Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

BIC-GA: Return-Aware Cooperative Exploration in Fragmented Environments via Directed Commitments and Gate Admission

Yao Xue

yao.xue@monash.edu

Monash University Malaysia

Chee Keong Tan

Monash University Malaysia

Article

Keywords:

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11248079/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

BIC-GA: Return-Aware Cooperative Exploration in Fragmented Environments via Directed Commitments and Gate Admission

Yao Xue[1,*] **and Chee Keong Tan**[1,2]

[1]School of Information Technology, Monash University Malaysia, 47500 Subang Jaya, Selangor, Malaysia
[2]Monash Climate-Resilient Infrastructure Research Hub (M-CRInfra), School of Engineering, Monash University Malaysia, 47500 Subang Jaya, Selangor, Malaysia
[*]yao.xue@monash.edu

ABSTRACT

Dual-robot exploration in fragmented unknown environments is constrained by incomplete sensing and complex free-space structure. In narrow passages, dead ends, and shared bottlenecks, frontier updates can destabilize assignments, while locally attractive targets may be infeasible once branch entry, service execution, and return requirements are considered. This study presents Branch-Intent Commitment with Gate Admission (BIC-GA) for return-feasible dual-robot target-coverage exploration. Instead of assigning raw frontier points, BIC-GA models each frontier component as a structured, directed commitment characterized by its entrance location, traversable entry orientation, expected information gain, entry difficulty, and predicted service workload. The robots select a commitment pair through a return-aware joint objective after predictive loop-budget screening, considering outbound, service, and return costs, overlap, short-prefix conflict, and workload balance. At local gates, exploratory changes require sufficient improvement over legal continuation, while an execution guard checks post-step return margins before motion to preserve modeled return feasibility. Experiments on fragmented grid maps show that, over jointly successful conditions, BIC-GA reduces mean team path length by 6.2–26.4%, turning effort by 34.9–50.4%, and composite motion cost by 7.8–27.8% relative to four tested baselines. Component tests further demonstrate more stable assignments, fewer reversals and conflict-repair interventions, and reliable exploration–return performance under budget constraints.

Keywords: Dual-robot exploration; Directed commitments; Return-aware planning; Gate admission; Fragmented environments

Introduction

Autonomous exploration in unknown environments is important for robots used in search-and-rescue, collapsed-building inspection, underground surveys, and other hazardous tasks. In these scenarios, robots must reveal unmapped space, avoid obstacles, and retain sufficient resources to return to their starting positions. Frontier-based exploration provides a natural starting point by selecting targets along the boundary between known free space and unknown space[1]. Although this approach has been widely adopted, frontier-based target selection can become less stable when the environment contains narrow passages, dead ends, and densely distributed obstacles.

Fragmented environments introduce difficulties that cannot be fully described by the location or information gain of a frontier alone. A nearby target may require a long detour, geometrically close frontiers may be separated by an obstacle, and different regions may share the same entrance. Changes in the observed frontier can also lead a robot to reconsider its target even when the intended exploration region remains unchanged. Such behavior may interrupt useful progress and cause repeated traversal. Exploration decisions therefore need to account for the structure of the selected region and maintain a sufficiently stable intent as new information becomes available.

Cooperative exploration can improve efficiency by distributing exploration tasks among robots. Coordinated frontier methods consider the target utility and travel cost to reduce redundant coverage[2], while market-based approaches allocate tasks according to the expected gain and execution cost[3]. At the planning level, hierarchical planners such as FUEL and TARE organize spatial information and combine local and global planning to improve exploration efficiency[4,5]. These methods provide effective ways to select and organize exploration targets. However, assigning different targets does not necessarily prevent interference in fragmented environments. Robots heading to separate regions may still depend on the same narrow passage, leading to waiting, repeated motion, or competing local decisions. The value of a target therefore depends partly on how its execution interacts with the tasks assigned to other robots.

Resource constraints further complicate this process. Energy-aware exploration has shown that motion and heading changes should influence target selection[6], while full-mission studies emphasize the importance of connecting exploration planning with reliable execution and mission completion[7]. In fragmented environments, a region that appears attractive may require substantial effort to enter, explore, and leave. Considering only the travel cost to a target can therefore overlook the resources needed to complete the task and return home. Return requirements should influence exploration decisions throughout the mission, rather than being considered only when the remaining budget becomes critical.

Stable local execution is equally important. Classical multi-agent path-finding methods coordinate routes for given start–goal configurations[8], whereas online exploration involves evolving map knowledge and target assignments. At a narrow entrance or junction, a robot may need to maintain its current intention, enter another region, or yield to its partner. Frequent changes in these decisions can create unnecessary reversals and reduce the benefit of coordinated target allocation. At the same time, maintaining an exploration intention should not prevent a necessary return. These difficulties motivate a closer connection between exploration intent, cooperative allocation, and local execution.

To address these problems, this paper presents Branch-Intent Commitment with Gate Admission (BIC-GA) for return-feasible dual-robot target-coverage exploration in fragmented unknown environments. The central idea is to maintain an explicit exploration commitment that describes the intended branch through its entrance and entry direction, while allowing the execution target to adapt as the map develops. These commitments provide a common task reference for return-aware joint selection and Gate Admission: the former evaluates predicted task burden and inter-robot interaction, while the latter regulates local changes relative to the selected intentions. Together with an execution-level return guard, these components connect target selection to stable local execution under modeled return requirements.

The main contributions are summarized as follows.

- A directed commitment representation is developed to describe branch-level exploration intent through its entrance, entry direction, and belief-derived task features, separating the selected exploration task from the frontier representative used for execution.
- A return-aware joint selection model is proposed to coordinate exploration commitments through predictive loop-budget screening and joint evaluation of exploration value, task burden, and inter-robot interaction. The task burden accounts for outbound travel, estimated service workload, and return cost.
- A commitment-guided Gate Admission mechanism is introduced to regulate exploratory changes relative to legal continuation, reducing unnecessary switching and reversal. An execution guard checks post-step return margins before motion to preserve per-step modeled return feasibility, with necessary return taking priority over exploratory commitments.

The remainder of this paper reviews the related work, presents the proposed methodology, and reports the experimental evaluation. The final section discusses the findings and limitations and concludes the paper.

Related Work

Autonomous exploration requires decisions about where to gather information, how to distribute the remaining work, and how to execute the selected tasks within the available resources. Research on frontier allocation, structured representations, learning-based cooperation, and resource-aware coordination addresses different parts of this process.

Frontier-Based Exploration and Task Allocation

Frontier-based exploration connects the uncertainty of the map with navigation by directing robots toward the boundary between known free space and unknown space. Yamauchi introduced this formulation for incremental exploration and subsequently extended it to multiple robots sharing map information[1], [9]. However, sharing observations alone does not determine how the robots should divide their next targets. Simmons et al. introduced explicit coordination based on the cost and expected information associated with candidate destinations[10]. Burgard et al. combined travel cost with target utility and reduced the utility of areas expected to be observed by other robots[2]. Market-based methods approached the same allocation problem through bids reflecting the task value and execution cost[3], with their broader coordination principles reviewed by Dias et al.[11]. These studies established that an exploration target should be evaluated in relation to the tasks assigned to the rest of the team.

Later, the methods refined the distribution and ranking of the frontier targets. MinPos uses a decentralized allocation rule that considers the spatial distribution of robots[12]. Colares and Chaimowicz combined information gain, travel cost, and a coordination term to discourage redundant exploration[13], while coExplore combined multiple frontier classifications to reduce the preference for large frontiers[14]. These approaches improve target allocation, but target separation and execution compatibility remain distinct considerations: different frontiers may still depend on the same narrow entrance or overlapping approach paths. In fragmented environments, task selection, therefore, needs to consider branch access alongside frontier value.

Topological and Hierarchical Exploration

Topological representations organize the connectivity of free space and provide a basis for reasoning beyond individual grid cells. Kuipers and Byun used a hierarchy of spatial representations for exploration and mapping[15], while Thrun combined metric and topological maps for indoor navigation[16]. Choset and Burdick developed the hierarchical generalized Voronoi graph for sensor-based exploration[17]. These approaches established the role of connectivity and meaningful locations in reducing the dependence of navigation on a dense metric representation.

The environmental structure has also been incorporated into the allocation and planning of exploration. Wurm et al. assigned robots to segments of the environment to reduce redundant exploration[18]. In subterranean environments, Dang et al. combined local and global graph planning to guide aerial and legged robots through branching spaces[19]. FUEL maintains an incremental frontier information structure for hierarchical planning[4], whereas TARE combines detailed local planning with a compact global representation[5]. RACER extends structured exploration to multi-UAV decentralized teams through online space decomposition and pairwise task allocation[20]. Together, these methods use spatial organization to guide exploration and manage planning efforts on different scales.

Entrances and room boundaries have also been used explicitly. Kim et al. extract geometric cues for doors and rooms and apply circular decomposition to coordinate multi-room exploration[21]. This connects the entrance geometry with the region-level coordination. For a partially observed branch, a complementary concern is how entrance-based structural information can support a continuing task as observations change, without equating every execution-target update with task reassignment.

Learning-Based Exploration and Cooperation

Learning-based methods adapt exploration decisions using observations and experience. Niroui et al. applied deep reinforcement learning to exploration in unknown cluttered environments motivated by search-and-rescue tasks[22]. Hu et al. combined Voronoi partitioning with deep reinforcement learning for multi-robot exploration[23]. Other work emphasizes the information needed for cooperation: H2GNN integrates information across graph-hop distances to support learned multi-robot exploration[24], while MADE-Net uses macro actions and teammate-intention reasoning for decentralized exploration under limited communication[25]. These studies show how spatial context and anticipated teammate behavior can guide decisions beyond a fixed frontier score.

The present study considers a planning-based formulation in which branch intent, estimated workload, and local action acceptance are evaluated using the shared belief map, robot states, and maintained commitments rather than an offline-trained exploration policy.

Energy-Aware Planning and Local Coordination

Energy constraints influence both the routes a robot can follow and the tasks it can afford to complete. Mei et al. incorporated motion energy and heading changes in exploration decisions[6]. Morales et al. modeled skid-steer robot power consumption[26], and Tiwari et al. estimated the operating range while accounting for maneuvering and other onboard consumers[27]. These studies motivate evaluating resource use beyond travel distance alone. Datsko et al. further combined multi-UAV coverage planning with flight-speed selection and battery constraints[28], linking route decisions with mission-level resource limits.

Unknown-environment exploration introduces the additional difficulty that the remaining task geometry is only partially observed. Karumanchi et al. combined frontier exploration, graph-based planning, and multi-robot assignment in a setting where robots discover and revisit charging stations, deriving sufficient conditions for recursive feasibility and finite-time termination[29]. This work directly addresses resource feasibility during exploration. The task considered here instead emphasizes reaching a coverage target and returning home within the available budget. Branch-level selection must anticipate approach, service, and return costs, while recognizing that service workload remains an estimate before the branch has been fully observed. Task prediction therefore guides allocation, whereas the execution guard checks the next proposed step against the modeled return requirement before motion.

Resource-aware task selection must also be supported by coordinated motion. The full-mission exploration work of Best et al. integrates sensing, planning, and onboard execution within an autonomy framework[7]. At the motion-planning level, cooperative pathfinding coordinates movements in space and time[30], and Safe Interval Path Planning represents collision-free timing through safe intervals[31]. ORCA provides reciprocal collision avoidance[32], while Conflict-Based Search resolves path conflicts through constraints on individual robots[8]. Stern et al. clarify the assumptions, objectives, and benchmarks underlying multi-agent pathfinding[33].

These methods address motion feasibility and coordination, but a further decision arises when exploration goals evolve online: whether a newly available local action offers sufficient improvement over continuing the selected exploration intention. A collision-free alternative can still introduce unnecessary switching or immediate reversal, while maintaining an exploration intention should not block a necessary return. The issue is therefore not only how to coordinate paths, but also when to accept exploratory changes during execution.

BIC-GA addresses this connection between task representation, allocation, and execution in fragmented dual-robot exploration by using directed commitments as a common reference for task selection and local action admission under modeled

return requirements.

Methodology

At each update, BIC-GA constructs directed commitments from the shared belief map and selects a commitment pair using predicted task burdens and inter-robot interactions. The selected commitments provide a common reference for Gate Admission, which evaluates exploratory changes relative to legal continuation. An execution guard then checks the post-step return margins of the proposed motion before execution, separating predictive task selection from per-step modeled return feasibility. Figure 1 provides a schematic overview of the three main stages.

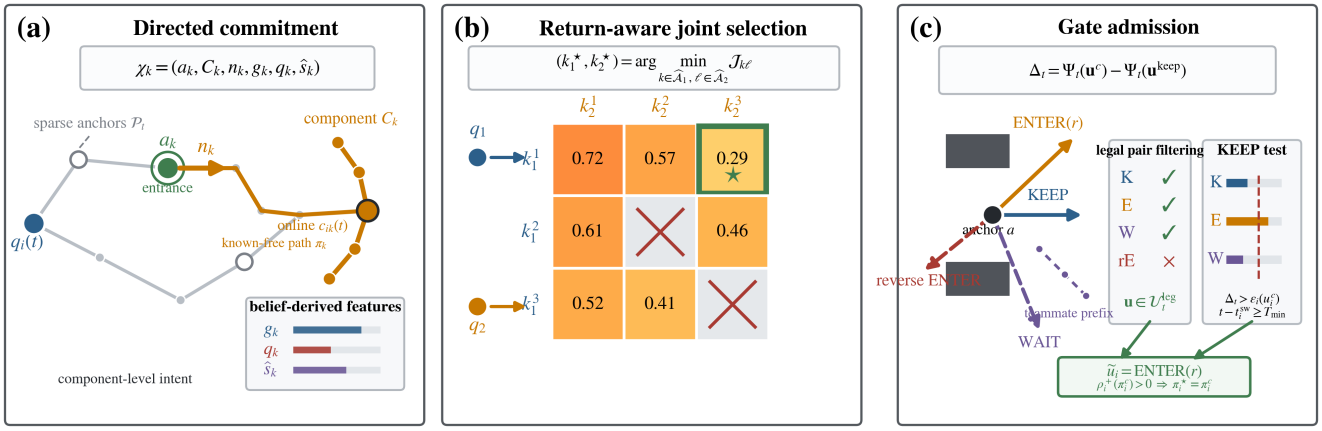


Figure 1. Schematic overview of BIC-GA. (a) Directed commitments encode entrance a_k, entry direction n_k, and belief-derived features for frontier component C_k, while the execution representative $c_{ik}(t)$ can change within it. (b) Return-aware joint selection after predictive loop-budget screening: crossed pairs are excluded, and the highlighted pair minimizes joint cost over $\widehat{\mathcal{D}}_t$ in Eq. (19). Values are illustrative. (c) Gate Admission compares exploratory changes with legal continuation under compatibility, improvement, and commitment-duration checks. The illustrated equality $\pi_i^\star = \pi_i^c$ assumes no return override or conflict repair. The full procedure includes joint revalidation, return guarding, prefix repair, and a final return-margin recheck.

Exploration Model and Return Requirement

Two robots explore a finite grid $\Omega \subset \mathbb{Z}^2$. Local observations update a shared belief map $\mathscr{M}_t$, whose cell labels are `unk`, `free`, or `occ`. The environment is static and observed cell labels are retained as the map develops. For robot $i \in \{1,2\}$, the state and pose are

$$x_i(t) = \big(p_i(t), \psi_i(t), e_i(t), h_i\big), \qquad q_i(t) = \big(p_i(t), \psi_i(t)\big), \tag{1}$$

where p_i, ψ_i, e_i, and h_i denote position, heading, remaining motion budget, and home position. The known-free graph $G_t = (V_t, E_t)$ is

$$\begin{aligned} V_t &= \{x \in \Omega \mid \mathscr{M}_t(x) = \texttt{free}\}, \\ E_t &= \{(x,y) \in V_t \times V_t \mid y \in \mathscr{N}_4(x)\}, \end{aligned} \tag{2}$$

where $\mathscr{N}_4(x)$ is the four-connected neighborhood within Ω. The approach, local-prefix, and home-return paths all use this graph. Unknown cells are not used as shortcuts, so a planned route relies on connections already verified by observation.

The task is to reach a prescribed coverage level η and return both robots home without budget depletion. Coverage is evaluated as

$$\Gamma_t = \frac{|\{x \in \Omega_{\text{eval}} \mid \mathscr{M}_t(x) \neq \texttt{unk}\}|}{|\Omega_{\text{eval}}|}, \tag{3}$$

where Ω_{eval} is the fixed evaluation set of the simulated map: free cells together with occupied cells adjacent to free space. This set is used for coverage evaluation and target-completion monitoring, not as prior geometric information for commitment construction or path planning, which use $\mathscr{M}_t$. Reaching $\Gamma_t \geq \eta$ satisfies the exploration target, while mission success also

requires both home returns. Efficiency is assessed over the resulting exploration–return task rather than by maximizing coverage after the target has been reached.

For a pose path $\pi = (q^0, \dots, q^n)$, with $q^\ell = (p^\ell, \psi^\ell)$, the motion cost is

$$J(\pi) = \sum_{\ell=1}^{n} \left[\mathbf{1}\{p^\ell \neq p^{\ell-1}\} + \lambda_\theta \Delta\theta(\psi^{\ell-1}, \psi^\ell) \right], \tag{4}$$

where $\Delta\theta$ measures the turning effort and λ_θ weights it relative to unit-cost translation. Thus, paths of equal length can have different costs when their turning requirements differ. The budget $e_i(t)$ uses the same units as J, and a stationary action with an unchanged heading has zero motion cost. This is a composite translation–turn model, not a measurement of electrical energy.

Let $\Pi_t(q, v)$ contain the pose paths in G_t that start at q and end at position v. The minimum known-free path cost is

$$D_t(q, v) = \min_{\pi \in \Pi_t(q,v)} J(\pi), \qquad \min \varnothing = +\infty, \tag{5}$$

with the terminal heading chosen by the path optimization. The current return reserve is

$$\rho_i(t) = e_i(t) - \kappa_s D_t\big(q_i(t), h_i\big) - e_0, \tag{6}$$

where $\kappa_s \geq 1$ inflates the return cost and $e_0 \geq 0$ is a retained budget margin. The modeled return requirement is $\rho_i(t) \geq 0$. A nonpositive reserve gives home return priority. However, a positive current reserve does not by itself justify another exploratory step: that step consumes budget and may increase the distance or turning effort needed to return. The selection model below estimates task-level budget demand, whereas the execution guard checks the post-step reserve of the proposed motion before execution. The stepwise return-feasibility argument assumes an initially return-feasible state, deterministic motion costs, and verified free cells that remain traversable.

Directed Exploration Commitments

A directed commitment separates branch-level exploration intent from the frontier representative used for motion, as shown in Figure 1(a). The frontier set and its connected components are

$$\begin{aligned} \mathscr{F}_t &= \{x \in V_t \mid \exists y \in \mathscr{N}_4(x), \mathscr{M}_t(y) = \texttt{unk}\}, \\ \mathscr{C}_t &= \mathrm{CC}_{G_t}(\mathscr{F}_t) = \{C_1, \dots, C_K\}, \end{aligned} \tag{7}$$

where connectivity is evaluated through frontier cells, not through arbitrary intervening free cells. A component provides a larger task object, but it still does not specify the entrance through which it should be served. This distinction is important when neighboring frontier segments lie behind different passages or distant targets share an entrance.

Entrances are drawn from structural anchors in the known-free graph. Let $\nu_t(v) = \deg_{G_t}(v)$, let $\mathrm{opp}_t(v)$ indicate opposite neighbors at a degree-two node, and let $w_t(v)$ denote the local known-free cross-section width. For a narrow-passage threshold w_c, define $w_c^+ = \max(2, w_c + 1)$. The anchor set is

$$\begin{aligned} \mathscr{P}_t = \{&v \in V_t \mid \nu_t(v) \in \{0,1,3\}\} \\ &\cup \{v \in V_t \mid \nu_t(v) = 2, [\neg \mathrm{opp}_t(v) \vee w_t(v) \leq w_c]\} \\ &\cup \{v \in V_t \mid \nu_t(v) = 4, w_t(v) \leq w_c^+\} \\ &\cup \{p_1(t), p_2(t), h_1, h_2\}. \end{aligned} \tag{8}$$

Endpoints, junctions, corners, and narrow passages are retained because they identify changes in access or direction. Unconstricted degree-four cells are omitted unless included as robot or home positions; they remain available in the planning graph.

For component C_k, define

$$\begin{aligned} d_{G_t}(a, C_k) &= \min_{x \in C_k} d_{G_t}(a, x), \\ \mathscr{P}_k^{\mathrm{r}} &= \{a \in \mathscr{P}_t \mid d_{G_t}(a, C_k) < \infty\}, \\ \mathscr{P}_k^{\mathrm{ent}} &= \begin{cases} \mathscr{P}_k^{\mathrm{r}} \setminus C_k, & \mathscr{P}_k^{\mathrm{r}} \setminus C_k \neq \varnothing, \\ \mathscr{P}_k^{\mathrm{r}}, & \text{otherwise.} \end{cases} \end{aligned} \tag{9}$$

The entrance is then

$$a_k = \arg\min_{a \in \mathscr{P}_k^{\mathrm{ent}}} d_{G_t}(a, C_k). \tag{10}$$

Here, d_{G_t} measures graph distance for entrance construction, whereas D_t includes heading-dependent motion cost. The preference for an external anchor separates the branch entrance from the frontier itself whenever the observed structure permits it. The reachable-anchor fallback retains a candidate when no external anchor is available.

Let $\pi_k = (v_0, \dots, v_m)$ be a shortest known-free path from $v_0 = a_k$ to a nearest $v_m \in C_k$. For $m \geq 1$, its first edge gives

$$n_k = \frac{v_1 - v_0}{\|v_1 - v_0\|_2}. \tag{11}$$

A centroid can lie across a wall from the entrance even when the component is reachable by a detour. Using the path's first edge therefore provides a locally traversable entry direction without requiring the rest of the route to remain straight. For a zero-length entrance path, the retained cardinal centroid-direction fallback supplies an orientation cue rather than certifying an outgoing free edge. Subsequent movement remains restricted to known-free connections.

The remaining features distinguish an informative branch from one that is expensive to enter and serve. Let m_k combine unknown support adjacent to C_k with a short forward cone, ρ_k be the occupied-to-known ratio near a_k, σ_k be local branching excess with frontier-spread support, and $w_k = w_t(a_k)$. The gain score, bounded entry difficulty, and predicted service workload are

$$\begin{aligned} g_k &= \alpha_m m_k + \alpha_C |C_k| + \alpha_\sigma \sigma_k + \alpha_w \max(w_k - 1, 0), \\ q_k &= \mathrm{sigm}\left(c_\rho \rho_k + \frac{c_w}{\max(w_k, 1)} - c_\sigma \sigma_k \right), \\ \widehat{s}_k &= a_m m_k + a_C |C_k| \\ &\quad + \lambda_\theta \left(a_\rho \rho_k + \frac{a_w}{\max(w_k, 1)} + \frac{a_\sigma}{\max(\sigma_k, 1)} \right), \end{aligned} \tag{12}$$

where $\mathrm{sigm}(z) = (1 + e^{-z})^{-1}$. Unknown support and frontier size contribute to both potential gain and service burden, reflecting the information available from the current task and the work required to obtain it. Density, narrowness, and limited branching raise the estimated entry or turning burden. The score q_k ranks local difficulty rather than estimating a failure probability, and $\widehat{s}_k$ is a belief-derived surrogate because the interior of the branch is not yet known.

The resulting task object is

$$\chi_k = (a_k, C_k, n_k, g_k, q_k, \widehat{s}_k), \qquad \mathscr{X}_t = \{\chi_1, \dots, \chi_K\}. \tag{13}$$

Each observed component produces at most one proxy in the current update; splits and merges reconstruct the candidates rather than implying a permanent partition of unknown space. Candidate indices are local to the current update. The entrance–direction signature (a_k, n_k) describes the semantic branch intent used in the commitment-switch analysis, while C_k and the associated features describe its current observed support.

During execution, robot i selects a reachable representative $c_{ik}(t) \in C_k$ using turn-aware cost, unknown local support, forward progress, and revisit penalties. Changing this representative or updating the belief-derived scores does not, by itself, change the entrance–direction signature. The representation therefore distinguishes execution-target updates from changes in branch intent. Ownership regularization discourages premature task handover, while Gate Admission regulates subsequent local action changes relative to legal continuation. The Algorithm 1 summarizes the construction.

Algorithm 1 Directed Commitment Construction

Require: Belief map $\mathscr{M}_t$, robot states, construction parameters
Ensure: Current commitment set $\mathscr{X}_t$
1: Extract $\mathscr{F}_t$ and $\mathscr{C}_t$ on the known-free graph G_t.
2: Construct $\mathscr{P}_t$ using Eq. (8); set $\mathscr{X}_t \leftarrow \varnothing$.
3: **for** each component $C_k \in \mathscr{C}_t$ **do**
4: Form the reachable entrance candidates in Eq. (9) and select a_k.
5: Find a shortest known-free path from a_k to C_k.
6: Set n_k from its first edge, with the centroid-direction fallback for a zero-length path.
7: Compute m_k, ρ_k, σ_k, and w_k from the current map.
8: Evaluate g_k, q_k, and $\widehat{s}_k$ using Eq. (12).
9: Form χ_k and add it to $\mathscr{X}_t$.
10: **end for**
11: **return** $\mathscr{X}_t$.

Return-Aware Joint Commitment Selection

Directed commitments expose the intended entrances, but selecting them independently can still send both robots toward the same branch or through a shared passage. Moreover, a branch with high gain may be affordable to reach, but too expensive to serve and leave. Joint selection first screens each robot's candidate tasks for predicted budget sufficiency, then evaluates the interaction between the remaining assignments, as illustrated in Figure 1(b).

For robot i and commitment k, let q^a_{ik} be the arrival pose of a known-free path of minimum-cost from $q_i(t)$ to a_k. The predicted loop burden is

$$\widehat{B}_{ik} = D_t\big(q_i(t), a_k\big) + \widehat{s}_k + D_t\big(q^a_{ik}, h_i\big). \tag{14}$$

The approach and return terms depend on the robot's pose and home location, while $\widehat{s}_k$ describes the estimated work associated with the branch. Retaining the arrival heading in q^a_{ik} accounts for its contribution to the return cost. If either path is not available, $\widehat{B}_{ik} = +\infty$. Since service is estimated and return is referenced to the entrance, this burden screens prospective tasks rather than certifying their complete future execution.

The predicted-admissible set is

$$\widehat{\mathscr{A}_i}(t) = \Big\{ k \in \{1,\dots,K\} \;\Big|\; \widehat{B}_{ik} < \infty, \quad e_i(t) - \kappa_s \widehat{B}_{ik} - e_0 \geq 0 \Big\}. \tag{15}$$

Applying the screen before scoring prevents high information gain from compensating for a predicted budget deficit. It can admit a branch for one robot and exclude it for the other, even though the branch features are shared. For each admitted candidate, the individual selection cost is

$$R_{ik} = -g_k + \lambda_B \widehat{B}_{ik} + \lambda_Q q_k + P_{\text{own}}(i,k). \tag{16}$$

The gain favors informative branches, while burden and difficulty discourage costly service. The ownership term penalizes premature reassignment of the partner's branch with a smaller penalty when relay is permitted. This discourages task exchange after minor score changes without making ownership permanent.

Individually favorable commitments do not necessarily form a favorable pair. Spatial separation can reduce duplicated service, but different destinations may still share an approach prefix. These effects enter the interaction vector

$$\mathbf{h}_{k\ell} = \begin{bmatrix} -S_{\text{sep}}(k,\ell) \\ P_{\text{ov}}(k,\ell) \\ P_{\text{CH}}(k,\ell) \\ \dfrac{|\widehat{B}_{1k} - \widehat{B}_{2\ell}|}{e_1(t) + e_2(t) + \varepsilon_e} \end{bmatrix}. \tag{17}$$

Here, S_{sep} rewards separation in the entrance location and the direction of the branch. The overlap term P_{ov} penalizes shared or nearby entrances and shared frontier support, whereas P_{CH} measures predicted vertex, head-on, and near-prefix interactions over the horizon H. The latter depends on robot 1 approaching k and robot 2 approaching ℓ, not on the two destination locations alone. The final term discourages unequal predicted task burdens relative to the current team budget; $\varepsilon_e > 0$ prevents a zero denominator. It does not require the robots to travel identical distances.

The pair cost is

$$\mathscr{J}_{k\ell} = R_{1k} + R_{2\ell} + \lambda_h^\top \mathbf{h}_{k\ell}, \tag{18}$$

where λ_h contains the weights of the interaction terms. A same-commitment pair is excluded unless ownership permits relay. Writing the admitted pair domain explicitly gives the following.

$$\begin{aligned} \widehat{\mathscr{Q}}_t &= \big\{ (k,\ell) \in \widehat{\mathscr{A}_1}(t) \times \widehat{\mathscr{A}_2}(t) \;\big|\; k \neq \ell \text{ or relay is permitted} \big\}, \\ (k_1^\star, k_2^\star) &= \operatorname*{arg\,min}_{(k,\ell)\in\widehat{\mathscr{Q}}_t} \mathscr{J}_{k\ell}. \end{aligned} \tag{19}$$

Joint minimization is used when an admitted pair is available. A robot with an empty individual set is assigned home return, and a remaining active robot minimizes R_{ik} over its own set. Returning or finished robots are not assigned a new exploration commitment.

The Algorithm 2 summarizes this stage. Its output specifies which branch each robot should pursue, but it does not commit the pair to an entire future trajectory. Short-prefix interactions remain predictive, and new observations can change the local choice at an entrance. The selected commitments therefore pass to Gate Admission, which evaluates whether a local exploratory change offers sufficient improvement over legal continuation and is compatible with the partner's action.

Algorithm 2 Return-Aware Joint Commitment Selection

Require: Commitments $\mathscr{X}_t$, graph G_t, robot states, ownership state
Ensure: Selected commitments or home-return commands
for each exploring robot i **do**
  Compute $\widehat{B}_{ik}$ and form $\widehat{\mathscr{A}_i}(t)$.
  if $\widehat{\mathscr{A}_i}(t) = \varnothing$ **then**
    Assign home return to robot i.
  else
    Compute R_{ik} for every $k \in \widehat{\mathscr{A}_i}(t)$.
  end if
end for
if both robots admit a joint assignment **then**
  Form $\widehat{\mathscr{Q}}_t$ using the ownership/relay restriction.
  Evaluate $\mathbf{h}_{k\ell}$ and $\mathscr{J}_{k\ell}$ over this domain.
  Select $(k_1^\star, k_2^\star)$ using Eq. (19).
else if only robot i remains active **then**
  Select $k_i^\star \leftarrow \arg\min_{k \in \widehat{\mathscr{A}_i}(t)} R_{ik}$.
end if
return selected commitments and home-return commands.

Commitment-Guided Gate Admission

A useful branch assignment can still be executed inefficiently. At an entrance, successive local observations may alternately favor different directions, producing unnecessary switching and immediate reversal. Gate Admission therefore asks whether a local change offers enough benefit over legal continuation of the selected intention, rather than simply choosing the largest instantaneous score. Figure 1(c) shows this connection between branch intent and local motion.

At the current anchor $z_i = p_i(t)$, an exploration robot considers:

$$\mathscr{U}_i(z_i) = \{\mathsf{KEEP}\} \cup \{\mathsf{ENTER}(d) \mid d \in \mathscr{D}_i^F(z_i)\} \cup \mathscr{W}_i(z_i), \tag{20}$$

where $\mathscr{D}_i^F$ contains known-free cardinal directions that satisfy token ownership constraints and short-term reverse-suppression constraints. The conditional set $\mathscr{W}_i \subseteq \{\mathsf{WAIT}, \mathsf{RELAY}\}$ is activated by right-of-way or same-branch coordination needs. KEEP follows the assigned prefix, while ENTER fixes its first direction. Reversal is treated as a reverse ENTER, not as a separate global task. A returning robot is fixed to RETURN; other non-exploring robots retain their current continuation.

For a local action u_i, let $\pi_i^{u_i}$ be its simulated prefix, $\delta E_i(u_i)$ its first-step cost, and $d_i(u_i)$ its first nonzero direction, set to zero for a zero-displacement prefix. The local gain and value are

$$\begin{aligned} G_i^{\mathrm{loc}}(u_i) &= \alpha_U \overline{U}_i(u_i) + \alpha_P \Delta_i^{\mathrm{prog}}(u_i) + \alpha_A [d_i(u_i)^\top n_{k_i^\star}]_+, \\ V_i(u_i) &= \eta_G G_i^{\mathrm{loc}}(u_i) - \eta_E \delta E_i(u_i) - \eta_R \widehat{r}_i^{\mathrm{ret}}(u_i) \\ &\quad - \eta_S s_i(u_i) - \eta_V v_i(u_i) - \eta_W \mathbf{1}\{u_i = \mathsf{WAIT}\}, \end{aligned} \tag{21}$$

where $[x]_+ = \max(x, 0)$. Unknown support $\overline{U}_i$ favors informative motion, progress Δ_i^{prog} favors advancing toward $C_{k_i^\star}$, and directional alignment links the action to the entry direction of the selected commitment. These terms favor local adaptation consistent with branch progress. The remaining terms penalize motion, estimated return-margin erosion $\widehat{r}_i^{\mathrm{ret}}$, local intent changes s_i, reversal v_i, and waiting. WAIT has zero gain and motion cost but incurs the idling penalty in the action value. Non-exploring robots contribute no exploration gain.

Let $\mathbf{u} = (u_1, u_2)$. Joint evaluation accounts for the fact that a locally attractive entry may obstruct the partner:

$$\Psi_t(\mathbf{u}) = V_1(u_1) + V_2(u_2) - \mu_O O_t(u_1, u_2) - \mu_C C_H(u_1, u_2), \tag{22}$$

where O_t penalizes local overlap and C_H scores predicted short-horizon vertex and head-on interactions. Pairs violating traversability, token ownership, reverse suppression, or immediate conflict constraints are removed, giving $\mathscr{U}_t^{\mathrm{leg}}$. Immediate legality is thus enforced as a constraint, while later short-prefix interactions influence ranking. The candidate pair is

$$\mathbf{u}^c = \arg\max_{\mathbf{u} \in \mathscr{U}_t^{\mathrm{leg}}} \Psi_t(\mathbf{u}). \tag{23}$$

The reference $\mathbf{u}^{\mathrm{keep}}$ continues the current KEEP or RETURN intentions when they are compatible. If needed, the lower-priority component is replaced by WAIT, RELAY, or RETURN to obtain a legal fallback. An unconditional $(\mathsf{KEEP}, \mathsf{KEEP})$ pair

would be an unsuitable reference when continuing both paths already creates a conflict. For a joint alternative and an individual action, respectively, define

$$\Delta_t(\mathbf{u}) = \Psi_t(\mathbf{u}) - \Psi_t(\mathbf{u}^{\mathrm{keep}}),$$
$$\varepsilon_i(u_i) = \begin{cases} \varepsilon_{\mathrm{rev}}, & v_i(u_i) = 1, \\ \varepsilon_{\mathrm{ent}}, & \text{otherwise}, \end{cases} \qquad \varepsilon_{\mathrm{rev}} > \varepsilon_{\mathrm{ent}}. \tag{24}$$

A switching penalty changes the ranking of candidates but does not determine whether the improvement over continuation is sufficient. The admission threshold makes this comparison explicit, with a higher requirement for reversal. A minimum commitment duration $T_{\min}$ further limits repeated local changes shortly after the last switch t_i^{sw}. For an exploring robot, the preliminary action is

$$\widetilde{u}_i = \begin{cases} \mathsf{RETURN}, & \rho_i(t) \le 0, \\ \mathsf{KEEP}, & \Delta_t(\mathbf{u}^c) \le \varepsilon_i(u_i^c), \\ \mathsf{KEEP}, & t - t_i^{\mathrm{sw}} < T_{\min}, \\ \mathsf{RELAY}, & \mathrm{relay}_i(\mathbf{u}^c) = 1, \\ u_i^c, & \text{otherwise}. \end{cases} \tag{25}$$

The relay condition identifies a permitted same-branch case in which the partner has the stronger action. Returners remain fixed to RETURN throughout; neither the improvement test nor the commitment lock can reopen exploration for them.

The preliminary pair $\widetilde{\mathbf{u}}$ may differ from $\mathbf{u}^c$. For example, one robot can remain locked to its current prefix while the other accepts entry. A legality or improvement test on the original candidate cannot certify this new combination. The threshold for its changed, non-return components is

$$\varepsilon(\mathbf{u}) = \max_{\substack{i \in \{1,2\}:\ u_i \neq u_i^{\mathrm{keep}} \\ u_i \neq \mathsf{RETURN}}} \varepsilon_i(u_i), \qquad \max \varnothing = 0. \tag{26}$$

The assembled pair is checked for membership in $\mathscr{U}_t^{\mathrm{leg}}$ and for its own advantage $\Delta_t(\widetilde{\mathbf{u}})$. If the pair is illegal or its advantage does not exceed $\varepsilon(\widetilde{\mathbf{u}})$, the accepted or safety-priority component is fixed and the best compatible partner action is selected. If no such alternative exists, the legal continuation is used. The continuation itself is the default and need not improve on itself. Writing this joint revalidation and repair as Close_t gives

$$\overline{\mathbf{u}} = \mathrm{Close}_t(\widetilde{\mathbf{u}}, \mathbf{u}^{\mathrm{keep}}) \in \mathscr{U}_t^{\mathrm{leg}}. \tag{27}$$

Thus, individual acceptance rules cannot bypass the joint compatibility check. The ownership term in assignment, the local switching cost, and this admission test act at different levels: they regulate task handover, action ranking, and whether a proposed exploratory change is admitted.

The planner converts $\overline{u}_i$ and $k_i^\star$ into a candidate prefix π_i^c; WAIT and RELAY produce zero-displacement prefixes. Let q_i^+ be the first resulting pose. The executable-step return margin is

$$\rho_i^+(\pi_i^c) = e_i(t) - \delta E_i(\pi_i^c) - \kappa_s D_t(q_i^+, h_i) - e_0. \tag{28}$$

This pre-execution check accounts for both the budget spent immediately and the new starting pose for home return. In particular,

$$\rho_i^+(\pi_i^c) - \rho_i(t) = -\delta E_i(\pi_i^c) - \kappa_s \left[D_t(q_i^+, h_i) - D_t(q_i(t), h_i) \right]. \tag{29}$$

An exploratory step can therefore erode the reserve even when its own movement cost is small. Unlike the predicted service workload used in selection, Eq. (28) is evaluated for the first step that is about to be taken.

The initial guarded prefix is

$$\pi_i^{\mathrm{g}} = \begin{cases} \pi_i^{\mathrm{home}}, & \overline{u}_i = \mathsf{RETURN} \text{ or } \rho_i(t) \le 0 \\ & \quad \text{or } \rho_i^+(\pi_i^c) \le 0, \\ \pi_i^c, & \text{otherwise}, \end{cases} \tag{30}$$

where π_i^{home} is a minimum-cost known-free home-return path. A return override clears the assignment and makes the return mode absorbing. Generated-prefix conflicts are repaired under return priority, and the margins of the first repaired steps are

checked again before obtaining executable prefixes $\pi_i^\star$. This recheck applies the return requirement to the repaired motion. Joint intention revalidation handles compatibility after local acceptance; prefix repair handles the concrete motion produced afterward.

For an unchanged first step on a minimum-cost home-return path, $D_t(q_i(t),h_i) = \delta E_i + D_t(q_i^+,h_i)$, and Eq. (29) reduces to

$$\rho_i^+(\pi_i^{\text{home}}) - \rho_i(t) = (\kappa_s - 1)\delta E_i \geq 0. \tag{31}$$

Such a return step does not reduce the modeled reserve. An exploratory step is executed only after satisfying the positive-margin check. Since new observations do not remove verified free connections under the stated model, they cannot increase the minimum return cost from the resulting pose. Starting from a return-feasible state, these checks preserve the modeled return requirement step by step; Eq. (31) is not a certificate for an arbitrary motion substituted during conflict repair.

Algorithm 3 completes the decision process. After the first steps are executed, the updated poses, budgets, and observations determine the next set of commitments. Joint selection and local admission enumerate at most $O(K^2)$ commitment pairs and $O(R^2)$ action pairs, respectively, for at most R actions per robot. These are pair-evaluation counts; graph construction, path search, and prefix evaluation contribute additional computation.

Algorithm 3 Commitment-Guided Gate Admission

Require: Selected commitments, G_t, robot states, ownership and commitment states
Ensure: Executable prefixes $(\pi_1^\star, \pi_2^\star)$

Build local action sets, fixing returners and other inactive robots to their continuation actions.
Construct $\mathscr{U}_t^{\text{leg}}$ and the legal continuation $\mathbf{u}^{\text{keep}}$.
Evaluate Ψ_t and select $\mathbf{u}^c$ using Eq. (23).
for each exploring robot i **do**
    Apply Eq. (25); retain fixed actions for non-exploring robots.
end for
if $\widetilde{\mathbf{u}} = \mathbf{u}^{\text{keep}}$ **then**
    Retain the legal continuation.
else if $\widetilde{\mathbf{u}}$ is illegal or $\Delta_t(\widetilde{\mathbf{u}}) \leq \varepsilon(\widetilde{\mathbf{u}})$ **then**
    Fix the accepted or safety-priority component and select its best compatible partner action.
    Use $\mathbf{u}^{\text{keep}}$ if no compatible alternative exists.
else
    Retain $\widetilde{\mathbf{u}}$.
end if
Denote the revalidated pair by $\overline{\mathbf{u}}$.
for $i = 1,2$ **do**
    Generate π_i^c and evaluate its post-step margin using Eq. (28).
    Form the guarded prefix π_i^g using Eq. (30).
    if home return overrides exploration **then**
        Clear the assignment and retain absorbing return mode.
    end if
end for
Repair remaining prefix conflicts under return priority.
Recheck the repaired first-step margins; reject unsafe exploratory motion under the return override.
return the resulting executable prefixes $(\pi_1^\star, \pi_2^\star)$.

The main model and decision parameters are summarized in Table 1.

Experimental Evaluation

The experiments examine whether BIC-GA completes the exploration–return task efficiently and how its components affect robot behavior. External comparisons assess overall performance, while component tests examine commitment stability, cooperative allocation, local coordination, and return-budget management.

Experimental Setup and Evaluation Protocol

The experiments use two robots on static, initially unknown 20×20 grid maps. The robots move on a four-connected grid, obtain obstacle-occluded observations within a radius of three cells, and update a shared belief map at each simulation step. Turn-aware A* plans paths through known free cells. Unless otherwise stated, translation has unit cost, turn-cost weight is

Table 1. Main parameters used in BIC-GA and their meanings.

Parameter	Meaning
Motion and return constraints	
η	Required exploration coverage.
λ_θ	Weight of turning effort relative to unit-cost translation.
κ_s	Safety inflation factor for return cost and predicted loop burden.
e_0	Reserved motion budget in task screening and return checks.
Directed commitment construction	
w_c	Narrow-passage width threshold used in anchor selection.
$\alpha_m, \alpha_C, \alpha_\sigma, \alpha_w$	Gain weights for unknown support, frontier size, branching, and entrance width.
c_ρ, c_w, c_σ	Entry-difficulty weights for obstacle density, narrowness, and branching.
$a_m, a_C, a_\rho, a_w, a_\sigma$	Service-workload weights for unknown support, frontier size, obstacle density, narrowness, and limited branching.
Joint commitment selection	
λ_B, λ_Q	Weights of predicted loop burden and entry difficulty in the individual selection cost.
λ_h	Interaction weights for separation, overlap, short-prefix conflict, and workload imbalance.
H	Prediction horizon for short-prefix interactions.
Gate Admission	
$\alpha_U, \alpha_P, \alpha_A$	Local-gain weights for unknown support, branch progress, and entry-direction alignment.
η_G, η_E, η_R	Weights of local gain, first-step motion cost, and estimated return-margin erosion.
η_S, η_V, η_W	Penalty weights for local intent switching, immediate reversal, and waiting.
μ_O, μ_C	Joint-action penalty weights for local overlap and short-prefix conflict.
$\varepsilon_{\text{ent}}, \varepsilon_{\text{rev}}$	Improvement thresholds relative to legal continuation for non-reverse actions and immediate reversals, respectively.
$T_{\min}$	Minimum commitment duration used by the local switching lock.

$\lambda_\theta = 0.20$, initial motion budgets are $(160, 160)$, coverage target is $\eta = 0.80$, return safety factor is $\kappa_s = 1.15$, and simulation limit is 1400 steps. The coverage is computed over Ω_{eval} as defined in the methodology.

Each method–setting combination uses 40 matched seeds, numbered 1–40. External comparisons cover random-fragmented maps at nominal fragmentation settings $c = 0.20$, 0.25, and 0.30, giving 120 conditions per method. Component tests use seed-dependent Y-junction, dead-end-nest, parallel-corridor, long-return-loop, and single-bottleneck layouts. Budget and safety-factor variations are introduced in the corresponding return tests. The experiment groups contain 2560 run records; the return-budget and safety-factor groups share one operating condition and are analyzed separately. The additional performance analyzes use these existing records.

BIC-GA is compared to Greedy Frontier, Burgard-style Coordinated Utility, MinPos-inspired allocation, and coExplore-inspired ranking[1,2,12,14]. Their core selection principles are adapted to the common grid setting. For each matched condition, BIC-GA and the baselines use the same underlying map and home positions, with common sensing, budgets, stopping limits, path planner, motion-cost accounting, execution-conflict handling, and return safeguard. Component tests remove or replace selected BIC-GA mechanisms, with the changes described alongside the corresponding results.

Target success requires reaching the prescribed coverage, return success requires both robots to arrive home without depletion, and mission success requires both. Success rates and final coverage include all runs. Motion-cost summaries use successful missions, and paired efficiency comparisons retain conditions in which both methods complete the task. The primary measures are team path length, turning effort, and composite motion cost J, accumulated during exploration and return. Coverage-normalized efficiency, completion steps, commitment switching, duplicate allocation, conflict-repair demand, and return reserve provide further information about the source of performance differences. These measures are defined in the result subsections where they are used.

Continuous outcomes are summarized by mean and standard deviation. Paired continuous and binary comparisons use two-sided Wilcoxon signed-rank and exact McNemar tests, respectively, with Holm correction within each comparison family at $\alpha = 0.05$. Sample sizes and confidence-interval procedures are stated with the relevant analyses.

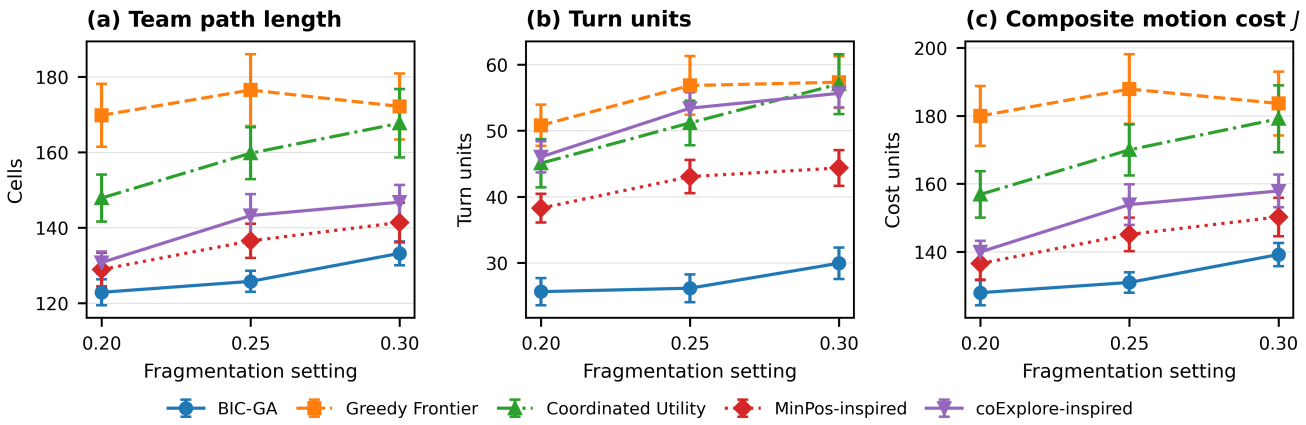


Figure 2. External comparison at fragmentation settings $c = 0.20$, 0.25, and 0.30. Panels report team path length, turn units, and composite motion cost. Markers denote means over successful missions; error bars show approximate 95% confidence intervals computed as the mean ± 1.96 standard errors.

External Baseline Comparison

Figure 2 compares BIC-GA with four frontier-allocation baselines in the three fragmentation settings, and Table 2 summarizes their pooled performance. BIC-GA, Greedy Frontier, and Coordinated Utility complete all 120 exploration–return missions. MinPos-inspired and coExplore-inspired each miss the coverage target once at the highest fragmentation setting, although both robots return successfully in every run. Mean final coverage ranges from 81.90% to 82.67%, and BIC-GA reaches 82.51%. Under the common return safeguard, the main distinction in this comparison is the motion required to complete the exploration–return task rather than return success alone.

Table 2. External comparison over 120 conditions per method. Coverage is reported over all runs; motion metrics use successful missions. Continuous values are mean $\pm$ standard deviation.

Method	Mission/Return (%)	Coverage (%)	Team path (grid steps)	Turn units	Cost J
BIC-GA	100/100	82.51 ± 1.88	$\mathbf{127.27 \pm 10.94}$	$\mathbf{27.26 \pm 7.25}$	$\mathbf{132.72 \pm 11.79}$
Greedy Frontier	100/100	82.39 ± 1.62	172.82 ± 28.57	54.98 ± 12.69	183.81 ± 30.62
Coordinated Utility	100/100	82.35 ± 1.68	158.43 ± 25.32	51.10 ± 13.35	168.65 ± 27.63
MinPos-inspired	99.2/100	82.67 ± 2.10	135.55 ± 16.05	41.87 ± 8.28	143.92 ± 17.25
coExplore-inspired	99.2/100	81.90 ± 5.03	140.18 ± 16.05	51.66 ± 8.36	150.52 ± 17.21

Motion summaries contain 120 successful runs for BIC-GA, Greedy Frontier, and Coordinated Utility, and 119 for each of the other methods. Relative improvements use conditions in which both compared methods succeed.

BIC-GA achieves the lowest mean path length, turning effort, and composite cost among the tested methods. In jointly successful conditions, the reductions relative to Greedy Frontier are 26.4%, 50.4% and 27.8%, respectively. Against Coordinated Utility, they are 19.7%, 46.7%, and 21.3%. Although Coordinated Utility improves on Greedy Frontier, BIC-GA further reduces both travel and turning, with larger relative gains in turning effort.

MinPos-inspired and coExplore-inspired paths provide closer comparisons in path length. Against MinPos-inspired, BIC-GA reduces team path length by 6.2%, turning effort by 34.9% and J by 7.8%. Against coExplore-inspired, the reductions are 9.3%, 47.3%, and 11.9%. Each comparison uses 119 jointly successful conditions. The paired differences in all three motion metrics remain significant against every baseline after Holm correction, with adjusted $p < 0.001$. The motion-efficiency advantage therefore remains evident against these closer baselines, particularly in turning effort.

The same pattern is observed at individual fragmentation levels in Figure 2. BIC-GA's mean path length increases from 122.85 to 125.75 and 133.20 grid steps as c increases from 0.20 to 0.30, while mean turn units increase from 25.65 to 26.18 and 29.95. Its path, turn, and cost means remain below those of the external baselines in all three settings. The pooled advantage is thus consistent across the tested fragmentation levels rather than being produced by a single setting.

Figure 3 examines whether the advantages persist after accounting for achieved coverage and path length. For each

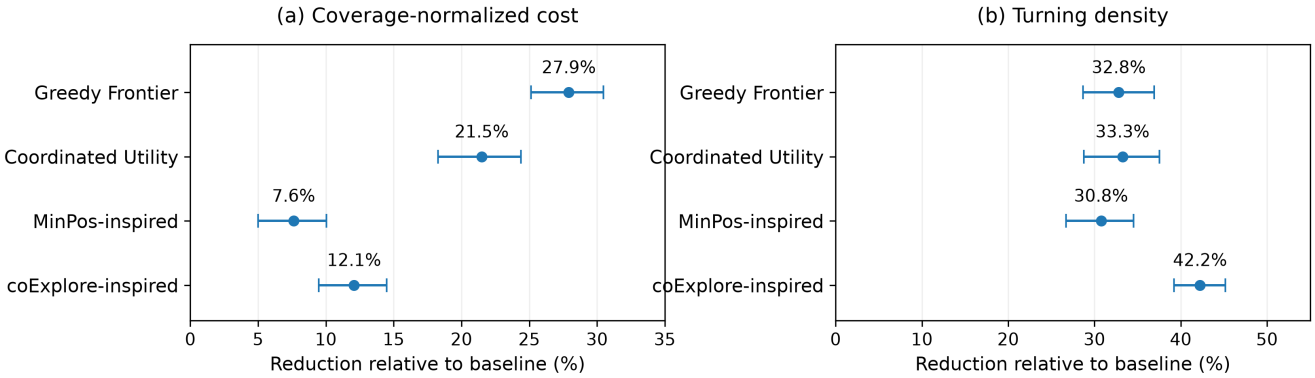


Figure 3. Paired improvements achieved by BIC-GA in (a) coverage-normalized composite cost and (b) turning density. Each row names the comparison baseline; its point represents BIC-GA's percentage reduction, not the baseline's absolute value. Positive values favor BIC-GA. Comparisons use 120 jointly successful conditions against Greedy Frontier and Coordinated Utility, and 119 against each remaining baseline. Error bars are 95% percentile intervals from 6000 seed-level bootstrap resamples, retaining matched fragmentation conditions within each seed.

successful run r, the coverage-normalized cost and turning density are

$$C_{J,r} = \frac{J_r}{100\Gamma_{\mathrm{f},r}}, \qquad D_{\mathrm{turn},r} = \frac{100T_r}{L_r}, \tag{32}$$

where $\Gamma_{\mathrm{f},r}$ is the final coverage as a fraction, L_r is the length of the team path, and T_r is the turning effort. The measures give the cost per percentage point of coverage and turn units per 100 translations, respectively. Ratios are calculated per run before averaging. For either measure X, the plotted reduction is $100(1-\overline{X}_{\text{BIC-GA}}/\overline{X}_{\text{baseline}})$, with both means evaluated under the same jointly successful conditions.

In Figure 3(a), BIC-GA reduces the coverage-normalized cost by 27.9%, 21.5%, 7.6% and 12.1% relative to Greedy Frontier, Coordinated Utility, MinPos-inspired and coExplore-inspired, respectively. All four bootstrap intervals remain above zero. Across each method's successful runs, the mean C_J is 1.610 for BIC-GA, compared to 1.741 for MinPos-inspired and 1.829 for coExplore-inspired. The cost advantage therefore persists after normalization for achieved coverage.

Figure 3(b) shows turning-density reductions of 32.8%, 33.3%, 30.8% and 42.2% against the same four baselines. Across each method's successful runs, BIC-GA averages 21.32 turn units per 100 translations, compared with 30.83 for MinPos-inspired and 36.86 for coExplore-inspired. The lower total turning effort is therefore accompanied by a lower turning effort per unit distance, rather than resulting only from shorter paths. This is consistent with maintaining branch intent while regulating local directional changes; component tests examine these behaviors directly.

The paired outcomes show how widely the gains occur. Of the 119 conditions shared with coExplore-inspired, BIC-GA has a shorter path in 89, an equal path in two, and a longer path in 28. It uses fewer turn units in 117 conditions and has a lower composite cost in 100. Against MinPos-inspired, it uses fewer turn units in 111 of 119 conditions. These results show that the turning benefit occurs across most matched conditions, rather than being confined to a few favorable cases.

In the same coExplore-inspired comparisons, the mean reduction of 17.88 cost units in J comprises 12.99 units from shorter travel and 4.89 units from weighted turning. Turning therefore contributes 27.3% of the gain at $\lambda_\theta = 0.20$. This decomposition shows that shorter travel and reduced turning contribute to the lower composite motion cost.

Exploration Progress and Mission Completion

The preceding cost comparison does not directly show how quickly the exploration task progresses, since the robots may move concurrently or wait during coordination. Figure 4 therefore examines coverage attainment and mission completion using the detailed external records at $c = 0.20$ and 0.30, for which both step measures are available, giving 80 runs per method.

Let $\tau_{\eta,r}$ be the first recorded step in which run r reaches the coverage target, and let $\tau_{\mathrm{m},r}$ be its completion step when both coverage and return requirements are satisfied. The corresponding empirical attainment fractions are

$$F_a(t) = \frac{1}{N}\sum_{r=1}^{N} \mathbf{1}\{S_{a,r} = 1,\ \tau_{a,r} \le t\}, \qquad a \in \{\eta, \mathrm{m}\}, \tag{33}$$

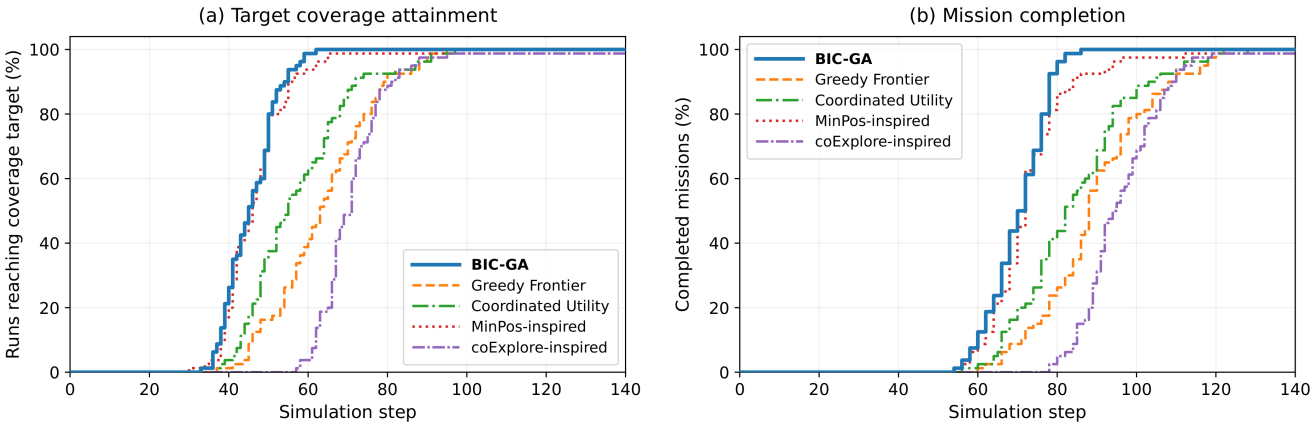


Figure 4. Empirical attainment curves for (a) reaching the coverage target $\eta = 0.80$ and (b) completing coverage and both home returns. Each curve includes all 80 runs of a method at fragmentation settings $c = 0.20$ and 0.30. Unsuccessful runs remain in the denominator but contribute no attainment event. The vertical axes show the percentage of runs attaining each outcome, not instantaneous spatial coverage.

where $S_{a,r}$ indicates whether the outcome a is achieved and $N = 80$. A higher curve means that more runs have attained the outcome by that step. Unsuccessful runs remain in the denominator. The horizontal axis represents the simulation steps, not the computation time or the physical elapsed time.

BIC-GA reaches the coverage target in an average of 45.60 steps and completes the mission in 70.18 steps over its 80 runs. Compared to Greedy Frontier, the mean target-attainment and mission-completion steps decrease by 28.4% and 21.5%, respectively; relative to Coordinated Utility, they decrease by 20.5% and 16.8%. All three methods complete every mission, so these comparisons use all 80 matched conditions. Their curves show that BIC-GA's lower motion burden is accompanied by earlier task completion.

Against coExplore-inspired, the 79 jointly successful conditions give mean target-attainment steps of 45.48 for BIC-GA and 70.63 for the baseline, and mission-completion steps of 70.13 and 95.65. The corresponding reductions are 35.6% and 26.7%, with BIC-GA reaching both outcomes earlier in every matched condition. Step reductions against Greedy Frontier, Coordinated Utility, and coExplore-inspired remain significant after Holm correction, with adjusted $p < 0.001$. The coExplore-inspired curves end at 98.75% because its unsuccessful run remains in the denominator.

MinPos-inspired is closer to BIC-GA in the completion steps. In their 79 jointly successful conditions, BIC-GA reduces the mean target-attainment and mission-completion steps by 1.5% and 3.3%, respectively, but neither difference is significant after correction. The advantage relative to this baseline is therefore clearer in the motion-cost and turning-density results of the preceding external comparison than in the task-completion steps.

The post-target task tail, $\tau_{m,r} - \tau_{\eta,r}$, further identifies where savings occur in the completion-step. It averages 24.65 steps for BIC-GA and 25.01 steps for coExplore-inspired on the same 79 pairs, with no significant difference. Most of the completion-step reduction therefore occurs before target attainment, rather than through a shorter post-target tail. This tail is not an individual robot's full return duration, since return may begin before the team reaches the coverage threshold.

Commitment Stability and Coordination

Figure 5 examines how task representation, joint selection, and local admission rules contribute to the performance reported above. Representation and allocation tests use the Y-junction, dead-end-nest, parallel-corridor and long-return loop layouts at nominal complexity $c = 0.30$, with default budgets and coverage target. Point-Frontier replaces directed commitments with frontier-cell targets, while Independent Commitment retains the representation but selects tasks separately. Each variant uses 40 matched seeds per topology, giving 160 runs per variant; all of these missions are completed successfully.

In Figure 5(a), BIC-GA reduces the mean semantic commitment-switch rate from 92.09 to 5.98 changes per 100 simulation steps relative to Point-Frontier. For directed commitments, semantic identity follows the entrance–direction signature rather than a transient component index; rates are calculated for each run before averaging. Lower switching is observed in all four topologies, while the mean turning effort decreases by 8.1%. The mean team path lengths remain close, with 93.40 grid steps for BIC-GA and 92.73 for Point-Frontier. The representation's clearest benefit is, therefore, more persistent exploration intent and lower turning effort at a comparable travel distance, consistent with separating execution-target updates from changes in branch intent.

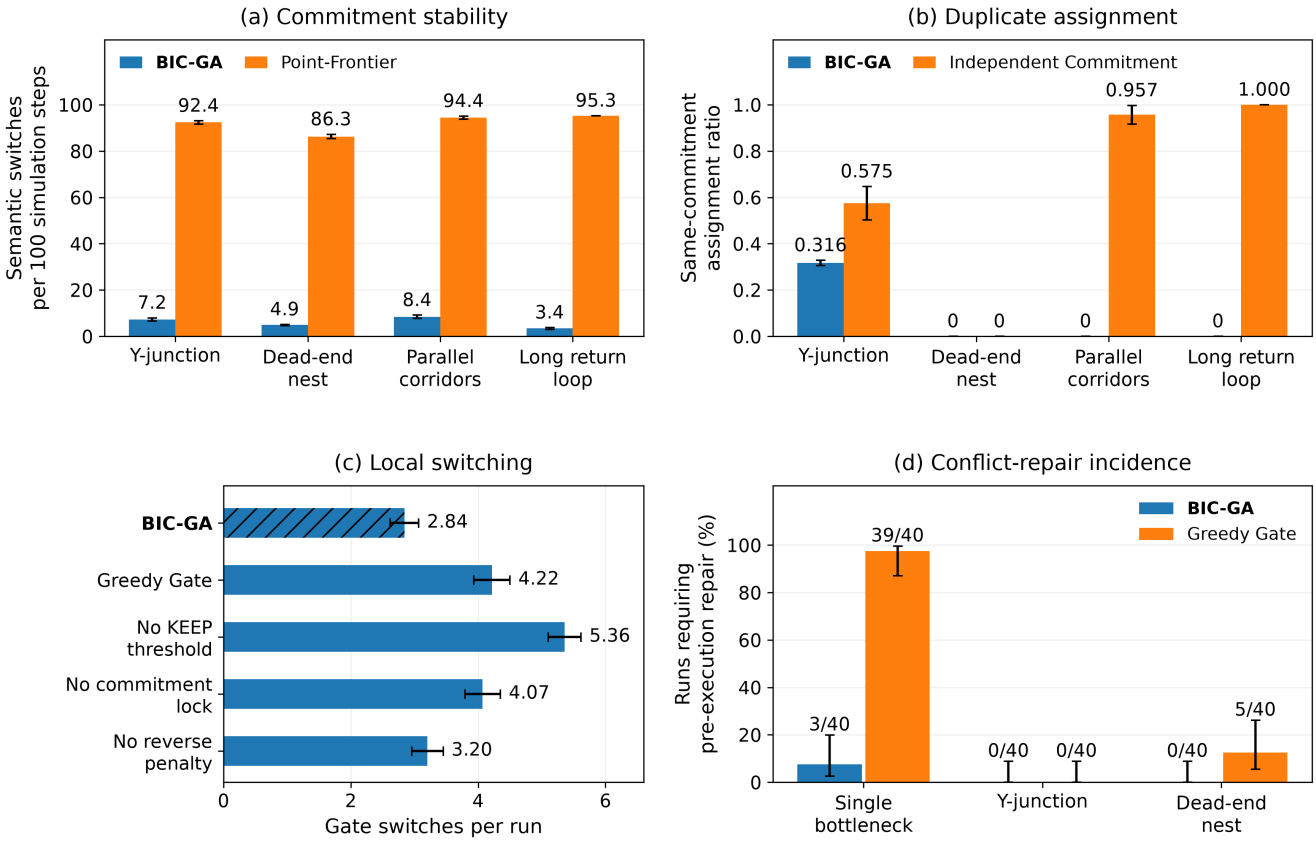


Figure 5. Commitment stability and coordination: (a) semantic commitment-switch rate; (b) same-commitment assignment ratio; (c) executed Gate switches; and (d) runs requiring pre-execution conflict repair. Panels (a,b,d) use 40 matched seeds per method and topology; panel (c) pools 120 runs per variant across three Gate layouts. Error bars are approximate 95% intervals using 1.96 standard errors in (a,b), percentile intervals from 6000 seed-level bootstrap resamples in (c), and Wilson intervals in (d). Zero labels denote observed zero values.

Joint selection uses these task objects to coordinate work between robots. The same-commitment assignment ratio measures the fraction of dual-assignment updates that select the same tracked commitment, rather than geometric trajectory overlap. BIC-GA reduces its pooled mean from 0.633 under Independent Commitment to 0.079 in Figure 5(b). The mean team path length decreases from 122.55 to 93.40 grid steps, a reduction of 23.8%, and the composite motion cost decreases by 23.7%. Reduced duplicate assignment is therefore accompanied by lower travel and composite cost, connecting the allocation statistic to mission-level efficiency.

The topology-specific results clarify where this benefit is most pronounced. In parallel corridors, the mean ratio of the same-commitment decreases from 0.957 to zero, and the length of the path decreases by 19.4%. In the long-return-loop layout, the ratio falls from 1.000 to zero, and the path length decreases by 45.8%, from 161.60 to 87.65 steps of the grid. In the dead-end nest layout, both methods already record zero duplicate assignments and the same mean path length. This pattern associates the largest allocation gains with layouts in which independent decisions repeatedly direct robots toward the same work.

Gate tests examine execution after assignment on single-bottleneck, Y-junction and dead-end-nest layouts at $c = 0.30$, with $\eta = 0.90$ and an 1800-step limit. Each variant has 120 successful runs, with high-level selection, budgets, return safeguards, and conflict repair that are otherwise retained. Figure 5(c) shows that BIC-GA reduces the number of Gate switches executed from 4.22 to 2.84 per run relative to Greedy Gate, a reduction of 32.6%. Relative to full BIC-GA, removing the KEEP-relative admission threshold increases switching by 88.6%, immediate reversals by 56.9% and turning effort by 14.0%, linking frequent local changes to additional movement burden. Removing the commitment lock increases switching by 43.1%, while removing reverse regularization increases executed immediate reversals by 20.1%. These effects support complementary roles for admission threshold, commitment lock, and reverse regularization in maintaining directional continuity.

Local coordination is also reflected in the need for corrective intervention. Figure 5(d) reports the fraction of runs requiring at least one pre-execution conflict repair, not collisions that have already occurred. In the single-bottleneck layout, the incidence falls from 39/40 runs under Greedy Gate to 3/40 runs under BIC-GA, or from 97.5% to 7.5%. Neither method requires repair in the Y-junction tests, while the dead-end nest layout produces five affected runs under Greedy Gate and none under BIC-GA. The largest reduction therefore occurs at the shared bottleneck. Together, the results of the components associate persistent task intent with a lower turning effort, joint allocation with less duplicate assignment and travel, and local admission with fewer switches and fewer runs requiring conflict repair. The final comparison examines whether this coordinated exploration also retains a sufficient return reserve under tighter budgets.

Return-Budget Management and Mission Reliability

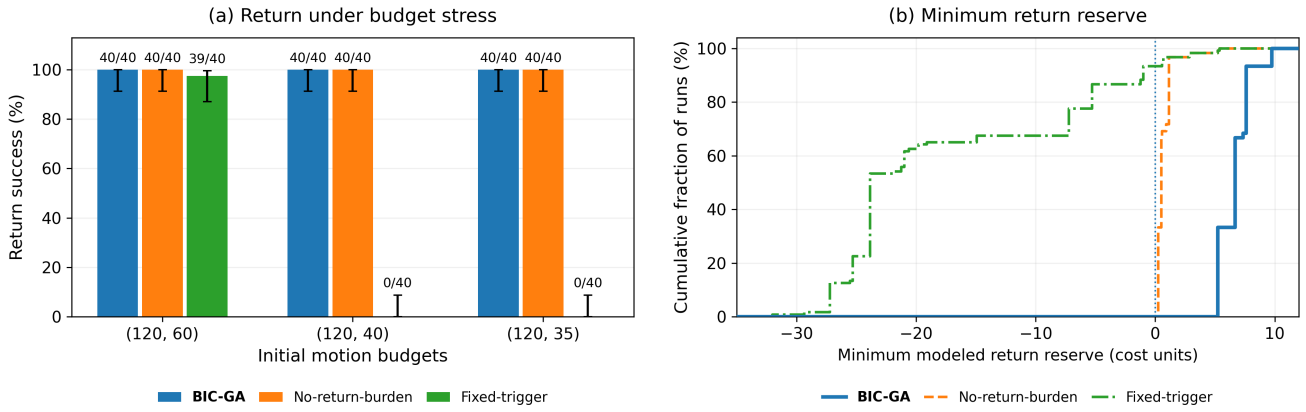


Figure 6. Return-budget performance: (a) return success for each initial budget pair, with counts out of 40 runs and 95% Wilson intervals; (b) empirical distributions of the minimum logged return reserve per run, pooled over the three budget pairs. Each curve includes all 120 runs of a strategy, including unsuccessful missions. The vertical dotted line marks zero modeled reserve.

Figure 6 examines whether coordinated exploration retains enough budget for return as resources become more restricted. The tests use the long-return-loop layout at $c = 0.35$, with $\eta = 0.95$, $\kappa_s = 1.15$ and a 1600-step limit. The initial budgets are $(120, 60)$, $(120, 40)$, and $(120, 35)$, with 40 seeds matched per strategy and budget pair. No-return-burden removes the return contribution from assignment while retaining dynamic monitoring and the execution guard. The Fixed-trigger instead uses a remaining-budget trigger without predictive return coupling or post-step return filtering.

BIC-GA completes 119 of the 120 missions and returns both robots successfully in every run. No-return-burden completes 114 missions, also with 120 successful returns. The Fixed-trigger reaches the coverage target in all 120 runs but returns successfully in only 39. These successes occur in 39 of the 40 runs with budgets $(120, 60)$; neither tighter budget pair produces a successful return under this strategy. The contrast shows that coverage attainment alone is insufficient for mission completion and highlights the value of considering return requirements throughout exploration.

The reserve distribution in Figure 6(b) distinguishes the two strategies that retain execution guarding. Each observation is the minimum modeled return reserve over both robots and all logged steps of a run. Negative values indicate a violation of the modeled return requirement, not necessarily budget depletion. BIC-GA has a mean minimum reserve of 6.65 cost units, compared with 0.71 for No-return-burden. The latter reaches a minimum of 0.24, with 86 of 120 runs falling to one cost unit or less; every BIC-GA run has a logged minimum of at least 5.24. Thus, including the predictive return burden is associated with more completed missions and a larger reserve, rather than conserving resources simply by abandoning exploration. The execution guard checks the next movement, while the return-aware selection anticipates the burden of the task being pursued.

The safety-factor tests fix the budgets at $(120, 40)$ and vary κ_s over 1.00, 1.15 and 1.30, with the remaining conditions unchanged. BIC-GA achieves 40 successful returns in each setting without recorded return-margin violations. The setting $\kappa_s = 1.15$ is shared with the budget comparison and is not counted as additional independent evidence. Together, budget-pressure comparisons and safety-factor tests support combining predictive task selection with execution-level return checks: BIC-GA maintains high mission completion under restricted budgets while preserving the modeled return requirement in the tested conditions.

Discussion and Conclusion

This paper presents BIC-GA for return-feasible dual-robot target-coverage exploration in fragmented unknown environments. The method connects directed branch commitments, return-aware joint selection, and local Gate Admission within one decision process. Under jointly successful conditions, BIC-GA reduces the mean team path length by 6.2–26.4%, turning effort by 34.9–50.4%, and composite motion cost by 7.8–27.8% relative to the four frontier-allocation baselines tested under a common return safeguard. The cost advantage persists after coverage normalization, while the lower turning density indicates that the reduced turning effort is not simply a consequence of shorter paths. The completion step analysis complements these findings by showing earlier task completion against several baselines, while the advantage over MinPos-inspired is more evident in motion efficiency than in completion steps.

The component results explain why task representation, allocation, and execution need to be considered together. A directed commitment separates branch intent from execution-target updates, supporting more persistent exploration intent and lower turning effort at a comparable travel distance. Joint selection uses this task structure to reduce duplicate assignment, with the largest motion gains occurring in layouts where independent choices compete for the same work. Gate Admission then limits unnecessary directional changes and reduces the incidence of pre-execution conflict repair, particularly in the single-bottleneck layout. These effects occur at different stages of exploration: a stable intention provides continuity, but its value also depends on how the work is divided and how the resulting actions interact. The evidence therefore supports the complementary roles for these components within the combined decision process.

The budget-pressure results also distinguish task anticipation from execution safeguarding. The predicted service workload is useful for selecting commitments, but cannot specify the exact cost of exploring an unknown branch. The execution guard instead checks the modeled post-step return margin before motion. In tested conditions, including the return burden in selection retains larger modeled reserves and completes more missions than removing that contribution while keeping the guard. This suggests that predictive selection improves the use of task-level budgets beyond the final return check, with larger reserves accompanied by more completed missions rather than reduced task attainment.

The present formulation and evaluation concern two robots on static grids with shared mapping and a composite translation–turn cost. Pairwise selection and Gate evaluation require graph searches and prefix calculations in addition to candidate enumeration; reported task steps describe mission progress rather than planning latency and do not establish real-time performance. The return-feasibility argument relies on deterministic costs and verified free connections that remain traversable. Therefore, it establishes a modeled return requirement under these assumptions, not a physical battery guaranty under localization, mapping, or cost-model errors.

Overall, BIC-GA provides an explicit connection between where a robot intends to explore, how the team evaluates and allocates that work, and which local actions are admitted under modeled return requirements. The results support this connection as a way to improve exploration efficiency through shorter travel and lower turning effort while preserving modeled return feasibility. Future work will investigate communication and localization uncertainty, coordination beyond robot pairs, and physical deployment with nonholonomic motion and more realistic energy models.

Data availability

Data supporting the findings of this study are available at https://doi.org/10.5281/zenodo.23098648. Additional simulation records and derived results not included in the public repository are available from the corresponding author upon reasonable request.

References

1. Yamauchi, B. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 146–151, DOI: 10.1109/CIRA.1997.613851 (1997).
2. Burgard, W., Moors, M., Stachniss, C. & Schneider, F. E. Coordinated multi-robot exploration. *IEEE Transactions on Robotics* **21**, 376–386, DOI: 10.1109/TRO.2004.839232 (2005).
3. Zlot, R., Stentz, A., Dias, M. B. & Thayer, S. Multi-robot exploration controlled by a market economy. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, vol. 3, 3016–3023, DOI: 10.1109/ROBOT.2002.1013690 (2002).
4. Zhou, B., Zhang, Y., Chen, X. & Shen, S. FUEL: Fast UAV exploration using incremental frontier structure and hierarchical planning. *IEEE Robotics Autom. Lett.* **6**, 779–786, DOI: 10.1109/LRA.2021.3051563 (2021).
5. Cao, C., Zhu, H., Choset, H. & Zhang, J. TARE: A hierarchical framework for efficiently exploring complex 3d environments. In *Robotics: Science and Systems*, DOI: 10.15607/RSS.2021.XVII.018 (2021).

6. Mei, Y., Lu, Y.-H., Lee, C. S. G. & Hu, Y. C. Energy-efficient mobile robot exploration. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 505–511, DOI: 10.1109/ROBOT.2006.1641761 (2006).

7. Best, G., Garg, R., Keller, J., Hollinger, G. A. & Scherer, S. Multi-robot, multi-sensor exploration of multifarious environments with full mission aerial autonomy. *The Int. J. Robotics Res.* **43**, 485–512, DOI: 10.1177/02783649231203342 (2024).

8. Sharon, G., Stern, R., Felner, A. & Sturtevant, N. R. Conflict-based search for optimal multi-agent pathfinding. *Artif. Intell.* **219**, 40–66, DOI: 10.1016/j.artint.2014.11.006 (2015).

9. Yamauchi, B. Frontier-based exploration using multiple robots. In *Proceedings of the Second International Conference on Autonomous Agents*, 47–53 (1998).

10. Simmons, R. *et al.* Coordination for multi-robot exploration and mapping. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, 852–858 (2000).

11. Dias, M. B., Zlot, R., Kalra, N. & Stentz, A. Market-based multirobot coordination: A survey and analysis. *Proc. IEEE* **94**, 1257–1270, DOI: 10.1109/JPROC.2006.876939 (2006).

12. Bautin, A., Simonin, O. & Charpillet, F. Minpos: A novel frontier allocation algorithm for multi-robot exploration. In *International conference on intelligent robotics and applications*, 496–508 (Springer, 2012).

13. Colares, R. G. & Chaimowicz, L. The next frontier: Combining information gain and distance cost for decentralized multi-robot exploration. In *Proceedings of the 31st annual ACM symposium on applied computing*, 268–274 (2016).

14. Scheler, I. & Dietrich, R. coExplore: Combining multiple rankings for multi-robot exploration (2023). 2303.17459.

15. Kuipers, B. & Byun, Y.-T. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics Auton. Syst.* **8**, 47–63, DOI: 10.1016/0921-8890(91)90014-C (1991).

16. Thrun, S. Learning metric-topological maps for indoor mobile robot navigation. *Artif. Intell.* **99**, 21–71, DOI: 10.1016/S0004-3702(97)00078-7 (1998).

17. Choset, H. & Burdick, J. Sensor-based exploration: The hierarchical generalized voronoi graph. *The Int. J. Robotics Res.* **19**, 96–125, DOI: 10.1177/02783640022066770 (2000).

18. Wurm, K. M., Stachniss, C. & Burgard, W. Coordinated multi-robot exploration using a segmentation of the environment. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1160–1165, DOI: 10.1109/IROS.2008.4650734 (2008).

19. Dang, T. *et al.* Graph-based subterranean exploration path planning using aerial and legged robots. *J. Field Robotics* **37**, 1363–1388, DOI: 10.1002/rob.21993 (2020).

20. Zhou, B., Xu, H. & Shen, S. Racer: Rapid collaborative exploration with a decentralized multi-uav system. *IEEE Transactions on Robotics* **39**, 1816–1835 (2023).

21. Kim, S., Corah, M., Keller, J., Best, G. & Scherer, S. Multi-robot multi-room exploration with geometric cue extraction and circular decomposition. *IEEE Robotics Autom. Lett.* **9**, 1190–1197, DOI: 10.1109/LRA.2023.3342553 (2024).

22. Niroui, F., Zhang, K., Kashino, Z. & Nejat, G. Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments. *IEEE Robotics Autom. Lett.* **4**, 610–617, DOI: 10.1109/LRA.2019.2891991 (2019).

23. Hu, J., Niu, H., Carrasco, J., Lennox, B. & Arvin, F. Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Veh. Technol.* **69**, 14413–14423, DOI: 10.1109/TVT.2020.3034800 (2020).

24. Zhang, H., Cheng, J., Zhang, L., Li, Y. & Zhang, W. H2GNN: Hierarchical-hops graph neural networks for multi-robot exploration in unknown environments. *IEEE Robotics Autom. Lett.* **7**, 3435–3442, DOI: 10.1109/LRA.2022.3146912 (2022).

25. Tan, A. H., Bejarano, F. P., Zhu, Y., Ren, R. & Nejat, G. Deep reinforcement learning for decentralized multi-robot exploration with macro actions. *IEEE Robotics Autom. Lett.* **8**, 272–279, DOI: 10.1109/LRA.2022.3224667 (2023).

26. Morales, J., Martínez, J. L., Mandow, A., García-Cerezo, A. J. & Pedraza, S. Power consumption modeling of skid-steer tracked mobile robots on rigid terrain. *IEEE Transactions on Robotics* **25**, 1098–1108, DOI: 10.1109/TRO.2009.2026502 (2009).

27. Tiwari, K., Xiao, X. & Chong, N. Y. Estimating achievable range of ground robots operating on single battery discharge for operational efficacy amelioration. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3991–3998, DOI: 10.1109/IROS.2018.8594450 (2018).

28. Datsko, D., Nekovar, F., Penicka, R. & Saska, M. Energy-aware multi-UAV coverage mission planning with optimal speed of flight. *IEEE Robotics Autom. Lett.* **9**, 2893–2900, DOI: 10.1109/LRA.2024.3358581 (2024).

29. Karumanchi, S., Rokaha, B., Schperberg, A. & Vinod, A. P. Energy-constrained multi-robot exploration for autonomous map building. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9154–9161, DOI: 10.1109/IROS60139.2025.11247332 (2025).

30. Silver, D. Cooperative pathfinding. In *Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 117–122 (2005).

31. Phillips, M. & Likhachev, M. SIPP: Safe interval path planning for dynamic environments. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, 5628–5635, DOI: 10.1109/ICRA.2011.5980306 (2011).

32. van den Berg, J., Guy, S. J., Lin, M. & Manocha, D. Reciprocal n-body collision avoidance. In *Robotics Research*, 3–19, DOI: 10.1007/978-3-642-19457-3_1 (Springer, 2011).

33. Stern, R. *et al.* Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Proceedings of the International Symposium on Combinatorial Search*, vol. 10, 151–158 (2019).

Funding

This research did not receive any funding.

Author contributions statement

Y.Xue: conceived the study, developed the methodology, collected the data, and wrote the original draft. C K.Tan: supervised the study and reviewed and edited the manuscript. All authors reviewed the manuscript.

Competing interests

The authors declare that they have no competing interests.