An Autonomous Navigation Framework for Item Retrieval in Unknown Environments

Alessio Mezzina[1,2][0009−0003−3130−121X], Carolina Crespi[1][0000−0003−1050−2453], Vincenzo Cutello[1][0000−0002−7521−3516], and Mario Pavone[1][0000−0003−3421−3293]

[1] Department of Mathematics and Computer Science, University of Catania
Viale Andrea Doria 6, 95125 Catania, Italy
alessio.mezzina@phd.unict.it, carolina.crespi@unict.it, cutello@unict.it, mpavone@dmi.unict.it

[2] Koexai S.R.L., Via Jose Maria Escriva 6, 95125 Catania, Italy
alessio.mezzina@koexai.com

Abstract. A multi-agents framework is proposed for exploration and target retrieval in grid-based mazes under partial observability, proximity-limited communication, and finite battery constraints. The framework integrates line-of-sight sensing, local map synchronization, and a physics-inspired energy model that penalizes stop-and-go motion. We evaluate multiple decentralized navigation strategies, from baseline target-oriented and exploration-driven behaviors to greedy distance-based heuristics, information-valued frontier selection, and an energy-aware, direction-sensitive shortest-path planner, considering both homogeneous and heterogeneous team compositions. Performance is assessed in terms of items retrieval and delivery; return feasibility; mission duration; and energy efficiency. The outcomes highlight a trade-off between delivery efficiency and agent survivability: strategies that prioritize rapid target acquisition tend to expose agents to higher energy consumption risk, whereas energy-aware approaches improve return feasibility while preserving competitive delivery performance. Heterogeneous teams further provide a robust compromise between task completion and agent safety.

Keywords: Autonomous Agents · Multi-Agent Systems · Autonomous Navigation · Unknown Environment Exploration · Partial Observability · Energy-Aware Navigation · Path Planning · Decentralized Coordination

1 Introduction

Exploring uncharted environments with multiple agents is a key challenge in autonomous navigation and robotics, with applications such as search and rescue and cave exploration in planetary missions [2]. Deploying a multi-robot system for these tasks, ranging from cooperative swarms to highly intelligent automata, provides advantages in mission speed, redundancy, and fault tolerance [26]. However, maximizing these benefits requires effective coordination architectures [32]. What if new constraints, such as limited communication range, arise? In these various scenarios, groups of autonomous agents should work individually, bringing out a collective behavior that could maximize the chances of success of the rescue mission, intended as item recovery.

At first glance, establishing a strategy might seem straightforward; however, the proposed setting arises from the intersection of several challenging subproblems: (*i*) agents can only observe their immediate surroundings; (*ii*) energy constraints can severely limit the operational range, forcing a trade-off between continued exploration and a safe return to the dropzone; and (*iii*) without a centralized "orchestrator", coordination is required to avoid redundant coverage and physical collisions [1, 19]. Conventional approaches to multi-agent exploration often assume one or more of the following: full observability; unlimited power; centralized coordination; and/or no physics-based model–assumptions that rarely hold in real scenarios.

Prior work has addressed subsets of these limitations. Seminal frontier-based exploration methods target the boundaries between known and unknown regions [31]. Other approaches focus on coordinated multi-robot exploration and distributed mapping, leveraging shared information and task allocation to reduce redundancy and improve coverage [2, 28, 16, 13]. Market-based methods further support distributed task allocation and goal assignment, providing scalable coordination mechanisms for heterogeneous teams [12, 14, 32]. Energy-aware planning has also been investigated to optimize motion and exploration under resource constraints [21, 17, 29]. Early studies addressed energy-efficient exploration decisions and information gathering [21], while subsequent work incorporated energy- or time-optimal trajectory planning, speed control, and stopping criteria to preserve return-to-base feasibility [18, 29, 15, 17]. Related studies have explored multi-agent navigation in unknown maze-like environments from complementary perspectives, including the role of memory and visibility in navigation strategies [4], how cooperative and competitive behaviors shape evacuation processes in path-finding scenarios [7, 9, 6, 5, 8], and how social and objective trust guides exploration and decision making [3].

To the best of our knowledge, a unified framework that simultaneously addresses partial observability, exclusively local communication, physical constraints, and energy-aware decision-making has yet to be proposed. In light of this, in this research paper we propose a model that similarly emphasizes return feasibility and explicitly models inertial surge (acceleration from rest) in the simulator. Specifically, we propose a unified simulation framework for multi-agent item recovery in unknown grid mazes and evaluate a spectrum of decentralized decision strategies under the same constraints, similar to benchmarks designed to assess the efficiency of stochastic search processes in [11]. Beyond a classic approach, we introduce a total of 5 different strategies and study whether heterogeneous team compositions could yield more robust mission outcomes.

2 Methodology

We considered a classic maze as the environment for the proposed framework, modeled as an undirected graph $G = (V, E)$ with 4-neighborhoods connectivity; a dropzone (start) location $d \in V$; and a set of targets $I = \{I_1, \ldots, I_m\}$. Environments can be generated as *perfect mazes* ($\rho = 0$) or *open mazes* with additional paths ($\rho > 0$), as shown in Figure 1.

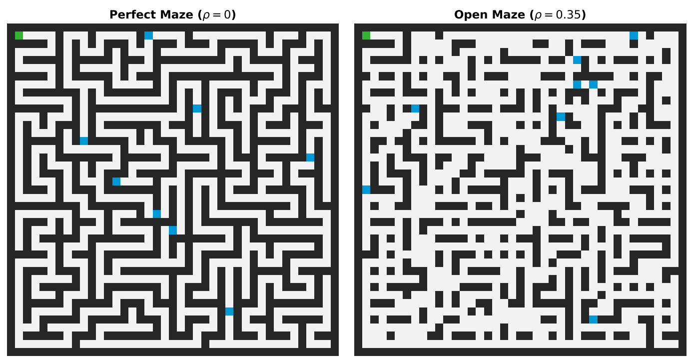


Fig. 1: Maze environments: Perfect maze ($\rho = 0$, left) and Open maze ($\rho = 0.35$, right). Green: Start/Dropzone; Blue: Targets; Red: Dropzone marker.

Each agent a_i, for $i \in \{1, \ldots, n\}$, is characterized by: (i) position $p_i(t) \in V$; (ii) battery level $B_i(t) \in [0, B_{\max}]$ and a prudence factor λ (safety margin of energy for a safe back to the dropzone); (iii) carrying load $c_i(t)$; (iv) visibility sensor range r_s; (v) communication range r_c; and (vi) physical parameters θ_i, such as for instance mass; and inertia coefficient. Additional flags track deployed state $\delta_i(t)$; exited state $\epsilon_i(t)$; and must-return status $\mu_i(t)$. An agent in this system is therefore encoded as a set of *dynamic state variables*, which change over time t, and *static parameters* that define its physical and operational limitations. The dynamic variables track where the agent is; its energy level; and its current task status (e.g., carrying an object or returning). The static parameters, instead, define its capabilities, such as how far it can sense and communicate, as well as its physical properties. A complete breakdown of this agent encoding is provided in Table 1.

The goal is to maximize the number of targets retrieval and delivered to the dropzone d. A target is considered *delivered* when an agent carrying it reaches d; while it is counted as safely *exited* when it is at d and not carrying a target and it declares to exit the environment. Note that if an agent retrieves an item but it runs out its energy before reaching the dropzone, then the item is considered lost, along of course with the agent. A few technical assumptions are adopted, worth of being stated, to keep the model reproducible and to isolate the effects of partial observability, local communication, and energy-aware navigation: (i) discrete time with synchronous actions, deterministic 4-neighborhood motion and collision handling: for the agent is possible to perform a swap to avoid

Table 1: Agent a_i Encoding Summary

Description	**Symbol**	**Domain**	**Type**
Position	$p_i(t)$	V	Dynamic
Battery Level	$B_i(t)$	$[0, B_{\max}]$	Dynamic
Prudence Factor	λ	Constant	Static
Carrying Load	$c_i(t)$	$\mathbb{N}_{\geq 0}$	Dynamic
Sensor Range	r_s	Constant	Static
Communication Range	r_c	Constant	Static
Physical Parameters	θ_i	*e.g., mass, inertia*	Static
Deployed State	$\delta_i(t)$	Boolean flag	Dynamic
Exited State	$\epsilon_i(t)$	Boolean flag	Dynamic
Must-Return Status	$\mu_i(t)$	Boolean flag	Dynamic

remain blocked in a *hallway*; (*ii*) static environments and stationary targets, no sensing or actuation noise; (*iii*) although it is possible to make configurations of agents with heterogeneous components, for the sake of brevity we only make combinations of homogeneous components.

To prevent congestion at the dropzone, agents deploy sequentially rather than simultaneously. The deployment mechanism enforces this behavior by allowing agent a_i to launch only when no previously indexed agent is still occupying the dropzone. Formally, agent a_i is allowed to deploy at time t if no agent a_j, with $j < i$, remains at the dropzone without having exited the mission area, as expressed in Eq. 1:

$$\text{CanLaunch}(a_i, t) = \neg \exists a_j : (j < i) \wedge (p_j(t) = d) \wedge \delta_j(t) \wedge \neg \epsilon_j(t). \tag{1}$$

In the simulation framework developed, time is modeled as a discrete sequence of atomic agent actions. The simulation variable *ticks* represents the cumulative count of individual actuation opportunities across the entire team, rather than synchronized global rounds. The system employs a sequential turn-based execution model. Within a single simulation cycle (round), the system iterates through all n active agents $(a_1, \ldots, a_n)$. Consequently, for a team of n agents, one global synchronization round corresponds to an increment of n ticks, as expressed in Equation (2):

$$T_{\text{round}} \approx \sum_{i=1}^{n} \text{action}(a_i). \tag{2}$$

Each agent maintains two separate knowledge maps: *personal knowledge* K_i^{p} (i.e. cells observed directly) and *shared knowledge* K_i^{s} (i.e. union of knowledge received through synchronization). The shared map is used for pathfinding, while the personal map determines mission completion. This separation aligns with common practice in distributed mapping in intermittent or local communication [13, 28, 25]. Cell states are: -1 (unknown), 0 (free), or 1 (wall). At each time step t, the agent a_i observes a subset of cells within its sensor range using line-of-sight visibility. We denote by $Vis_i(t)$ the set of cells that are within Manhattan distance r_s, from the current position $p_i(t)$ and not occluded by obstacles. Formally, the visibility set is defined as: $Vis_i(t) = \{v \in V : \|v - p_i(t)\|_1 \leq r_s \wedge \text{LoS}(p_i(t), v)\}$, where LoS (Line of Sight) indicates that there is no wall between the agent position and the vertex.

When two agents are within mutual communication range ($\|p_i(t)-p_j(t)\|_1 \leq r_c^{(i)} + r_c^{(j)}$),they perform bidirectional knowledge synchronization, exchanging map information and merging known target sets. Each agent receives an *independent copy* of merged data, preventing instantaneous global propagation. Note that target positions are updated only through direct observation or explicit communication, and they are not instantaneously shared. Figure 2 illustrates how four agents exploring different regions build individual knowledge maps.

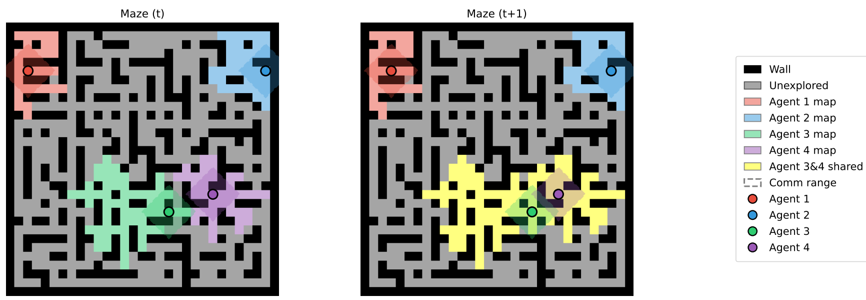


Fig. 2: Partial observability and map merging. Left: Individual local maps at time t. Right: Merged shared knowledge (yellow) after agents 3 and 4 communicate at $t+1$.

The energy consumption for a single move follows a physics-inspired model incorporating friction and inertia, in the spirit of energy-aware motion planning and speed control [20, 30, 22]. We define with E_{base} the base energy cost per step; μ friction coefficient; $m = m_{a_i} + c_i(t) \times m_{\text{target}}$ the agent's mass (robot plus target if carrying); $v = 1$ velocity magnitude; $v_{\text{prev}} \in \{0, 1\}$ the agent's velocity from the previous step (0 if stationary); and α the inertia coefficient for the acceleration penalty. The surge term $(\alpha \times m \times (v - v_{\text{prev}})^2)$ applies only when accelerating from rest, penalizing stop-and-go motion; therefore, the energy cost E_{step} for a movement action is defined as:

$$E_{\text{step}} = E_{\text{base}} + (\mu \times m \times v) + \alpha\, m \,(\max(0, v - v_{\text{prev}}))^2 \tag{3}$$

For a path estimation of k steps, the first step cost depends on the current previous speed v_{prev} (so the surge term may be zero if already moving), and subsequent steps use the steady cost. This estimation is formalized as follows:

$$E_{\text{path}}(k \mid v_{\text{prev}}) = E_{\text{step}}(v_{\text{prev}}) + (k-1) \cdot E_{\text{step}}^{(\text{steady})}. \tag{4}$$

Agents employ a return policy, motivated by prior work on energy-aware exploration and energy-based termination criteria [24, 27]. A *hard threshold* sets the must return flag $\mu_i(t)$ when battery drops below 50 units. *Prudent return* uses optimistic path estimates with $\lambda \geq 1$ that we call *prudence factor*, a value that establish how is important for an agent to exit the environment; once set, must-return is permanent. Return policy is formalized as:

$$E_{\text{reserve}} = E_{\text{path}}(d_{\text{home}}^{\text{opt}}) + \lambda \cdot (E_{\text{step}}^{(\text{surge})} + E_{\text{step}}^{(\text{steady})}) \tag{5}$$

Unless otherwise stated, an agent ends the execution when at the dropzone with no cargo with battery critically low (checking $\mu_i(t)$ flag); or with personal map fully explored with no known targets remaining to retrieval. Exit decisions use personal knowledge only. If the energy battery is insufficient for a move, the agent remains stuck, and the agent is considered dead.

2.1 Navigation Strategy

Five increasingly sophisticated decentralized navigation strategies are proposed. All strategies share the same energy management, return logic and collision avoidance mechanisms, and differ only in how they select exploration goals and prioritize target recovery, inspired by classical multi-robot exploration and foraging formulations [31, 2, 23]. All strategies follow a hierarchical decision structure, whose pseudocode is reported in Algorithm 1 (hereafter referred to as CBS). Although the framework supports multi-item carrying capacities ($c_i(t) > 1$), in the evaluated strategies agents adopt a secure return policy, delivering items immediately upon collection to minimize the risk of cargo loss due to battery depletion.

Algorithm 1 Common Decision Framework

function ChooseMove(agent a)
    if $B(t) < B_{\text{critical}}$ **then**
        $\mu \leftarrow 1$ ▷ Set must-return flag
    end if
    $d_{\text{home}}^{\text{opt}} \leftarrow$ ShortestPath($p(t)$, d, assumeUnknownFree=true)
    if $\mu = 1$ *or* ShouldReturnPrudently() **then**
        return PathToDropzone
    end if
    if $c(t) > 0$ **then** ▷ Carrying target
        return PathToDropzone
    end if
    return StrategySpecificDecision ▷ Varies by strategy
end function

Distance-based strategies use Breadth-First Search (BFS) on the agent's current knowledge grid, while the *Energy-Optimal* strategy replaces BFS with an energy-aware Dijkstra planner that minimizes estimated energy rather than hop count. The planner's step cost combines the base cost, the friction proportional to mass, and an inertial surge term. To better match the simulator, the planner tracks a proxy for previous speed in its planning state so that the surge estimates reflect stop-and-go behavior. In both cases we support two map assumptions: a *conservative* setting that restricts planning to known-free cells; and an *optimistic* setting that treats unknown cells as traversable and replans as new walls are observed. In our experiments, agents use optimistic planning for exploration and target pursuit but use conservative planning for return-to-dropzone execution. Additionally, several strategies estimate the return cost using an optimistic home path for the prudence check so that agents do not over-commit when the map is incomplete. When multiple equivalent paths exist, the strategies prefer moves that do not return to the immediately previous position.

Base Strategy. The Base strategy (Alg. 2) implements target-prioritized exploration with greedy nearest-first selection, similar in spirit to simple foraging and pickup-and-delivery heuristics. Target selection uses BFS to find the

Algorithm 2 Base Strategy

```
function BaseDecision(agent a)
    if T^known ≠ ∅ then
        return PathToNearestTarget(T^known)
    end if
    u* ← NearestUnknownCell
    if u* ≠ null then
        return PathTo(u*)
    end if
    if FullyExplored and |T^known| = 0 then
        return PathToDropzone                          ▷ Mission complete
    end if
    return null                                        ▷ Idle at dropzone
end function
```

nearest reachable known target: $t^* = \arg\min_{t \in \mathcal{T}^{\text{known}}} d_{\text{BFS}}(p(t), t)$. For exploration, instead, the strategy seeks the nearest cell with an unknown state: $u^* = \arg\min_{v:K(v)=-1} d_{\text{BFS}}(p(t), v)$. By combining greedy target retrieval with nearest-first exploration, the strategy maintains a complexity of $O(|V| + |E|)$ per decision, although this approach may inadvertently lead to agent clustering near the dropzone.

Explorer Strategy. The Explorer strategy inverts the priority hierarchy, favoring complete map coverage over immediate target collection (Alg. 3) . The

Algorithm 3 Explorer Strategy

```
function ExplorerDecision(agent a)
    u* ← NearestUnknownCell
    if u* ≠ null then
        return PathTo(u*)                              ▷ Explore first
    end if
    if T^known ≠ ∅ then
        return PathToNearestTarget(T^known)
    end if
    return PathToDropzone
end function
```

exploration priority is given by:

$$\pi_{\text{explorer}} = \begin{cases} \text{explore}, & \text{if } \exists v : K(v) = -1; \\ \text{collect}, & \text{if } |\mathcal{T}^{\text{known}}| > 0; \\ \text{return}, & \text{otherwise.} \end{cases} \tag{6}$$

By prioritizing maximum map coverage over immediate collection, this strategy ensures complete target discovery before retrieval commences (limiting retrieval to targets encountered directly on the path). Although this could yield higher exploration efficiency suitable for environments with sparse or hidden targets, it may trade off retrieval speed for completeness. While the CBS can evaluate the prudence return criterion using an optimistic home-path estimate (treating unknown cells as traversable), the implementation of the Explorer strategy computes the prudence check using a conservative home path restricted to known-free cells.

Greedy Strategy. The Greedy strategy (Alg. 4) extends Base with distance-aware exploration tie-breaking to improve natural agent dispersion, leveraging ideas related to decentralized dispersion and cooperative foraging. Among

Algorithm 4 Greedy Exploration

function GreedyExplore(agent a)
    $d_{\min} \leftarrow \infty$
    $U_{\text{nearest}} \leftarrow \emptyset$
    for all v in BFS order from $p(t)$ **do**
        if $K(v) = -1$ **then**
            if $d_{\text{BFS}}(p(t), v) < d_{\min}$ **then**
                $d_{\min} \leftarrow d_{\text{BFS}}(p(t), v)$
                $U_{\text{nearest}} \leftarrow \{v\}$
            else if $d_{\text{BFS}}(p(t), v) = d_{\min}$ **then**
                $U_{\text{nearest}} \leftarrow U_{\text{nearest}} \cup \{v\}$
            else
                break ▷ Past minimum distance
            end if
        end if
    end for
    $k \leftarrow \max\left(1, \left\lfloor \frac{|U_{\text{nearest}}|}{3} \right\rfloor\right)$
    $U_{\text{top}} \leftarrow \text{TopK}_{\|v-d\|_1}(U_{\text{nearest}}, k)$ ▷ Furthest from dropzone
    return RandomChoice(U_{top}) ▷ Random tie-break among top candidates
end function

equally-close unknown cells, Greedy ranks candidates by Manhattan distance from the dropzone and applies randomized tie-breaking: $u^* \sim \text{Unif}(U_{\text{top}})$

Frontier Strategy. The Frontier strategy (Alg. 5) employs information-theoretic exploration, prioritizing boundary cells that maximize the expected information gain [31]. A cell v is a *frontier* if it is known-free and has at least one unknown neighbor:

$$\text{Frontier}(v) = \mathbb{1}[K(v) = 0] \times \mathbb{1}\left[\exists u \in N_4(v) : K(u) = -1\right]. \tag{7}$$

The value of a frontier cell is the count of unknown neighbors, i.e.:

$$\text{Value}(v) = \sum_{u \in N_4(v)} \mathbb{1}[K(u) = -1] \in \{1, 2, 3, 4\}. \tag{8}$$

To keep computation bounded, the frontier search considers only candidates within $D_{\max}$ BFS steps (set to 20 in our implementation):

$$f^* = \max_{\substack{v \in \text{Frontier} \\ d(p,v) \leq D_{\max}}} \Big(\text{Value}(v), -d(p, v)\Big), \tag{9}$$

Algorithm 5 Frontier Strategy

function FindBestFrontier(agent a)
  $D_{\max} \leftarrow 20$ ▷ Search horizon (implementation constant)
  $\mathcal{F} \leftarrow \emptyset$
  for all v in BFS order from $p(t)$ while $d_{\mathrm{BFS}} \leq D_{\max}$ **do**
    if Frontier(v) **then**
      $\mathcal{F} \leftarrow \mathcal{F} \cup \{(v, \mathrm{Value}(v), d_{\mathrm{BFS}}(p(t), v))\}$
    end if
  end for
  Sort $\mathcal{F}$ by (Value descending, distance ascending)
  return first element of $\mathcal{F}$
end function

using lexicographic ordering (value first, then distance). The selected frontier balances information gain against travel cost.

Energy-Optimal Strategy. The Energy-Optimal strategy uses energy-aware pathfinding with Dijkstra's algorithm, computing paths that minimize total energy expenditure rather than hop count (Alg. 6). For exploration, the direction-

Algorithm 6 Energy-Optimal Strategy (hybrid energy-aware Dijkstra)

function EnergyOptimalDecision(agent a)
  if $\mathcal{T}^{\mathrm{known}} \neq \emptyset$ **then**
    $P \leftarrow$ EnergyAwareDijkstra($p(t)$, $\mathcal{T}^{\mathrm{known}}$, OptimisticPlanning)
    if $P \neq$ null **then**
      return first step of P
    end if
  end if
  $P \leftarrow$ EnergyAwareDijkstra($p(t)$, $\{v : K(v) = -1\}$, OptimisticPlanning)
  if $P \neq$ null **then**
    return first step of P
  end if
  return null
end function

function EnergyAwareDijkstra(start, goal set G, model, OptimisticPlanning)
  Run Dijkstra on an augmented state and return a minimum-cost path to the first reached goal in G
end function

mode planner selects the reachable unknown cell minimizing planning cost:

$$u^* = \underset{\substack{v:K(v)=-1 \\ \mathrm{reachable}(v)}}{\arg\min} J_{\mathrm{dir}}^{\mathrm{Dijkstra}}(p(t), v) \tag{10}$$

Key differences between these strategies are summarized in Table 2.

Table 2: Strategy characteristics. Smooth*: implicit dispersion via energy-efficient paths.

Property	*Base*	*Explorer*	*Greedy*	*Frontier*	*Energy-Opt*
Target Priority	High	Low	High	High	High
Exploration Priority	Low	High	Medium	Medium	Medium
Agent Dispersion	None	None	Distance	Value	Smooth*
Pathfinding	BFS	BFS	BFS	BFS	Dijkstra
Complexity	$O(\|V\| + \|E\|)$	$O(\|V\| + \|E\|)$	$O(\|V\| + \|E\|)$	$O(\|V\| + \|E\|)$	$O((\|V\| + \|E\|) \log \|V\|)$

3 Experimental Setup

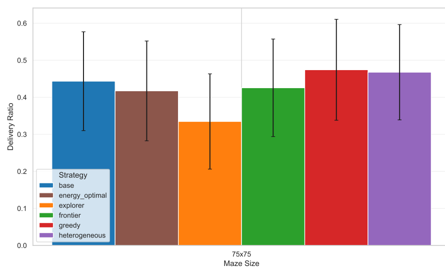
We Evaluate our navigation framework and strategies through Extensive simulation experiments have been carried out in order to assess performance of the proposed navigatin framework in a challenging environment and across team compositions. The proposed experiments, and then the experimental methodology, including the simulation framework, parameter configurations, and evaluation metrics, have been specifically designed for answering the following research questions: (1) how do different decentralized strategies trade off delivery and return feasibility under tight energy budgets? (2) how do strategies compare in efficiency (ticks/target, energy/target) and exploration coverage? (3) does heterogeneous team composition improve robustness compared to homogeneous teams?

All experiments have been conducted in a grid-based environment 75×75 ($5,625$ cells) with an obstacle density of $\rho = 0.35$, and 12 targets. Each simulation involves a team of 10 agents, all initialized with a maximum battery capacity of $B_{\max} = 500$. The agents are equipped with a sensor range of $r_s = 3$ and a communication range of $r_c = 2$. The maximum simulation length is set to $10,000$ steps. The robot mass is $m_{\text{robot}} = 1.0$, while each target has a mass $m_{\text{target}} = 0.5$. Movement energy consumption is regulated by a base energy cost $E_{\text{base}} = 1.0$; a friction coefficient $\mu = 0.5$; an inertia coefficient $\alpha = 1.0$; and a prudence factor $\lambda = 1.5$, which modulates conservative energy-aware behavior.

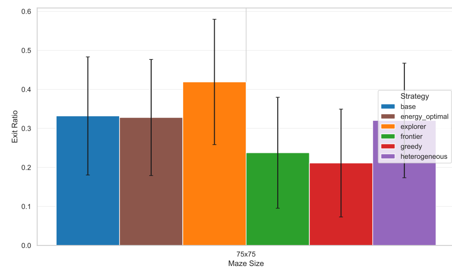
Six different configurations have been evaluated: five homogeneous teams (10 agents each using Base, Explorer, Greedy, Frontier, or Energy-Optimal) and one heterogeneous team (2 agents per strategy). The heterogeneous configuration tests whether diversity could improve robustness. For each of the 6 configurations, $2,000$ independent trials are performed, resulting in a total of $12,000$ runs. Strategy performance is evaluated using several metrics: (*i*) *delivery ratio*, defined as the fraction of targets successfully delivered; (*ii*) *exit ratio*, measuring the proportion of agents that successfully leave the environment; (*iii*) *average number of ticks* required per delivered target; (*iv*) *average energy consumption* per delivered target; and (*v*) *explored ratio*, quantifying the fraction of open cells visited during exploration. The reported results are aggregated over $2,000$ tests per configuration and presented as mean values with standard deviation to capture variability across runs.

4 Results

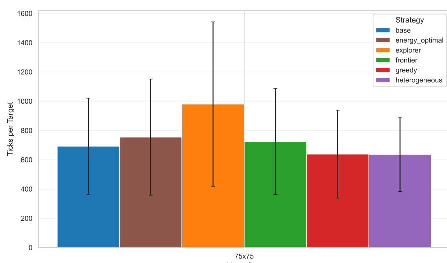
In figure 3 we display and summarize the results performance with respect to the delivery, the return feasibility and efficiency. Inspecting the figure, it is possible to assert how greedy achieves the best delivery ratio (0.474) and the highest explored ratio (0.553); but, at the same time, shows the lowest exit ratio (0.211), indicating frequent stranding. Heterogeneous teams offer the best team compromise, with near-greedy delivery (0.468); an improved exit ratio (with respect to the greedy strategy) (0.320); and a similar explored ratio (0.534), while minimizing ticks and energy metrics.



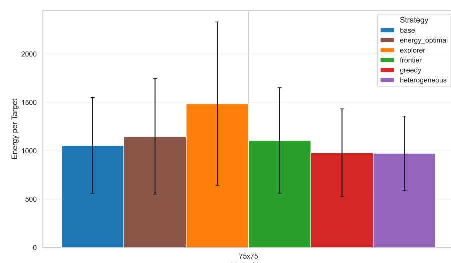
(a) Delivery ratio (↑)



(b) Exit ratio (↑)



(c) Ticks/target (↓)



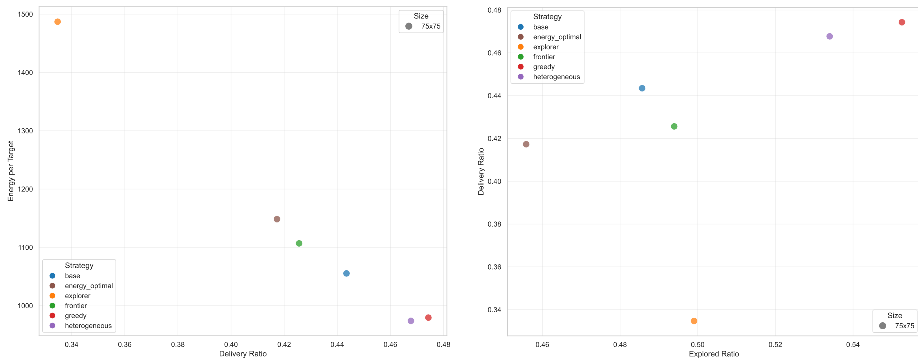
(d) Energy/target (↓)

Fig. 3: Performance summary (mean $\pm$ std over $2,000$ trials).

Figure 4 visualizes per-run relationships. Because energy/ticks are normalized by the number of delivered targets, higher delivery is typically associated with lower energy/target and ticks/target. In contrast, explored ratio and delivery exhibit only a weak association, suggesting that maximizing coverage alone is insufficient under tight energy budgets.

To aggregate heterogeneous performance indicators into a single composite score, we first map each metric to a direction-aligned normalized scale $z \in [0, 1]$, where larger is better. Let $x_m(s)$ denote the raw value of the metric m for the strategy s. For benefit metrics to be maximized (Delivery, Exit, Explored), we use min–max scaling:

$$z_m(s) = \frac{x_m(s) - \min_{s'} x_m(s')}{\max_{s'} x_m(s') - \min_{s'} x_m(s')}. \tag{11}$$



(a) Delivery vs. energy/target (b) Explored ratio vs. delivery

Fig. 4: Run-level trade-offs.

For cost metrics to be minimized (Ticks/Target, Energy/Target), we invert the scale:

$$z_m(s) = \frac{\max_{s'} x_m(s') - x_m(s)}{\max_{s'} x_m(s') - \min_{s'} x_m(s')}. \tag{12}$$

This ensures $z_m(s) = 1$ for the best-performing strategy and $z_m(s) = 0$ for the worst-performing strategy on metric m within the compared set.

Composite Utility Score (CUS). We then compute a single composite utility score as a weighted average of the normalized metrics:

$$\mathrm{CUS}(s) = \sum_{m\in\mathcal{M}} w_m z_m(s), \qquad \sum_{m\in\mathcal{M}} w_m = 1, \tag{13}$$

using uniform weights: $w_m = 1/|\mathcal{M}|$ over $\mathcal{M} = \{$Delivery, Exit, Explored, Ticks/Target, Energy/Target$\}$. CUS provides a compact summary of strategy performance by jointly accounting for delivery effectiveness; return feasibility; and efficiency-related costs. By aggregating normalized metrics with uniform weights, CUS avoids privileging a single performance dimension, highlighting instead balanced behaviors across different objectives. Table 3 reports the normalized metrics and the resulting CUS for each strategy. Although the Greedy strategy achieves the highest delivery, its very low exit score penalizes its overall utility. In contrast, the heterogeneous configuration attains the highest CUS by leading on efficiency (ticks/target and energy/target) and staying close to the best on delivery and explored ratio, despite a non-maximal delivery, exit, and explored ratio. The base configuration achieves the second highest composite score without dominating any single metric, indicating a well-balanced behavior that remains consistently competitive across delivery, survivability, and efficiency objectives.

Table 3: Overall strategy comparison (2,000 trials). Scores $z \in [0, 1]$ are normalized (1=best); CUS is the uniform composite score.

Strategy	*Delivery*↑	*Exit*↑	*Explored Ratio*↑	*Ticks/Target*↓	*Energy/Target*↓	$z_D \uparrow$	$z_X \uparrow$	$z_E \uparrow$	$z_T \uparrow$	$z_{En} \uparrow$	CUS↑
base	0.444	0.332	0.486	692.3	1055.3	0.779	0.581	0.309	0.838	0.841	0.670
energy_optimal	0.417	0.328	0.456	754.6	1148.3	0.592	0.562	0.000	0.657	0.660	0.494
explorer	0.335	**0.419**	0.499	980.3	1487.0	0.000	**1.000**	0.447	0.000	0.000	0.289
frontier	0.426	0.238	0.494	724.5	1106.8	0.651	0.127	0.394	0.745	0.741	0.532
greedy	**0.474**	0.211	**0.553**	638.5	979.5	**1.000**	0.000	**1.000**	0.995	0.989	0.797
heterogeneous	0.468	0.320	0.534	**636.8**	**973.9**	0.953	0.525	0.808	**1.000**	**1.000**	**0.857**

5 Conclusion

A decentralized multi-agent framework is proposed for item recovery in unknown grid environments under partial observability, proximity-limited communication, and finite battery budgets. The framework implements line of sight sensing, local map synchronization, and a physics-inspired energy model that penalizes stop-and-go motion and supports multiple navigation strategies through a shared decision architecture. Through extensive simulation on homogeneous and heterogeneous team configurations, the results highlight a trade-off between delivery efficiency and survivability: delivery-focused strategies increase deliveries but strand agents more often, whereas more conservative exploration/return behaviors improve exit feasibility at the cost of delivery. In addition, heterogeneous teams encourage complementary behaviors and provide the best overall compromise between task completion and agent safety.

Crucially, the framework is designed to be highly modular, allowing systematic modification of both physical parameters (e.g., mass, inertia, energy budgets) and logical components (e.g., sensing, communication, and decision policies). This design choice enables controlled exploration of alternative configurations and facilitates the integration of adaptive strategies and dynamic environments, such as the one presented in [10], within the same experimental setup. Future work aims to exploit this modularity to incorporate dynamic environments, adaptive decision-making, and more realistic sensing and communication models.

References

1. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of markov decision processes. Mathematics of Operations Research **27**(4), 819–840 (2002). https://doi.org/10.1287/moor.27.4.819.297
2. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration. IEEE Transactions on Robotics **21**(3), 376–386 (2005). https://doi.org/10.1109/TRO.2004.839232
3. Cangelosi, A., Crespi, C., Pavone, M., Romeo, M.: Modeling trust dynamics on a multi-agent system. In: International Conference on Optimization and Learning. pp. 215–226. Springer (2025)
4. Crespi, C., Cutello, V., Pavone, M., Zito, F.: An agent framework to explore pathfinding strategies in maze navigation problem. Le Matematiche **79**(2), 555–583 (2024)

5. Crespi, C., Fargetta, G., Pavone, M., Scollo, R.A.: An agent-based model for crowd simulation. In: Italian Workshop on Artificial Life and Evolutionary Computation. pp. 15–26. Springer (2022)
6. Crespi, C., Fargetta, G., Pavone, M., Scollo, R.A.: An agent-based model to investigate different behaviours in a crowd simulation. In: International Conference on Bioinspired Optimization Methods and Their Applications. pp. 1–14. Springer (2022)
7. Crespi, C., Fargetta, G., Pavone, M., Scollo, R.A., Scrimali, L.: A game theory approach for crowd evacuation modelling. In: International Conference on Bioinspired Methods and Their Applications. pp. 228–239. Springer (2020)
8. Crespi, C., Pavone, M.: Does a group's size affect the behavior of a crowd? an analysis based on an agent model. In: Conference of the European Social Simulation Association. pp. 411–422. Springer (2023)
9. Crespi, C., Scollo, R.A., Fargetta, G., Pavone, M.: A sensitivity analysis of parameters in an agent-based model for crowd simulations. Applied Soft Computing **146**, 110684 (2023)
10. Cutello, V., Mezzina, A., Pavone, M., Zito, F.: A real-time adaptive tabu search for handling zoom in/out in map labeling problem. In: Festa, P., Ferone, D., Pastore, T., Pisacane, O. (eds.) Learning and Intelligent Optimization. pp. 108–122. Springer Nature Switzerland, Cham (2025)
11. Cutello, V., Mezzina, A., Pavone, M., Zito, F.: A weighted binary string benchmark to assess the efficiency of stochastic search processes. In: Tan, Y., Shi, Y. (eds.) Advances in Swarm Intelligence. pp. 116–127. Springer Nature Singapore, Singapore (2026)
12. Dias, M., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: A survey and analysis. Proceedings of the IEEE **94**(7), 1257–1270 (2006). https://doi.org/10.1109/JPROC.2006.876939
13. Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., Stewart, B.: Distributed multirobot exploration and mapping. Proceedings of the IEEE **94**(7), 1325–1339 (2006). https://doi.org/10.1109/JPROC.2006.876927
14. Gerkey, B.P., Matarić, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research **23**(9), 939–954 (2004). https://doi.org/10.1177/0278364904045564
15. Hesami, S., De Cauwer, C., Rombaut, E., Vanhaverbeke, L., Coosemans, T.: Energy-optimal speed control for autonomous electric vehicles up- and downstream of a signalized intersection. World Electric Vehicle Journal **14**(2) (2023). https://doi.org/10.3390/wevj14020055
16. Howard, A.: Multi-robot simultaneous localization and mapping using particle filters. The International Journal of Robotics Research **25**(12), 1243–1256 (2006). https://doi.org/10.1177/0278364906072250
17. Kabir, R.H., Lee, K.: Efficient multi-robot exploration with energy constraint based on optimal transport theory (2020)
18. Liu, X., Qiu, C., Zeng, Q., Li, A., Xie, N.: Time-energy optimal trajectory planning for collaborative welding robot with multiple manipulators. Procedia Manufacturing **43**, 527–534 (2020). https://doi.org/https://doi.org/10.1016/j.promfg.2020.02.174, sustainable Manufacturing - Hand in Hand to Sustainability on Globe: Proceedings of the 17th Global Conference on Sustainable Manufacturing
19. Matignon, L., Jeanpierre, L., Mouaddib, A.I.: Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes.

Proceedings of the AAAI Conference on Artificial Intelligence **26**(1), 2017–2023 (Sep 2021). https://doi.org/10.1609/aaai.v26i1.8380
20. Mei, Y., Lu, Y.H., Hu, Y., Lee, C.: Energy-efficient motion planning for mobile robots. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004. vol. 5, pp. 4344–4349 Vol.5 (2004). https://doi.org/10.1109/ROBOT.2004.1302401
21. Mei, Y., Lu, Y.H., Lee, C., Hu, Y.: Energy-efficient mobile robot exploration. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. pp. 505–511 (2006). https://doi.org/10.1109/ROBOT.2006.1641761
22. Morales, J., Martinez, J.L., Mandow, A., Garcia-Cerezo, A.J., Pedraza, S.: Power consumption modeling of skid-steer tracked mobile robots on rigid terrain. IEEE Transactions on Robotics **25**(5), 1098–1108 (2009). https://doi.org/10.1109/TRO.2009.2026499
23. Parker, L.: Alliance: an architecture for fault tolerant multirobot cooperation. IEEE Transactions on Robotics and Automation **14**(2), 220–240 (1998). https://doi.org/10.1109/70.681242
24. Rappaport, M.: Energy-aware mobile robot exploration with adaptive decision thresholds. In: Proceedings of ISR 2016: 47st International Symposium on Robotics. pp. 1–8 (2016)
25. Saeedi, S., Trentini, M., Seto, M., Li, H.: Multiple-robot simultaneous localization and mapping: A review. Journal of Field Robotics **33**(1), 3–46 (2016). https://doi.org/https://doi.org/10.1002/rob.21620
26. Schultz, A.C., Parker, L.E., Schneider, F.: Multi-robot systems: from swarms to intelligent automata. Springer (2002)
27. Shnaps, I., Rimon, E.: Online coverage of planar environments by a battery powered autonomous mobile robot. IEEE Transactions on Automation Science and Engineering **13**(2), 425–436 (2016). https://doi.org/10.1109/TASE.2016.2515165
28. Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., Younes, H.: Coordination for multi-robot exploration and mapping. pp. 852–858 (01 2000)
29. Tokekar, P., Karnad, N., Isler, V.: Energy-optimal velocity profiles for car-like robots. In: 2011 IEEE International Conference on Robotics and Automation. pp. 1457–1462 (2011). https://doi.org/10.1109/ICRA.2011.5980374
30. Tokekar, P., Karnad, N., Isler, V.: Energy-optimal trajectory planning for car-like robots. Autonomous Robots **37**(3), 279–300 (Oct 2014). https://doi.org/10.1007/s10514-014-9390-3
31. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'. pp. 146–151 (1997). https://doi.org/10.1109/CIRA.1997.613851
32. Yan, Z., Jouandeau, N., Cherif, A.A.: A survey and analysis of multi-robot coordination. International Journal of Advanced Robotic Systems **10**(12), 399 (2013). https://doi.org/10.5772/57313