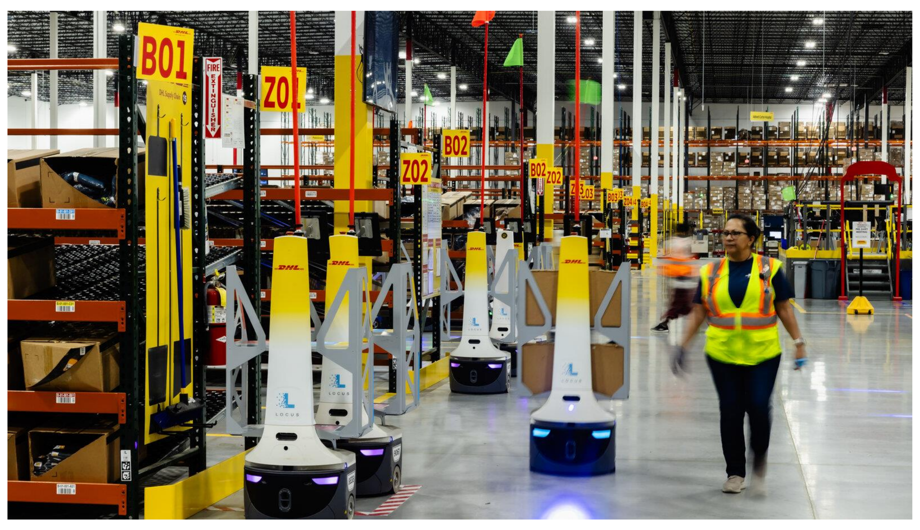
Multi-Agent Pickup and Delivery in Human-Populated Environments

Benedetta Flammini[1], Leo D'Amato[2], and Francesco Amigoni[1]

[1]Politecnico di Milano, Milan, Italy
[2]National Research Council of Italy, Rome, Italy

INTRODUCTION

Multi-Agent Pickup and Delivery (MAPD) [Ma et al., 2017] problem: **finding collision-free paths for a team of agents** executing dynamically incoming pickup and delivery tasks.



Standard MAPD solvers treat **robots as the only dynamic entities**, which is not realistic in many real world applications, such as hospitals, shopping malls, and warehouses.

PROBLEM STATEMENT

Setting

- Environment $G = (V, E)$, a 4-connected grid (vertices = locations, edges = connections)
- Team agents $\mathcal{A} = \{a_1, a_2, \dots, a_k\}$ that have to complete a stream of pickup and delivery tasks
- Time is discrete and, at each time step, an agent can move to an adjacent vertex or wait at its current vertex
- Set of humans $\mathcal{H}$ coexisting in G; $|\mathcal{H}|$ is not fixed, humans enter and exit freely and can jump multiple cells per time step and move diagonally

Assumptions

- Agents cannot communicate with or restrict human movements
- Delivery and parking locations cannot be accessed by humans
- Each agent observes humans within a predefinied observation radius obs (based on Manhattan distance)
- A reactive safety layer is enforced at execution time: each agent observes humans within obs, and a planned move is executed only if its Chebyshev distance from every observed human is greater than one

We introduce **MAPD in Human-Populated Environment (MAPD-HPE)**, where agents have to complete a stream of **pickup and delivery tasks** in an environment **shared with humans**.

The aim for agents is **to complete all tasks while minimizing completion time**, by finding collision-free paths not only with respect to other agents but also with respect to humans, thus **reducing any disturbance** caused to them.

PROPOSED SOLUTION

We propose an algorithm in which agents, in addition to reactively avoiding humans, **observe humans and build a model of their behaviour**, exploiting the fact that human movements are often predictable. Agents then incorporate this information in their planning phase.

Graph Neural Network Model

Each grid cell $v \in V$ is assigned a feature vector at each time step t:

$$\boldsymbol{x}_v^t = (\boldsymbol{o}_v^t, \boldsymbol{tod}^t, \boldsymbol{dow}^t) \in \mathbb{R}^3$$

where $o_v^t \in \{0,1\}$ represents binary occupancy, tod^t is encoded time-of-day $\in [0,1]$, and dow^t is encoded day-of-week $\in [0,1]$, at time t.

Given $T_{in} = 20$ past snapshots, at time step t the GNN predicts:

$$\boldsymbol{P}^{t+1:t+T_{out}} = \boldsymbol{f_\theta}(\boldsymbol{X}^{t-T_{in}+1:t}, \boldsymbol{G})$$

that is the occupancy probability over the next $T_{out} = 15$ future steps, where $X^t \in \mathbb{R}^{|V|\times 3}$ collects the feature vectors of all vertices at time step t, and θ contains the parameters of the model.

Model and Plan Integration

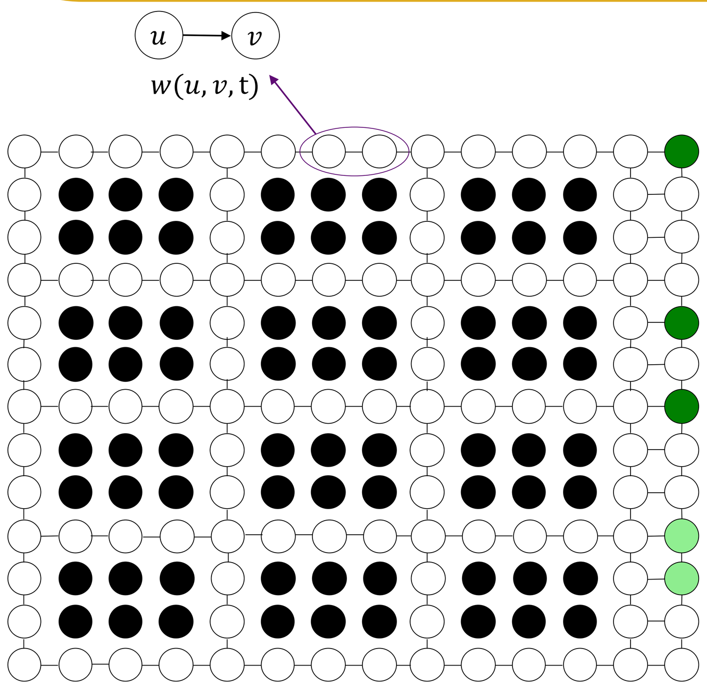
Objective: plan paths **avoiding cells with high occupation probability while keeping a reasonable path length**.

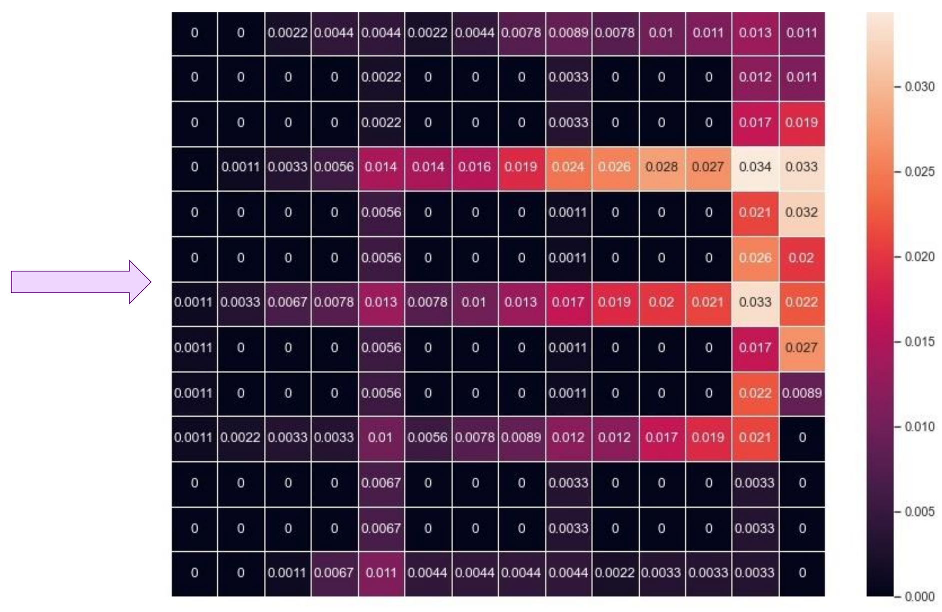
Agents plan their paths using time-extended A* on a graph with weighted edges:

$$w(u,v,t) = (1-\alpha)\frac{d(u,v)}{diam(G)} + \alpha p_v^{t+1}$$

Linear combination of **distance** and **occupation probability**

where p_v^{t+1} is the GNN–predicted occupation probability of v at time step $t+1$, $d(u,v)$ the distance between vertices u and v, $diam(G)$ the diameter of graph G, and $\alpha \in [0,1]$.





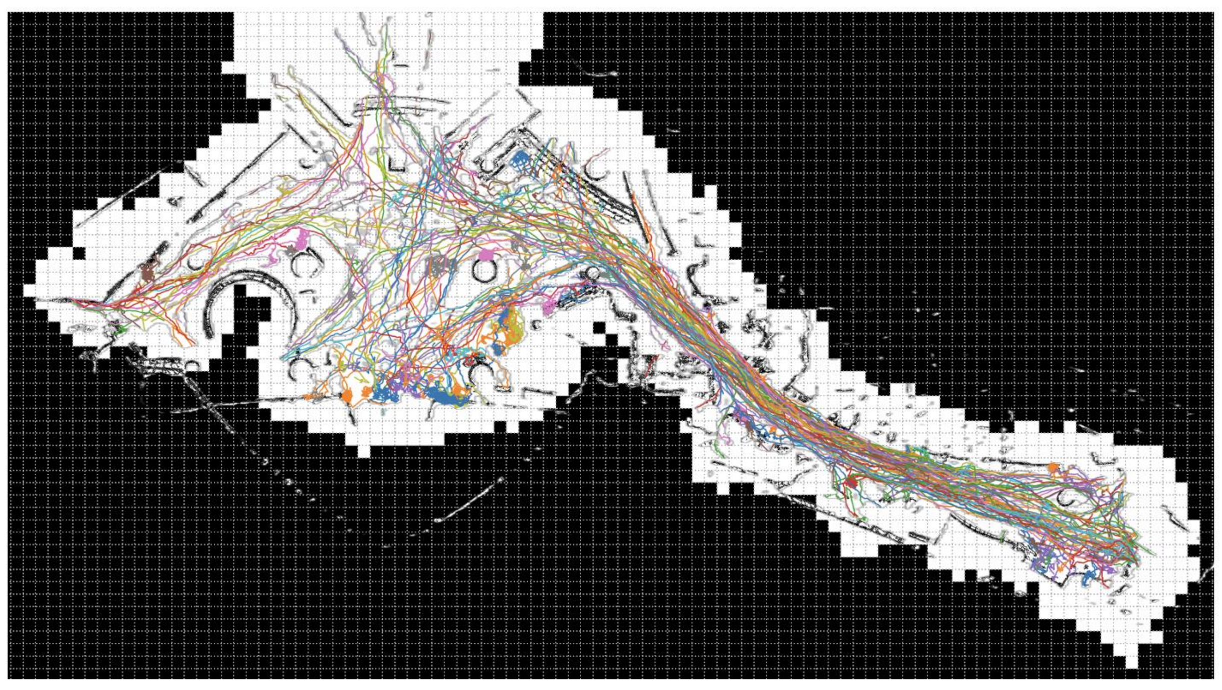
Token Passing + Rolling Horizon

The planning framework operates at two levels.

At the **coordination level**, Token Passing (TP) [Ma et al., 2017] manages task assignment and high-level conflict resolution: agents share a common token storing all current paths and task assignments, and each agent acquires it in turn to compute a conflict-free path, treating all other planned paths as obstacles.

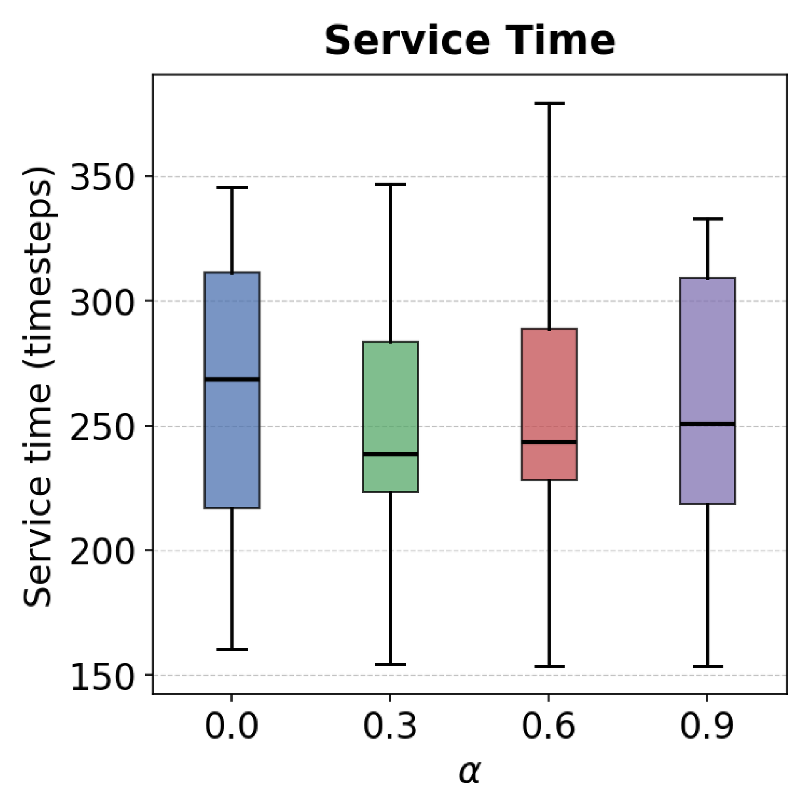
A **rolling horizon strategy** (plan only the first H steps) ensures that updated GNN predictions are continuously exploited: agents replan whenever fewer than ρ steps remain in their current path.

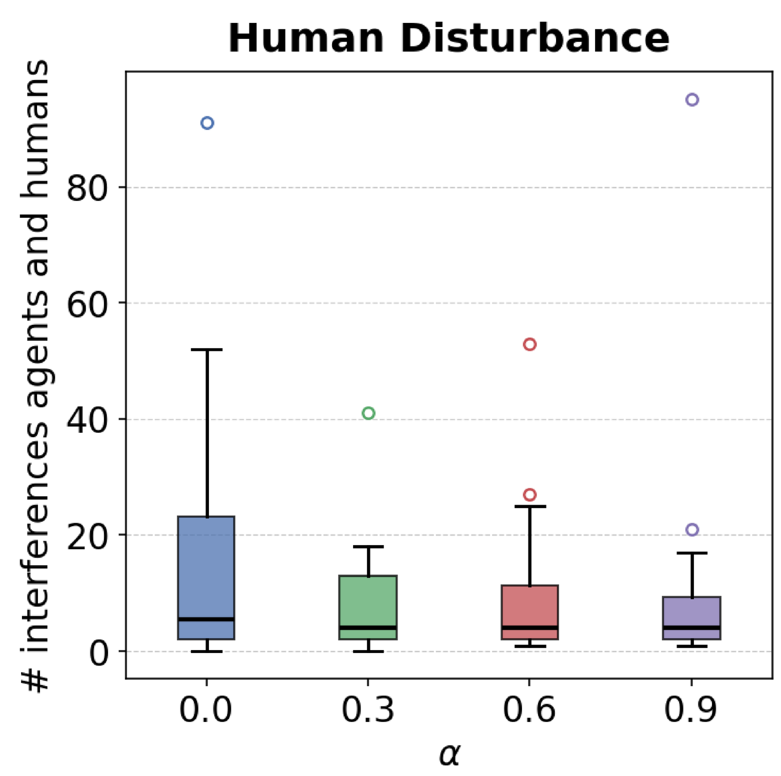
EXPERIMENTAL RESULTS



The dataset [Bršćić et al., 2013] describes the behaviour of people moving in the ATC business and shopping centre in Osaka, Japan.

Data were collected on Wednesdays and Sundays, from morning until evening.





Setting

- Number of robots: $k = 10$
- Tasks per run: 50
- Observation radius $obs = 10$ cells
- Planning horizon $H = 30$ steps
- Replan threshold $\rho = 15$ steps
- α values tested: $\{0, 0.3, 0.6, 0.9\}$
- Runs per configuration: 25
- Deliveries are located at the borders
- Pickups are sparsely diffused
- 92 days (82 training, 10 testing)

Metrics

- **Service time:** average #steps to complete a task since its appearance
- **Human disturbance:** #times paths of agents interfere with those of humans

Algorithms

- **Token Passing + Rolling Horizon**
- **Purely reactive approach** ($\alpha = 0$)

- All **human-aware** configs ($\alpha > 0$) outperform the **purely reactive baseline** ($\alpha = 0$)
- Human interferences drop
- Higher over-weights avoidance → slight path inefficiency
- $\alpha = \mathbf{0.3}$ gives best trade-off between efficiency and safety

FUTURE DIRECTIONS

- Evaluate our approach on **additional environments and datasets**
- Compare with **other crowd prediction models**
- Scale to **larger agent fleets** and task sets
- Integrate **online model updates** to handle distributional shift

{benedetta.flammini; francesco.amigoni}@polimi.it; leodamato@cnr.it