IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 July 2025, accepted 9 August 2025, date of publication 18 August 2025, date of current version 25 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3599858

RESEARCH ARTICLE

CBS-HT: Prioritized Safe Interval Path-Planning Algorithm for Heterogeneous Agricultural Robot Team

YUSEUNG JO[1,2] **AND HYOUNG IL SON**[1,2,3], **(Senior Member, IEEE)**
[1]Department of Convergence Biosystems Engineering, Chonnam National University, Gwangju 61186, Republic of Korea
[2]Interdisciplinary Program in IT-Bio Convergence System, Chonnam National University, Gwangju 61186, Republic of Korea
[3]Research Center for Biological Cybernetics, Chonnam National University, Gwangju 61186, Republic of Korea

Corresponding author: Hyoung Il Son (hison@jnu.ac.kr)

This work was supported in part by the Ministry of Education and National Research Foundation of Korea through the ''Convergence and Open Sharing System'' Project; and in part by the Technology Innovation Program (Development of AI Autonomous Manufacturing System for the Production Process of High-Difficulty Curved Ship Blocks) funded by the Ministry of Trade, Industry and Energy (MOTIE), South Korea, under Grant RS-2024-00511865.

ABSTRACT Existing multi-agent path-finding (MAPF) methods either assume homogeneous robots or require inter-robot communication, limiting their deployment in large-scale agricultural settings. We present conflict-based search for heterogeneous tasks (CBS-HT), a fixed-priority path planner for unpiloted ground vehicles (UGVs) performing monitoring, harvesting, and transportation. CBS-HT assigns initial priorities according to robot type and inter-workspace movement (e.g., field, sorting, warehouse) and integrates this priority scheme directly into a CBS-inspired high-level search, constraining only the lower-priority agent at each conflict. By design, CBS-HT relaxes strict completeness and optimality guarantees in exchange for practical real-time performance on large robot teams, reducing search complexity while preserving collision-free coordination. Three evaluation tiers–simulation, lab-scale, and orchard—test scalability, mission-level efficiency, and real-world robustness. Compared with an uncoordinated baseline, CBS-HT shortens high-priority robots' travel distance by up to 19.1 m (13.5 %) and mission time by 28.4 s (13.2 %), while maintaining a 100 % collision-free success rate for tested instances. These results demonstrate that priority-aware safe-interval planning can deliver practical, communication-free coordination of heterogeneous UGV fleets in commercial orchards and is adaptable to diverse multi-robot agricultural operations.

INDEX TERMS Agricultural robot, heterogeneous robot team, safe interval path planning, multi-agent pathfinding, priority.

I. INTRODUCTION

Efficient coordination of heterogeneous agricultural robot teams is essential for enhancing productivity in large-scale farming operations. In such environments, multiple tasks such as harvesting, transportation, and sorting occur concurrently within shared workspaces, where narrow passages, asymmetric task durations, and suboptimal local path planning can lead to congestion and conflicts. These challenges highlight the need for designing task-specific robots using various robotic platforms, such as unpiloted ground vehicles (UGVs) [1], unpiloted aerial vehicles (UAVs) [2], [3], mobile manipulators [4], and aerial manipulators [5]. Among these, UGVs, as reliable platforms, have been extensively studied, and UGV-based agricultural robots with autonomous driving capabilities have reached the commercialization stage.

Existing research has predominantly concentrated on evaluating the performance of individual robots and advancing the capabilities of singular robotic entities. The next imperative step lies in the development of multi-robot systems (MRSs) to enhance overall task efficiency [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Alexander Kocian.

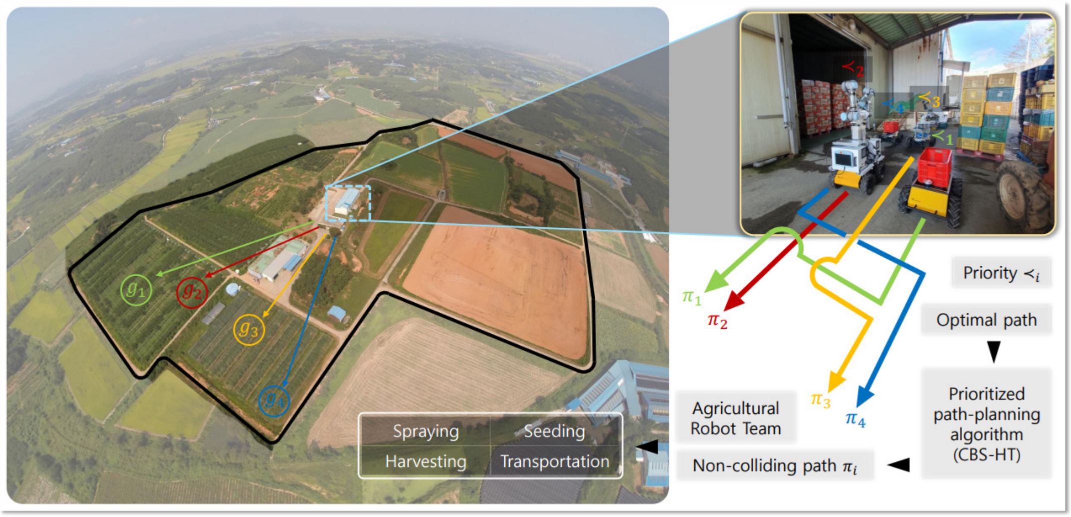


FIGURE 1. The prioritized path planning algorithm for heterogeneous agricultural robot team to maximize work efficiency.

In agricultural automation, incorporating the collaborative nature of human workers striving for efficiency in diverse agricultural tasks is crucial. For labor-intensive activities such as diagnosis, seeding, harvesting, and transportation, heterogeneous teams of human workers emerge, each dedicated to specific tasks like harvesting or transportation. Their collaboration is vital for optimizing task execution [7]. Given these aspects, deploying heterogeneous robot teams is an attractive solution [8], [9]. However, in practical automation systems, where the heterogeneous robot team must operate concurrently within a shared workspace, congestion may occur in confined passages or due to suboptimal local path planning. As shown in Fig. 1, strategically deploying heterogeneous MRSs is essential for enhancing operational efficiency [7].

The control and deployment of heterogeneous MRSs require careful investigation due to the significant increase in system complexity compared to individual robots [10], [11]. Robot teams can be regulated using centralized or decentralized strategies. Decentralized control offers several advantages, such as leveraging local information and scalability, but encounters limitations related to competition and local minima in task optimization within the agricultural context [12], [13]. In contrast, centralized control ensures consistency in system behavior and facilitates globally optimized decision-making. These benefits enable robot teams to maximize work efficiency, considering constraints such as optimal path planning and obstacle avoidance. Notably, centralized control relies on a central controller, which may pose fewer challenges in restricted workspaces, such as those encountered in agricultural scenarios. Agricultural settings, characterized by clearly defined workspaces and effective communication solutions—both satellite-based [14], [15] and local-network-based [16], [17]—have been extensively explored for the application of centralized control design [18].

However, decision-making aimed at maximizing the task efficiency of heterogeneous agricultural robots remains limited. Addressing this challenge is crucial for the efficient deployment of heterogeneous MRSs. To this end, the multi-agent pathfinding (MAPF) problem has been introduced. The MAPF problem has proven effective in planning optimal paths for multiple agents concurrently in dynamic environments. Moreover, it facilitates the configuration of MRSs across diverse scenarios without necessitating inter-robot communication, avoiding an increase in system complexity. Additionally, the design of a task-specific cost function for the MAPF solution allows for the planning of a path that is suitable for the key perspectives of the application (i.e., optimality, computational efficiency, and path continuity) [19], [20], [21].

This paper proposes conflict-based search for heterogeneous tasks (CBS-HT), a prioritized path-planning algorithm designed for heterogeneous agricultural robot teams performing diverse tasks such as monitoring, seeding, harvesting, and transportation. By integrating task-specific priorities into the MAPF framework, CBS-HT enables efficient coordination of robots with different roles, minimizing congestion and optimizing task execution order in shared workspaces [22]. Unlike conventional MAPF approaches that assume homogeneous agents, the proposed algorithm considers the unique operational constraints of agricultural environments, allowing high-priority tasks to be completed without unnecessary delays. Furthermore, CBS-HT is designed to function without inter-robot communication, enhancing its scalability and applicability to real-world autonomous farming systems.

A. RELATED WORK

MAPF is a fundamental problem in MRS, where the goal is to compute collision-free paths for multiple agents while optimizing specific cost functions. These cost functions typically consider factors such as optimality, computational efficiency, and path continuity [23], [24], [25]. Various algorithms have been developed to address this problem, each with distinct strengths and limitations.

A*-based global search is a well-established approach, which offers optimal solutions in static environments but suffers from high computational complexity as the number of agents increases. To improve scalability, decoupled approaches such as hierarchical cooperative A* (HCA*) have been introduced, where agents are assigned fixed priorities, and paths are sequentially computed [23]. Alternatively, optimal reciprocal collision avoidance (ORCA) adopts a decentralized approach by adjusting the velocities of agents in real time to avoid collisions, making it well-suited for dynamic environments [25].

Driven by hardware advancements, machine-learning (ML)-based MAPF solvers have gained significant attention over the past decade [26]. In particular, improvements in high-performance GPUs and parallel computing technologies have greatly enhanced the practicality of ML-based approaches for solving complex multi-agent path planning problems. Learning-based approaches such as PRIMAL and CLE enable adaptive decision-making by learning navigation policies in simulated environments [24], [27].

These traditional approaches have been widely applied in various domains, including warehouse automation, autonomous vehicle coordination, and swarm robotics. However, agricultural tasks present unique challenges due to their sequential and interdependent nature. Unlike industrial environments, where tasks can often be performed independently, agricultural workflows require specific tasks to be completed before subsequent ones can begin.

To address these domain-specific challenges, several extensions of MAPF algorithms have been proposed to enhance planning efficiency and flexibility under real-world constraints. Among them, conflict-based search with priorities (CBSw/P) and priority-based search (PBS) extend the original CBS framework by dynamically assigning and updating agent priorities during the planning process [28]. These methods are effective in reducing search space and resolving conflicts adaptively.

However, such approaches typically focus on homogeneous agents and assume uniform task roles. This assumption limits their applicability in domains like agriculture, where agent capabilities and task priorities often differ. For such domains, planning frameworks that incorporate fixed, role-aware priority structures may offer improved coordination and interpretability, especially when task dependencies are predefined.

Moreover, agricultural environments are inherently unstructured and dynamic, with frequent occurrences of task failures, robot malfunctions, and environmental changes. To address these challenges, adaptive mechanisms, which can respond to real-time situational changes, must be incorporated in MRS path planning. Efforts must be dedicated to developing MAPF algorithms capable of dynamically adjusting their planning strategies to enhance robustness and efficiency in unstructured agricultural environments.

The Safe Interval Path Planning (SIPP) is particularly noteworthy for its ability to incorporate temporal constraints into path computation [29]. While SIPP was originally proposed for individual agents, its formulation based on time-safe occupancy intervals makes it highly effective in constrained and dynamic environments. In multi-agent settings, SIPP is often adopted as a low-level planner to ensure temporal feasibility during sequential plan generation. Its integration into higher-level MAPF frameworks contributes to more efficient and collision-free coordination.

For example, a prioritized variant of SIPP tailored for continuous-time multi-agent pathfinding on 2D roadmaps is proposed [30]. While their method is effective for continuous-time planning on 2D roadmaps, it operates under the assumption of fixed agent priorities and relies solely on prioritized SIPP calls without hierarchical conflict resolution. In contrast, domains like agriculture require scalable coordination frameworks that can handle task-specific priorities, heterogeneous capabilities, and temporally structured workflows.

B. CONTRIBUTION

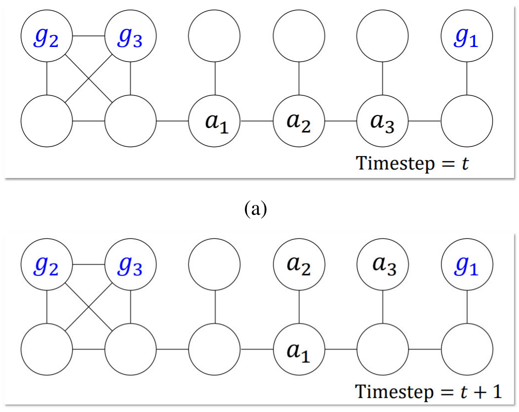
The contributions and novelty of this study can be summarized as follows:

1) We propose CBS-HT, a prioritized path-planning algorithm that integrates task-specific priorities into MAPF, improving coordination efficiency among heterogeneous agricultural robots while minimizing congestion.
2) We introduce a priority-aware safe interval mechanism for collision-free path planning in continuous environments, enabling deployment without inter-robot communication.
3) We validate CBS-HT's scalability and applicability through simulations and field experiments, demonstrating its effectiveness in real-world agricultural tasks.

II. PRIORITIZED SAFE INTERVAL PATH-PLANNING ALGORITHM

A. PROBLEM DESCRIPTION

To overcome the practical challenges associated with heterogeneous agricultural task automation, we propose and implement a heterogeneous agricultural robot team. Agricultural environments, characterized by high density and numerous obstacles, often restrict the mobility and maneuverability of multiple robots due to narrow pathways. Therefore, efficiently planning the paths for a large number of robots is crucial in such settings. The complexity of this task is compounded by the heterogeneous nature of agricultural environments, each exhibiting distinct features. Thus, for efficient path planning, it is essential to generalize the agricultural operational environment and corresponding tasks. This generalization can be accomplished by



(a)

(b)

FIGURE 2. Three agents travel to their destination *g* on undirected graph *G*: (a) in time step *t*, (b) in time step $t + 1$.

discretizing agricultural environments and representing them as simplified undirected graphs. The heterogeneous agricultural robotics team, designed for efficient agricultural automation, must travel paths within this graph while avoiding collisions. Moreover, every robot must perceive others as dynamic obstacles and adapt to unpredictable dynamic factors (i.e., human operators). During this process, robots must traverse discretized environments in discrete time steps. For simplicity, all grid transitions, including horizontal, vertical, and diagonal, are assumed to consume one discrete time step (Δt), regardless of their geometric length.

Notably, the challenge lies not only in optimizing individual robot paths but also in maximizing global work efficiency. Agricultural tasks typically exhibit sequential characteristics (e.g., sampling precedes diagnosis and harvesting precedes transportation). Thus, assigning priority is an effective strategy to address this challenge. Consequently, this study introduces an algorithm to determine the priorities of multiple UGVs and plan robot paths based on these priorities. For example, as shown in Fig. 2, three agents (agent$_1$, agent$_2$, and agent$_3$) are required to move toward their goals (g$_1$, g$_2$, and g$_3$, respectively). In the absence of coordination behavior, if all robots attempt to navigate optimal paths, a collision may occur between agent$_1$ and agent$_2$ at time-step $= t + 1$. As depicted in Fig. 2, the priorities $\prec_i$ help control the behavior of agents and efficiently prevent conflicts and congestion.

Traditionally, this problem is addressed using MAPF frameworks [31]. MAPF algorithms are versatile, can accommodate various tasks, and facilitate flexible task distributions. They are capable of handling both static and dynamic obstacles, enhancing adaptability. In MAPF, the path planning of each agent is conducted in parallel, with conflict resolution strategies employed when collisions occur. Optimal search MAPF variants guarantee completeness and cost optimality. However, their runtime increases rapidly as the team size grows, and practical variants usually relax one or both guarantees to achieve real time performance.

The MAPF problem is a generalized extension of the single-agent pathfinding problem [22]. In the single-agent scenario, algorithms such as A* search for paths between two vertices s_1 and g_1 in an environment represented as a mapped graph G. Similarly, the MAPF problem, involving k agents, can be represented as the tuple $< G, s, g >$. These entities can be defined as follows:

$$G = (V, E), \tag{1}$$

$$A = \left\{\text{agent}_1, \cdots, \text{agent}_k\right\}, \tag{2}$$

$$s, g : [1, \cdots, k] \to V. \tag{3}$$

Here, V represents a vertex, and E represents an edge. The environment is mapped onto a graph G, where vertices can be occupied by agents. s denotes the initial positions of the agents, and g represents their respective goal positions. The MAPF problem aims to find a set A of collision-free paths π_k for k agents.

Let $\pi_i = \{v_i^0, v_i^1, \ldots, v_i^{T_i}\}$ denote the discrete-time path of agent a_i, where each $v_i^t \in \mathcal{V}$ represents the vertex occupied at time-step t.

$$\pi_i = \{v_i^0, v_i^1, \ldots, v_i^{T_i}\}, \quad v_i^t \in \mathcal{V}, \quad \forall t \in \{0, \ldots, T_i\} \tag{4}$$

The set of all collision-free paths is defined as:

$$P = \{\pi_1, \pi_2, \ldots, \pi_k\}. \tag{5}$$

Each path satisfies the goal condition:

$$\pi_i[T_i] = g(i), \quad \forall i \in \{1, \ldots, k\}. \tag{6}$$

Throughout the manuscript, $\pi_i[t]$ always denotes the vertex location of agent a_i at discrete time-step t. All cost and conflict computations in the algorithm refer to spatial positions derived from vertex-based paths π_i.

This problem involves finding a set of collision-free paths P for agents in an environment mapped onto a graph $G(V, E)$. Here, π_i represents a path of agent i, consisting of a set of actions a_n of agent i. $\pi_i[t]$ denotes the position of agent i at discrete time-step t, $g(i)$ denotes the goal position of agent i, with $i, j \in \{1, \cdots, k\}$. As illustrated in Fig. 3, collisions between agents are defined based on overlapping vertex positions at the same or adjacent time-steps. Since each $\pi_i[t]$ denotes the spatial location of agent a_i at time-step t, the following types of conflicts are defined [31], [32]:

$$\pi_i\left[t\right] = \pi_j\left[t\right], \tag{7}$$

$$\begin{cases} \pi_i\left[t\right] = \pi_j\left[t\right], \\ \pi_i\left[t+1\right] = \pi_j\left[t+1\right]. \end{cases} \tag{8}$$

$$\pi_i\left[t+1\right] = \pi_j\left[t\right], \tag{9}$$

$$\begin{cases} \pi_i\left[t\right] = \pi_j\left[t+1\right], \\ \pi_i\left[t+1\right] = \pi_j\left[t\right]. \end{cases} \tag{10}$$

The MAPF problem is defined by a task-specific cost function [31]. Conventional scenarios typically involve

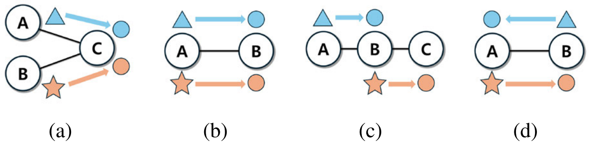


FIGURE 3. Four types of collisions in the discretized MAPF problem: (a) Vertex conflict; Equation (7), (b) Edge conflict; Equation (8), (c) Following conflict; Equation (9), (d) Swapping conflict; Equation (10).

minimizinge either the total sum (Equation (11)), referred to as the sum of costs (SoC) [33], or the maximum cost (Equation (12)), referred to as makespan [34], associated with the agents [35].

$$J(x) = \text{argmin}(\sum_{i \in A} |\pi_i|), \tag{11}$$

$$J(x) = \text{argmin}(\max_{i \in A} |\pi_i|). \tag{12}$$

Given that this study aims to optimize efficiency in the context of overall task performance, minimizing the total travel distance is rational. Consequently, the SoC cost function is used.

B. DESIGN OF PATH-PLANNING ALGORITHM

Agricultural tasks, such as monitoring, spraying, harvesting, and transportation, necessitate collaborative work. Agricultural tasks in a specific period can be divided into several sub-tasks (e.g., harvesting, transportation, and sorting). Accordingly, the workspace can be divided into sub-workspaces. This divisibility enables the cooperation of workers. Similarly, a heterogeneous agricultural robot team operates in their respective sub-workspaces. When managing such diverse tasks, prioritized planning emerges as an effective approach for addressing MAPF problems [36]. This planning strategy involves assigning priorities to agents during the path-planning process, guiding the MRS to navigate paths that minimize conflicts, from agents with higher priorities to those with lower priorities. The ultimate goal is to minimize conflicts between agents and seek efficient paths.

The proposed algorithm draws inspiration from the conflict-based search (CBS) algorithm, a widely used conventional MAPF algorithm [37]. CBS seeks a conflict-free path by conducting a global search for each agent while considering constraints $C = (a_i, a_j, v, t)$. This approach calculates collision-free paths by utilizing a constraint tree based on the optimal path of a single robot. Notably, this approach does not revolve around specific behaviors but discovers paths through global exploration. This characteristic facilitates algorithmic improvements, such as diversifying search strategies or exploring limited nodes.

The proposed algorithm leverages these advantages, relying on a search-based approach [38]. Unlike conventional CBS, CBS-HT does not branch the constraint tree on both agents in a conflict. Instead, it selectively imposes constraints only on the lower-priority agent, effectively integrating priority-based resolution directly into the high-level search process. This structural deviation from CBS enables more efficient resolution in heterogeneous task settings.

Algorithm 1 CBS-HT: Conflict-Based Search for Heterogeneous Tasks

```
Root.constraint ← ∅
for $i = 0$ to $N$ do
    allocate $v_i$ to each workspace
end for
Root.solution ← find individual paths by low-level()
Root.cost ← sum of individual costs in Root.solution
Insert Root into OPEN // initially OPEN is empty
while OPEN is not empty do
    $P$ ← best node from OPEN // lowest cost
    Validate paths in $P$ until a conflict occurs
    if $P$ exhibits no conflict then
        return P.solution // Goal node found
    end if
    $C$ ← first conflict $(a_i, a_j, v, t)$ in $P$
    $\prec_{main}$ ← ComparePriority$(a_i, a_j, \prec_{main}, P)$
    Identify agent $a_{low}$ with lower priority from updated $\prec_{main}$
    $A$ ← new node
    A.constraint ← P.constraint ∪ $\{(a_{low}, v, t)\}$
    A.solution ← recompute paths given constraints in A.constraint
    A.cost ← sum of individual costs in A.solution
    if A.cost $< \infty$ then
        Insert $A$ into OPEN
    end if
end while
return No solution found
```

Fig. 4 shows the concept and schematic of the CBS algorithm. a_i and a_j denote agents i and j, respectively. A collision is considered to occur between agents at a specific vertex v and a specific time-step t. In CBS, the optimal solution is derived through a global search by comparing sets of individual optimal paths iteratively. Conversely, CBS-HT modifies this process by assigning priorities to agents, thereby reducing the search space and computational complexity.

Unlike classical Conflict-Based Search (CBS), which generates constraints for both agents involved in a conflict and explores all possible priority orderings, CBS-HT introduces a selective constraint mechanism. It enforces constraints only on the lower-priority agent and resolves conflicts using a fixed or locally-adjusted priority ordering. While this reduces the branching factor and improves runtime efficiency, it sacrifices the completeness guarantees inherent to CBS. Therefore, CBS-HT should be regarded as a hybrid framework that inherits CBS's high-level search structure but adopts a prioritized resolution policy inspired by fixed-priority planning.

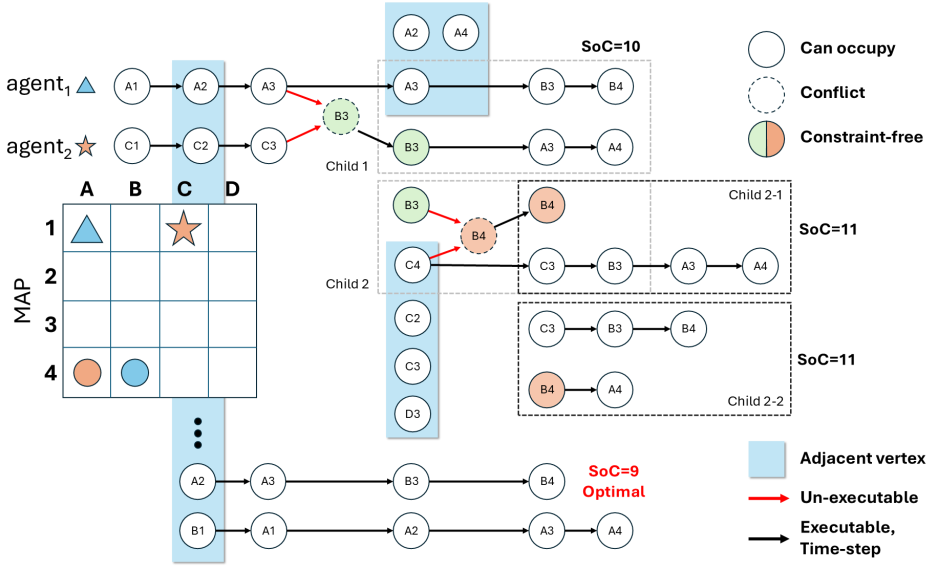


FIGURE 4. Brief sample of a constraint tree in the CBS algorithm.

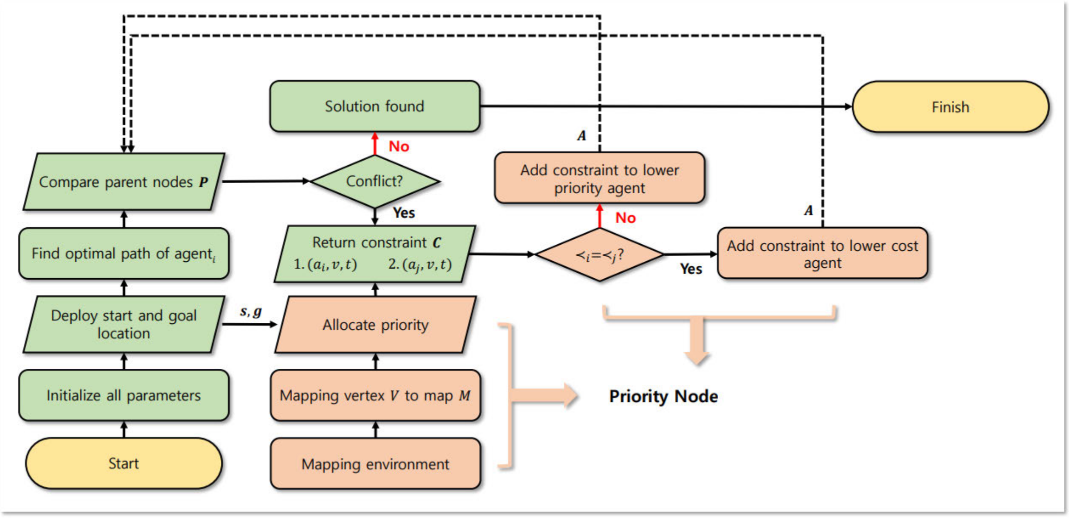


FIGURE 5. Process flow of the proposed prioritized path planning algorithm.

The pseudo-code for CBS-HT is outlined in Algorithm 1, and its procedural flow is depicted in Fig. 5.In contrast to the constraint tree used in CBS (see Fig. 4), our method in Fig. 5 follows a single-branching strategy by enforcing constraints only on the lower-priority agent. This not only reduces the branching factor but also aligns with the task-driven priority schema central to heterogeneous agricultural workflows. Initially, workspaces are allocated to each agent based on predefined task priorities (lines 2–4), followed by computing initial paths for all agents without constraints (line 5). Note that each agent's path is represented as a sequence of vertex positions over discrete time steps, denoted by $\pi_i = \{v_i^0, \ldots, v_i^{T_i}\}$, and stored in solution sets such as *Root.solution* or *A.solution*. The algorithm iteratively explores the best node from the OPEN set (line 9), validating paths until a conflict arises (lines 10–11). When a conflict is identified between two agents at a given vertex and time step (line 13), the priorities of the conflicting agents are compared to identify the lower-priority agent (lines 14–15). Constraints are then selectively imposed only on the lower-priority agent (lines 16–17), prompting path recalculations under the updated constraints (line 18). If a feasible solution emerges, it is reinserted into the OPEN set for further exploration (lines 19–21). If no valid path can be found for the constrained agent, the node is simply discarded without insertion. This effectively prunes the infeasible branch. This behavior aligns with the design choice of enforcing constraints only on lower-priority agents, and has no effect on the algorithm's intended

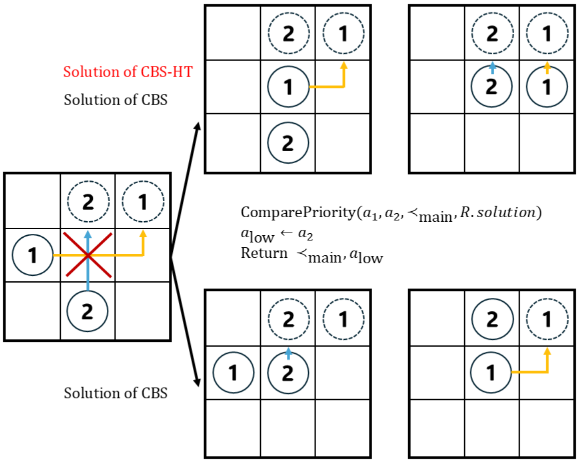

Solution of CBS-HT
Solution of CBS
ComparePriority($a_1, a_2, \prec_{\text{main}}, R.solution$)
$a_{\text{low}} \leftarrow a_2$
Return $\prec_{\text{main}}, a_{\text{low}}$
Solution of CBS
2
1
1
2

FIGURE 6. Simplified concept of the CBS-HT: allocating constraints to low-priority agents.

Algorithm 2 Assign and Compare Priority

```
procedure ComparePriority($a_i$, $a_j$, $\prec_{main}$, $R.solution$)
    if $\prec_{main,i} \neq \prec_{main,j}$ then
        $a_{low}$ ← agent with lower priority (higher numerical $\prec_{main}$ value)
    else
        if $a_i = a_t$ and $a_j = a_h$ then
            $\prec_{main,i} \leftarrow \prec_{main,i} - 1$
            $a_{low} \leftarrow a_i$
        else
            $cost_i \leftarrow R.solution.cost(a_i)$, $cost_j \leftarrow R.solution.cost(a_j)$
            if $cost_i > cost_j$ then
                $\prec_{main,i} \leftarrow \prec_{main,i} - 1$
                $a_{low} \leftarrow a_i$
            else if $cost_j > cost_i$ then
                $\prec_{main,j} \leftarrow \prec_{main,j} - 1$
                $a_{low} \leftarrow a_j$
            else
                (tie-break arbitrarily) $a_{low} \leftarrow a_j$
            end if
        end if
    end if
    return $\prec_{main}$, $a_{low}$
end procedure
```

operation. This is because CBS-HT does not aim to explore all possible constraint combinations.

To clarify the theoretical properties of CBS-HT, we briefly state three guarantees that follow from its structure and planning strategy.

Lemma 1 (Collision-Freedom Guarantee): With a fixed total priority order $\succ$ over the agents, the Safe-Interval scheduler ensures that:

(a) no two agents occupy the same vertex v at the same discrete time-step, and
(b) no two agents traverse the same undirected edge $\langle u, v \rangle$ in opposite directions during the same time-step.

Proof sketch. Planning proceeds in priority order. Each agent is routed only through currently unoccupied safe intervals. By induction on the priority list, vertex overlap is impossible. Because edge traversal requires both endpoints to be free at the same time-step, simultaneous opposite crossings are likewise excluded.

Lemma 2 (Polynomial Search-Complexity Bound): For n agents on a grid of $|V|$ vertices, a CBS-HT planning episode expands at most $\mathcal{O}(n \cdot |V| \log |V|)$ A* nodes.

Proof sketch. The root computes n independent A* plans ($\mathcal{O}(n|V| \log |V|)$). Each conflict triggers at most one replan for the lower-priority agent, which is never revisited after conflict resolution. Total replans are at most n, each with A* cost $\mathcal{O}(|V| \log |V|)$.

Proposition 1 (Approximation-Ratio Bound): Let $C^{\star}$ denote the optimal total path cost, and C_{HT} the cost returned by CBS-HT. Assuming no priority inversions, we obtain the approximation guarantee $C_{\text{HT}} \leq (1 + \mu) \cdot C^{\star}$, where μ is the maximal per-agent detour cost due to waiting.

Proof sketch. Each low-priority agent may wait behind higher-priority agents, incurring bounded detour costs. The worst-case detour μ upper-bounds the total additive cost.

Remark. These results provide safety, polynomial-time planning, and bounded sub-optimality. Completeness remains an open issue under fixed priorities and is discussed in Section V-D.

The CBS algorithm is a representative NP-hard problem [39]. This means that conducting a search for all comparable nodes exponentially increases the complexity of the algorithm as the number of agents and number of vertices that can be traversed increase. Thus, this extensive search may be unsuitable when constructing navigation systems designed for use in dense spaces or for multi-robot teams. Unlike conventional CBS, as depicted in Fig. 6, CBS-HT strategically reduces complexity by preventing constraint additions to high-priority robots, thereby decreasing the absolute travel distance and enhancing overall operational efficiency. This targeted constraint approach is particularly beneficial in dense operational environments and scenarios involving heterogeneous robot teams.

Initial priority assignments aim to optimize overall efficiency; however, the current framework may face challenges when robots share identical priority levels. To address this limitation, CBS-HT incorporates nuanced priority decision strategies, considering robot type and associated costs, as detailed in the subsequent section.

C. PRIORITY DECISION STRATEGY

The determination and adjustment of priorities for heterogeneous tasks are essential within the proposed algorithm. For priority establishment, the agricultural environment is subdivided into discrete workspace units. The environment is consistently divided into three primary workspace categories: (1) warehouse, (2) sorting workspace, and (3) field-processing workspace. Specifically, the warehouse acts as a standby workspace, where robots await further instructions before initiating their tasks [40].

Vertices representing these workspaces can be formally defined as follows:

$$s, g \rightarrow V, V = \left\{v_w, v_s, v_f\right\}. \tag{13}$$

Consequently, movements m between workspaces form a set M consisting of six possible transitions, explicitly defined as follows:

$$m \rightarrow M, M = \left\{v_{ws}, v_{wf}, v_{sf}, v_{sw}, v_{fw}, v_{fs}\right\}. \tag{14}$$

Robots are categorized into two distinct types: harvesting agents (a_h) and transportation agents a_t, formally defined as

$$r \rightarrow R, R = \{\mathrm{a}_h, \mathrm{a}_t\}. \tag{15}$$

The robot heading toward the field processing workspace (e.g., monitoring, spraying, and harvesting robot) is deemed the most crucial for operational efficiency in tasks. Therefore, the priority coefficient k_{init} is defined as 1 for priority determination. When defining the types of robots and their movements for priority determination, the priority matrix for $\prec_{init,1}$ is structured as follows:

$$r = \begin{bmatrix} a_t \\ a_h \end{bmatrix}, m = [v_{wf}, v_{sf}], \tag{16}$$

$$r \times m = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}. \tag{17}$$

The priority determination involves incorporating constraints. Specifically, constraints are introduced to consider both the movements (m) between workspaces and robot types (r). The priority coefficient k_{init}, crucial for operational efficiency in tasks, is set as 1. The priority matrix for $\prec_{init,1}$, which defines the initial priorities, is structured as follows:

$$\prec_{init,1} = k_{init} \times r \times m = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}. \tag{18}$$

Notably, the robot heading toward the warehouse does not considerably affect the efficiency of mission execution, as it is directed to the warehouse for reasons such as battery replacement or unpredictable error. Therefore, the priority coefficient k_{init} is defined as 3 for this case. The third priority, $\prec_{init,3}$, is derived as

$$r = \begin{bmatrix} a_t \\ a_h \end{bmatrix}, m = [v_{sw}, v_{fw}], \tag{19}$$

$$r \times m = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \tag{20}$$

$$\prec_{init,3} = k_{init} \times r \times m = \begin{bmatrix} 3 & 3 \\ 0 & 3 \end{bmatrix}. \tag{21}$$

Similarly, constraints are introduced in matrices r and m to indicate that robots (in this case, harvesting robots) are not associated with the sorting workspace. The three-level priority scale is explicitly defined as follows: priority 1 is assigned to agents with time-sensitive tasks, priority 3 to agents with more flexible schedules, and priority 2 to intermediate roles not falling into either category. The second priority can therefore be unambiguously determined by excluding the first and third priorities in all instances. The pseudo-code for this priority decision strategy is outlined in Algorithm 2. As mentioned, we adjust the priorities considering the harvesting and transportation scenarios to evaluate the algorithm under heterogeneous agricultural task settings. Homogeneous or heterogeneous task combinations (e.g., monitoring and spraying) can also be accommodated by adapting the same adjustment logic.

TABLE 1. Initial priority levels based on robot type and movement intention.

Robot Type	To Field (v_f)	To Sorting (v_s)	To Warehouse (v_w)
Harvest (a_h)	1	–	3
Transport (a_t)	1	2	3

Table 1 summarizes the initial priority levels assigned to agents based on their movement between workspace types. As shown in Table 1, harvesting robots (a_h) are assigned the highest priority (1) when moving toward the field workspace due to the time-sensitive nature of harvesting. They are assigned the lowest priority (3) when returning to the warehouse. Transportation robot (a_t) also receive high priority (1) when headed to the field, but receive medium priority (2) for sorting tasks and low priority (3) when moving toward the warehouse.

In this study, priorities among agents were predefined based on their operational roles; specifically, harvesting robots (a_h) were given higher priority compared to transporting robots (a_t). This prioritization reflects practical considerations in agricultural operations, where timely harvesting is critical. While this predefined prioritization simplifies conflict resolution, determining optimal priorities tailored to specific agricultural applications was considered beyond the scope of the current research.

D. SAFE INTERVAL MRS CONFIGURATION

The proposed algorithm is expected to be deployed in agricultural environments with noise and rough road surfaces. Therefore, we configure a multi-robot navigation system for field evaluation. However, because the proposed algorithm considers a discrete environment and time, it cannot be immediately applied in the continuous real world due to field characteristics (e.g., slope, slip, and sensory noise). To address this, we introduce a drive completion feedback-loop that maintains a safe interval between robots in the configured system. The overview and detailed architecture is depicted in Figs. 7 and 8. Each robot is assigned a unique starting point, destination, and tasks. A LiDAR-based simultaneous localization and mapping (SLAM) and a navigation system

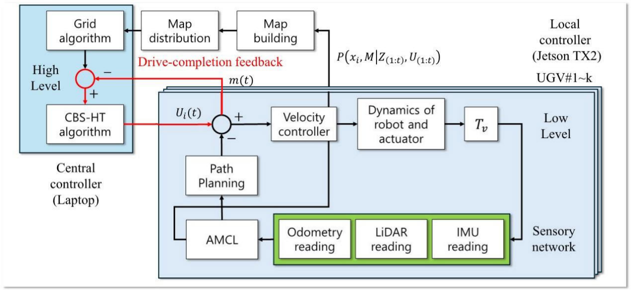


FIGURE 7. **Architecture of the configured multi-robot navigation system. The drive-completion feedback loop (red) can address potential problems in a continuous world.**

are installed on all agents. Multiple individual controllers calculate the optimal path for each robot, and a central controller then synthesizes these paths into a collision-free path that all robots can follow. After the collision-free path is distributed to each robot, the robots do not need to actively recognize and avoid each other.

Thus, in the ideal case, communication between robots is not essential. However, practical challenges exist in the real world. As discussed, the proposed algorithm considers a discretized environment and time. Owing to potential problems (e.g., dynamic environment, sensory noise, and robot slip) that may lead to excessive obstacle avoidance or undesired travel in agricultural environments, the performance of the navigation system may be degraded.

To solve this challenge, we design a safe interval controller so that a multi-robot system can traverse the path calculated through the proposed algorithm in continuous time and environment. One robot, chosen at random, maps the environment and shares the resulting map with all other robots. Concurrently, robot localization and map updates occur and are refined using Bayesian-filter-based beliefs [41].

The control strategy for the four-wheeled mobile robots used in this study can be expressed as follows:

$$U_{(1:t)} = \left(V_{(1:t)}, w_{(1:t)}\right), \tag{22}$$

where $U_{(1:t)}$ represents control inputs, and V and w denote the linear and angular velocities of a robot, respectively. The process of updating the states and maps of the robots can be expressed as follows:

$$P(X_t|Z_{(1:t)}, U_{(1:t)}), \tag{23}$$

$$P(M|Z_{(1:t)}, X_{(1:t)}), \tag{24}$$

where $Z_{(1:t)}$ represents sensor measurements, $X_{(1:t)}$ denotes robot pose estimates, and M denotes the environment map. The Bayesian filter-based beliefs are updated as follows:

$$P(X_t, M|Z_{(1:t)}, U_{(1:t)}). \tag{25}$$

Algorithm 3 Create Grid Map

```
function CREATEGRIDMAP(width, height, cellSize)
    gridMap ← empty 2D array with dimensions (width/cellSize) × (height/cellSize)
    for $x$ from 0 to (width/cellSize) − 1 do
        for $y$ from 0 to (height/cellSize) − 1 do
            gridMap[$x$ × (height/cellSize) + $y$] ← CREATEGRIDCELL($x$ × cellSize, $y$ × cellSize, cellSize)
        end for
    end for
    return gridMap
end function
```

The MAPF problem assumes an environment represented by a grid. Therefore, the generated map must be divided into grids. The pseudo-code of the grid-partitioning algorithm is presented as Algorithm 3. Here, "width" and "height" refer to the horizontal and vertical dimensions of the entire map. The variable "cellSize" indicates the size of each grid cell, determined based on the dimensions of the mobile robots.

As depicted in Fig. 7, within the configured system, no direct communication occurs among the robots. Instead, the robots communicate exclusively with a central control node. This open-loop navigation system exhibits limitations in terms of adapting to environmental changes and coordinating with other robots, especially when operating within a discretized time domain where synchronization-related challenges may arise. To address these limitations, we implement a closed-loop control system using the robot operating system (ROS). This ensures that all robots detect their arrival at waypoints and proceed to the next waypoint. The control input $U_i(t)$ for agent i at time-step t can be expressed as follows:

$$U_i(t) = K(P_i, \theta_i, X_i^{\text{way}}(t+1), m(t)), \tag{26}$$

$$m(t) \in \{0, 1\}. \tag{27}$$

Here, m indicates whether all the robots have reached waypoint $X_i^{\text{way}}(t)$ at time-step t. If m is 1, agent i needs to adjust its position P_i and orientation θ_i to reach the next waypoint $X_i^{\text{way}}(t+1)$. K denotes the control algorithm, with PID control leveraged in this study.

E. COMMUNICATION

A middleware-based communication framework is implemented to enable centralized control of three autonomous robots over an LTE-based IEEE 802.11 wireless network. The system leverages the ROS for distributed message passing, while the network infrastructure is established via an LTE-backed WLAN. This setup ensures real-time bidirectional data exchange with minimal latency and robust connectivity across the robotic unit and central controller.

To establish a secure and reliable communication channel, secure shell (SSH) over TCP/IP is adopted, enabling remote access to each robotic platform. The network layer uses IEEE 802.11n operating in the infrastructure mode, with the LTE gateway functioning as the primary access point. TCPROS

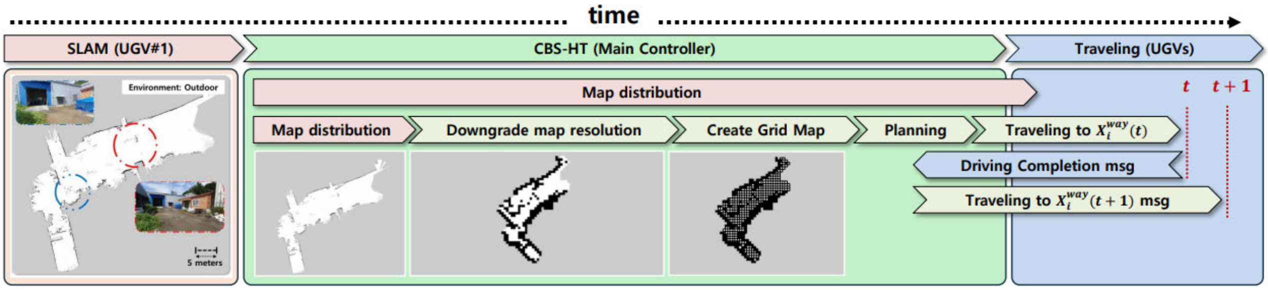


FIGURE 8. Process flow of the configured multi-robot navigation system.

is used for deterministic data transfer, ensuring packet integrity and in-order message delivery, while UDPROS facilitates low-latency communication for time-sensitive control signals.

To enhance network performance and mitigate potential congestion in the LTE-backed IEEE 802.11 infrastructure, adaptive bitrate selection is used. These techniques ensure prioritized data transmission for real-time robotic operations. This architecture provides a scalable, secure, and low-latency communication framework, enabling efficient multi-robot coordination in dynamic and decentralized environments.

III. EXPERIMENTAL SETUP

A. ENVIRONMENTS AND SCENARIOS

Experiments are carefully structured to rigorously evaluate the effectiveness, performance, and robustness of the proposed CBS-HT algorithm across three experimental scenarios: simulation, lab-scale field test, and orchard evaluation. Each experimental setup includes two distinct comparative scenarios: Baseline and CBS-HT. The Baseline scenario employs an uncoordinated MRS, where individual robots independently navigate using the A* algorithm without centralized coordination or inter-robot communication. Consequently, robots consider each other solely as dynamic obstacles identified via onboard sensors. A concise overview of the key characteristics of each scenario is provided below. Detailed protocols for each scenario are presented in the subsequent subsections.

- Scalability validation: A 32 × 32 four-connected grid world with randomized obstacle densities of 10% and 20% and robot teams ranging from 20 to 50 units stress-tests CBS-HT for scalability, measured by runtime and success rate, and evaluates its ability to resolve conflicts.
- Lab-scale test: In a lab-scale outdoor arena, three robots (two harvesters, one transporter) operate under "face" and "crossover" congestion patterns; mission-level efficiency is gauged via travel distance, task-completion time, and collision count.
- Orchard evaluation: Using the three-robot team in a commercial pear orchard with two single-width corridors, we again measure mission-level efficiency to validate real-world applicability under uneven terrain and intermittent GNSS.

1) SCALABILITY VALIDATION

This simulation study has two objectives: (i) quantifying how the proposed CBS-HT planner scales as team size and obstacle density increase, and (ii) estimating its success rate, defined as the percentage of instances where a collision-free plan is returned within a 60 s time budget.

The proposed algorithm assumes a heterogeneous agricultural robot team capable of performing monitoring, harvesting, or transportation tasks by adapting tools or effectors. Specifically, the evaluation focuses on the most challenging scenario involving harvesting and transportation robots. The average agricultural environment size considered is based on realistic conditions. According to the USApple Association's 2024 economic impact report, the average apple farm size was 14.47 acres in 2021. Additionally, the University of Kentucky reports an average apple yield of approximately 300 bushels per acre. Considering our harvesting robot's performance of harvesting a 300 g fruit within 13.5 s, approximately 33 harvesting robots would be required to harvest a typical farm within 72 h [42]. Because deploying such a large fleet physically is impractical, we use simulation to examine scalability and empirical success under representative workloads.

To validate scalability and empirical success, simulations are performed in 32 × 32 four-connected grid worlds with randomized obstacle densities of 10% and 20%. Robot team size ranges from 20 to 50 in increments of ten. Each run is limited to 60 s; instances that exceed this limit are counted as failures. For every problem instance, planner runtime and success rate are recorded. Fig. 9 shows representative grids at obstacle densities of (a) 10% and (b) 20% used in the evaluation. In the simulation experiments, both the spatial layout and agent priorities are randomized. Specifically, obstacle fields are procedurally generated using uniform random sampling, and each agent is assigned a random priority index independent of task roles.

2) LAB-SCALE TEST

This experiment targets mission-level efficiency—how quickly and smoothly a small heterogeneous team accomplishes its task under real-world sensing and congestion limits.

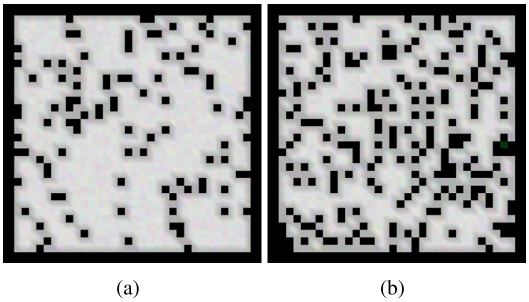


FIGURE 9. Sample instances with obstacle density of (a) 10% and (b) 20%. The black cell represent an unoccupiable state (e.g., obstacle in actual world), and the white gray cell represents an empty state in four-connected grids.

In the lab-scale environment, the team consists of three robots: two harvesting robots and one transport robot. The objective is to simulate coordination among multiple robots, a scenario common in high-density agricultural environments. The features of the outdoor environment are sparser than those of the indoor environment, and uncertain and unpredictable features such as bushes are used for mapping. This may lower the mapping quality. However, we confirm that the localization error is within 0.09 m in this environment. This localization error is smaller than the grid size of 990 mm by 699 mm, as determined by the robot size. Each robot is assigned a fixed and predefined priority to reflect typical field hierarchies.

We establish two scenarios to simulate robot congestion in an agricultural environment: face and crossover scenarios. In the face scenario, three robots face each other, a situation that often occurs between multiple transportation robots, leading to overlapping optimal paths. In the crossover scenario, one robot crosses the path of the other two robots, representative of common agricultural congestion situations for various robots. This scenario can aid the comprehensive evaluation of the coordination performance of the path and robot according to predefined priorities. For both patterns, mission-level metrics are recorded for Baseline and CBS-HT to quantify efficiency gains attributable to coordinated planning.

3) ORCHARD EVALUATION

Unlike the lab-scale test, the orchard test is conducted in a commercialized pear orchard located in Bonghwang-myeon, Naju-si, Jeollanam-do, Republic of Korea. Its primary purpose is to evaluate mission-level efficiency (i.e., travel distance, completion time, and collision count) under real farming conditions.

The proposed system can perform most agricultural tasks with UGVs (including monitoring, seeding, and spraying). However, to evaluate the applicability of the proposed algorithm to agricultural scenarios, we assume the most challenging heterogeneous tasks in a pear orchard: harvesting and transportation. The experiment simulates the complete task process of the agricultural robot team. Initially, all robots wait in the sorting workspace and are then deployed to the first lane of the pear orchard to commence heterogeneous tasks, coordinated through a central controller. The robot team consists of two harvesting robots and one transport robot. The three robots pass through two one-robot-width narrow corridors, potentially leading to path congestion.

B. HARDWARE

The experimental setup involves three mobile robotic platforms: two Clearpath Husky robots and one AgileX Scout robot. Each platform is equipped with a Velodyne VLP-16 LiDAR sensor and an inertial measurement unit (IMU) to implement a standard simultaneous localization and mapping (SLAM) system alongside navigation functionalities. The LiDAR sensor provides precise environmental perception, facilitating accurate mapping and localization, while the IMU enhances the system robustness by providing complementary orientation and acceleration data. The integration of these sensors enables reliable autonomous navigation within the defined experimental workspace. As this hardware configuration and sensor integration follow conventional SLAM methodologies without significant technical innovation, the implementation details are briefly summarized here to establish a clear context for the experimental evaluations.

C. PERFORMANCE METRICS

Unlike discrete-time algorithms, robotic systems inherently operate within a continuous-time framework. Consequently, conventional discrete-time cost metrics (π) alone are insufficient for comprehensive evaluation of the proposed algorithm in robotic implementations. Therefore, additional quantitative performance metrics are utilized.

The performance evaluation primarily employs the cumulative travel distance D_i for each robot agent i:

$$D_i = \int_0^t \sqrt{\frac{\mathrm{d}x_i^2}{\mathrm{d}t} + \frac{\mathrm{d}y_i^2}{\mathrm{d}t}}, dt, \tag{28}$$

where x_i and y_i represent the Cartesian coordinates of robot agent a_i within the operational workspace. As the algorithm incorporates strategic waiting periods to mitigate collisions, distance alone is insufficient as a sole performance indicator. Hence, the average task completion time, denoted by T_i, is also considered:

$$T_i = \frac{1}{N} \sum_{1}^{N} t_i, \tag{29}$$

where N represents the number of experimental repetitions.

Additionally, to ensure robust evaluation aligned with the MAPF problem framework, the collision frequency metric N_c is considered:

$$N_c = \sum_{1}^{N} n_c, \tag{30}$$

with n_c being the number of collisions recorded in each individual experimental instance.

TABLE 2. Results of simulation-based validation.

Algorithm	Runtime	N	π	π_g	A	Algorithm	Runtime	N	π	π_g	A
Proposed	0.02917	50	1168	1082	1.08	CBS algorithm	14.02170	50	1147	1082	1.06
	0.00886	40	841	819	1.03		0.02934	40	837	819	1.02
	0.00293	30	638	622	1.03		0.00597	30	637	622	1.02
	0.00139	20	413	405	1.02		0.00282	20	413	405	1.02

Experiments are systematically repeated five times each for both indoor and outdoor scenarios to validate the consistency and reliability of results.

IV. EXPERIMENTAL RESULTS

A. SCALABILITY VALIDATION

The scalability and empirical success of the proposed algorithm are evaluated through simulations using heterogeneous robot teams in harvesting and transportation scenarios. Through simulations, we can assess the g-value, which includes the success rate and runtime. The g-value $\pi(g)$ is the sum of the optimal paths of k agents and can be calculated as follows. The additionality A is calculated as follows:

$$\pi(g) = \sum_{i=1}^{k} \pi_{\text{optimal}}, \quad A = \frac{\pi}{\pi(g)} \tag{31}$$

The results presented in Table 2 demonstrate the scalability and high success rate of the proposed algorithm in scenarios involving varying robot team sizes and obstacle densities. First, in terms of scalability, the algorithm exhibits consistent performance improvements as the number of robots (N) increases from 20 to 50. The runtime is significantly lower than that of the CBS algorithm, with this improvement particularly evident at higher robot counts (e.g., runtime of 0.02917 s for the proposed algorithm compared with 14.02170 s for CBS at $N = 50$). These outcomes indicate robust scalability, suggesting that the proposed algorithm effectively manages complexity without exponential computational growth.

Across all cases, the proposed algorithm consistently achieves near-optimal path efficiency, reflected in the additionality (A) values varying in a narrow range of 1.02–1.08. These values imply that the calculated paths are very close to the optimal solutions (π_g), confirming the absence of critical inefficiencies or gaps. Furthermore, the marginal differences observed in path lengths (π versus π_g) reinforce that the algorithm efficiently exploits the available workspace while maintaining high-quality solutions.

In summary, the proposed approach not only maintains its computational efficiency as robot numbers scale but also consistently delivers near-optimal solutions, affirming its practical applicability and robustness in diverse and complex multi-agent agricultural environments.

B. LAB-SCALE TEST

The simulation-based validation shows robust scalability and a high success rate for the proposed algorithm in randomized instance. Nevertheless, further validation in realistic agricultural conditions with environmental disturbances and challenging terrains is necessary to comprehensively evaluate performance.

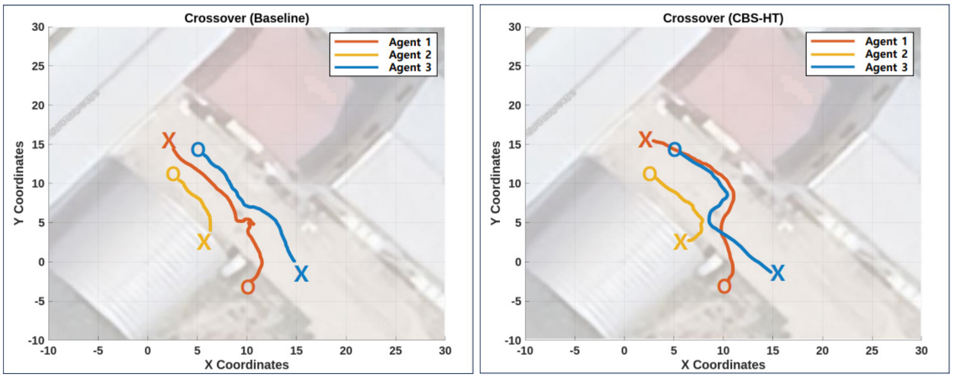
Detailed results from lab-scale tests are summarized in Table 3 and visualized in Figs. 10 and 11. These figures clearly demonstrate path planning differences between scenarios. In the ‘‘Crossover’’ scenario (Fig. 10), CBS-HT improves the path efficiency of the highest-priority Agent 1, significantly reducing travel distance (by 3.6 m) and travel time (by 5.2 s). However, Agent 3, with the lowest priority, experiences increased travel distance (5.3 m longer) and time (7.5 s longer) owing to additional maneuvers required for yielding.

In the ‘‘Face’’ scenario (Fig. 11), all agents benefit from CBS-HT, achieving reductions in both travel distance and time. Specifically, Agent 1 exhibits the most significant improvements (distance reduced by 5.6 m, time reduced by 6.3 s). These results highlight that CBS-HT effectively alleviates congestion by minimizing instances of agent interaction and potential conflict.

Additionally, these outcomes emphasize the effectiveness of priority assignment in optimizing path planning performance. Higher-priority agents experience substantial performance gains, while lower-priority agents manage acceptable trade-offs. This further underlines CBS-HT’s suitability and potential effectiveness for real-world applications involving complex multi-agent coordination in agricultural environments.

C. ORCHARD EVALUATION

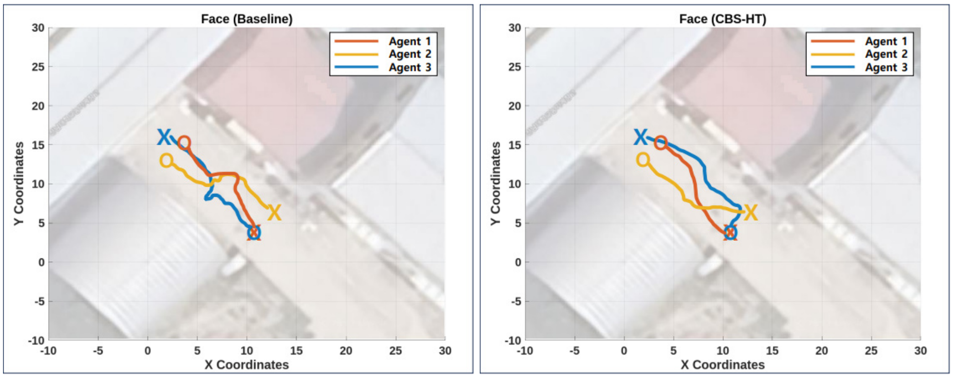
The sequential snapshots provided in Fig. 12. And the orchard experiment results are summarized in Table 4 and 5, with corresponding trajectory analyses shown in Fig. 13. The predefined ground-truth trajectory from the initial position to the intended destination measures 121.2 m. Analysis of systems operating without the proposed algorithm reveals an average increase in travel distance of approximately 10 m (e.g., 141.5 m for Agent 1 in baseline compared with 122.4 m with the algorithm in Table 4), attributable to route conflicts induced by simultaneous departures toward identical destinations. In contrast, the introduction of the priority-based algorithm significantly alleviates these congestion issues, as quantified by the sum of positive and negative distance deviations, which result in an overall negative net increment, thus confirming improved navigational efficiency aligned with agent priorities.



(a)

(b)

FIGURE 10. Result for crossover scenario in lab-scale test. O and X denote the starting point and destination, respectively: (a) baseline, (b) CBS-HT.



(a)

(b)

FIGURE 11. Result for face scenario in lab-scale test. O and X denote the starting point and destination, respectively: (a) baseline, (b) CBS-HT.

TABLE 3. Results of lab-scale tests.

Environment	Scenario	Agent (≺)	Case	D_i (m)	T_i (s)	N_c (times)
Outdoor	Crossover	Agent 1 (1st)	Baseline	31.8	29.3	0
			CBS-HT	28.2 (-3.6)	24.1 (-5.2)	0
		Agent 2 (2nd)	Baseline	13.5	17.6	0
			CBS-HT	14.7 (+1.2)	19.2 (+1.6)	0
		Agent 3 (3rd)	Baseline	23.1	22.5	0
			CBS-HT	28.4 (+5.3)	30.0 (+7.5)	0
	Face	Agent 1 (1st)	Baseline	18.1	20.3	0
			CBS-HT	12.5 (-5.6)	14.0 (-6.3)	0
		Agent 2 (2nd)	Baseline	17.8	20.6	0
			CBS-HT	15.3 (-2.5)	18.4 (-2.2)	0
		Agent 3 (3rd)	Baseline	19.7	26.1	0
			CBS-HT	17.0 (-2.7)	20.3 (-5.8)	0

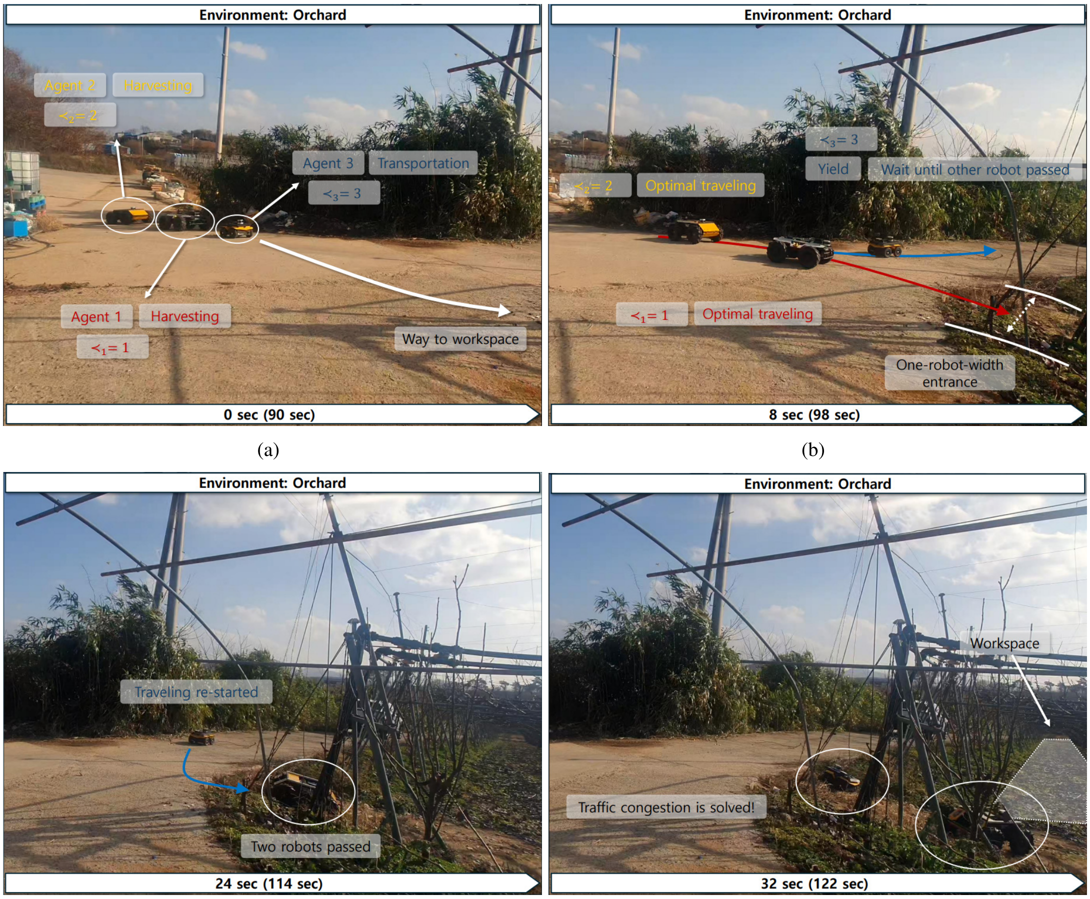
A detailed inspection of Case 3, where improvements are most pronounced, provides deeper insights. The highest-priority agent, Agent 1, implements a strategic detour of 19.1 m (from 141.5 m baseline to 122.4 m with CBS-HT). Despite the longer path, Agent 1 achieves earlier access to the field workspace compared with the baseline by approximately 28.4 s (from 215.6 s baseline to 187.2 s with CBS-HT), reflecting optimized conflict resolution and improved operational sequencing. Similarly, Agent 2, performing analogous harvesting tasks, exhibits only marginal increments in travel distance (2.3 m increase, from 127.8 m to 130.1 m) and travel time (2.6 s increase,

TABLE 4. Averaged results for orchard tests.

Environment	Agent ($\prec$)	Case	D_i (m)	T_i (s)	N_c (times)
Orchard	Agent 1 (1st)	Baseline	141.5	215.6	0
		CBS-HT	122.4 (-19.1)	187.2 (-28.4)	0
	Agent 2 (2nd)	Baseline	127.8	190.3	0
		CBS-HT	130.1 (+2.3)	192.9 (+2.6)	0
	Agent 3 (3rd)	Baseline	121.1	184.1	0
		CBS-HT	135.2 (+14.1)	200.7 (+16.6)	0

TABLE 5. Representative results of orchard tests (case#3).

Environment	Agent ($\prec$)	Case	D_i (m)	T_i (s)	N_c (times)
Orchard	Agent 1 (1st)	Baseline	133.2	210.0	0
		CBS-HT	122.7 (-10.5)	189.2 (-20.8)	0
	Agent 2 (2nd)	Baseline	129.9	215.7	0
		CBS-HT	130.4 (+0.5)	217.3 (+1.6)	0
	Agent 3 (3rd)	Baseline	127.3	188.5	0
		CBS-HT	135.3 (+8.0)	197.3 (+8.8)	0



(a) (b) (c) (d)

FIGURE 12. Snapshot of the orchard test: (a) at 0 s (90 s in instance), (b) at 8 s (98 s in instance), (c) at 24 seconds (114 s in instance), (d) at 32 seconds (122 s in instance). A full instance video is available at https://youtu.be/wtIaAGrEN4I?si=7R4bfz1y9xkJmKBf

from 190.3 s to 192.9 s), highlighting the algorithm's effectiveness in maintaining operational consistency. Notably, the incremental travel time observed for Agent 2 remains disproportionately smaller compared with the incremental

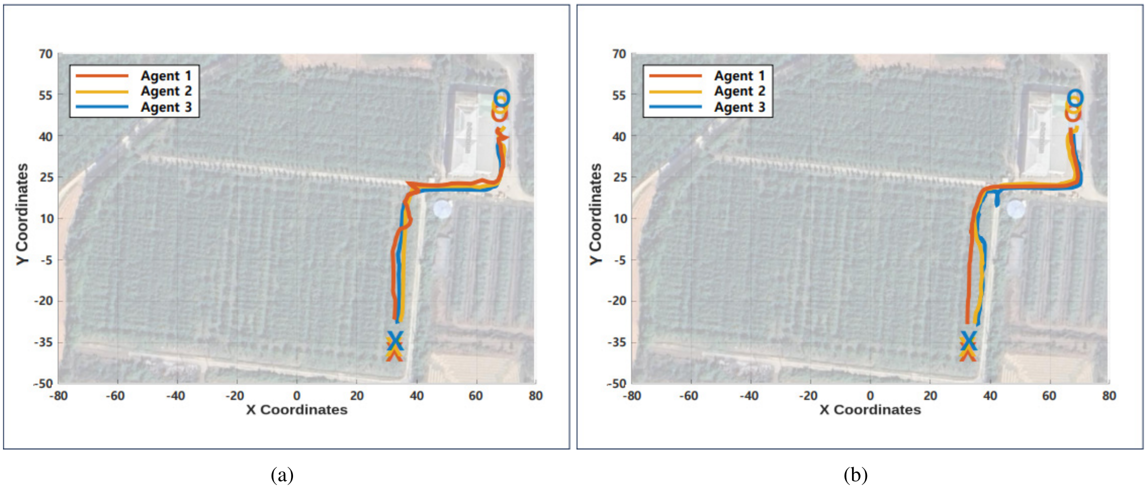


FIGURE 13. Results of orchard test. O and X denote the starting point and destination, respectively: (a) baseline, (b) CBS-HT.

distance, suggesting enhanced route selection efficiency and minimized acceleration–deceleration cycles.

The sequential snapshots provided in Fig. 12 further substantiate these findings, offering a time-resolved visualization of agent movements and interactions, clearly illustrating the strategic rerouting decisions enabled by the algorithm. Specifically, at 0 s (initial state), agents depart simultaneously toward the workspace, resulting in immediate potential congestion (Figure 12a). At 8 s, Agent 1 executes optimal traveling while Agent 2 temporarily waits, demonstrating the algorithm's real-time conflict management capabilities (Figure 12b). By 24 s, both Agent 1 and Agent 2 have successfully navigated the narrow entrance, allowing Agent 3 to resume its route (Figure 12c). Finally, at 32 s, congestion is entirely resolved, showcasing the algorithm's robust dynamic conflict resolution capabilities in complex scenarios (Figure 12d).

The trajectory comparison provided in Fig. 13 confirms these numerical improvements, clearly illustrating optimized routes generated by the CBS-HT algorithm compared with the baseline. Overall, these quantified and visual results underscore the substantial operational advantages of the priority-based algorithm, emphasizing its capability to systematically mitigate congestion and significantly enhance navigational efficiency within complex orchard environments.

V. DISCUSSION

A. LIMITATIONS OF CENTRALIZED CONTROL SYSTEM

Centralized control systems inherently depend on the robustness and reliability of communication networks owing to their dependence on continuous data transmission to and from a central coordinator. The proposed path-planning algorithm operates on such a centralized framework, making the system highly sensitive to network disruptions, latency, and bandwidth constraints [43]. In scenarios where agricultural robots must navigate complex or unstructured terrains, including areas with dense vegetation, significant obstacles, or varying topography, maintaining stable connectivity becomes challenging [44]. Although typical agricultural settings have bounded and defined operational areas, facilitating network management, unexpected conditions such as environmental interferences or topological complexity may compromise network integrity.

This study adopts an IEEE 802.11 wireless network, a practical but limited choice that does not fully guarantee high reliability in extended or obstacle-dense environments. To mitigate these limitations, integrating satellite-based communication infrastructure, which provides broader coverage and improved reliability for outdoor environments, should be considered. Satellite communication can significantly enhance the scalability and robustness of centralized control systems, enabling seamless real-time data exchanges even in extensive agricultural landscapes. However, the deployment of satellite networks requires addressing challenges such as signal delay, power management, and cost-effectiveness, which require detailed investigation for successful practical implementations.

B. ADVANTAGES AND LIMITATIONS

The proposed CBS-HT planner offers several advantages. It generates collision-free paths for up to fifty robots in under sixty seconds, satisfying real-time requirements for orchard operations. Runtime grows almost in proportion to team size, which demonstrates practical scalability rather than exponential growth. Because coordination is handled centrally, the method works without inter-robot communication, so packet loss and radio congestion do not reduce performance. The planner also supports heterogeneous tasks, for example harvesting and transport, by encoding static priority levels that give higher-value robots shorter and less congested routes.

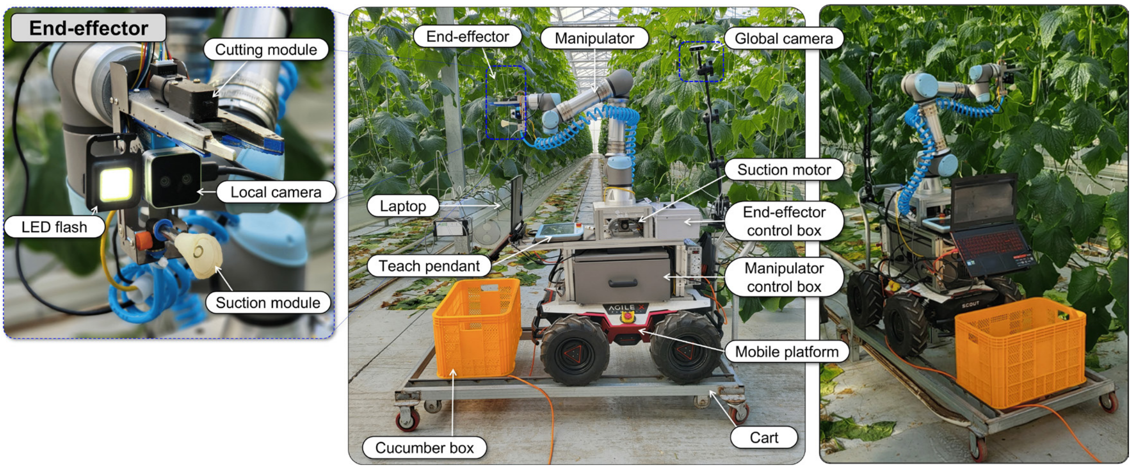


FIGURE 14. Developed harvesting robot, composed of heterogeneous robot teams [4].

These strengths come with limitations. CBS-HT offers no formal guarantees of completeness or cost optimality under fixed priorities, so it may fail to find a path in environments that are highly cluttered. Priorities are assigned offline and remain unchanged during execution, which prevents the system from reacting when mission urgency shifts, such as when a low-battery transporter should take precedence. The success rate falls when obstacle density exceeds thirty percent, suggesting that richer heuristics or local replanning would help. The evaluation used a grid resolution of seven hundred millimetres and only three physical robots, so performance with finer maps and much larger fleets remains untested. Future work will explore dynamic priority adaptation, adaptive grid refinement, and broader field trials to address these issues.

C. DECISION ON PRIORITIES

The proposed algorithm introduces generalized prioritized path-planning for coordinating multiple heterogeneous agricultural robots. Although harvesting and transportation scenarios are selected empirically for evaluating the algorithm's performance, the method itself remains task-agnostic and universally applicable to various agricultural operations requiring coordinated robotic teams [40], [45]. The priority scheme adopted in this study, while arbitrary, demonstrates the algorithm's capability to manage task assignments and robot coordination effectively.

Although CBS-HT assumes a fixed priority ordering among agents, the algorithm's resolution mechanism—where only the lower-priority agent is constrained at each conflict—ensures that the overall search structure remains stable across different priority configurations. This structural consistency suggests that minor changes in the priority order (e.g., reversing harvesters and transporters) are unlikely to significantly degrade solution quality or feasibility. However, we acknowledge that certain priority choices may have environment-dependent effects, particularly in constrained or asymmetric maps.

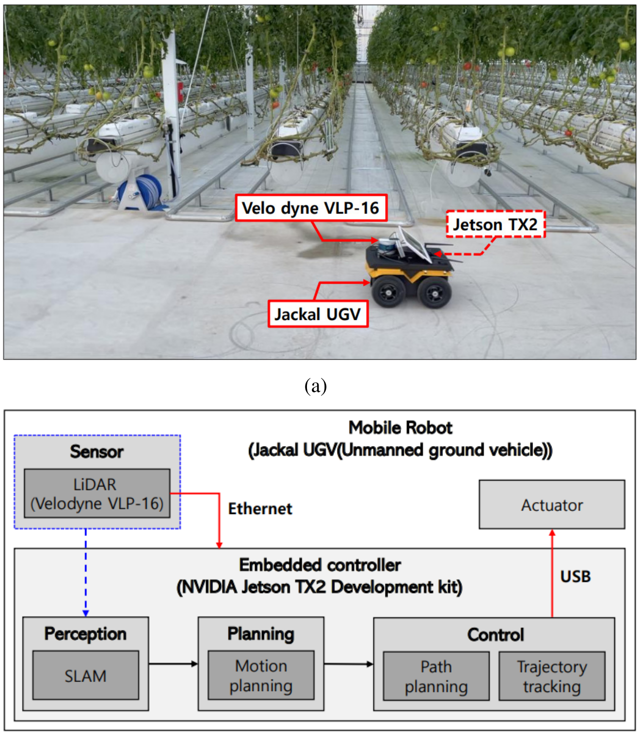
To address this, we conceptually analyze the potential impact of alternative fixed-priority schemes in this section. While empirical evaluation is left for future work, preliminary reasoning indicates that CBS-HT is robust to such variations due to its local conflict resolution mechanism. Nevertheless, developing context-aware or adaptive priority assignment strategies remains a promising extension. Future work will explore online adjustment of agent priorities based on task urgency, congestion level, or learned policies.

Future work can be aimed at developing systematic and context-aware frameworks for priority assignment, clearly defining criteria reflective of specific agricultural operations and environmental conditions. Advanced methodologies such as multi-criteria optimization, reinforcement-learning-based prioritization, or auction-based task allocation could be utilized to dynamically adapt priorities, thus enhancing efficiency and operational flexibility in diverse agricultural environments.

D. DIRECTIONS FOR FUTURE WORK

Insights from this study suggest several avenues for advancing agricultural robotic systems. Although the evaluation in this work is focused on harvesting and transportation, the prioritization algorithm is generalizable and can be applied in broader contexts. The priority scheme adopted in this study, while empirical, demonstrates the algorithm's capability to manage task assignments and robot coordination effectively. Future studies can be aimed at developing quantitative performance metrics for practical validation, assessing aspects such as operational robustness, communication reliability, task completion efficiency, and adaptability to unforeseen environmental changes.

Expanding communication infrastructure through satellite networks represents a critical direction for improving scalability and operational reliability. Given the need for



(a)

(b)

FIGURE 15. Developed transportation robot, composed of heterogeneous robot teams: (a) hardware, (b) software architecture [46].

agricultural robot teams to operate reliably at larger scales, future research will explicitly focus on deploying CBS-HT in significantly expanded fleets of over 100 robots, parallelizing the solver to efficiently manage complex scenarios, and rigorously evaluating performance in realistic agricultural environments. Furthermore, GPS-denied environments such as dense orchards require robust alternative localization approaches. Therefore, future studies will investigate sensor fusion methods integrating LiDAR, visual odometry, and IMUs to ensure consistent navigation performance alongside satellite-based communication systems.

Also, extending the validation of the proposed system to actual field experiments is essential. The integration of previously developed autonomous robots, in our research group, for harvesting (Fig. 14), transportation (Fig. 15), and spraying tasks [4], [46], [47] can enable comprehensive validation. Moreover, quantitative evaluation metrics, such as task completion time, collision frequency, path optimality, and recovery robustness, must be defined and assessed to evaluate autonomous multi-task agricultural systems in real-world settings.

Although CBS-HT is conceptually distinguished from existing priority-based methods, a direct quantitative comparison was beyond the scope of this study due to system-level constraints. Future research will focus on implementing and benchmarking these planners in shared environments to validate their relative performance in agricultural contexts.

The proposed algorithm is evaluated assuming harvest and transportation scenarios in this work, and thus, the actual agricultural task is not performed. However, as depicted in Figs. 14 and 15, our research group has previously studied not only the autonomous harvesting [4], [47] and transportation robots [46] covered in this study but also autonomous spraying robots. The heterogeneous agricultural robot team can thus be composed of existing agricultural robots [7]. In future work, the proposed algorithm can be integrated with the configured heterogeneous agricultural team. Additionally, fully autonomous monitoring, spraying, transportation, and harvesting tasks can be performed in the orchard.

VI. CONCLUSION

This study proposes CBS-HT, a prioritized safe-interval path-planning algorithm developed for coordinating heterogeneous agricultural robot teams. By assigning distinct priorities to robots performing diverse agricultural tasks, CBS-HT effectively addresses multi-agent pathfinding challenges, significantly enhancing operational efficiency by reducing congestion and unnecessary maneuvers. Through rigorous validation, including simulations, lab-scale experiments, and orchard field tests, the algorithm achieved the initial objectives of ensuring collision-free coordination, improving mission-level efficiency, and maintaining robust scalability across different operating conditions. Although centralized control introduces potential vulnerabilities in communication reliability, CBS-HT successfully optimized travel distances and minimized conflict scenarios without requiring inter-robot communication.

Future work will extend real-world testing to larger heterogeneous teams and more varied agricultural task sets, including monitoring and spraying. Additional directions include developing systematic frameworks for adaptive priority assignment that respond to real-time factors such as task urgency, battery status, and congestion level, as well as expanding communication infrastructure—e.g., integrating satellite-based networks—to improve reliability in large-scale deployments. These advancements will further position CBS-HT as a scalable and adaptable coordination framework for complex, multi-robot agricultural operations.

REFERENCES

[1] J. Kim, J. Seol, S. Lee, S.-W. Hong, and H. I. Son, "An intelligent spraying system with deep learning-based semantic segmentation of fruit trees in orchards," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3923–3929, doi: 10.1109/ICRA40945.2020.9197556.

[2] C. Ju and H. I. Son, "Multiple UAV systems for agricultural applications: Control, implementation, and evaluation," *Electronics*, vol. 7, no. 9, p. 162, Aug. 2018, doi: 10.3390/electronics7090162.

[3] J. Kim, S. Kim, C. Ju, and H. I. Son, "Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications," *IEEE Access*, vol. 7, pp. 105100–105115, 2019, doi: 10.1109/ACCESS.2019.2932119.

[4] Y. Park, J. Seol, J. Pak, Y. Jo, C. Kim, and H. I. Son, "Human-centered approach for an efficient cucumber harvesting robot system: Harvest ordering, visual servoing, and end-effector," *Comput. Electron. Agricult.*, vol. 212, Sep. 2023, Art. no. 108116, doi: 10.1016/j.compag.2023.108116.

[5] C. D. Bellicoso, L. R. Buonocore, V. Lippiello, and B. Siciliano, ''Design, modeling and control of a 5-DoF light-weight robot arm for aerial manipulation,'' in *Proc. 23rd Medit. Conf. Control Autom. (MED)*, Jun. 2015, pp. 853–858, doi: 10.1109/MED.2015.7158852.

[6] C. Ju and H. I. Son, ''Modeling and control of heterogeneous agricultural field robots based on Ramadge–Wonham theory,'' *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 48–55, Jan. 2020, doi: 10.1109/LRA.2019.2941178.

[7] C. Ju, J. Kim, J. Seol, and H. I. Son, ''A review on multirobot systems in agriculture,'' *Comput. Electron. Agricult.*, vol. 202, Nov. 2022, Art. no. 107336, doi: 10.1016/j.compag.2022.107336.

[8] A. Mohiuddin, T. Tarek, Y. Zweiri, and D. Gan, ''A survey of single and multi-UAV aerial manipulation,'' *Unmanned Syst.*, vol. 8, no. 2, pp. 119–147, Apr. 2020, doi: 10.1142/s2301385020500089.

[9] C. Ju and H. I. Son, ''A hybrid systems-based hierarchical control architecture for heterogeneous field robot teams,'' *IEEE Trans. Cybern.*, vol. 53, no. 3, pp. 1802–1815, Mar. 2023, doi: 10.1109/TCYB.2021.3133631.

[10] J. Yu and S. LaValle, ''Structure and intractability of optimal multi-robot path planning on graphs,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 27, 2013, pp. 1443–1449, doi: 10.1609/aaai.v27i1.8541.

[11] J. Sang, D. Ma, X. Xie, and X. Hu, ''Group-aggregation of hierarchical containment control for homogeneous multi-agent systems in precision agriculture,'' *Int. J. Control, Autom. Syst.*, vol. 22, no. 4, pp. 1400–1408, Apr. 2024, doi: 10.1007/s12555-022-1071-y.

[12] S. Patil, V. Vyatkin, and M. Sorouri, ''Formal verification of intelligent mechatronic systems with decentralized control logic,'' in *Proc. IEEE 17th Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2012, pp. 1–7, doi: 10.1109/ETFA.2012.6489678.

[13] I. Draganjac, D. Miklic, Z. Kovacic, G. Vasiljevic, and S. Bogdan, ''Decentralized control of multi-AGV systems in autonomous warehousing applications,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 4, pp. 1433–1447, Oct. 2016, doi: 10.1109/TASE.2016.2603781.

[14] J. Li, L. Chen, M. Cao, and C. Li, ''Satellite formation flying control by using only angle measurements,'' *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 2, pp. 1439–1451, Apr. 2023, doi: 10.1109/TAES.2022.3200345.

[15] T. Jiang, Y. Liu, L. Xiao, W. Liu, and G. Liu, ''PCC polar codes for future wireless communications: Potential applications and design guidelines,'' *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 414–420, Jun. 2024, doi: 10.1109/MWC.017.2200586.

[16] R. Janssen, R. van der Molengraft, H. Bruyninckx, and M. Steinbuch, ''Cloud based centralized task control for human domain multi-robot operations,'' *Intell. Service Robot.*, vol. 9, no. 1, pp. 63–77, Jan. 2016, doi: 10.1007/s11370-015-0185-y.

[17] F. Matoui, B. Boussaid, B. Metoui, and M. N. Abdelkrim, ''Contribution to the path planning of a multi-robot system: Centralized architecture,'' *Intell. Service Robot.*, vol. 13, no. 1, pp. 147–158, Jan. 2020, doi: 10.1007/s11370-019-00302-w.

[18] F. J. Mendiburu, M. R. A. Morais, and A. M. N. Lima, ''Behavior coordination in multi-robot systems,'' in *Proc. IEEE Int. Conf. Autom. (ICA-ACCA)*, Oct. 2016, pp. 1–7, doi: 10.1109/ICA-ACCA.2016.7778506.

[19] B. Nebel, ''The computational complexity of multi-agent pathfinding on directed graphs,'' *Artif. Intell.*, vol. 328, Mar. 2024, Art. no. 104063, doi: 10.1016/j.artint.2023.104063.

[20] R. N. Gómez, C. Hernández, and J. A. Baier, ''A compact answer set programming encoding of multi-agent pathfinding,'' *IEEE Access*, vol. 9, pp. 26886–26901, 2021, doi: 10.1109/ACCESS.2021.3053547.

[21] Y. Bai, B. Lindqvist, S. Nordström, C. Kanellakis, and G. Nikolakopoulos, ''Cluster-based multi-robot task assignment, planning, and control,'' *Int. J. Control, Autom. Syst.*, vol. 22, no. 8, pp. 2537–2550, Aug. 2024, doi: 10.1007/s12555-023-0745-4.

[22] A. Felner, R. Stern, S. E. Shimony, E. Boyarski, M. Goldenberg, G. Sharon, N. Sturtevant, G. Wagner, and P. Surynek, ''Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges,'' in *Proc. Int. Symp. Combinat. Search*, vol. 8, 2021, pp. 29–37, doi: 10.1609/socs.v8i1.18423.

[23] D. Silver, ''Cooperative pathfinding,'' in *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, vol. 1, 2021, pp. 117–122, doi: 10.1609/aiide.v1i1.18726.

[24] G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T. K. S. Kumar, S. Koenig, and H. Choset, ''PRIMAL: Pathfinding via reinforcement and imitation multi-agent learning,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2378–2385, Jul. 2019, doi: 10.1109/LRA.2019.2903261.

[25] J. van der Berg, S. J. Guy, M. Lin, and D. Manocha, ''Reciprocal n-body collision avoidance,'' in *Proc. 14th Int. Symp. Robot. Res.* Cham, Switzerland: Springer, 2011, pp. 3–19, doi: 10.1007/978-3-642-19457-3_1.

[26] J.-M. Alkazzi and K. Okumura, ''A comprehensive review on leveraging machine learning for multi-agent path finding,'' *IEEE Access*, vol. 12, pp. 57390–57409, 2024, doi: 10.1109/ACCESS.2024.3392305.

[27] T. Wang, ''CLE: An integrated framework of CNN, LSTM, and enhanced A3C for addressing multi-agent pathfinding challenges in warehousing systems,'' *IEEE Access*, vol. 12, pp. 88904–88912, 2024, doi: 10.1109/ACCESS.2024.3416111.

[28] H. Ma, D. Harabor, P. J. Stuckey, J. Li, and S. Koenig, ''Searching with consistent prioritization for multi-agent path finding,'' in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 188–189.

[29] M. Phillips and M. Likhachev, ''SIPP: Safe interval path planning for dynamic environments,'' in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 5628–5635.

[30] K. Kasaura, M. Nishimura, and R. Yonetani, ''Prioritized safe interval path planning for multi-agent pathfinding with continuous time on 2D roadmaps,'' *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10494–10501, Oct. 2022.

[31] R. Stern, N. Sturtevant, A. Felner, S. Koenig, H. Ma, T. T. Walker, J. Li, D. Atzmon, L. Cohen, T. K. S. Kumar, E. Boyarski, and R. Barták, ''Multi-agent pathfinding: Definitions, variants, and benchmarks,'' in *Proc. Int. Symp. Combinat. Search*, 2019, pp. 151–158.

[32] A. Zapata, J. Godoy, and R. Asín-Achá, ''Anytime automatic algorithm selection for the multi-agent path finding problem,'' *IEEE Access*, vol. 12, pp. 62177–62188, 2024, doi: 10.1109/ACCESS.2024.3395495.

[33] R. Barták and J. Švancara, ''On SAT-based approaches for multi-agent path finding with the sum-of-costs objective,'' in *Proc. Int. Symp. Combinat. Search*, vol. 10, 2021, pp. 10–17, doi: 10.1609/socs.v10i1.18497.

[34] L. Crombez, G. D. da Fonseca, Y. Gerard, A. Gonzalez-Lorenzo, P. Lafourcade, and L. Libralesso, ''Shadoks approach to low-makespan coordinated motion planning,'' *ACM J. Experim. Algorithmics*, vol. 27, pp. 1–17, Dec. 2022, doi: 10.1145/3524133.

[35] P. Surynek, A. Felner, R. Stern, and E. Boyarski, ''An empirical comparison of the hardness of multi-agent path finding under the makespan and the sum of costs objectives,'' in *Proc. Int. Symp. Combinat. Search*, vol. 7, 2021, pp. 145–146, doi: 10.1609/socs.v7i1.18407.

[36] M. Cáp, P. Novák, A. Kleiner, and M. Selecký, ''Prioritized planning algorithms for trajectory coordination of multiple mobile robots,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 835–849, Jul. 2015, doi: 10.1109/TASE.2015.2445780.

[37] W. Hönig, S. Kiesel, A. Tinka, J. W. Durham, and N. Ayanian, ''Conflict-based search with optimal task assignment,'' in *Proc. Int. Joint Conf. Auto. Agents Multiagent Syst.*, 2018, pp. 757–765.

[38] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, ''Conflict-based search for optimal multi-agent pathfinding,'' *Artif. Intell.*, vol. 219, pp. 40–66, Feb. 2015, doi: 10.1016/j.artint.2014.11.006.

[39] M. Rahman, M. A. Alam, M. M. Islam, I. Rahman, M. M. Khan, and T. Iqbal, ''An adaptive agent-specific sub-optimal bounding approach for multi-agent path finding,'' *IEEE Access*, vol. 10, pp. 22226–22237, 2022, doi: 10.1109/ACCESS.2022.3151092.

[40] Y. Jo and H. I. Son, ''Field evaluation of a prioritized path-planning algorithm for heterogeneous agricultural tasks of multi-UGVs,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 11891–11897, doi: 10.1109/ICRA57147.2024.10610857.

[41] G. Grisetti, C. Stachniss, and W. Burgard, ''Improved techniques for grid mapping with rao-blackwellized particle filters,'' *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007, doi: 10.1109/TRO.2006.889486.

[42] Y. Park, J. Seol, J. Pak, Y. Jo, J. Jun, and H. I. Son, ''A novel end-effector for a fruit and vegetable harvesting robot: Mechanism and field experiment,'' *Precis. Agricult.*, vol. 24, no. 3, pp. 948–970, Jun. 2023, doi: 10.1007/s11119-022-09981-5.

[43] H. Li, Z. Chen, B. Fu, and M. Sun, ''Nonlinear control of heterogeneous vehicle platoon with time-varying delays and limited communication range,'' *Int. J. Control, Autom. Syst.*, vol. 21, no. 6, pp. 1727–1738, Jun. 2023, doi: 10.1007/s12555-021-1109-6.

[44] J. Seol, C. Ju, and H. I. Son, ''Leader–follower control of multi-unmanned aerial vehicle based on supervisory control theory for a broad tributary area mapping scenario,'' *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 237, no. 10, pp. 1765–1776, Nov. 2023, doi: 10.1177/09596518231173765.

[45] Y. Jo and H. I. Son, ''A path planning and coordination algorithm for heterogeneous tasks of multi-UGV in smart farm: Work in progress,'' in *Proc. 22nd Int. Conf. Control, Autom. Syst. (ICCAS)*, Nov. 2022, pp. 1387–1390, doi: 10.23919/ICCAS55662.2022.10003824.

[46] J. Pak, J. Kim, Y. Park, and H. I. Son, ''Field evaluation of path-planning algorithms for autonomous mobile robot in smart farms,'' *IEEE Access*, vol. 10, pp. 60253–60266, 2022, doi: 10.1109/ACCESS.2022.3181131.

[47] J. Jun, J. Kim, J. Seol, J. Kim, and H. I. Son, ''Towards an efficient tomato harvesting robot: 3D perception, manipulation, and end-effector,'' *IEEE Access*, vol. 9, pp. 17631–17640, 2021, doi: 10.1109/ACCESS.2021.3052240.

YUSEUNG JO received the B.S. degree from the Department of Rural and Biosystems Engineering, Chonnam National University, Republic of Korea, in 2022, and the M.S. degree from the Department of Convergence Biosystems Engineering, Chonnam National University, in 2024, where he is currently pursuing the Ph.D. degree. His research interests include field robotics, agricultural robotics, multi-robot systems, and neural networks.

HYOUNG IL SON (Senior Member, IEEE) received the B.S. and M.S. degrees from the Department of Mechanical Engineering, Pusan National University, Republic of Korea, in 1998 and 2000, respectively, and the Ph.D. degree from the Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea, in 2010. He had several appointments both academia and industry as a Senior Researcher with LG Electronics, Pyungtaek, Republic of Korea, from 2003 to 2005, and Samsung Electronics, Cheonan, Republic of Korea, from 2005 to 2009. He was a Research Associate with the Institute of Industrial Science, The University of Tokyo, Tokyo, Japan, in 2010. He was a Research Scientist with the Max Planck Institute for Biological Cybernetics, Tbingen, Germany, from 2010 to 2012. From 2012 to 2015, he led the Telerobotics Group, Central Research Institute, Samsung Heavy Industries, Daejeon, Republic of Korea, as a Principal Researcher. In 2015, he joined the Department of Convergence Biosystems Engineering, Chonnam National University, Gwangju, Republic of Korea, as a Faculty Member, where he is currently a Professor. He is also an Adjunct Professor with the Department of Robotics Engineering, Chonnam National University. His research interests include field robotics, hybrid systems, haptics, teleoperation, and agricultural robotics.