PushAround: Collaborative Path Clearing via Physics-Informed Hybrid Search

***Abstract*— In unstructured environments, the passage of large external vehicles is frequently blocked by movable obstacles, which motivates the deployment of mobile robot teams to proactively clear traversable corridors. Existing approaches to Navigation Among Movable Obstacles (NAMO) primarily plan sequences of obstacle manipulations but typically neglect physical feasibility, overlooking essential factors such as robot dimensions, mass, applied forces, and the coupled dynamics of pushing. This limitation often results in strategies that cannot be executed reliably in practice. A physics-informed framework for collaborative multi-robot pushing is introduced, enabling the active construction of physically feasible paths. The framework jointly searches the sequence of obstacles to be displaced, the transit motions of robots between obstacles, and the contact points and forces required for effective execution. Core components include a W-Clearance Connectivity Graph (WCCG) for reasoning about vehicle passage under width constraints, a gap-ranking strategy for prioritizing obstacle-clearing actions, and a simulation-in-the-loop search to validate push feasibility. Extensive simulations and hardware experiments demonstrate robust success, scalability, and efficiency compared to baseline NAMO methods.**

I. Introduction

In many real-world scenarios, the path of a large vehicle or transport unit is obstructed by movable obstacles such as pallets, boxes, or equipment, which motivates the use of a fleet of mobile robots to actively clear traversable corridors and enable safe passage. This capability is especially critical in unstructured or cluttered workspaces, including warehouses, disaster sites, and crowded urban environments, where traditional path planning approaches assume static obstacles and therefore fail to provide feasible solutions when direct paths are blocked []. Research on Navigation Among Movable Obstacles (NAMO) has proposed strategies for pushing, pulling, or rotating objects to create new routes [], but these methods often abstract away physical feasibility, ignoring key factors such as robot dimensions, obstacle size and mass, contact geometry, applied forces, and the coupled dynamics of pushing. As a result, the generated plans may be theoretically valid but physically unrealizable in practice. This problem is particularly challenging because a single large robot is often unable to independently clear a corridor; smaller robots are constrained by their limited size and reach, making some obstacle boundaries inaccessible; and pushing actions introduce strong physical coupling, where a single push may trigger chained motions of multiple obstacles, further complicating prediction and planning.

A. Related Work

Navigation Among Movable Obstacles (NAMO) has been studied as a core extension of motion planning in cluttered

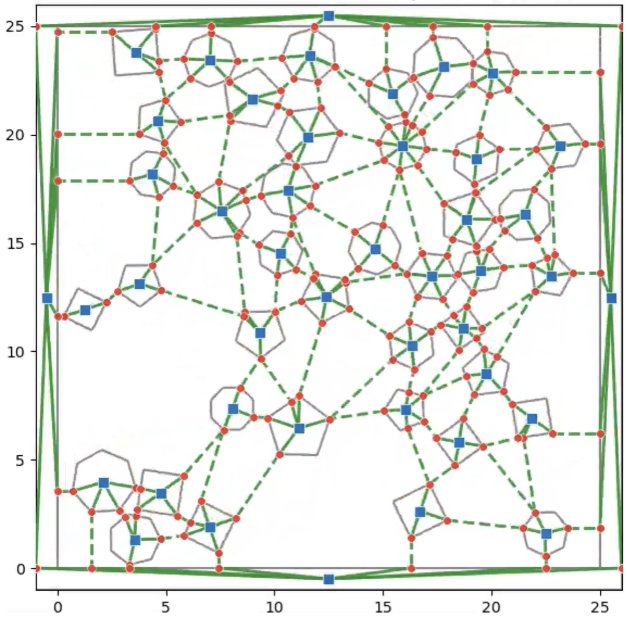

Fig. 1.

environments. Early approaches planned sequences of obstacle displacements through pushing, pulling, or rotating actions []. Heuristic search methods improved scalability by prioritizing which obstacles to move [], and graph- or sampling-based algorithms enabled planning in larger workspaces []. Multi-agent variants have also been explored []. Despite these advances, most NAMO approaches neglect physical feasibility, typically treating robots as point agents with unlimited power and ignoring dimensions, object mass, contact geometry, and force constraints. As a result, many generated plans remain physically unrealizable [].

Collaborative pushing has also been investigated in the context of multi-robot manipulation. Prior work has addressed synchronization of forces [], stable contact maintenance [], and cooperative transport of objects in structured settings []. These approaches demonstrate effective coordination but generally focus on a single object type and assume strict collision avoidance with surrounding obstacles. Such assumptions exclude the possibility of exploiting inter-object interactions, and the challenges of coupled dynamics and chain reactions remain largely unaddressed [].

Physics-informed planning has recently emerged to incorporate realistic dynamics into motion planning. Some methods employ physics engines to validate candidate manipulations [] or simulate contact dynamics during planning []. While these works highlight the benefits of embedding physics, they are mostly limited to single-object manipulation

or grasp planning. In addition, many approaches decouple abstract planning from low-level physics checks [], which reduces efficiency, and simulation-in-the-loop techniques face scalability challenges in multi-robot, cluttered scenarios []. A unified framework that couples collaborative planning with physics-based feasibility for robust path clearing has yet to be established.

B. Our Method

This work introduces a physics-informed framework for collaborative multi-robot pushing that actively constructs traversable corridors for large vehicles in cluttered environments. The approach integrates three key components: a W-Clearance Connectivity Graph (WCCG) to represent workspace connectivity under vehicle width constraints and to identify blocking frontier gaps; a gap-ranking strategy that estimates the cost of clearing each gap and prioritizes obstacles for coordinated pushing; and a simulation-in-the-loop path construction scheme that incrementally expands a configuration-space search tree and validates each pushing mode through parallel physical simulation. This combination ensures that planned actions account for robot dimensions, contact geometry, applied forces, and chained obstacle motions, enabling physically feasible execution. The framework is evaluated through extensive 2D and large-scale simulations as well as hardware experiments, demonstrating robustness under various dense obstacle configurations, and varying team sizes.

The contributions of this work are twofold: (I) it presents the first unified framework that couples multi-robot collaborative pushing with simulation-in-the-loop planning to construct physically feasible paths in cluttered environments; (II) it demonstrates significant improvements in feasibility, efficiency, and scalability compared to existing NAMO and collaborative pushing approaches.

II. Problem Description

A. Model of Workspace and Robots

Consider a 2D workspace $\mathcal{W} \subset \mathbb{R}^2$ cluttered with a set of immovable obstacles $\mathcal{O}^{\texttt{fix}}$ and a set of $M > 0$ movable obstacles $\mathbf{\Omega} \triangleq \{\Omega_1, \dots, \Omega_M\} \subset \mathcal{W}$. Each movable obstacle Ω_m is modeled as a rigid polygon of arbitrary shape, with mass M_m, inertia I_m, and frictional parameters []. Its state at time t is $\mathbf{s}_m(t) \triangleq (\mathbf{x}_m(t), \psi_m(t))$, and the occupied area is denoted by $\Omega_m(t)$.

A team of N robots $\mathcal{R} \triangleq \{R_1, \dots, R_N\}$ operates in the workspace. Each robot R_n has state $\mathbf{s}_n(t) \triangleq (\mathbf{x}_n(t), \psi_n(t))$ with position and orientation, and occupies region $R_n(t) \subset \mathcal{W}$. Robots are equipped with low-level velocity tracking and can apply contact forces when interacting with obstacles. The free space at time t is thus

$$\widehat{\mathcal{W}}(t) \triangleq \mathcal{W} \backslash \big(\mathcal{O}^{\texttt{fix}} \cup \{\Omega_m(t)\}_{m \in \mathcal{M}} \cup \{R_n(t)\}_{n \in \mathcal{N}} \big).$$

B. Collaborative Pushing Modes

Robots interact with movable obstacles through *pushing modes*. For an obstacle Ω_m, a pushing mode is defined as $\boldsymbol{\xi}_m \triangleq (\xi_m, \mathbf{F}_m, \mathcal{N}_m)$, where (I) $\mathcal{N}_m \subseteq \mathcal{N}$ is the subgroup of robots assigned to push Ω_m; (II) ξ_m specifies the set of contact points $\mathbf{c}_n \in \partial\Omega_m$ for each robot; and (III) $\mathbf{F}_m$ denotes the set of contact forces applied at these points. The complete set of possible pushing modes is denoted by Ξ_m. Different pushing modes induce different obstacle motions, depending on contact geometry, forces, and frictional parameters [].

C. Problem Statement

The goal is to compute a hybrid plan that reconfigures the movable obstacles so that a collision-free path of width at least $W > 0$ exists between a given start $\mathbf{s}^{\mathrm{S}}$ and goal $\mathbf{s}^{\mathrm{G}}$. The plan specifies the sequence of pushing modes, robot assignments, and corresponding trajectories. Formally:

$$\begin{aligned} \underset{\substack{T,\, \{\mathbf{s}_n(t),\, \forall n\},\\ \{\boldsymbol{\xi}_m(t),\, \forall m\}}}{\textbf{min}} \quad & T + \alpha \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} \mathrm{J}_m\big(\boldsymbol{\xi}_m(t), \mathbf{S}_{\mathcal{N}}(t), \mathbf{s}_m(t)\big) \\ \textbf{s.t.} \quad & \exists\, \mathcal{P}_W :\ \mathbf{s}^{\mathrm{S}} \rightsquigarrow \mathbf{s}^{\mathrm{G}} \ \text{ with clearance } \geqslant W, \\ & \mathcal{N}_{m_1}(t) \cap \mathcal{N}_{m_2}(t) = \varnothing,\ \forall m_1 \neq m_2,\ \forall t, \\ & R_n(t) \cap R_{n'}(t) = \varnothing,\ \Omega_m(t) \cap \Omega_{m'}(t) = \varnothing,\ \forall t, \\ & R_n(t) \subset \widehat{\mathcal{W}},\ \Omega_m(t) \subset \widehat{\mathcal{W}},\ \forall n, m,\ \forall t. \end{aligned} \tag{1}$$

Here, $T > 0$ is the task duration, $\mathcal{T} \triangleq \{0, 1, \cdots, T\}$ the timeline, and $\alpha > 0$ balances duration against effort. The cost $\mathrm{J}_m(\cdot)$ evaluates the feasibility, stability, and control cost of choosing a particular pushing mode. The clearance constraint ensures that the final arrangement of obstacles permits a continuous path $\mathcal{P}_W$ with width at least W from start to goal.

III. Proposed Solution

This section presents a unified framework for collaborative multi-robot pushing. The approach consists of three main steps. First, a W-Clearance Connectivity Graph (Sec. III-A) is constructed to test the existence of a path of width W and identify frontier gaps when blocked. Second, a gap-ranking strategy (Sec. III-B) estimates the cost of clearing candidate gaps and guides robot coordination. Third, a simulation-in-the-loop search (Sec. III-C) expands a configuration-space tree while validating pushing actions through parallel physical simulation. The overall execution flow and generalizations are summarized in Sec. III-D.

A. W-Clearance Connectivity Graph (WCCG)

To reason about the existence of a traversable path of minimum width W, a W-Clearance Connectivity Graph (WCCG) is constructed. The key idea is to represent the free space $\widehat{\mathcal{W}}(t)$ as a graph whose edges encode feasible passages with clearance at least W, and whose nodes correspond to connected free regions. This allows efficient connectivity queries between start $\mathbf{s}^{\mathrm{S}}$ and goal $\mathbf{s}^{\mathrm{G}}$.

1) Free-Space Decomposition: The first step is to describe the free space available for navigation when movable and static obstacles are present. Since obstacles may be arbitrarily positioned, it is necessary to explicitly exclude their occupied areas to obtain the traversable region. At time t, the set of movable obstacles is denoted by $\mathbf{\Omega}(t) \triangleq \{\Omega_1(t), \ldots, \Omega_M(t)\}$, and the instantaneous free space is defined as

$$\widehat{\mathcal{W}}(t) \triangleq \mathcal{W} \backslash \big(\mathcal{O}^{\mathtt{fix}} \cup \{\Omega_m(t)\}_{m \in \mathcal{M}}\big), \tag{2}$$

where $\mathcal{O}^{\mathtt{fix}}$ are static obstacles and $\Omega_m(t)$ are movable obstacles.

However, this free space does not yet guarantee that a large vehicle of width W can traverse through it. To capture this requirement, the free space is eroded by a disk of radius $W/2$, so that any valid path maintains clearance of at least W. The resulting clearance-valid free space is

$$\widehat{\mathcal{W}}^{W}(t) \triangleq \{p \in \widehat{\mathcal{W}}(t) \mid \mathtt{dist}(p, \partial\widehat{\mathcal{W}}(t)) \geqslant W/2\}, \tag{3}$$

where $\mathtt{dist}(p, \partial\widehat{\mathcal{W}}(t))$ is the Euclidean distance from point p to the nearest boundary of $\widehat{\mathcal{W}}(t)$.

2) Graph Construction and Connectivity: Once the clearance-valid free space is obtained, it is represented in graph form to facilitate connectivity analysis. Each connected region of $\widehat{\mathcal{W}}^{W}(t)$ is treated as a node, and narrow gaps between regions are modeled as edges if they are wide enough for traversal. The WCCG is thus defined as

$$\mathcal{G}^{W}(t) \triangleq \big(\mathcal{V}^{W}(t), \mathcal{E}^{W}(t)\big), \tag{4}$$

where $\mathcal{V}^{W}(t)$ is the set of nodes representing connected components of $\widehat{\mathcal{W}}^{W}(t)$, and $\mathcal{E}^{W}(t)$ is the set of edges corresponding to separating gaps. Each gap is denoted by g with geometric width $w(g,t) \in \mathbb{R}_+$, and an edge exists between two nodes if $w(g,t) \geqslant W$.

Using this graph, the existence of a path from the start to the goal can be reduced to a standard connectivity check. Denote by $v^{\mathrm{S}}, v^{\mathrm{G}} \in \mathcal{V}^{W}(t)$ the nodes containing $\mathbf{s}^{\mathrm{S}}$ and $\mathbf{s}^{\mathrm{G}}$, respectively. Then a feasible path exists if and only if

$$\exists\, \mathcal{P}_W \subseteq \mathcal{G}^{W}(t) \quad \text{s.t.} \quad v^{\mathrm{S}} \rightsquigarrow v^{\mathrm{G}}, \tag{5}$$

where $\mathcal{P}_W$ denotes a connected subgraph of $\mathcal{G}^{W}(t)$ linking start and goal nodes. When no such path exists, the blocking frontier gaps are collected in the critical gap set

$$\mathcal{G}^{\mathtt{crit}}(t) \triangleq \{g \in \mathcal{G}(t) \mid w(g,t) < W,\ g \text{ lies on all paths between } v^{\mathrm{S}}, v^{\mathrm{G}}\}, \tag{6}$$

where $\mathcal{G}^{\mathtt{crit}}(t)$ contains exactly the narrow gaps that must be widened by pushing nearby obstacles, and thus provides the natural targets for subsequent ranking and obstacle-clearing steps.

B. Gap Ranking Strategy

When no clearance-W path exists, the critical gaps identified from the WCCG must be prioritized for obstacle-clearing actions. Since multiple gaps may block connectivity, it is necessary to rank them according to the estimated cost of clearing, so that robots can focus on the most promising ones. The gap-ranking strategy assigns a heuristic score to each gap by considering both the physical effort required to widen it and the expected benefit of restoring connectivity.

1) Gap Clearing Cost: The first element of the ranking strategy is to estimate the physical difficulty of widening a gap. Intuitively, narrow gaps require more clearance, and heavy obstacles are harder to move. Thus, the cost of clearing should depend on both the clearance deficiency and the blocking objects. For a gap $g \in \mathcal{G}^{\mathtt{crit}}(t)$ with width $w(g,t) < W$, the additional clearance required is defined as $\Delta w(g,t) \triangleq W - w(g,t)$. Let $\mathcal{O}(g,t) \subseteq \mathbf{\Omega}(t)$ denote the subset of movable obstacles adjacent to g. The clearing cost is then formulated as

$$\mathsf{C}(g,t) \triangleq \beta_1 \Delta w(g,t) + \beta_2 \sum_{\Omega_m \in \mathcal{O}(g,t)} \mathsf{M}_m, \tag{7}$$

where $\beta_1, \beta_2 > 0$ are weighting parameters, and M_m is the mass of obstacle Ω_m. The first term measures the geometric effort required to widen the gap, while the second reflects the physical effort of moving adjacent obstacles.

2) Gap Ranking Rule: The second element of the strategy is to evaluate how much clearing a specific gap improves connectivity between the start and the goal. If a gap is cleared, the effective shortest path through the WCCG may become shorter or even newly feasible. Denote the current shortest-path length between v^{S} and v^{G} by $d^{\mathtt{cur}}(t)$ (possibly infinite if no path exists). Suppose gap g is hypothetically cleared, yielding a modified graph $\mathcal{G}^{W}_{+g}(t)$ with updated shortest-path length $d^{+g}(t)$. The connectivity benefit of clearing gap g is then defined as

$$\mathsf{H}(g,t) \triangleq d^{\mathtt{cur}}(t) - d^{+g}(t), \tag{8}$$

where $\mathsf{H}(g,t) > 0$ indicates a reduction in shortest-path length after clearing g.

Finally, the ranking score combines benefit and cost into a single measure:

$$\mathsf{R}(g,t) \triangleq \frac{\mathsf{H}(g,t)}{\mathsf{C}(g,t) + \varepsilon}, \tag{9}$$

where $\varepsilon > 0$ is a small constant for numerical stability. Gaps with larger $\mathsf{R}(g,t)$ yield greater connectivity improvement per unit of clearing effort, and are therefore prioritized for obstacle-clearing actions in the next step.

C. Simulation-in-the-Loop Search

The final step integrates physical feasibility directly into the planning process through a simulation-in-the-loop search scheme. Unlike pure graph-based reasoning, which abstracts away dynamics, this method validates candidate pushing actions by embedding a fast parallelized physics engine inside the search loop. In this way, both the sequence of obstacle pushes and their dynamic feasibility are determined consistently, leading to robust execution in realistic settings. The procedure consists of three phases: (i) node selection based on a priority queue; (ii) expansion through candidate pushing actions validated in simulation; (iii) termination upon reaching a feasible clearance-W path.

Algorithm 1: Simulation-in-the-Loop Search (SiLS).

Input: Initial obstacle states $\mathbf{\Omega}(0)$, start $\mathbf{s}^{\mathrm{S}}$, goal $\mathbf{s}^{\mathrm{G}}$.
Output: Feasible clearance-W plan $\nu^\star$.

Initialize priority queue $Q = \{\nu_0\}$ with $\nu_0 = (\mathbf{\Omega}(0), \varnothing)$;
while $Q \neq \varnothing$ **do**
  /* **Selection** */
  Pop $\nu^\star = \mathbf{argmin}_{\nu \in Q}\{\mathrm{C}(\nu) + \mathrm{H}(\nu)\}$;
  Check path existence in $\mathcal{G}^W(t)$; if feasible, return $\nu^\star$;
  /* **Expansion** */
  Identify $\mathcal{G}^{\mathtt{crit}}(t)$ from WCCG;
  foreach $g \in \mathcal{G}^{crit}(t)$ **do**
    Generate candidate pushing modes $\xi \in \Xi(g)$;
    Simulate next state $\mathbf{\Omega}(t + \Delta t) = \mathtt{SimDyn}(\Omega_m, \xi, \Delta t)$;
    if *state is collision-free and improves* $w(g, t)$ **then**
      Insert new node ν^+ into Q;
Return best $\nu^\star$ if time limit reached;

1) Node Representation and Cost: Each search node represents the current configuration of movable obstacles together with the path status between start and goal. Formally, let $\nu = (\mathbf{\Omega}(t), \mathcal{P}_W(t))$, where $\mathbf{\Omega}(t)$ is the current set of obstacle states and $\mathcal{P}_W(t)$ denotes the existence of a clearance-W path in the WCCG. The cost of a node is defined as

$$\mathrm{C}(\nu) \triangleq T(\nu) + \alpha \sum_{g \in \mathcal{G}^{\mathtt{crit}}(t)} \mathsf{C}(g, t), \tag{10}$$

where $T(\nu)$ is the elapsed task duration, $\mathcal{G}^{\mathtt{crit}}(t)$ is the set of critical gaps, and $\mathsf{C}(g, t)$ is the clearing cost defined in (7). This balances planning time against clearing effort.

2) Simulation-Guided Expansion: When expanding a node, the planner hypothesizes pushing one of the blocking obstacles adjacent to a critical gap. Let $\xi = (\mathbf{c}, \mathbf{f})$ denote a candidate pushing mode, with contact point $\mathbf{c} \in \partial\Omega_m$ and force $\mathbf{f} \in \mathbb{R}^2$ applied by a subgroup of robots. The candidate action is validated through a short-horizon simulation

$$\Omega_m(t + \Delta t) \leftarrow \mathtt{SimDyn}\big(\Omega_m(t), \xi, \Delta t\big), \tag{11}$$

where $\mathtt{SimDyn}(\cdot)$ integrates the coupled robot-object dynamics with contact forces and friction. If the simulated trajectory avoids collisions and improves the clearance of at least one critical gap, the resulting state $\mathbf{\Omega}(t + \Delta t)$ is inserted into the queue as a new node.

3) Correctness and Completeness: The simulation-in-the-loop search inherits both correctness and probabilistic completeness under mild assumptions. First, correctness follows since each expansion is validated by a dynamics simulator that enforces physical constraints, guaranteeing that any returned plan is dynamically feasible.

Lemma 1 (Correctness)**.** *Any plan $\nu^\star$ returned by Algorithm 1 is feasible for the physical system, as every action has been validated through forward simulation with collision checking.*

Completeness is ensured by the hybrid search structure: as long as the set of pushing modes Ξ is sufficiently rich and the simulation horizon Δt is bounded below, all feasible solutions are eventually explored.

Theorem 2 (Probabilistic Completeness)**.** *If a clearance-W path exists, Algorithm 1 finds a feasible plan $\nu^\star$ with probability approaching* 1 *as the number of node expansions tends to infinity.*

The proofs follow by extending standard arguments from randomized search algorithms with embedded feasibility oracles, noting that the simulator acts as a physical feasibility oracle in this case.

D. Overall Analyses

1) Online Execution and Adaptation: Once the hybrid pushing plan $\nu^\star$ is generated, the robot fleet executes the task in real-time by following the guiding path $\overline{\mathbf{S}} = \{\mathbf{s}_0, \ldots, \mathbf{s}_L\}$. Each robot tracks the assigned trajectory segments, pushing the target along the planned path while adjusting its motion based on the current target state $\mathbf{s}_m(t)$ and interaction mode $\xi_n(t)$. The robots apply the desired forces at the contact points $\mathbf{c}_n(t)$ using the control law:

$$\widehat{\mathbf{v}}_n = K_{\mathtt{vel}}\left(\widehat{\mathbf{c}}_n - \mathbf{c}_n\right), \quad \widehat{\omega}_n = K_{\mathtt{rot}}\left(\widehat{\psi}_n - \psi_n\right),$$

to guide the object from the start to the goal while ensuring contact force consistency. Transitioning between obstacles is handled according to the hybrid plan. Specifically, as each robot pushes the object, it follows a **collision-free path** to the next set of contact points on the next obstacle, ensuring that all movement respects the planned interaction modes, and the sequence of obstacles is maintained. This transition is dynamically adjusted as the robots continue pushing the target toward the goal.

In case of failures, such as when the object deviates from its expected position, the algorithm triggers adaptation. If the object fails to reach the desired position, the system checks the tracking error:

$$\mathrm{dist}(\mathbf{s}_m(t'), \varrho_\ell) \geqslant \delta_{\mathtt{f}}, \quad \|\mathbf{s}_m(t') - \mathbf{s}_m(t'')\| < r_{\mathtt{stuck}},$$

and re-computes the entire hybrid plan based on the current system state. This includes recalculating the sequence of movable obstacles to push, the contact points, and the pushing forces to be applied. The updated plan is then executed with the control law in (**??**) to guide the robots toward the goal despite disruptions, ensuring the task is completed even under unforeseen circumstances.

2) Complexity Analysis: The time complexity of the proposed method is dominated by three main components: the W-Clearance Connectivity Graph (WCCG), gap-ranking strategy, and simulation-in-the-loop hybrid search. Constructing the WCCG has a complexity of $\mathcal{O}(M^2 \log M)$, where M is the number of movable obstacles, due to the graph search and clearance checks. The gap-ranking strategy, which

TABLE I

COMPARISON WITH BASELINES

Env. N	Env. M	Env. J	Methods	Resp. Time[s]	Ave/Max Plan Time[s]	T/W/X Agents	Succ. Rate[%]
80	30	492	**Ours**	62.7	1.4/4.9	20/11/37	100
			C-MILP-1	88.5	100/192	16/6/26	100
			C-MILP-2	28.3	19/64	12/10/25	86
			SAMP-1	108.3	54/92	19/8/24	100
			SAMP-2	40.7	5.1/16	23/5/20	85
			Inf-H	77.5	$>$ 15min	24/17/31	100
			Greedy	112.1	0.3/0.6	33/9/30	100

involves evaluating each critical gap and ranking them based on their clearance and the obstacles' mass, has a complexity of $\mathcal{O}(G \log G + GM)$, where G is the number of critical gaps and M is the number of obstacles. The most computationally intensive part is the simulation-in-the-loop hybrid search, which checks the feasibility of each mode by simulating robot dynamics. The complexity for each simulation step is $\mathcal{O}(N^{3.5})$, where N is the number of robots, and the overall complexity of the hybrid search is $\mathcal{O}(N^{3.5} \cdot M \cdot L)$, where L is the number of keyframes in the path. Hence, the overall complexity is primarily determined by the hybrid search algorithm.

3) Generalization: The proposed framework can be generalized in several directions. (I) *Heterogeneous robots*: When robots have varying capabilities, such as different maximum forces, the hybrid plan $\nu^\star$ should be adjusted by assigning more powerful robots to obstacles that require higher forces, with smaller robots handling less demanding tasks. This adjustment is reflected in the interaction modes $\boldsymbol{\xi} = (\mathbf{c}_1, \mathbf{f}_1, R_1), \ldots, (\mathbf{c}_N, \mathbf{f}_N, R_N)$, where $\mathbf{f}_n$ represents the force for each robot. (II) *Simultaneous clearing*: The framework can be extended to allow multiple obstacles to be pushed simultaneously, improving the efficiency of path clearing. This requires modifying the WCCG, $\mathcal{G}^W(t)$, to support parallel tasks and ensuring collision-free paths for multiple robots. (III) *Dynamic obstacle movement*: The method can handle dynamic obstacles by incorporating real-time sensor data and updating the WCCG, allowing the hybrid search algorithm in Section III-C to adapt the plan on-the-fly to avoid collisions and clear the path efficiently, even with moving obstacles.

IV. NUMERICAL EXPERIMENTS

To numerical validations, the proposed method is implemented in `Python3` and tested on a laptop with an Intel Core i5-12500H CPU. The solver `GLOP` [1] is adopted for linear and integer optimization. Simulation videos can be found in the supplementary files.

A. System Description

B. Results

As shown in Fig. 1

C. Comparisons

The proposed method is compared against **six** baselines:

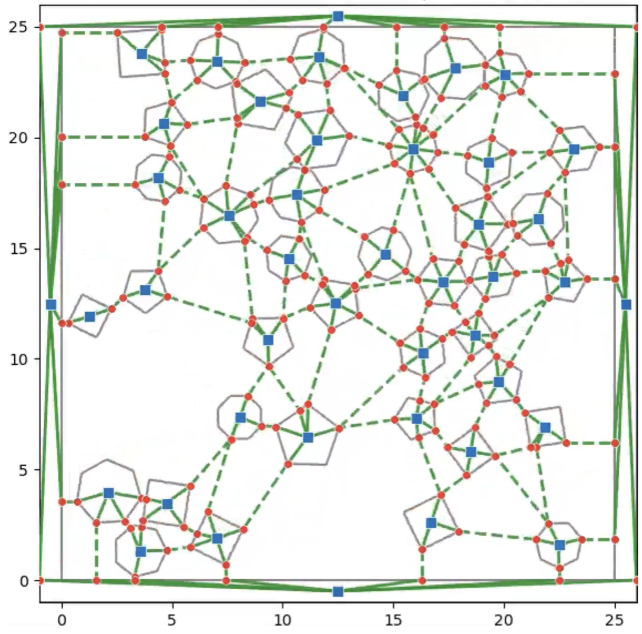

Fig. 2. **Top**: the number of subteams and their composition; **Bottom**: the status of agents in different modes and the number of tasks.

TABLE II

SCALABILITY ANALYSIS

Env. N	Env. M	Env. J	α	Resp. Time[s]	Ave/Max Plan Time [s]	T/W/X Agents	Succ. Rate[%]
120	50	824	0.05	95	1.6/4.6	33/10/39	100
			0.1	102	1.7/4.9	26/9/35	100
150	80	1319	0.05	153	1.8/5.1	37/12/37	100
			0.1	165	2.1/6.2	12/11/34	97

D. Generalization

For further validation, the fleet size is further increased

V. CONCLUSION

This work introduces a multi-robot approach for path clearing in unstructured environments, utilizing a hybrid search algorithm to plan and execute the sequence of obstacles, contact points, and forces efficiently. The framework demonstrates real-time adaptability to dynamic scenarios. Future work will address uncertainties such as estimating obstacle positions without external monitoring, handling uncertain obstacle masses, and managing partial robot visibility to improve robustness and performance in real-world, dynamic environments.

REFERENCES

[1] G. L. O. Solver, https://developers.google.com/optimization/lp.