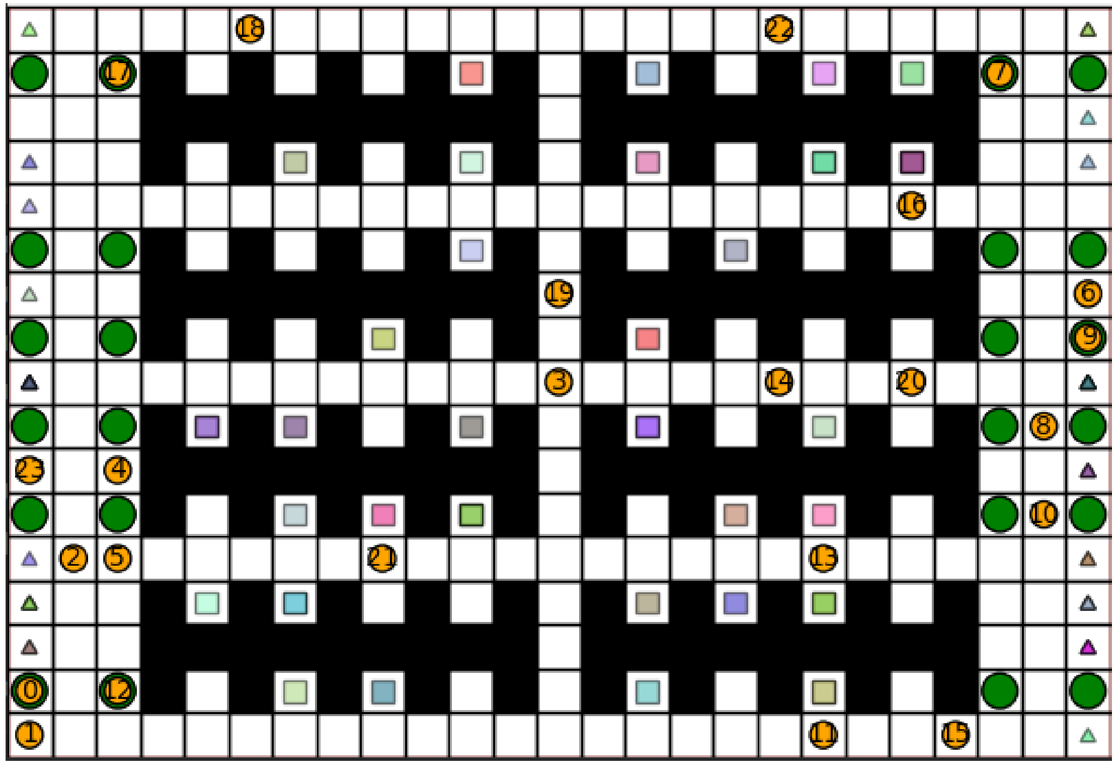
- *T:* set of unexecuted ***tasks***
- A pickup location and a delivery location
- Free agents are assigned to available tasks



# MAPD

- Solution: a plan (i.e., a set of paths) executing all tasks in a bounded amount of time

- Objective functions:
  - Makespan: $T_f - T_i$
  - Service time: $\frac{\sum_{t \in T}(t_f - t_i)}{N}$



# MAPD Algorithms – Online Approach

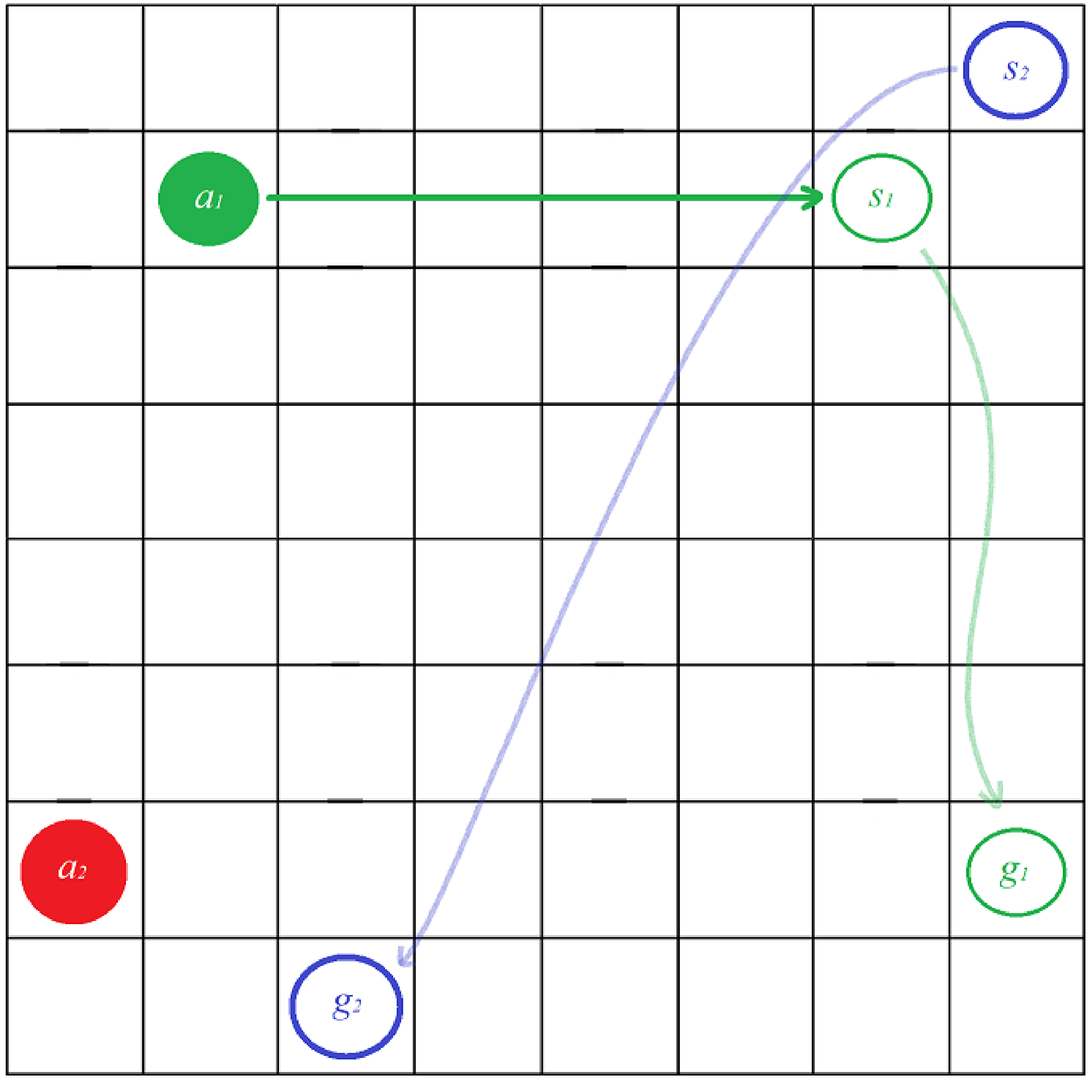
- Task set not known *a priori*
- Agents need to continously replan their path
- Online MAPD →No information is known until the task is added to the task set

# Goal of the research

- In some real applications information about future tasks can be estimated
- Main goal: reduce the service time by using task probability distribution in the phases of
  - ***Task Assignment***
  - ***Path Planning***

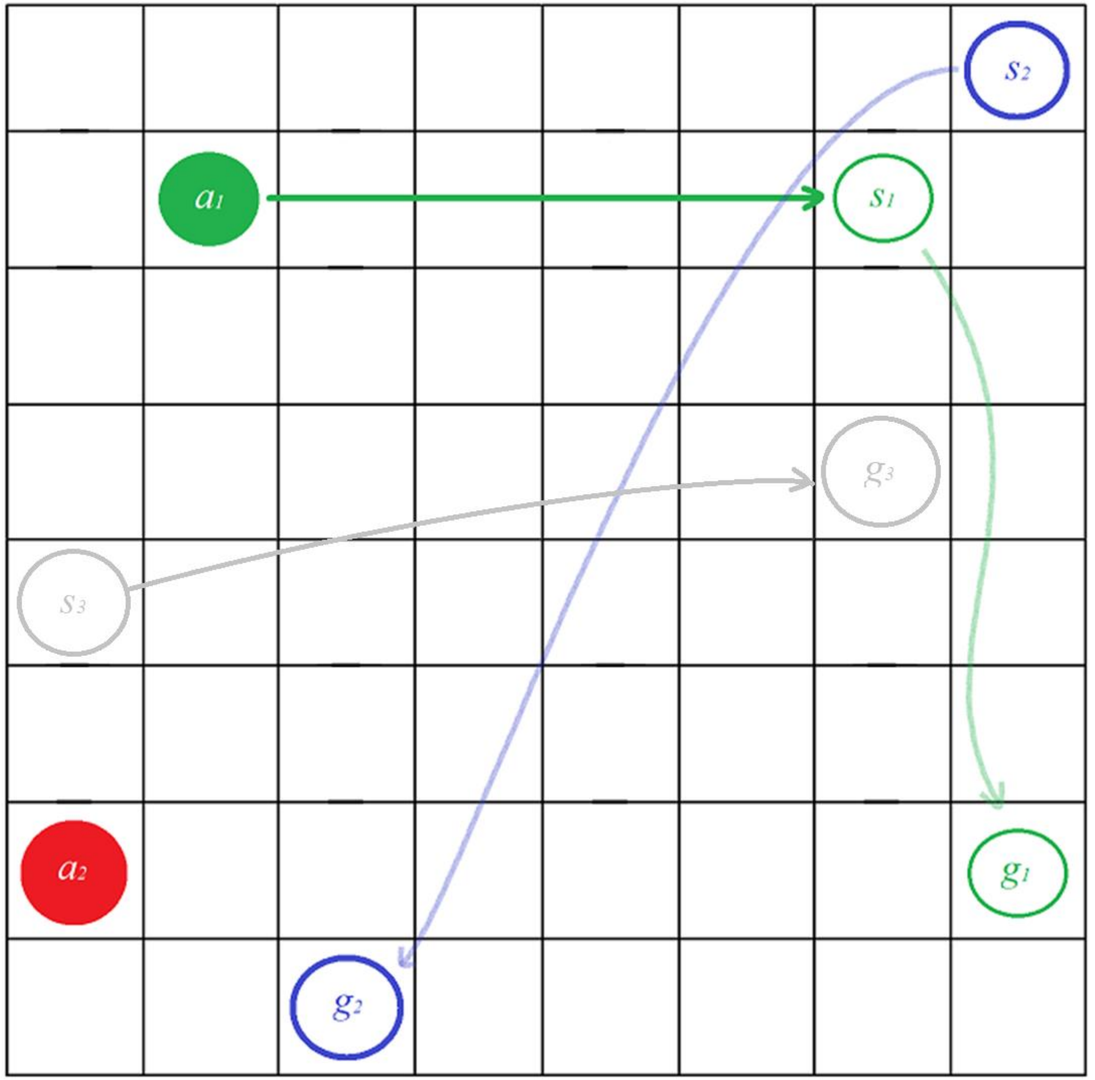
# Task Assignment

How should an agent decide whether to be assigned to a new task?



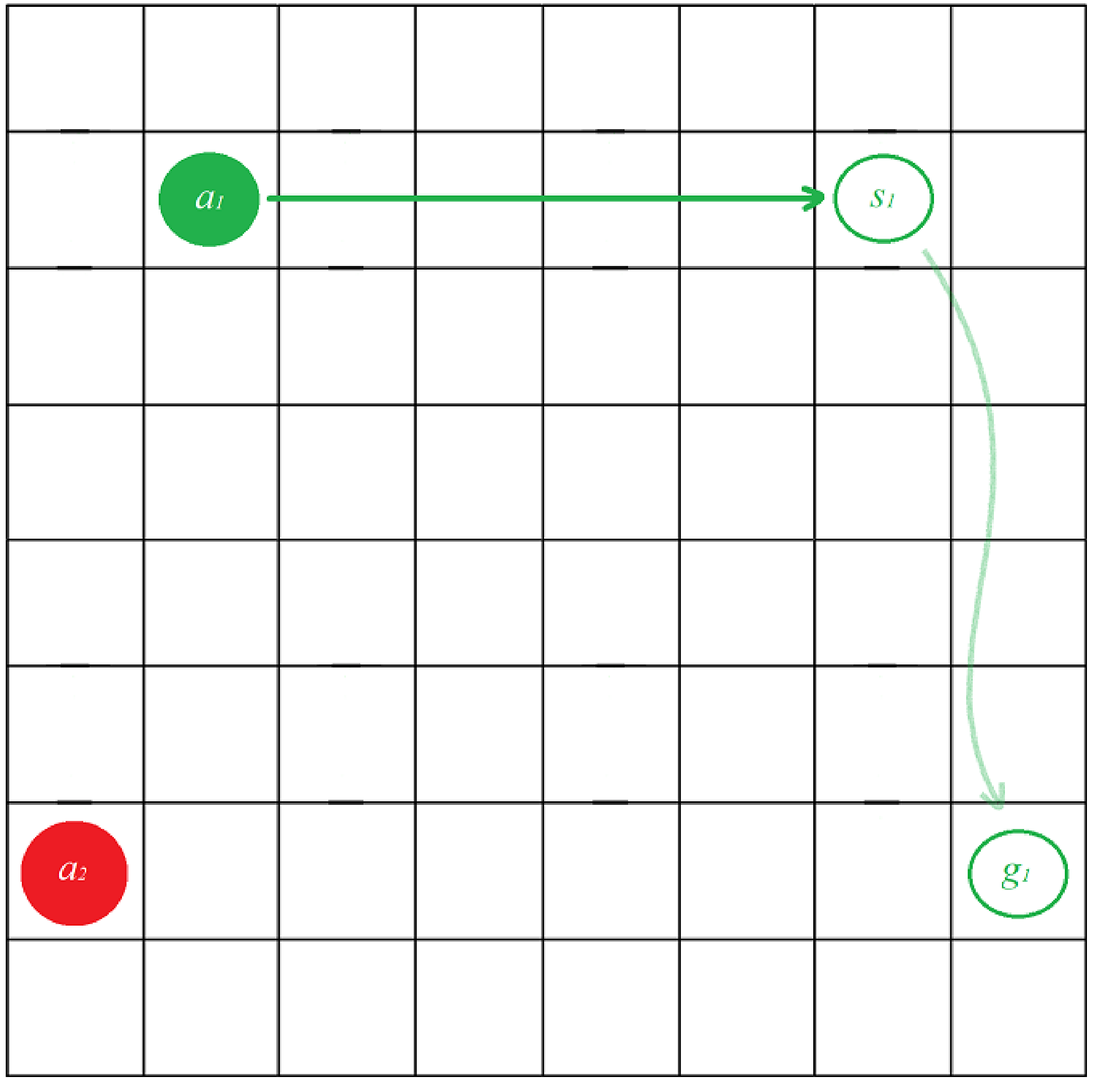
# Task Assignment

How should an agent decide whether to be assigned to a new task?



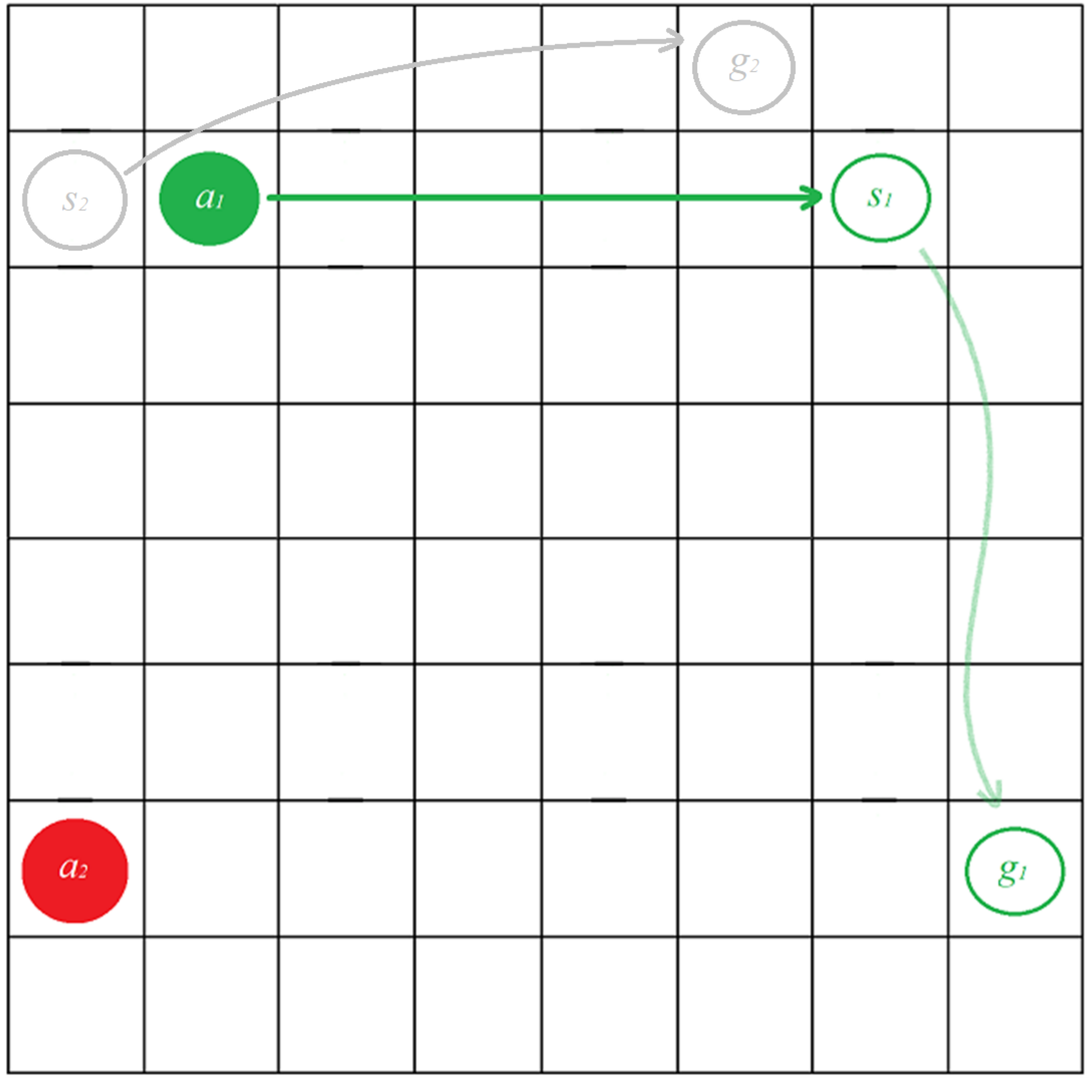
# Path Planning

What should a free agent do when the task set is empty?



# Path Planning

What should a free agent do when the task set is empty?



# MAPD-P

- Multi-Agent Pickup and Delivery with task probability distribution
- Possible pickup and delivery locations known in advance
- Function $P$: $N_0 \times V \times V \rightarrow [0, 1]$ → probability of appearance of a task
- $P_1$: $N_0 \times V \rightarrow [0, 1]$ → probability that a task with a given pickup location will appear at a time step $t$

# Token Passing (TP)

- Token shared among the agents

# Token Passing (TP)

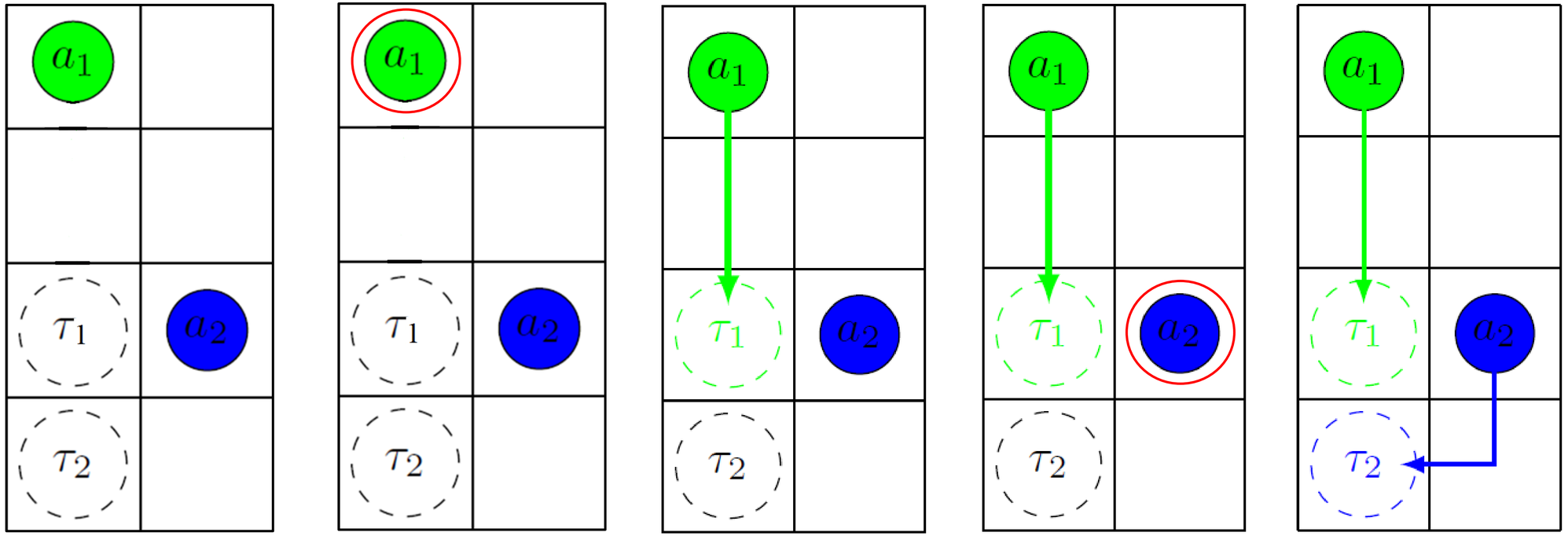
- Token shared among the agents
- Each free agent requests the token once per time step

# Token Passing (TP)

- Token shared among the agents
- Each free agent requests the token once per time step
- Agent is assigned to the available task with minimum cost path

# Token Passing (TP)

- Token shared among the agents
- Each free agent requests the token once per time step
- Agent is assigned to the available task with minimum cost path




*From [Ma et al., 2017]*

# Token Passing (TP)

**How should an agent decide whether to be assigned to a new task?**

**What should a free agent do when the task set is empty?**

# Token Passing (TP)

**How should an agent decide whether to be assigned to a new task?**

- Free agents always assigned to new tasks

**What should a free agent do when the task set is empty?**

- Free agents move to the closest non-conflicting endpoint

# Token Passing (TP)

**How should an agent decide whether to be assigned to a new task?**

- Free agents always assigned to new tasks

**What should a free agent do when the task set is empty?**

- Free agents move to the closest non-conflicting endpoint

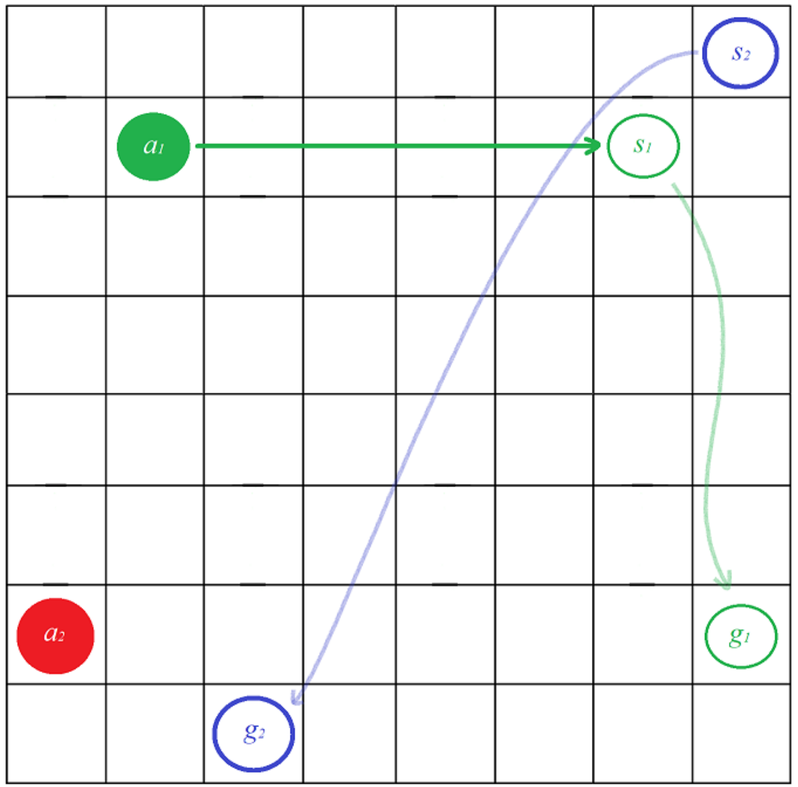
# Token Passing (TP)

**How should an agent decide whether to be assigned to a new task?**

- Free agents always assigned to new tasks

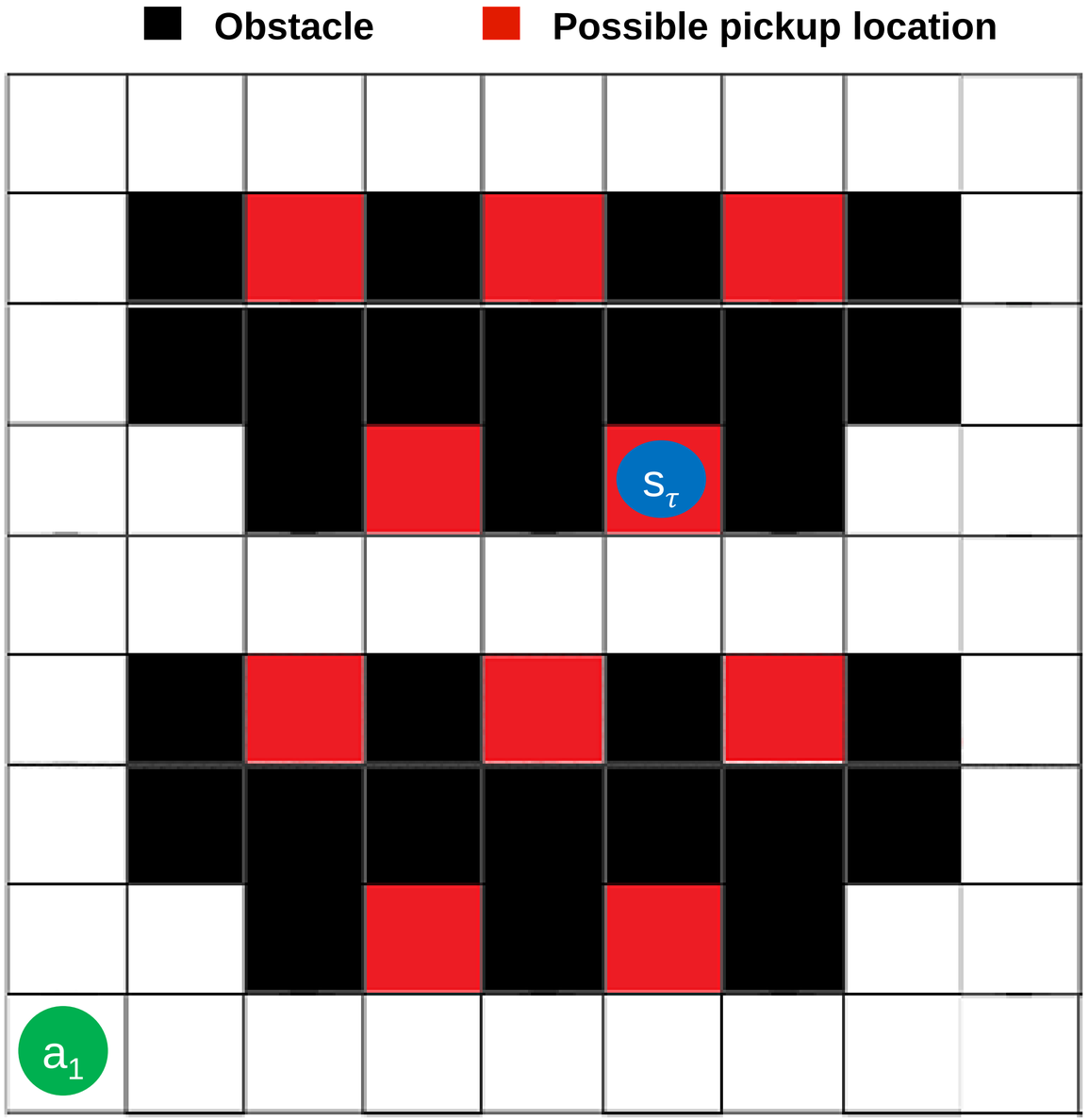
**What should a free agent do when the task set is empty?**

- Free agents move to the closest non-conflicting endpoint



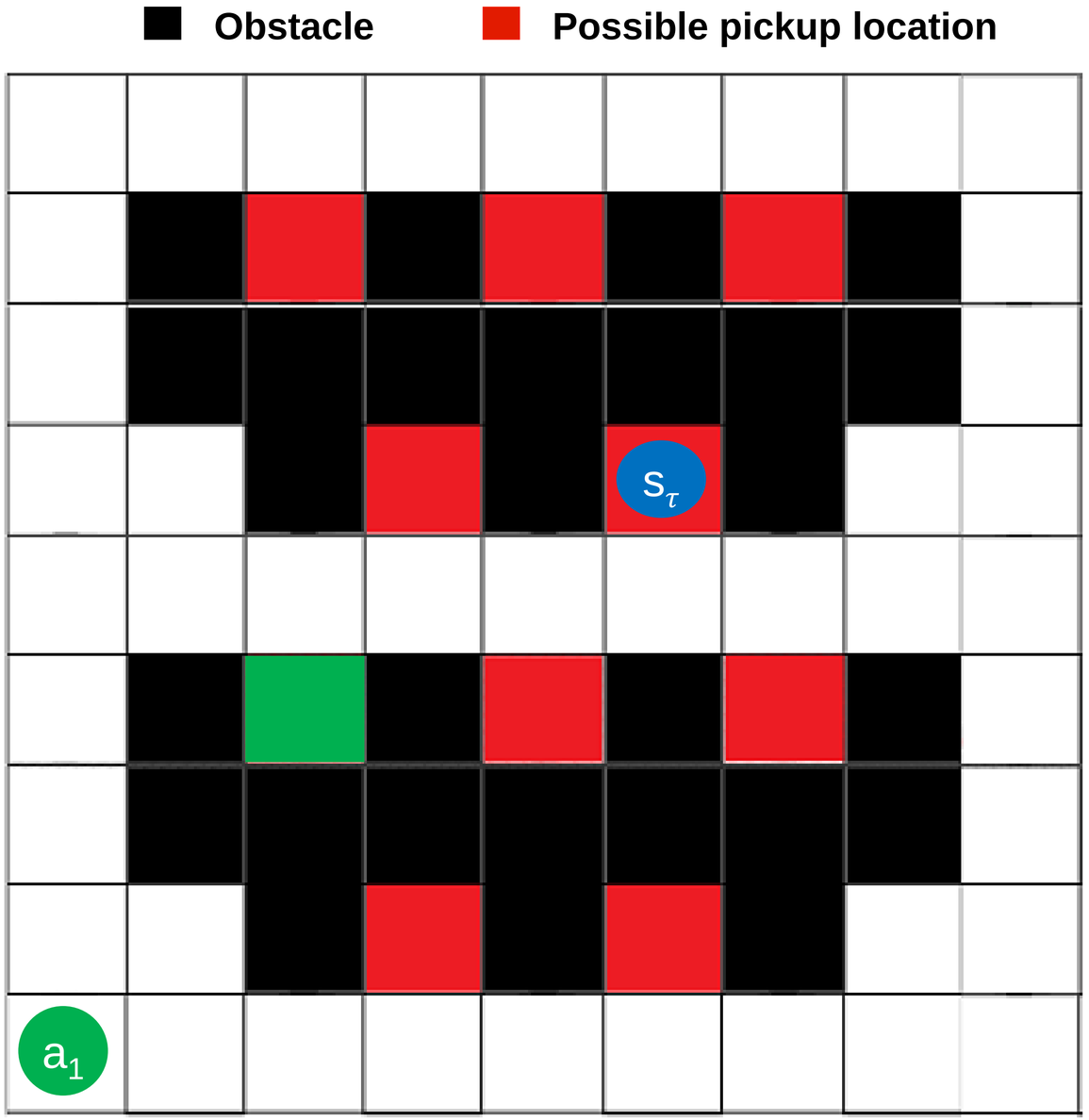
# TP-m1

- Functions $p$ and $t$ provide attractiveness of empty and not empty pickup locations



# TP-m1

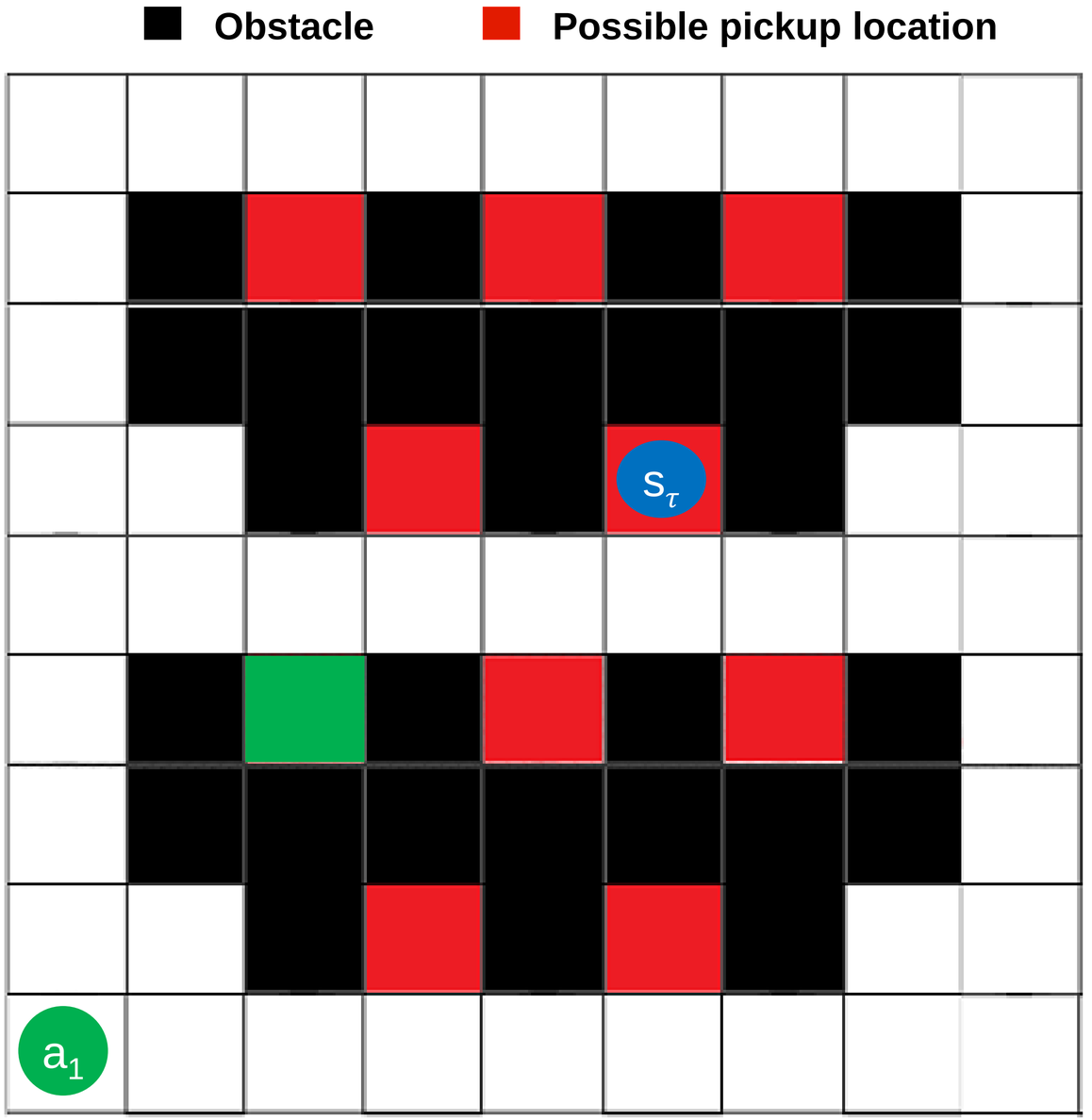
- Functions $p$ and $t$ provide attractiveness of empty and not empty pickup locations
- Most attractive empty pickup location $s$ is selected according to a function $p$
- $s$ is chosen as destination if $s$ is closer than the task $\tau$ and the value of function $p$ for $s$ is greater than the value of function $t$ for $\tau$



# TP-m1

$$p(T, s, a_i) = \frac{\sum_{t=T+1}^{T+1+h(loc(a_i),s)} P_1(t, s)}{h(loc(a_i), s) + 1}$$

$$t(a_i, \tau) = \frac{1}{h(loc(a_i), s_\tau) + 1}$$



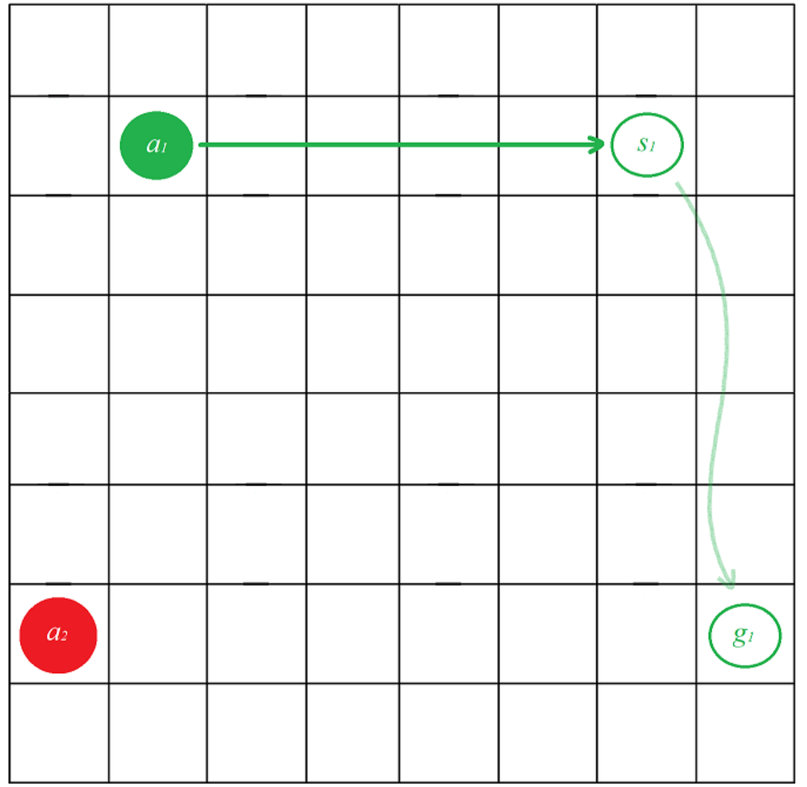
# Token Passing (TP)

How should an agent decide whether to be assigned to a new task?

- Free agents always assigned to new tasks

**What should a free agent do when the task set is empty?**

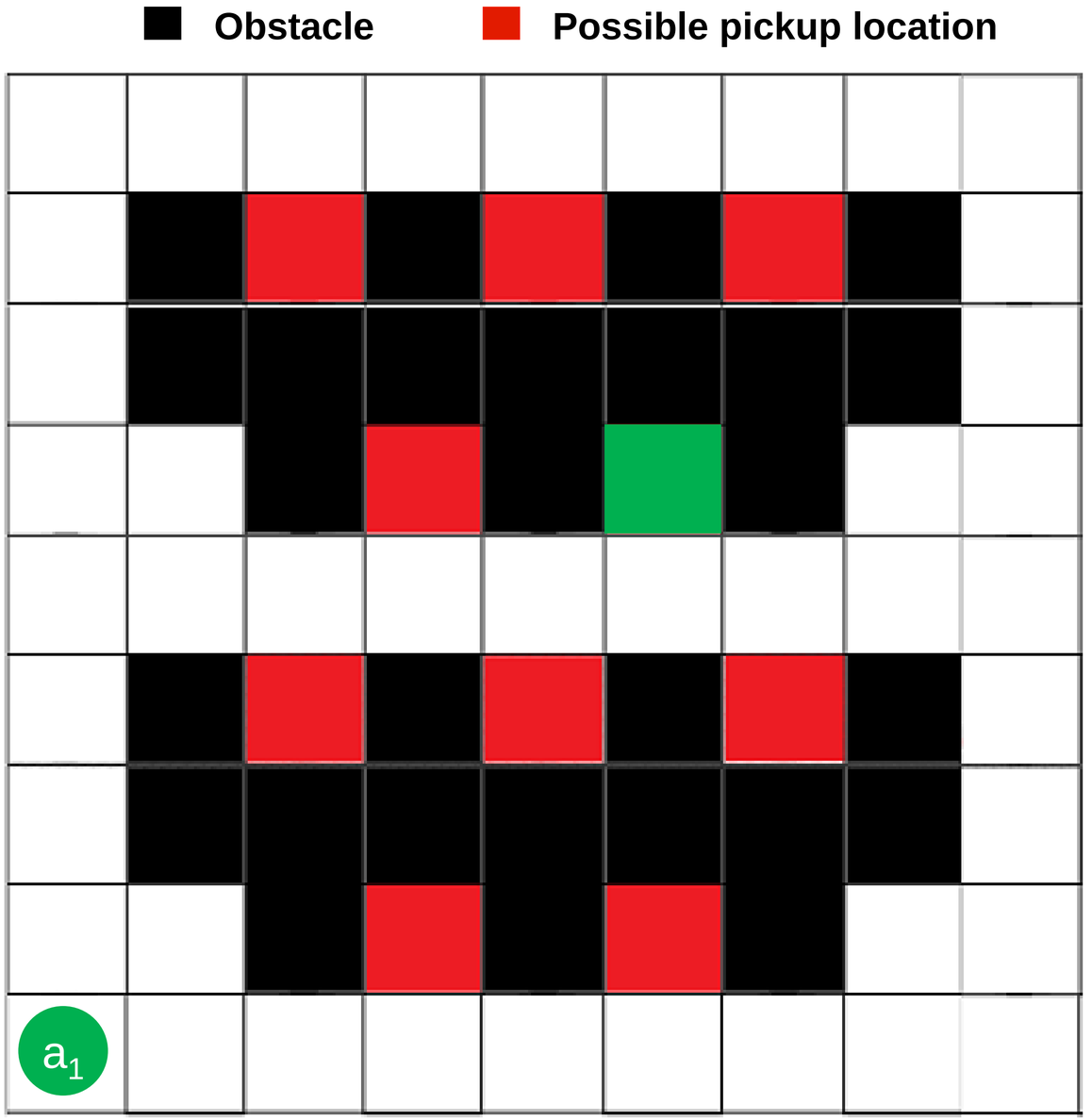
- Free agents move to the closest non-conflicting endpoint



# TP-m2

- All the possible reachable pickup locations are analysed and the most attractive location $s$ is selected

- If no pickup location is available, the logic of TP is followed



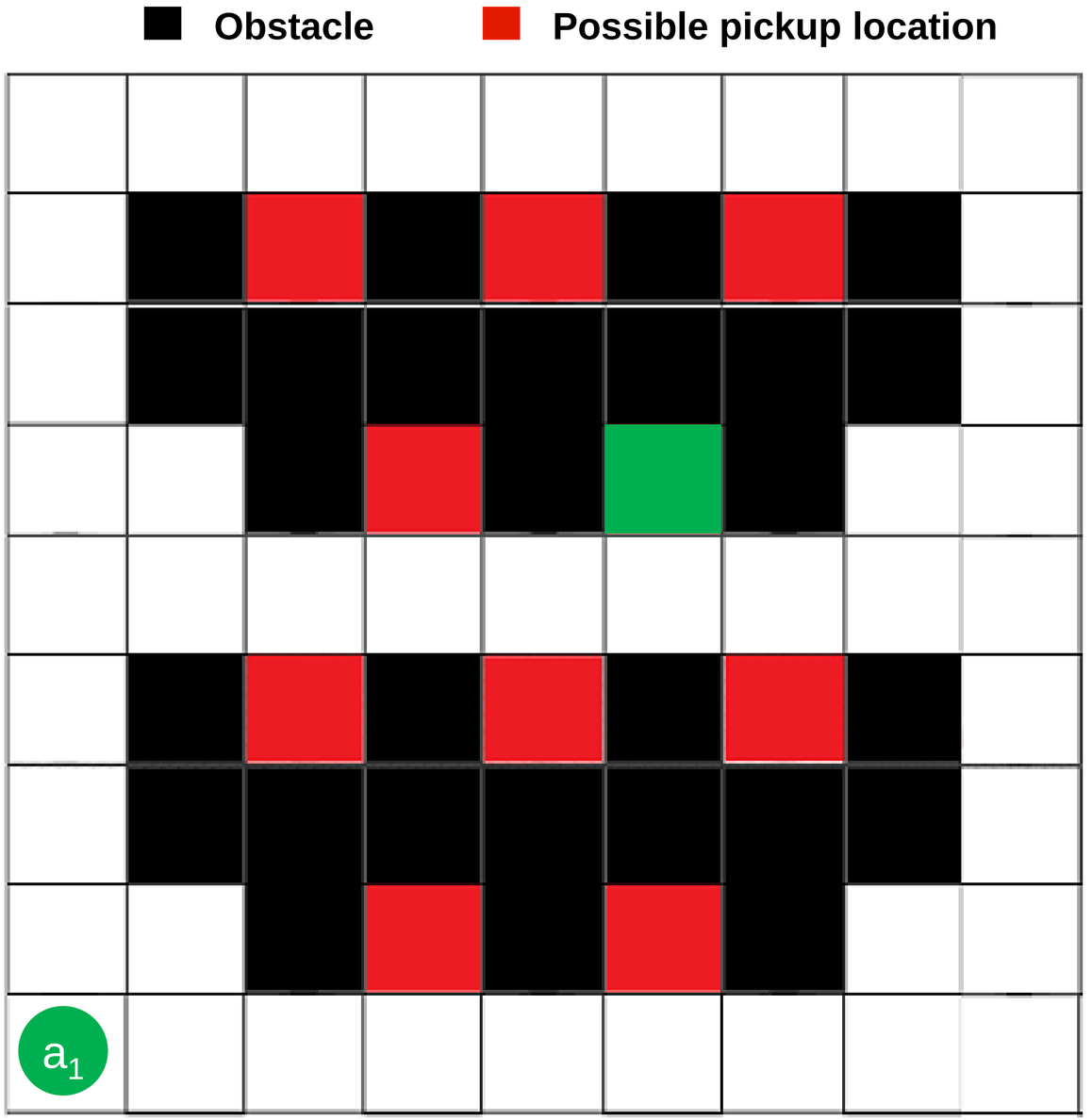
# TP-m2

- All the possible reachable pickup locations are analysed and the most attractive location $s$ is selected

- If no pickup location is available, the logic of TP is followed

Combined with TP-m1

**TP-m**



# TP-m

- ✓ How should an agent decide whether to be assigned to a new task?
- ✓ What should a free agent do when the task set is empty?

# TP-m

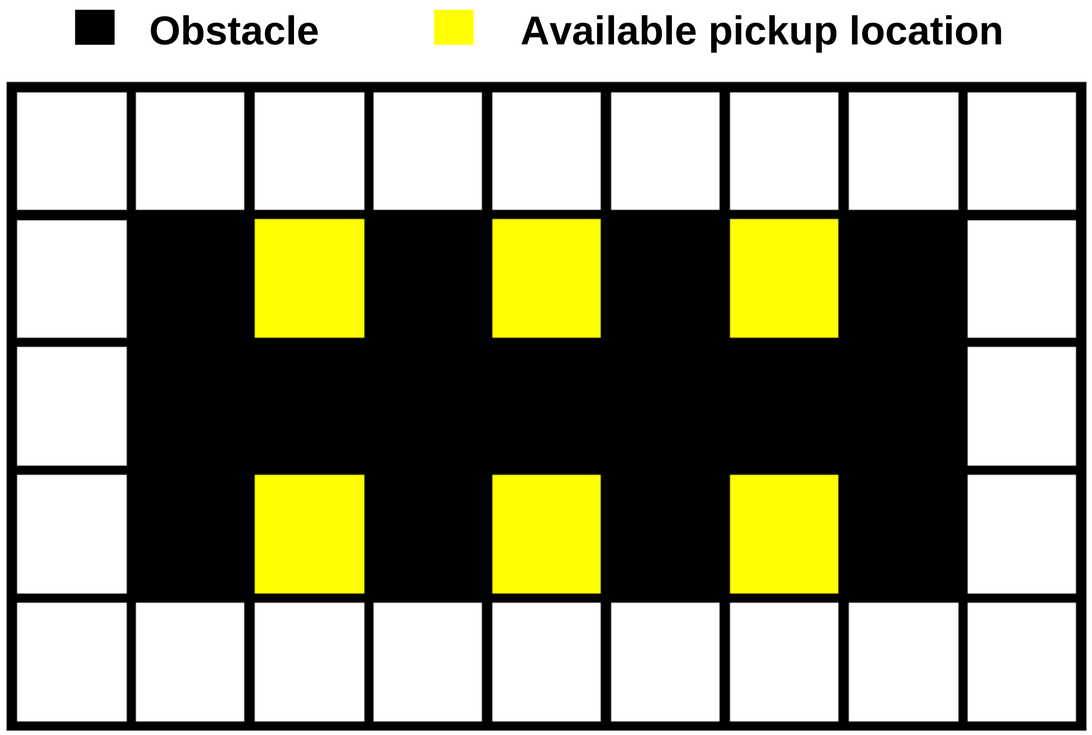
✓How should an agent decide whether to be assigned to a new task?

✓What should a free agent do when the task set is empty?

- What happens if the prediction does not come true?

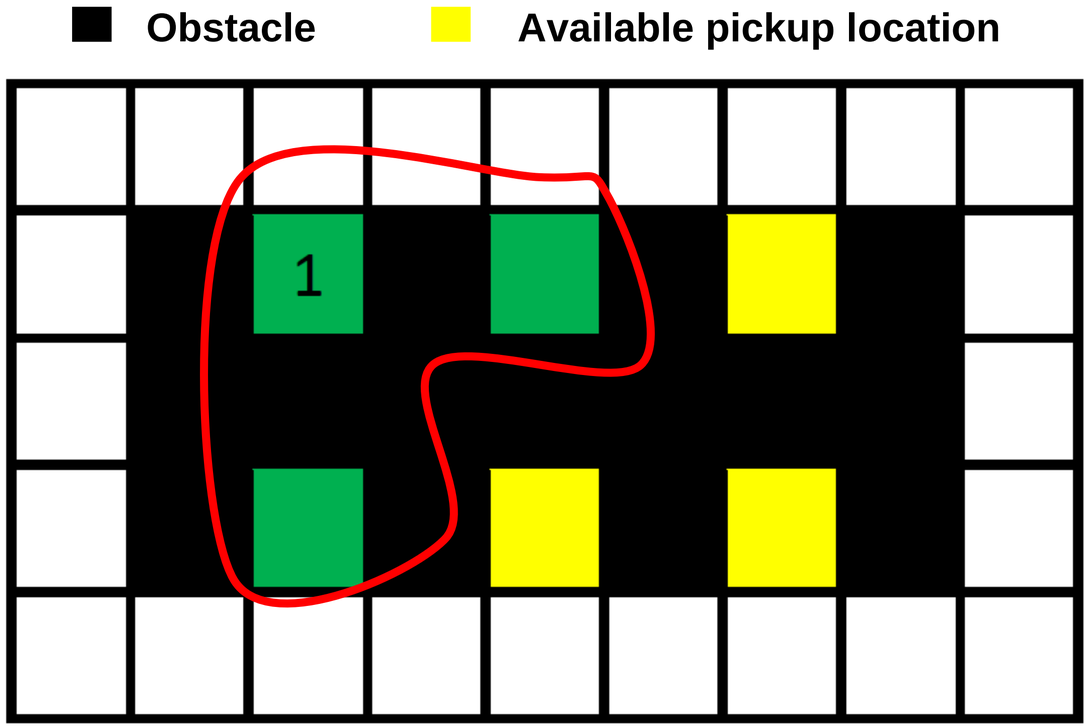
# TP-m with preemption

- Two parameters:
  - Preemption distance
  - Preemption duration



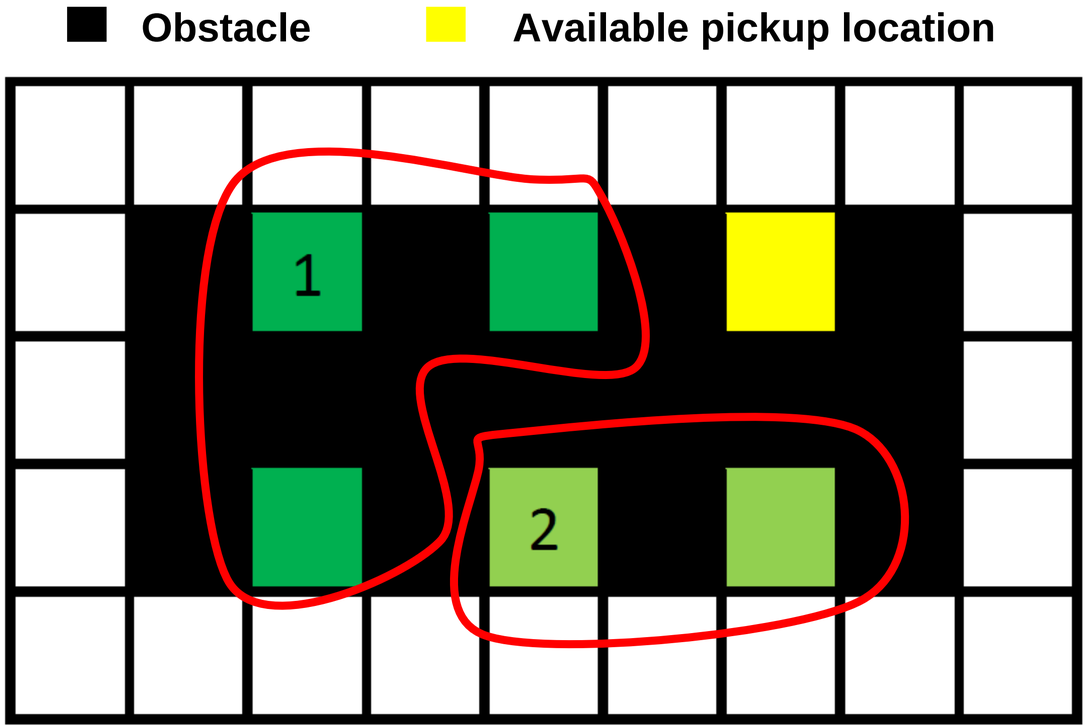
# TP-m with preemption

- Two parameters:
  - Preemption distance
  - Preemption duration



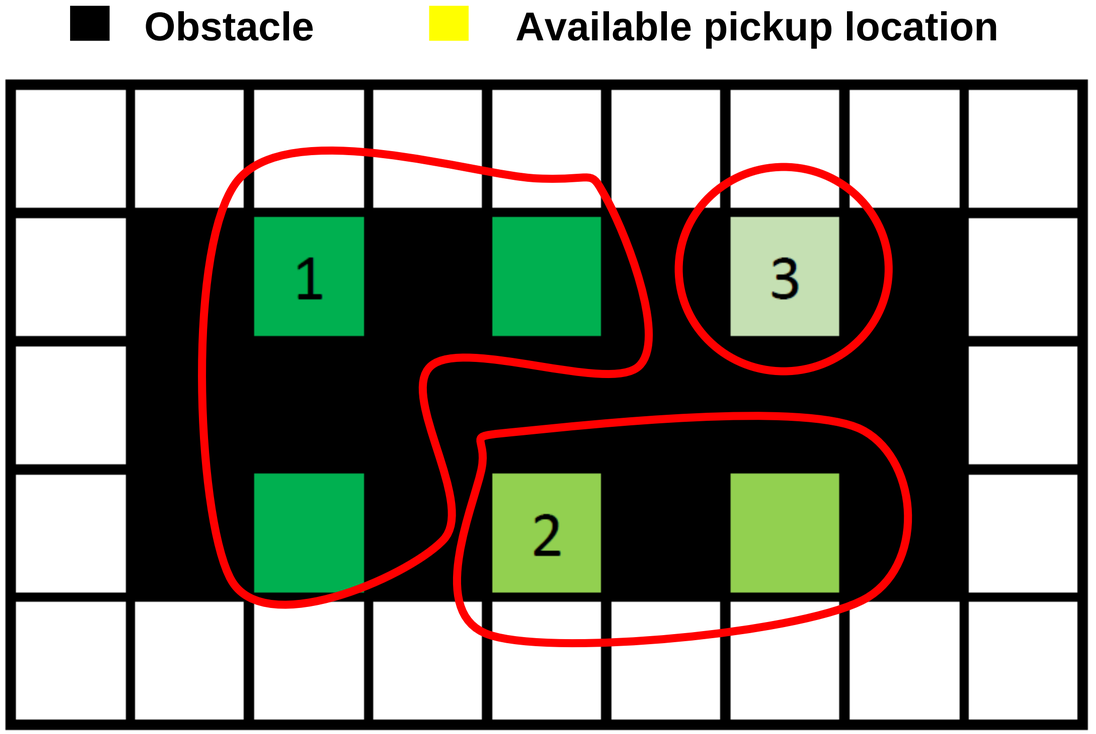
# TP-m with preemption

- Two parameters:
  - Preemption distance
  - Preemption duration



# TP-m with preemption

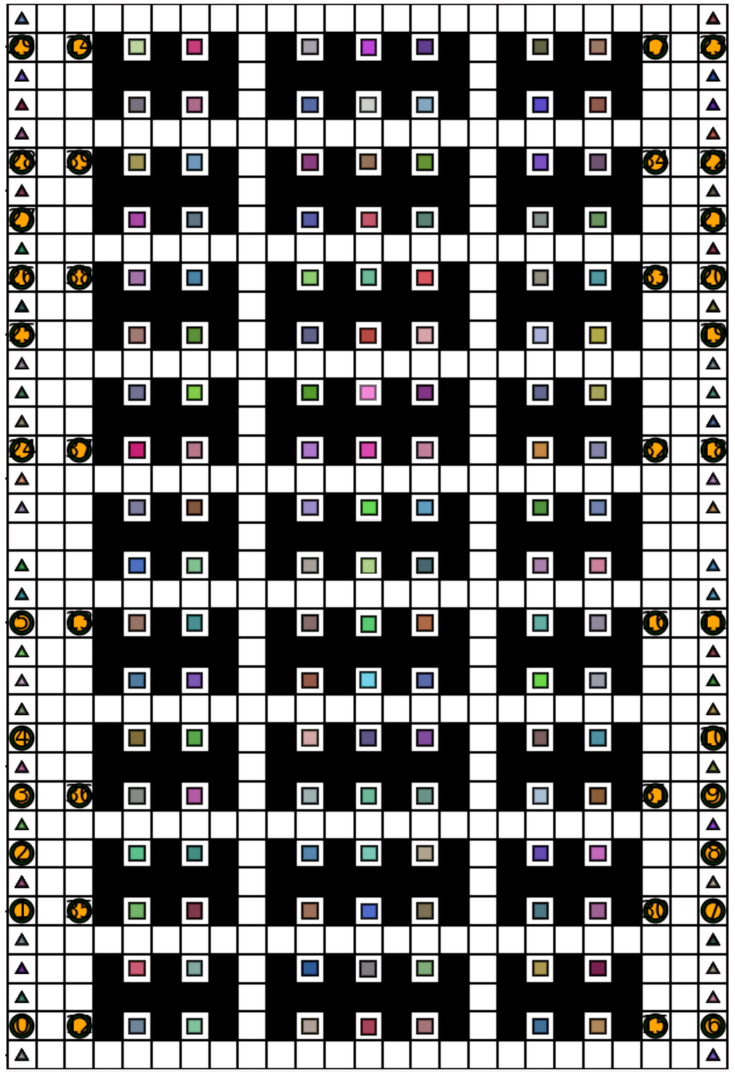
- Two parameters:
  - Preemption distance
  - Preemption duration

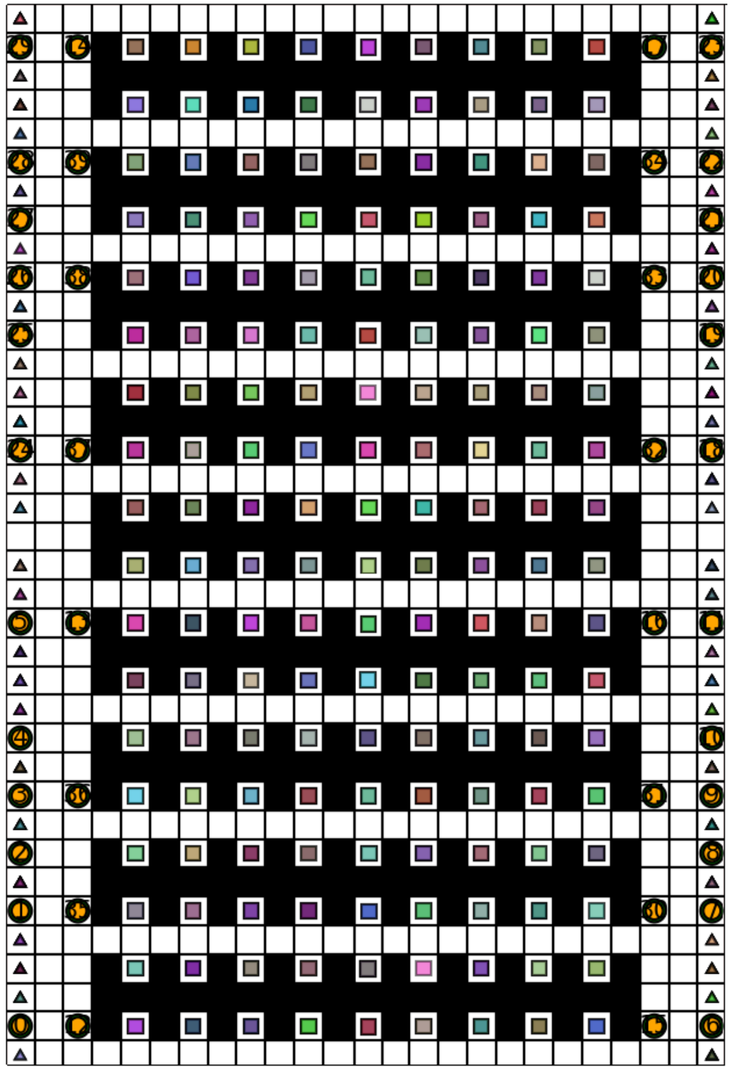


# Experimental results

## Evaluation metrics

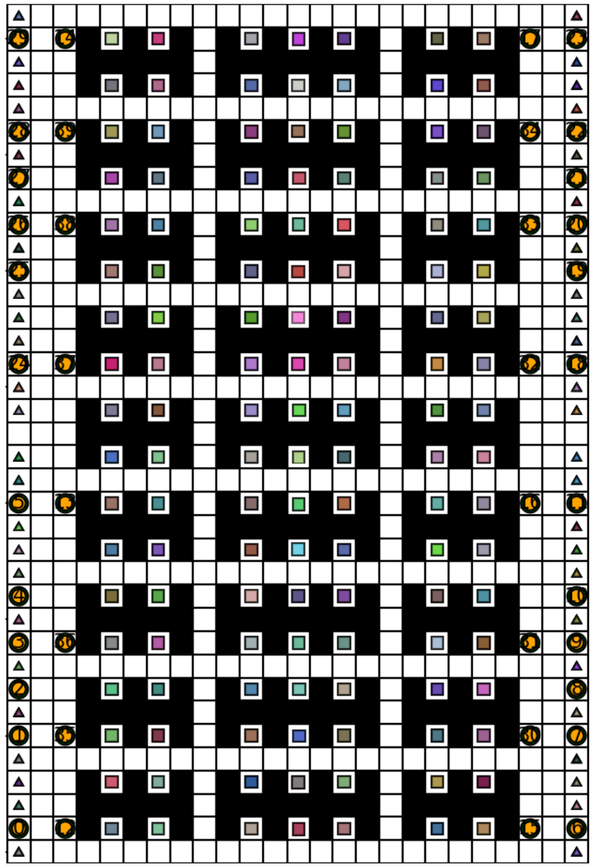
- **Service time**
- **Cost of the solution per task**
- Runtime cost
- Makespan
- Throughput





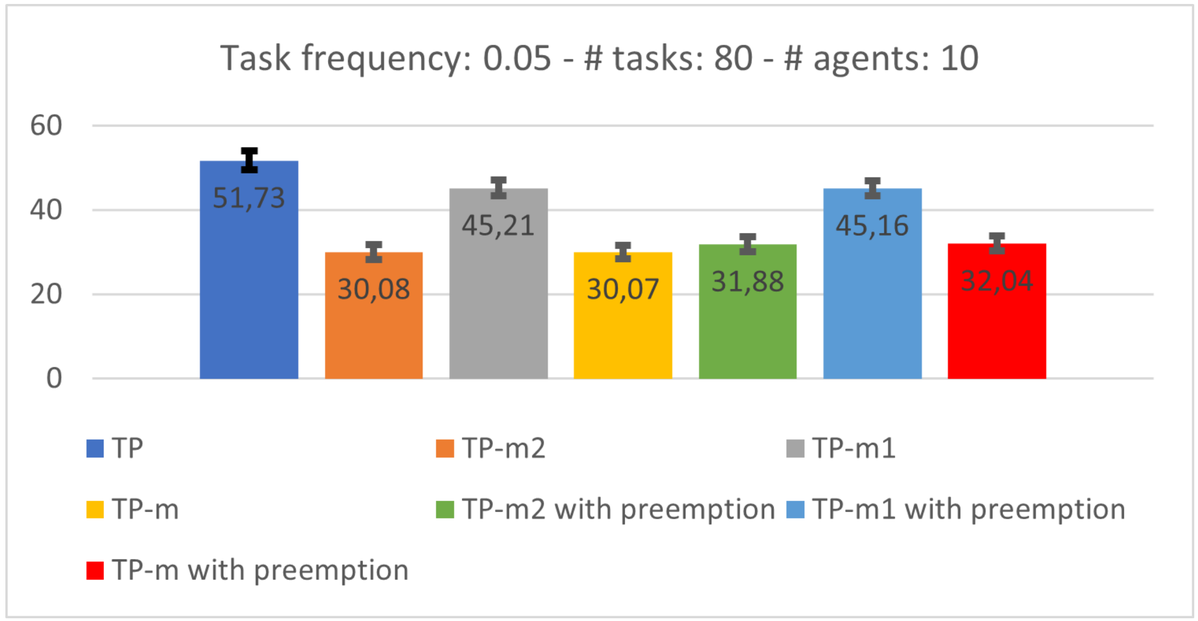
# Experimental results

- Task frequencies: 0.05, 0.1, 0.2, 0.5
- Agents: 10, 20, 30, 40
- Tasks: 80
- Number of runs: 20
- Preemption distance: 3
- Preemption duration: 3



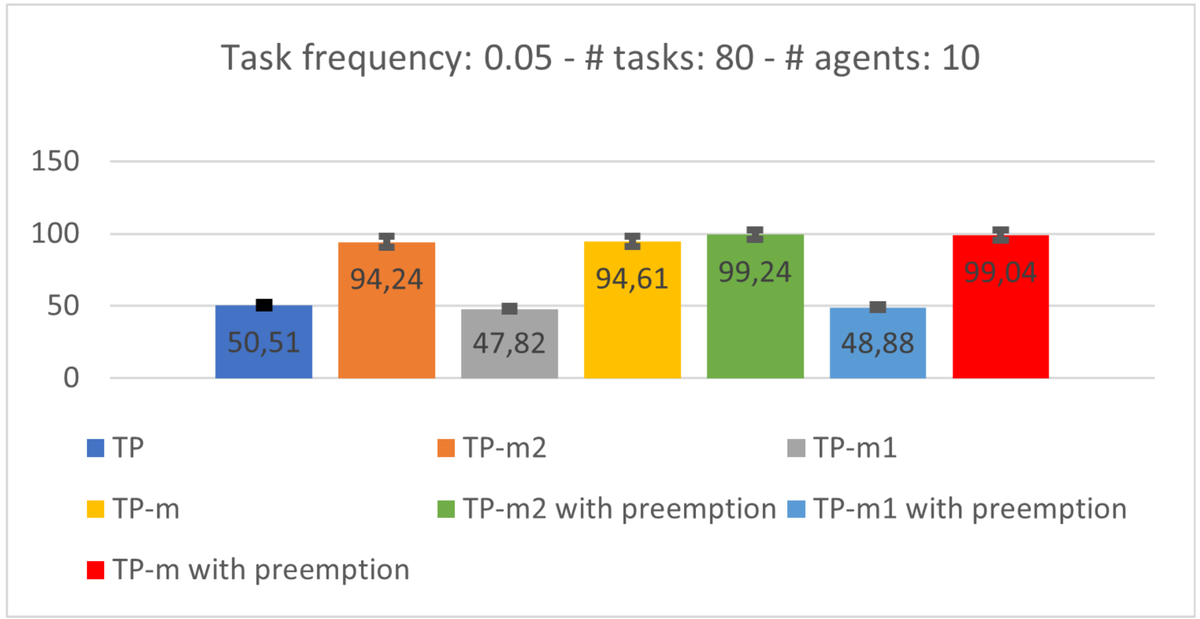
- Perfect model of the tasks used to define the probability distribution

# Experimental results: low task frequency

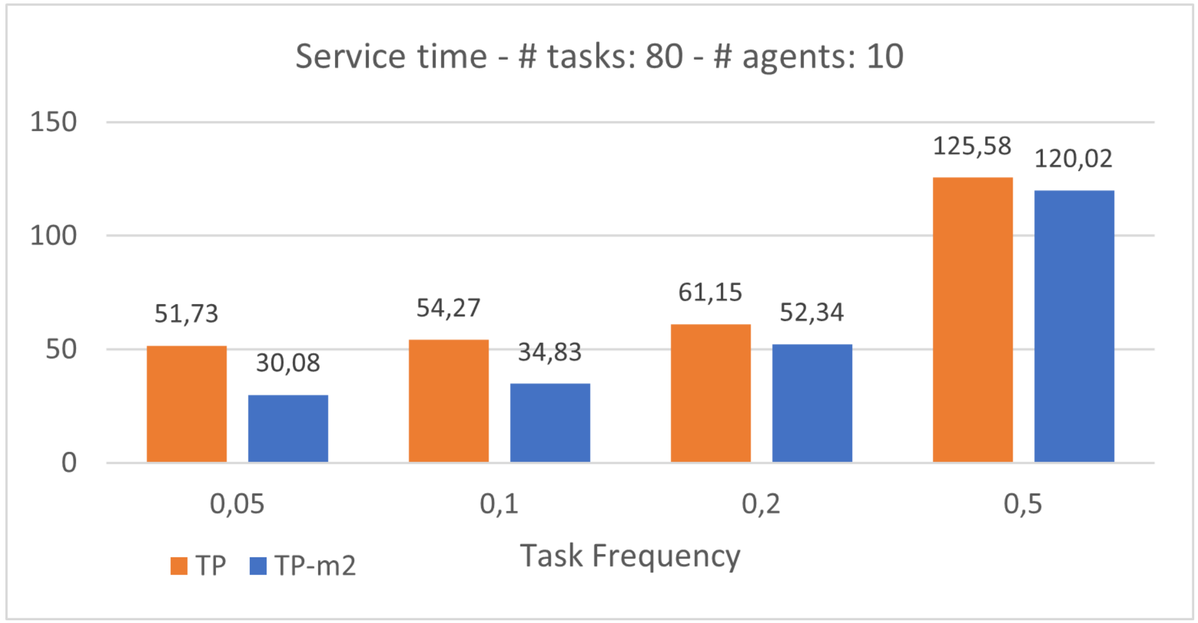


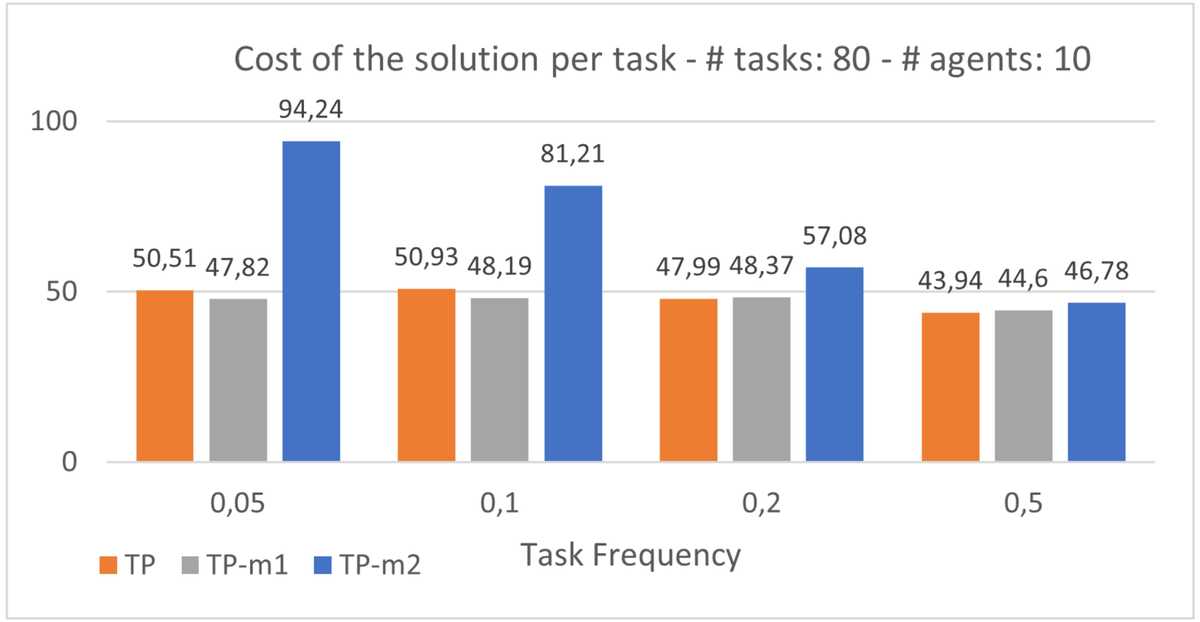

Service time

Cost of the solution per task

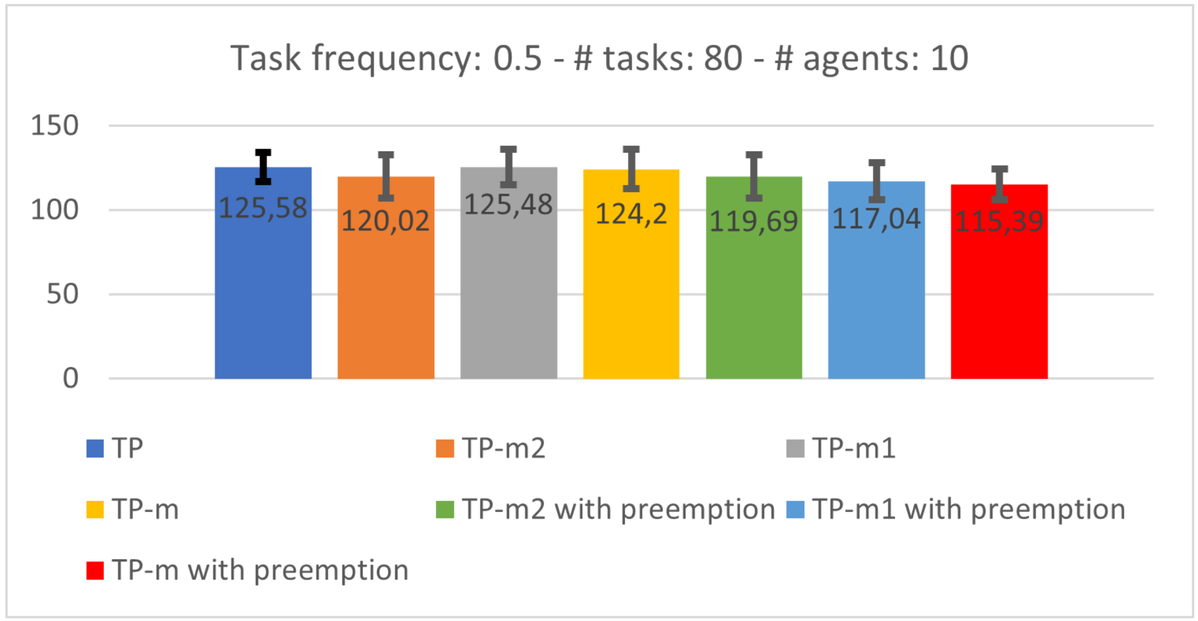


# Experimental results



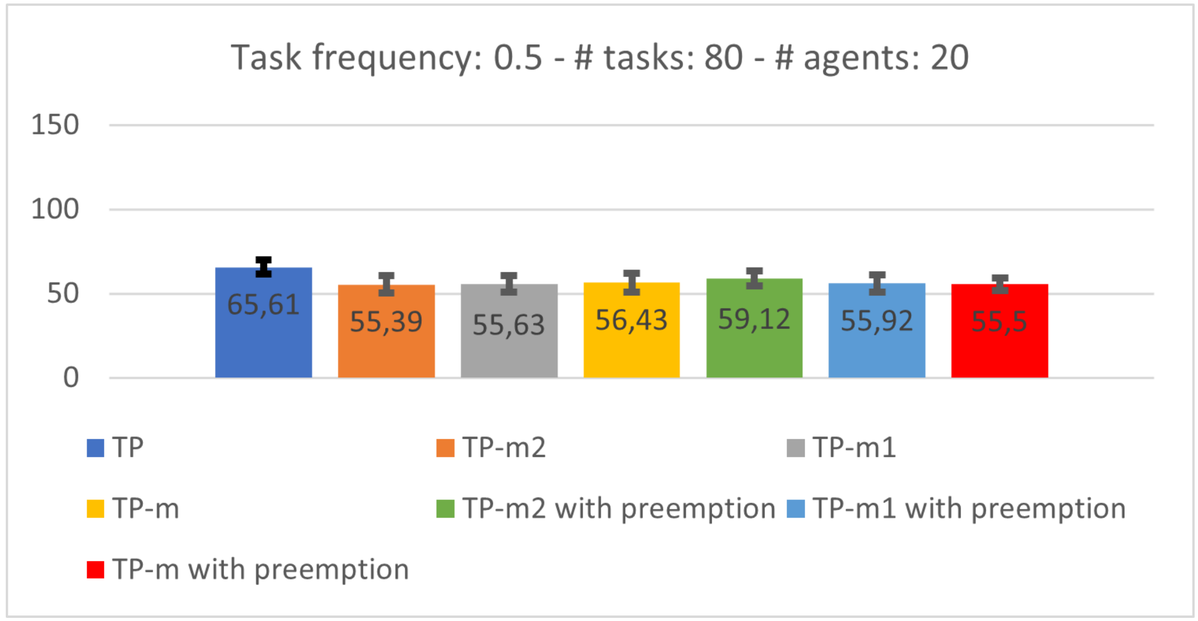



# Experimental results: high task frequency

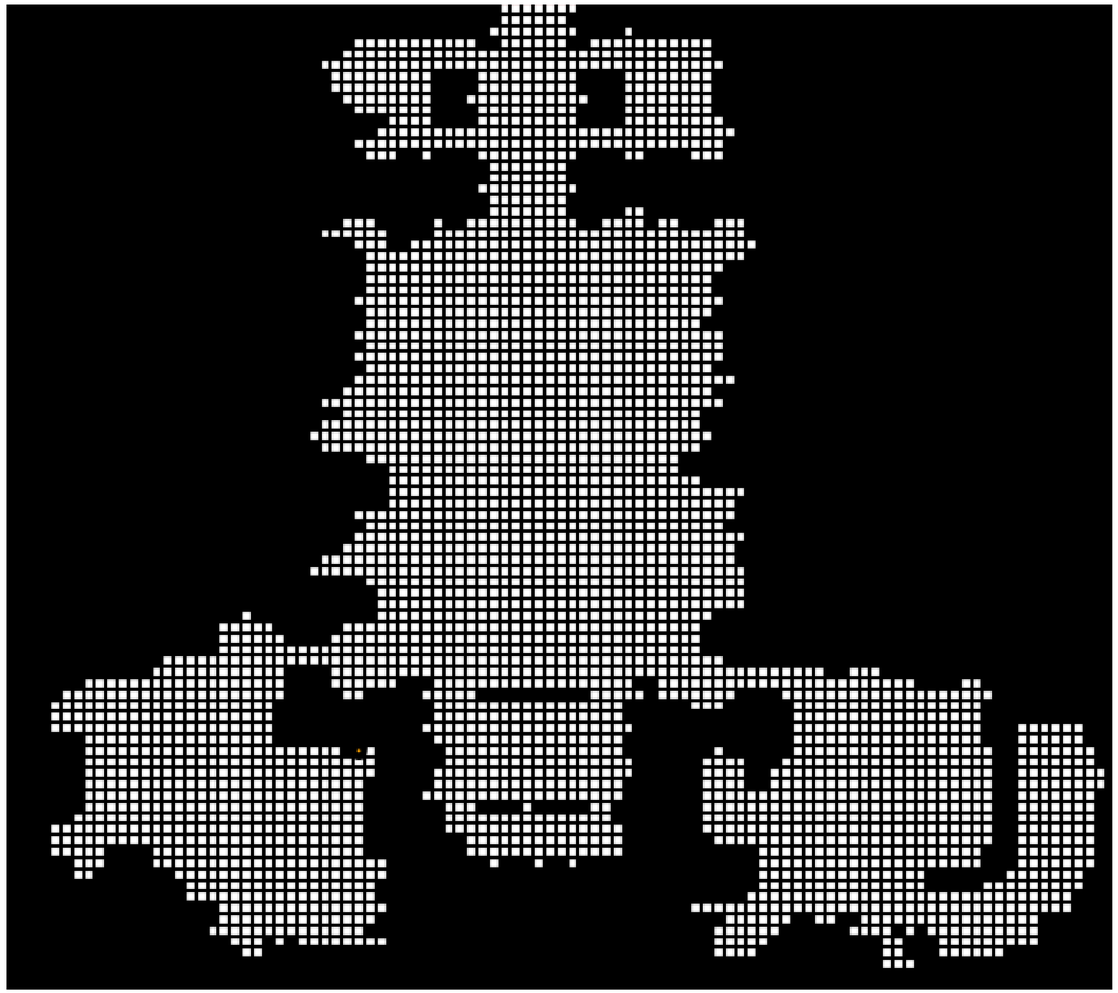



Service time

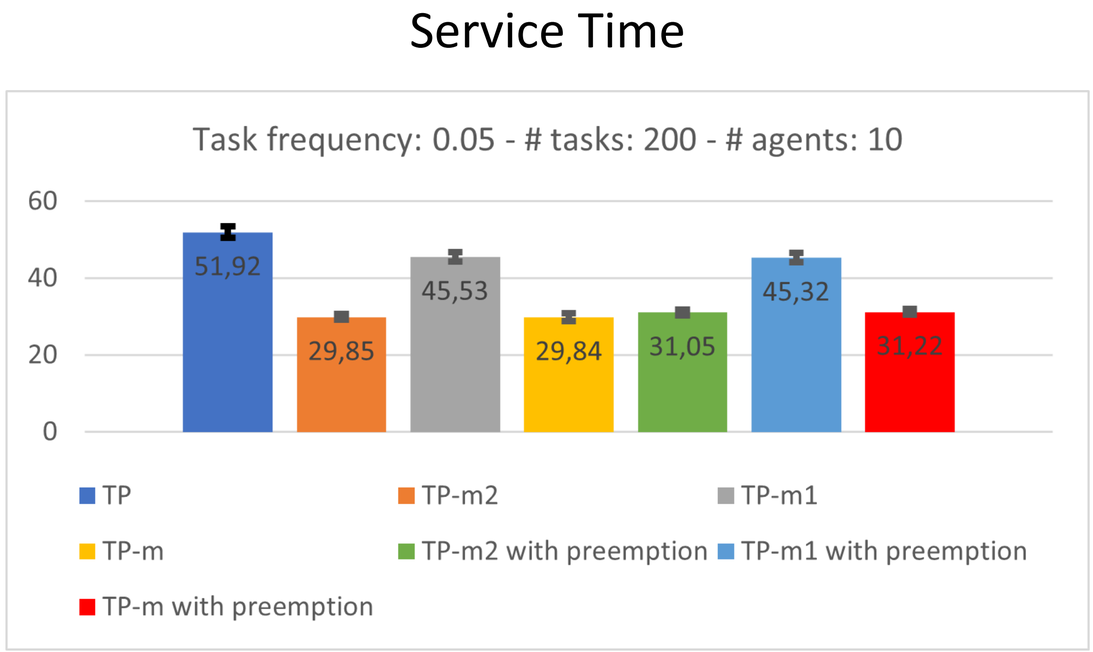
Service Time



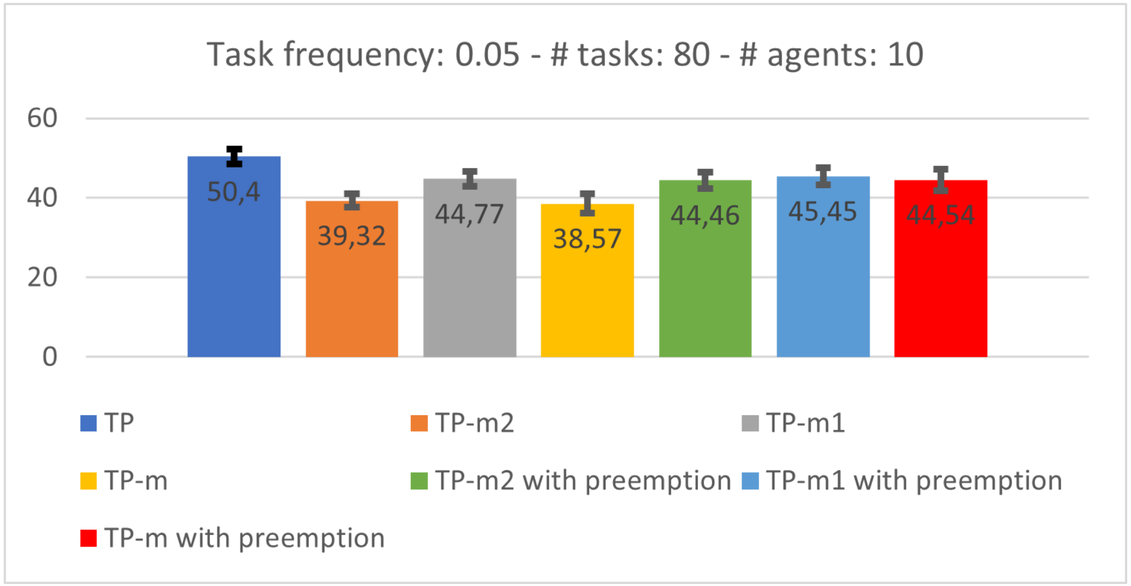
# Experimental results: scalability



Videogame benchmark from *mapf.info*

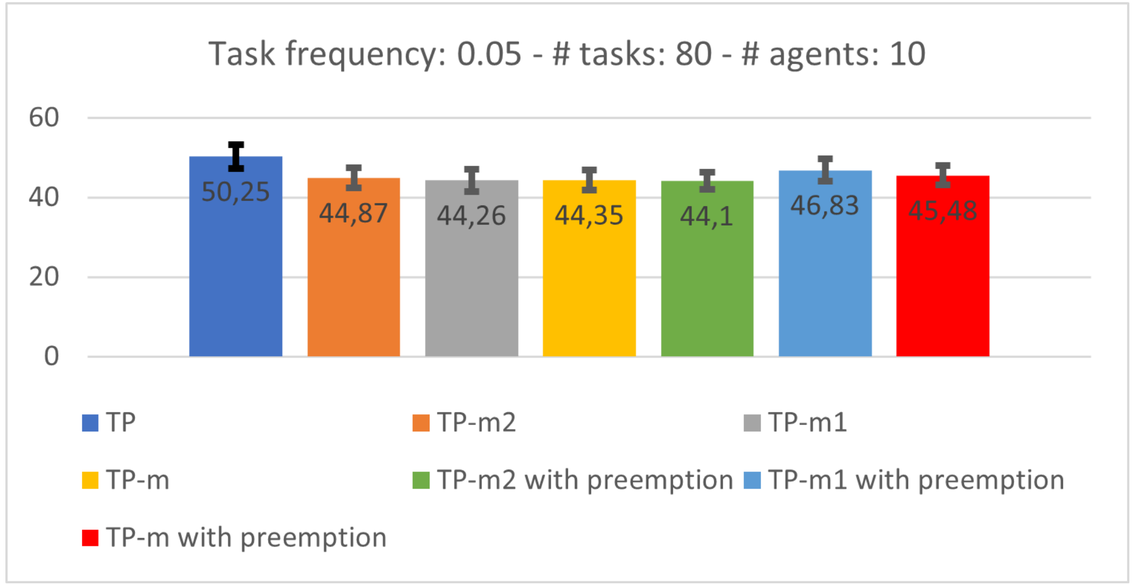


# Experimental results: inaccurate task probability distribution




Small Noise - Service Time

Large Noise - Service Time



# Conclusion and future work

- Defining a new model for MAPD that includes the task probability distribution
- Using the task probability distribution for task assignment and path planning
- Evaluating the effects on service time and cost metrics
  - TP-m2 → High impact on service time, significant increase in cost of the solution
  - TP-m1 → Lower impact on service time, stable cost of the solution. Trade-off between different metrics
- Future work
  - Testing other configurations
  - Allowing free moving agents to request the token

Thank you for your attention!