sensors

MDPI

Article

Multi-Robot System for Cooperative Tidying Up with Mobile Manipulators and Transport Agents

Jae-Bong Yi, Shady Nasrat, Dongwoon Song, Joonyoung Kim and Seung-Joon Yi *

Department of Electrical Engineering, Pusan National University, Busan 46241, Republic of Korea; niteofhunter@pusan.ac.kr (J.-B.Y.); shadyloai@pusan.ac.kr (S.N.); dongwoon@pusan.ac.kr (D.S.); kjykjy98@pusan.ac.kr (J.K.)
* Correspondence: seungjoon.yi@pusan.ac.kr; Tel.: +82-51-510-7917

Abstract: This paper presents a system in which mobile manipulators and transport agents cooperate to solve a multi-agent pickup and delivery (MAPD) problem. The primary objective is to allocate appropriate tasks to heterogeneous robots by considering their capabilities and states. Unlike previous studies that focused on homogeneous teams or assigned distinct roles to heterogeneous robots, this work emphasizes synergy through cooperative task execution. A key feature of the proposed system is that mobile manipulators behave differently depending on whether they are paired with a transport agent. Additionally, rather than generating a full trajectory from start to end, the system plans partial trajectories, allowing dynamic re-pairing of transport agents through an auction algorithm. After re-pairing, new starting nodes are defined, and the following trajectory is updated accordingly. The proposed system is validated through simulations, and its effectiveness is demonstrated by comparing it against a baseline system without dynamic pairing.

Keywords: heterogeneous multi-agent pickup and delivery; cooperating robots; multi-robot task allocation

check for updates

Academic Editor: Jiaxi Chen

Received: 17 April 2025
Revised: 14 May 2025
Accepted: 21 May 2025
Published: 22 May 2025

Citation: Yi, J.-B.; Nasrat, S.; Song, D.; Kim, J.; Yi, S.-J. Multi-Robot System for Cooperative Tidying Up with Mobile Manipulators and Transport Agents. *Sensors* **2025**, *25*, 3269. https://doi.org/10.3390/s25113269

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, multi-robot systems are utilized in various fields, such as logistics [1,2], rescue [3,4], medical care [5,6], and agriculture [7,8]. For these domains, which require the coordination of multiple robots, solving MAPD problems is an important part of the process. Current research in MAPD focuses on developing efficient algorithms for task allocation and path planning to optimize efficiency [9,10]. These studies aim to minimize travel distance, task completion time, and resource consumption while ensuring a collision-free trajectory for multiple agents [11–13].

However, most existing approaches assume homogeneous robot teams [7,14] or assigning heterogeneous robots to distinct tasks [15–18]. Although some studies address heterogeneous robots performing the same task, they rarely focus on performing assigned tasks cooperatively [3,5,19].

In this context, we propose a multi-robot system that performs tidying-up tasks by integrating two heterogeneous robot types: mobile manipulators and transport agents. The system is designed to support synergistic collaboration between these robots.

Figure 1 shows the capabilities of mobile manipulators and transport agents. In this system, a mobile manipulator can pick up an object and deliver it to the destination or hand it to a transport agent. In contrast, a transport agent cannot pick up objects themselves but can receive them from manipulators. However, unlike mobile manipulators, they can deliver multiple objects to the destination in a single trip.

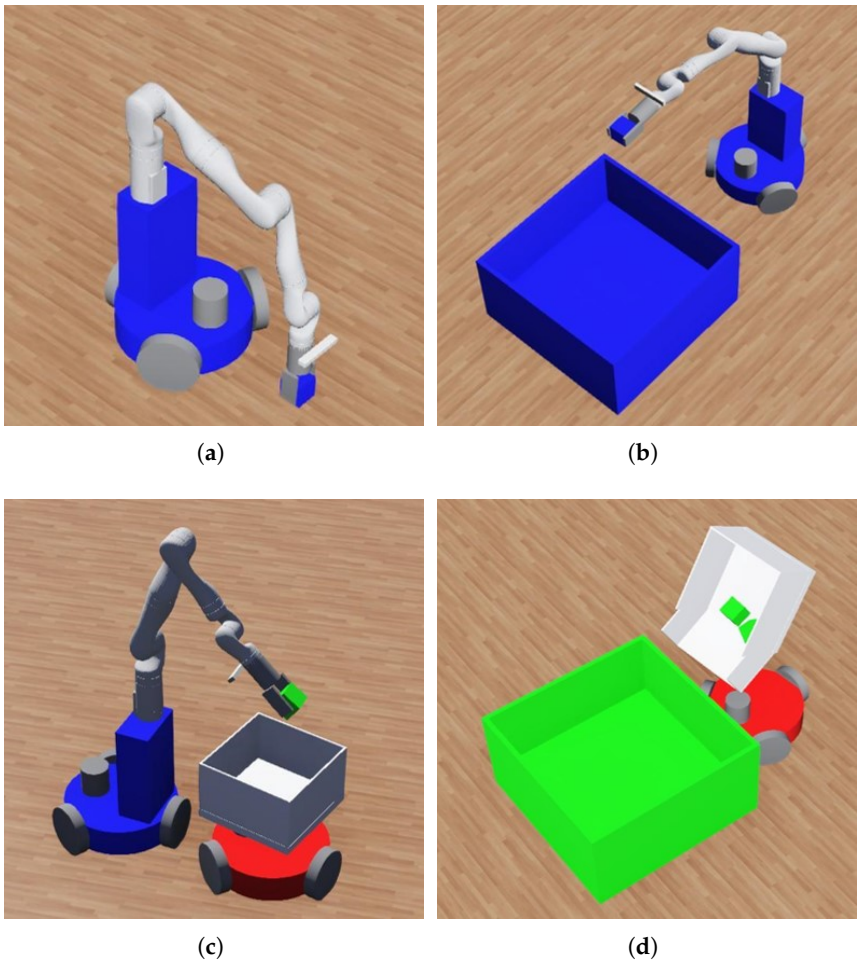


Figure 1. Capabilities of the mobile manipulator (**a**,**b**) and the transport agent (**c**,**d**): (**a**) picking up an object; (**b**) delivering an object at the destination; (**c**) receiving an object from a paired mobile manipulator; (**d**) delivering multiple objects to the destination.

Unlike previous approaches that assign independent tasks to each robot type, this system enables pairing these two types to form collaborative units and create complementary capabilities by integrating trajectory optimization and auction-based task allocation.

In generating an optimized trajectory, the objective function consists of traveling and object-handling time without considering the process of finding new partners. The planning conditions are different depending on whether a transport agent is paired (carrying multiple objects) or unpaired (carrying only one object). Additionally, the system generates a partial trajectory corresponding to the number of objects to deliver in a single expectation.

After matching a new pairing partner and generating the sequence trajectory, new starting nodes are defined at the positions where robots are expected to be located after a certain amount of time. From these nodes, a new trajectory is generated.

1.1. Related Work

Several works have been proposed to solve the MAPD problem. Authors in [14] present the MAPD approach that robots can run in a peer-to-peer fashion. By distributing computation to multiple robots using primal decomposition [20–22], this method enables a reduction in computations without a central node. However, this work assumes the use of homogeneous robots. Additionally, duplicate access to delivery points is not considered in the optimization, and all robots should start their task at the same starting point. Furthermore, they conducted only abbreviated experiments, without pick-and-place, using mobile robots

without manipulators. Authors in [23] show a game-theoric multi-robot task allocation framework for multi-robot trash collection in dynamically changing environments. In this work, robots choose their working area based on a decision-making algorithm consisting of a payoff (reward) mechanism, Poisson clock, and task revision protocol. Another MAPD framework introduced in [7] shows a practical example of using a multi-objective discrete artificial bee colony (MODABC) algorithm in orchards. However, both [7,23] favor the use of homogeneous robots in solving MAPD problems.

Meanwhile, other studies deal with multi-robot systems with heterogeneous robots. Authors in [3] show task allocation with heterogenous robots by considering their capabilities and states in a rescue scenario, and they prove its validity by applying it to *Capture the Flag* game. Authors in [5] present task allocation and path planning of heterogeneous robots in medical scenarios. This work introduces Intensive Inter-task Relationship Tree Search (IIRTS) to perform fast task allocation and enable real-time implementation. However, they only present the methods to place heterogeneous robots in appropriate places and do not present how to perform the tasks at the arrival point.

Research in [15–18] deals with a topic similar to this work that solves the MAPD problem with mobile manipulators and transport agents. Although they solve this problem in various ways (e.g., weighted block sequence graph, constructive heuristics, and recurrent open shop scheduling), these works have limitations in restricting the movement range of mobile manipulators to near storage units. For this reason, these robots only take out items from storage at designated points. Furthermore, as this work assumes that all robots are operated in a structured factory setting with a grid path, applying them in non-structured environments is challenging.

1.2. Contributions

In order to implement the MAPD using only optimization, the timing and location of pairings should be considered in the optimization condition. However, as this method requires an extremely high amount of computation, it is impossible to use in practice. To address this issue, we introduce a multi-robot control system that combines optimization and auction algorithms. This work provides the following contributions.

1. We propose a novel MAPD system that enables cooperative task execution between mobile manipulators and transport agents in the same space.
2. The system integrates trajectory optimization and an auction-based task allocation mechanism to adjust the pairing of robots dynamically.
3. Unlike previous full-trajectory approaches, our method uses a horizon-based partial trajectory planning strategy that enables dynamic pairing and re-optimization during execution with reduced computational costs.
4. The system applies different planning conditions to mobile manipulators depending on whether they are currently paired with a transport agent or are operating independently. On the other hand, many previous works assume robots have fixed roles or behave in the same way.
5. The proposed system is validated through realistic simulations, demonstrating the algorithm's effectiveness.

2. System Overview

The overall system structure, shown in Figure 2, is composed of four main components: the initializer, trajectory planner, pairing planner, and predictor.

At the beginning of an operation, the initializer processes the current states of the environment—such as robot positions and object distributions derived from GPS sensors—and derives the initial states necessary for trajectory planning. These are

passed to the trajectory planner, which generates a robot's optimal path up to a predefined planning horizon without considering pairing updates.

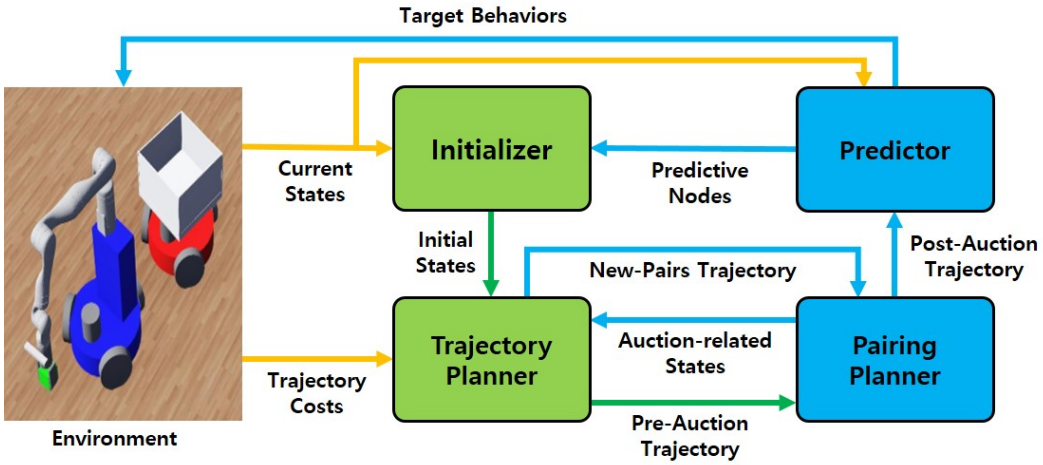


Figure 2. Overview of system structure: yellow arrows, green arrows, and blue arrows present the state flows starting from the environment, initializer, and predictor, respectively.

Next, the pairing planner uses an auction-based algorithm to determine new pairing partners for mobile manipulators and transport agents. It evaluates pairing options based on cost and availability and generates a trajectory that reflects these new pairings. These are then merged with the initial pre-auction trajectory to form the post-auction trajectory, which is passed to the predictor.

The predictor analyzes the post-auction trajectory and estimates predictive nodes, which are used by the initializer in the next planning cycle. It also contributes to determining the target behaviors of the robots by combining trajectory information with the current environmental context.

Finally, using the combined trajectory and current state information, the system derives the target actions for each robot.

3. Pre-Auction Trajectory Planning

We formulate the problem using the notations defined in Table 1 to generate optimal trajectories up to a predefined planning horizon without considering pairing updates.

An example of such a trajectory is illustrated in Figure 3, which includes three types of nodes: s for starting positions, j for object locations, and k for delivery destinations. In this example, mobile manipulators i_1 and i_3, which start at s_1 and s_3, are initially paired with transport agents, while i_2, starting at s_2, is not.

Table 1. Definitions of the notations.

	Parameters
$I = \{i_1, i_2, .., i_n\}$	Set of mobile manipulator i.
$I_p \subseteq I$	Set of mobile manipulators paired with transport agent.
$J = \{j_1, j_2, .., j_m\}$	Set of object node j.
$w \in [0, 1]$	Completion time weight.
$w_\tau \in [0, 1]$	Separated transport agent cost weight.
k	Destination node.
$H \in \mathbb{N}$	Object number horizon.

Table 1. *Cont.*

Constraints	
$l_i^{\text{init}} \in \mathbb{Z}_{\geq 0}$	Initial loaded object of the paired transport agent of the mobile manipulator i.
$L^{\max} \in \mathbb{N}$	Maximum carrying capacity of a transport agent.
$c_i^{\text{init}} \in \mathbb{R}_{\geq 0}$	Initial cost of the mobile manipulator i.
$c_i^{sj} \in \mathbb{R}_{\geq 0}$	The cost when traveling (s_i, j) and handling j.
$c^{jk} \in \mathbb{R}_{\geq 0}$	The cost when traveling (j, k) and placing the object j at its destination node k.
$c^{kj} \in \mathbb{R}_{\geq 0}$	The cost when traveling (k, j) and handling j.
$c^{jj'} \in \mathbb{R}_{\geq 0}$	The cost when traveling (j, j') and handling j'.
Decision variables	
$x_i^{sj} \in \{0, 1\}$	1 if i travels (s_i, j) and picks up j, 0 otherwise.
$x_i^{jkj'} \in \{0, 1\}$	1 if i travels (j, k), places the j at its destination node k, travels (k, j'), and picks up j', 0 otherwise.
$x_i^{jj'} \in \{0, 1\}$	1 if i travels (j, j') and picks up j', 0 otherwise.
$x_i^{kjj'} \in \{0, 1\}$	1 if i separates its paired transport and the transport agent travels to k, the destination node of j, and travels (j, j') and picks up j', 0 otherwise.
$x_i^{jk} \in \{0, 1\}$	1 if i travels (j, k) at last and places the j at its destination node k, 0 otherwise.
$o_i^j \in \mathbb{R}_{\geq 0}$	The order of picking up j among the tasks of i.
$C_i \in \mathbb{R}_{\geq 0}$	The total cost of i.

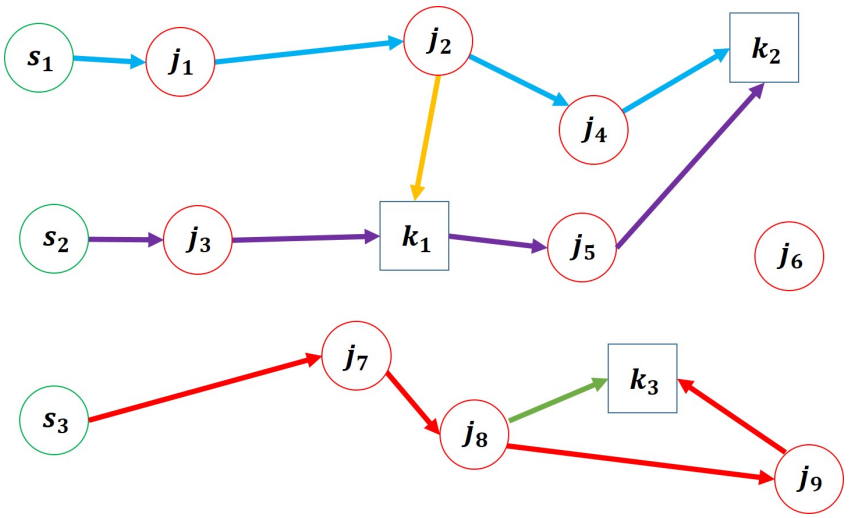


Figure 3. Pre-auction trajectory with $H = 8$: blue arrows, purple arrows, and red arrows represent the trajectory of mobile manipulators i_1, i_2, and i_3, respectively, and yellow and green arrows represent the trajectory of transport agents separated from i_1 and i_3, respectively.

The destination node k_1 is assigned to the objects from j_1 to j_3, k_2 to the objects from j_4 to j_6, and k_3 to those from j_7 to j_9.

3.1. Initializer

Before computing the optimized trajectory, it is necessary to initialize the state of each mobile manipulator i. Algorithm 1 outlines this process, where the initial cost c_i^{init}, starting node s_i, paired transport agent τ_i, and initially loaded objects l_i^{init} are determined.

Algorithm 1 Initializer.

for $i \in I$ **do**
  if predictive nodes exist **then**
    $c_i^{\text{init}} \leftarrow \text{GETINITIALCOST}(s_i^{\text{pred}}, \Pi_i^{\text{prev}})$
    $s_i \leftarrow \text{GETLASTNODEINPREDICTION}(s_i^{\text{pred}}, \Pi_i^{\text{prev}})$
    $\tau_i \leftarrow \text{GETPAIREDTRANSPORTAGENT}(\tau_i^{\text{pred}}, \Pi_i^{\text{prev}})$
    $l_i^{\text{init}} \leftarrow l_i^{\text{pred}}$
  else
    c_i^{init}, s_i, τ_i, $l_i^{\text{init}} \leftarrow 0$, $\mathbf{p}_i^{\text{now}}$, τ_i^{now}, l_i^{now}
end for

If predictive states are available, c_i^{init} is computed based on the predictive node s_i^{pred} in the previous trajectory Π_i^{prev}; s_i is defined as the last accessing node during the prediction period; τ_i is considered paired with i if τ_i^{pred} arrives at the pairing node within the prediction range; and l_i^{init} is set to its predictive value l_i^{pred}. (This process is explained in more detail in Section 4.2).

In contrast, if predictive states are unavailable, c_i^{init} is set to zero; s_i is assigned to the data received at the current position from an embedded GPS $\mathbf{p}_i^{\text{now}}$; and τ_i and l_i^{init} are also set to their current states τ_i^{now} and l_i^{now}, respectively.

3.2. Trajectory Planner

After determining the initial states, the following objective function and constraints are defined to generate the trajectory.

3.2.1. Objective Function

$$\min_{i \in I,\, j \in J} \{w \max_{i \in I} C_i + (1-w) \sum_{i \in I} C_i\} \tag{1}$$

Equation (1) describes the objective function used to minimize the completion time and the total cost of all robots. These two components can be balanced using the weight parameter w. A higher value of w emphasizes minimizing the maximum completion time, while a lower value prioritizes the reduction of the total cost.

3.2.2. Traveling Constraints

$$x_i^{j_a k j_b} = 0,\ x_i^{j_a j_b} = 0,\ x_i^{k j_a j_b} = 0, \qquad \text{where } j_a = j_b, \forall i \in I \tag{2}$$

$$x_i^{j_a k j_b} x_i^{j_b k j_a} = 0,\ x_i^{j_a j_b} x_i^{j_b j_a} = 0,\ x_i^{k j_a j_b} x_i^{k j_b j_a} = 0, \qquad \forall i \in I \tag{3}$$

$$x_{\neg i_p}^{j_a j_b} = 0,\ x_{\neg i_p}^{k j_a j_b} = 0, \qquad \forall i_p \in I_p \tag{4}$$

$$x_{i_p}^{j_a j_b} = 0, \qquad \forall i_p \in I_p, \qquad \text{where } k_{j_a} \neq k_{j_b} \tag{5}$$

$$x_{i_p}^{s j_a} x_{i_p}^{j_a k j_b} = 0,\ \sum_{j_a, j_b \in J} x_{i_p}^{k j_a j_b} \leq 1, \qquad \forall i_p \in I_p \tag{6}$$

Equations (2) and (3) show that the mobile manipulators cannot stop at a specific node and cannot return to the previously visited node.

Equations (4) and (5) describe the conditions when a mobile manipulator travels directly between object nodes. Equation (4) describes that performing such direct travels is restricted for the mobile manipulators not paired with the transport agents. However, even if paired, as described in Equation (5), object nodes j_a and j_b should have the same destination node in performing direct travels without separation.

Equation (6) describes the conditions for the mobile manipulators paired with transport agents: they cannot travel to an object node via a destination node immediately after its first movement. Additionally, separating tasks can only be performed once for the paired mobile manipulators.

3.2.3. Manipulation Constraints

$$\sum_{j\in J} x_i^{sj} \leq 1, \qquad \forall i \in I \tag{7}$$

$$\sum_{i\in I} x_i^{jk} \leq 1, \qquad \forall j \in J \tag{8}$$

$$x_i^{sj} = x_i^{sj} \sum_{j'\in J} x_i^{j'k}, \qquad \forall i \in I, \forall j \in J \tag{9}$$

$$\{\sum_{i\in I} x_i^{sj_a} + \sum_{i\in I}\sum_{j_b\in J}(x_i^{j_bj_a} + x_i^{kj_bj_a} + x_i^{j_bkj_a})\} \leq 1, \qquad \forall j_a \in J \tag{10}$$

$$\sum_{j_a\in J}\{\sum_{i\in I} x_i^{sj_a} + \sum_{i\in I}\sum_{j_b\in J}(x_i^{j_bj_a} + x_i^{kj_bj_a} + x_i^{j_bkj_a})\} = H \tag{11}$$

Equations (7)–(9) describe that if a mobile manipulator starts a task, it can only execute one starting task x_i^{sj} and one final task x_i^{jk}. However, these constraints alone may result in multiple robots performing the same task. This issue is addressed by Equation (10), which prevents such overlaps by enforcing exclusivity in task execution. Moreover, Equation (10) is extended to Equation (11) by summing over all object nodes, thereby defining the horizon that the number of objects handled in a single planning. In Equation (11), the horizon is determined based on available computing power.

3.2.4. Connecting Constraints

$$x_i^{sj_a} = x_i^{sj_a}\{\sum_{j_b\in J}(x_i^{kj_aj_b} + x_i^{j_aj_b} + x_i^{j_akj_b}) + x_i^{j_ak}\}, \qquad \forall j_a \in J, \forall i \in I \tag{12}$$

$$x_i^{j_ak} = x_i^{j_ak}\{\sum_{j_b\in J}(x_i^{kj_bj_a} + x_i^{j_bj_a} + x_i^{j_bkj_a}) + x_i^{sj_a}\}, \qquad \forall j_a \in J, \forall i \in I \tag{13}$$

$$x_i^{j_akj_b} = x_i^{j_akj_b}\{x_i^{sj_a} + \sum_{j_c\in J}(x_i^{kj_cj_a} + x_i^{j_ckj_a})\}, \qquad \forall j_a \in J, \forall i \in I \tag{14}$$

$$x_i^{j_bkj_a} = x_i^{j_bkj_a}\{x_i^{j_ak} + \sum_{j_c\in J} x_i^{j_akj_c}\}, \qquad \forall j_a \in J, \forall i \in I \tag{15}$$

$$x_i^{j_aj_b} = x_i^{j_aj_b}\{x_i^{sj_a} + \sum_{j_c\in J} x_i^{j_cj_a}\}, \qquad \forall j_a \in J, \forall i \in I \tag{16}$$

$$x_i^{j_bj_a} = x_i^{j_bj_a}\{x_i^{j_ak} + \sum_{j_c\in J}(x_i^{j_aj_c} + x_i^{kj_aj_c})\}, \qquad \forall j_a \in J, \forall i \in I \tag{17}$$

$$x_i^{kj_aj_b} = x_i^{kj_aj_b}\{x_i^{sj_a} + \sum_{j_c\in J} x_i^{j_cj_a}\}, \qquad \forall j_a \in J, \forall i \in I \tag{18}$$

$$x_i^{kj_bj_a} = x_i^{kj_bj_a}\{x_i^{j_ak} + \sum_{j_c\in J} x_i^{j_akj_c}\}, \qquad \forall j_a \in J, \forall i \in I \tag{19}$$

Equations (12)–(19) define the connecting constraint that connected edges should share the same node. These constraints enforce the continuity of movement by requiring the connections between nodes to align with respective task assignments. Additionally,

Equation (19) specifies that if the mobile manipulators separate from their paired transport agents, they are no longer allowed to move directly between object nodes.

3.2.5. Ordering and Capacity Constraints

$$o_i^{j_b} = o_i^{j_a} + 1 - n(J)(1 - x_i^{j_a j_b} - x_i^{j_a k j_b}), \qquad \forall j_a \in J, \forall i \in I \tag{20}$$

$$\sum_{j_a \in J} \sum_{j_b \in J} x_i^{j_a j_b} < (L^{\max} - l_i^{\text{init}}), \qquad \forall i \in I \tag{21}$$

Equation (20) defines the ordering constraints to prevent the generation of infinite loops in the trajectory (e.g., $x_i^{j_a k j_b} = 1$, $x_i^{j_b k j_c} = 1$, and $x_i^{j_c k j_a} = 1$), by assigning an execution order using the Miller–Tucker–Zemlin (MTZ) formulation [24].

Additionally, since the number of loaded objects increases by one whenever the robot travels directly between two object locations, the number of such direct travels for i is bounded by its maximum carrying capacity and the number of initially loaded objects, as expressed in Equation (21).

3.2.6. Total Cost

$$C_i = c_i^{\text{init}} + \sum_{j \in J} c_i^{sj} x_i^{sj} + \sum_{j \in J} c^{jk} x_i^{jk} + \sum_{j_a, j_b \in J} (c^{j_a k} + c^{k j_b}) x_i^{j_a k j_b} + \sum_{j_a, j_b \in J} c^{j_a j_b} x_i^{j_a j_b} + \sum_{j_a, j_b \in J} (w_\tau c^{j_a k} + c^{j_a j_b}) x_i^{k j_a j_b}, \qquad \forall i \in I \tag{22}$$

Equation (22) represents the deriving total cost of i. When $x_i^{k j_a j_b}$ is performed, the cost of the separated transport agent is adjusted by weight w_τ.

4. Post-Auction Trajectory Planning

After generating the pre-auction trajectory, each transport agent selects a new pairing partner and node and regenerates its trajectory accordingly.

4.1. Auction-Based Pairing Planner

Algorithm 2 shows the overall process. After initializing the best pairing candidate set P_{best} to an empty state, the set of pairing candidates P is generated from the pre-auction trajectory Π^{pre}. Each candidate comprises a transport agent separated from its original partner, a target mobile manipulator, and a pairing index indicating a node on the target manipulator's trajectory.

Algorithm 2 Pairing planner.

$P_{\text{best}} \leftarrow \emptyset, P \leftarrow \text{GetPairingCandidate}(\Pi^{\text{pre}})$
while $P \neq \emptyset$ **do**
  p_{best}, $J_{\text{best}}^{\text{done}} \leftarrow \text{SelectBestPair}(P, P_{\text{best}}, \Pi^{\text{pre}}, J, H)$
  $P \leftarrow \text{ExcludeSelectedAgents}(P, p_{\text{best}}.i, p_{\text{best}}.\tau)$
  $J \leftarrow J - J_{\text{best}}^{\text{done}}, H \leftarrow H - n(J_{\text{best}}^{\text{done}})$,
  $P_{\text{best}} \leftarrow P_{\text{best}} + p_{\text{best}}$
end while
C^{init}, $S^{\text{new}} \leftarrow \text{GetInitialStateAfterPairing}(P_{\text{best}})$
$\Pi^{\text{new}} \leftarrow \text{TrjPlanner}(P_{\text{best}}, C^{\text{init}}, S^{\text{new}}, H, J)$
$\Pi^{\text{post}} \leftarrow \text{ReplaceTrj}(\Pi^{\text{pre}}, S^{\text{new}}, \Pi^{\text{new}})$

The algorithm iteratively selects the best available candidate until all separated transport agents in P are paired with new mobile manipulators. Once the best candidate p_{best} is identified, the robots included in p_{best} and the objects handled at SelectBestPair $J_{\text{best}}^{\text{done}}$

are excluded from further consideration. The process is then repeated with the remaining pairings to determine the next optimal pairing state.

After all pairings are finalized, the new pairing trajectory Π^{new} is generated from new starting nodes S^{new}. By replacing the pre-auction trajectory after S^{new} to Π^{new}, the post-auction trajectory Π^{post} is generated.

4.1.1. Pairing Candidates

Algorithm 3 demonstrates the procedure for generating pairing candidates. For each transport agent τ and mobile manipulator i, the algorithm evaluates every node π_i^n along i's trajectory Π_i. If i is initially unpaired, the evaluation begins from the first node; otherwise, it starts from the separation index n_{sep}.

Algorithm 3 GETPAIRINGCANDIDATE

$P \leftarrow \emptyset$
for $\tau \in T,\ i \in I$ **do**
  $P_{i,\tau} \leftarrow \emptyset$
  for $n = 0, 1, ..., N_{\Pi_i} - 1$ **do**
    if $i \in \neg I_p$ **or** $n \geq n_{\text{sep}}$ **then**
      $t_\tau^{\pi_i^n} \leftarrow t_\tau^{\text{pre-sep}} + t_\tau^{j_{\text{sep}},k_{\text{sep}}} + t_{\text{handle}} + t_\tau^{k_{\text{sep}},\pi_i^n}$
      if $t_\tau^{\pi_i^n} \leq t_i^{\pi_i^n} + t_A$ **then**
        $P_{i,\tau} \leftarrow P_{i,\tau} + (\tau, i, n, t_\tau^{\pi_i^n})$
      end if
    end if
  end for
  $P \leftarrow P +$ FILTERFASTESTMEETINGPAIRS$(P_{i,\tau}, N)$
end for
return P

At each node, a pairing with τ is considered feasible if the total estimated time for the transport agent to arrive—including the pre-separation time $t_\tau^{\text{pre-sep}}$, travel time from the separation point to the destination node $t_\tau^{j_{\text{sep}},k_{\text{sep}}}$, object handling time t_{handle}, and travel time from the destination node to the n-th node on the trajectory $t_\tau^{k_{\text{sep}},\pi_i^n}$—does not exceed the sum of the arrival/handling time of the target mobile manipulator at the n-th node $t_i^{\pi_i^n}$, and the allowable waiting time t_A.

Among all feasible candidates in $P_{i,\tau}$, the top N candidates with the earliest meeting times $t_\tau^{\pi_i^n}$ are selected using FILTERFASTESTMEETINGPAIRS, and the corresponding values (τ, i, n) are added to the global candidate set P.

4.1.2. Selecting the Best Pair with Auction

To determine the optimal pairs, each candidate is evaluated using the process described in Algorithm 4.

In each evaluation, i in the subject candidate p is assumed to be paired with a transport agent at the n-th node on its pre-auction trajectory $\Pi_{p.i}^{\text{pre}}$. Additionally, the previously selected candidates P_{best} are also assumed to be paired at their respective pairing nodes. Moreover, $C_{\text{eval}}^{\text{init}}$ and S_{eval} represent the initial cost and starting positions of all agents, and $J_p.i^{\text{done}}$ represents the set of objects already handled by $p.i$ before the n-th node, and they are derived using GETINITIALSTATES.

Figure 4 shows an example of how the initial states are set when evaluating the subject candidate p_1. In this example, the mobile manipulator i_1 is paired with a transport agent at node j_2. Other relevant robots, such as i_2 and i_3, are also considered in the evaluation: i_2 is a non-subject mobile manipulator, and i_3 is a previously selected mobile manipulator

that is paired with a transport agent at node j_5. When i_1 arrives at j_2, each of these robots is positioned at its respective coincident node, which is marked with a red line in the figure.

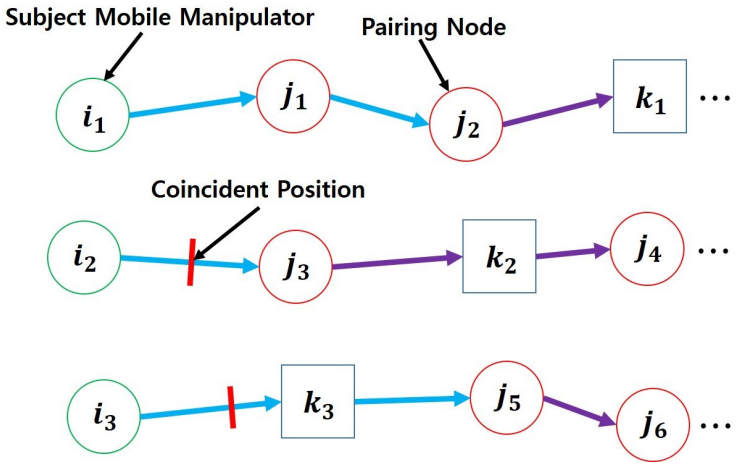


Figure 4. Evaluation of the subject candidate: Initial cost trajectories are depicted with blue arrows, while evaluation trajectories are represented with purple arrows.

Algorithm 4 SELECTBESTPAIR

$C_{\text{eval}} \leftarrow \emptyset$
for $p \in P$ **do**
  $\text{SETPAIRINGS}(p.i, p.n, P_{\text{best}}, \Pi^{\text{pre}}_{p.i})$
  $C^{\text{init}}_{\text{eval}}, S^{\text{eval}}, J^{\text{done}}_{p.i} \leftarrow \text{GETINITIALSTATES}(p, P_{\text{best}}, \Pi^{\text{pre}})$
  $J_{\text{eval}} \leftarrow J - J^{\text{done}}_{p.i}, \ H_{\text{eval}} \leftarrow H - n(J^{\text{done}}_{p.i})$
  $\Pi_p \leftarrow \text{TRJPLANNER}(p, P_{\text{best}}, C^{\text{init}}_{\text{eval}}, S_{\text{eval}}, H_{\text{eval}}, J_{\text{eval}})$
  $C_{\text{eval}} \leftarrow C_{\text{eval}} + \Pi_p.c$
end for
$p_{\text{best}}, \ J^{\text{done}}_{\text{best}} \leftarrow \text{GETMINCOSTPAIR}(C^{\text{eval}})$
return $p_{\text{best}}, \ J^{\text{done}}_{\text{best}}$

In this scenario, j_1 is the object handled by the subject candidate, and j_2 becomes the starting node for i_1. For each non-subject agent, the starting node is defined as the first node on its trajectory after passing the coincident point. However, if the agent belongs to a previously selected candidate, its pairing node is used as its starting point.

Based on these updated starting nodes, the initial cost of each candidate is calculated as the sum of travel and handling times from their original starting node to the updated one, presented with blue arrows.

After defining the initial states, the candidate trajectory Π_p is generated using TRJPLANNER (Section 3.2), assuming that non-subject mobile manipulators not in the previously selected set are unpaired. The resulting cost $\Pi_p.c$ is then computed from Π_p and stored in the evaluation set C_{eval}.

This evaluation process is repeated for all candidates. Among them, the candidate with the lowest cost is selected as the best pair p_{best}, and its handled object set $J^{\text{done}}_{\text{best}}$ is excluded from consideration in the subsequent sequence auctions.

4.1.3. Post-Auction Trajectory

The mobile manipulators included in P_{best} are designated as paired at their respective pairing nodes. Conversely, those not included in P_{best} are designated as unpaired, starting from their separating node if they were initially paired or from their original starting node if they were initially unpaired.

At these new starting nodes S^{new}, the cost from each robot's original starting node to its new starting node is set as the initial cost C^{init}. Additionally, the object nodes between these starting nodes are omitted when planning the new trajectory.

With these initial states, the trajectory Π^{new} is generated using TRJPLANNER, and the post-auction trajectory Π^{post} is obtained by replacing the segment after S^{new} in Π^{pre} with Π^{new}.

4.2. Predictor

After generating the post-auction trajectory, we can create a new trajectory from the points concerning the specific time that has passed in reference to the previous starting points. As shown in Figure 5, the starting nodes move through the trajectory for the predefined prediction time, which is indicated with a dotted arrow, and the predictive starting node is set as s_i^{pred}. From these nodes, a new trajectory is generated.

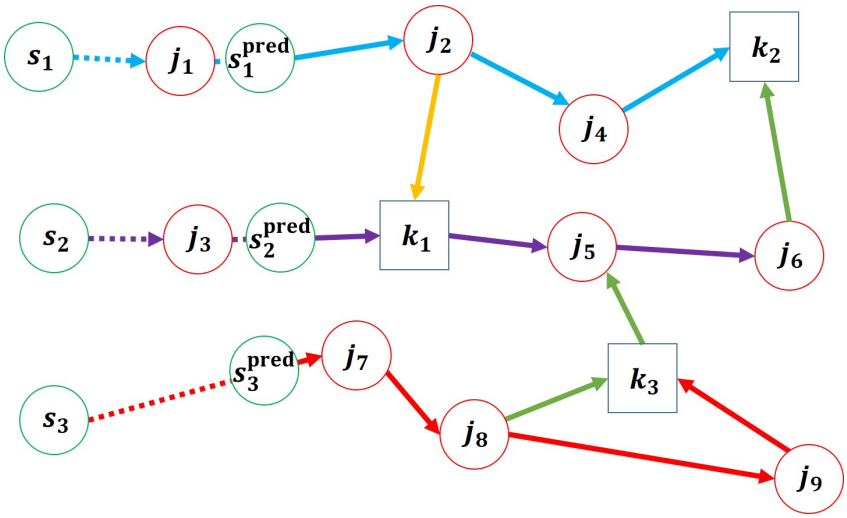


Figure 5. Post-auction trajectory generated from new starting nodes.

As performed in Section 4.1.2, each mobile manipulator designates its last accessing node during the prediction time as its predictive initial accessing nodes, and the sum of the traveling time and handling time from each predictive starting node to the initial accessing node is defined as the initial cost c_i^{init}. For instance, in Figure 5, j_2, k_1, and j_7 are assigned as the new starting nodes for i_1, i_2, and i_3, respectively, and the costs from each s_i^{pred} to its corresponding starting node are set as c_i^{init}.

If the initial accessing node for i is an object node j, then x_i^{sj} is set to 1. In addition, object nodes handled by each paired robot within the prediction range are added to its initially loaded objects l_i^{init} in the new trajectory. As shown in Figure 5, j_1 and j_3 are added to the loaded objects of i_1 and i_2, respectively, and $x_{i_1}^{sj_2}$ and $x_{i_3}^{sj_7}$ are set to 1.

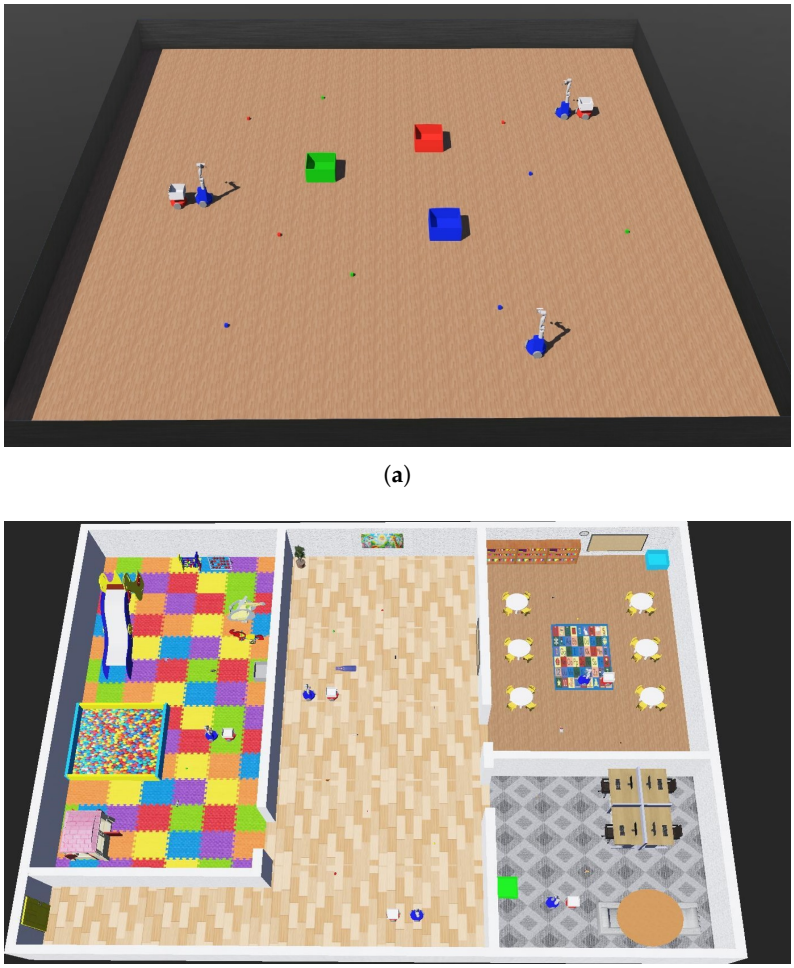
5. Experiments

The experiments were conducted using Webots [25], an open-source robotic simulator integrated with ROS2 [26].

To generate navigation trajectories, we employed the A* [27] algorithm to compute travel paths and estimate the costs between waypoints, including object positions, destinations, and robot locations. Based on the computed costs, the Gurobi Optimizer [28] was used to generate optimized trajectories as described in Section 3.2, where the weight parameter w was set to 0.7. In addition, ORCA (Optimal Reciprocal Collision Avoidance) [29] was applied during navigation to prevent collisions between robots.

The experiments are designed to evaluate the proposed system from two perspectives—its scalability in different task settings, and the impact of the auction-based pairing algorithm on overall performance.

In evaluating its scalability, the experiments were conducted in two environments: a simplified environment and a kindergarten-themed environment—both shown in Figure 6. The simplified environment consists of cube-shaped objects and three destinations without obstacles. In contrast, the kindergarten-themed environment includes objects commonly found in kindergartens, four destinations, and various obstacles.



(**a**)

(**b**)

Figure 6. Simulation environment used in the experiments: (**a**) simplified environment; (**b**) kindergarten-themed environment.

Meanwhile, to evaluate the effectiveness of the auction-based pairing algorithm, we compared the proposed system to a baseline configuration that does not include this mechanism. In the baseline system, each mobile manipulator operates independently with fixed storage, and no dynamic role reassignment with transport agents occurs.

Both experiments were conducted under varying numbers of objects N_j and transport agents N_τ, assuming $L^{\max} = 2$ for all transport agents.

For a visual representation of the experiments, all corresponding video recordings can be accessed at the following link: (https://youtube.com/playlist?list=PLB1pUAsYGpRHK9mhHIeu827JLSoLhcznD&si=97sXqhfHEqkMkMc0, accessed on 20 May 2025).

5.1. Scalability Evaluation

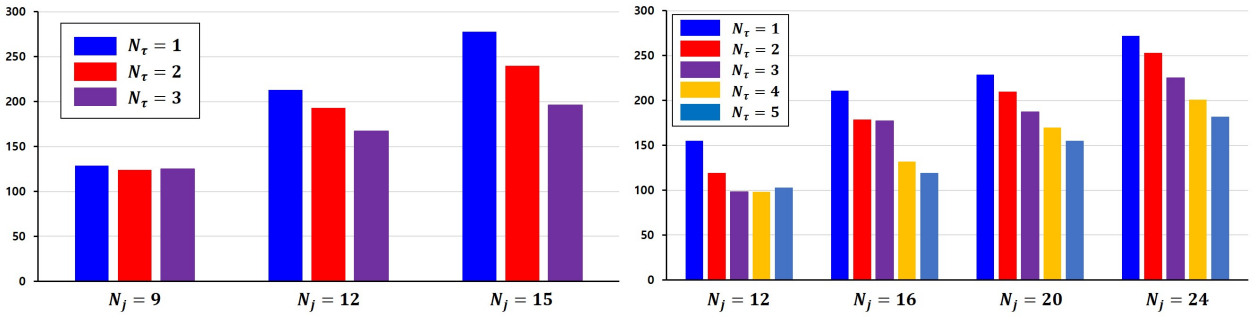
5.1.1. Experimental Scenarios

To evaluate the scalability of the proposed system, we test its performance under two different environments:

- **Simplified Environment**: A minimal environment shown in Figure 6a, where three mobile manipulators and a varying number of transport agents collaborate to tidy up the workspace by placing cube-shaped objects into the destination zones of matching colors.
- **Kindergarten-Themed Environment**: A realistic environment shown in Figure 6b, which consists of four distinct areas: a playroom, a corridor, a classroom, and a teacher's room. In this environment, five mobile manipulators and a varying number of transport agents work together to tidy up four different types of objects—balls, toys, teaching aids, and stationery—by delivering them to their designated locations. To ensure consistent handling and loading behaviors, all objects were standardized to have a weight of 50 g and were encapsulated in a cubic bounding box with a side length of 0.07 m.

5.1.2. Performance Comparison

We measured task completion time in both environments while increasing N_j and N_τ. The results, summarized in Figure 7 and Table 2, show that the proposed system scales well in both structured and complex environments.



(**a**) (**b**)

Figure 7. Changes in task completion time depend on the number of transport agents and objects: (**a**) simplified environment; (**b**) kindergarten-themed environment.

In the simplified environment, the system shows relatively minor gains when N_j is small. For example, at $N_j = 9$, increasing N_τ from 2 to 3 results in only a 1-second increase, suggesting possible coordination overhead. However, as the number of objects increases to $N_j = 15$, the benefit becomes more evident, with up to 44 s of reduction when increasing N_τ from 2 to 3.

In the kindergarten-themed environment, which introduces additional complexity, such as obstacles and spatial constraints, similar trends are observed. For $N_j = 12$, increasing the number of transport agents from one to two reduces the completion time significantly by 36 s. A further increase to three units continues to reduce the time to 98 s. However, increasing N_τ from 3 to 4 does not yield any improvement, and adding a fifth transport agent increases the completion time by 5 s, possibly due to the coordination overhead in a low-demand scenario.

For larger workloads, $N_j = 20$ and $N_j = 24$, the performance improves more consistently as the number of transport agents increases, following a similar trend observed in the simplified environment. For $N_j = 20$, task completion time drops from 229 s with one unit

to 155 s with five units, with the average decrease being 18.5 s per unit. A similar pattern is observed at $N_j = 24$, where time reduces from 272 to 182 s as the number of transport agents increases, with the average decrease being 22.5 s per unit.

These results demonstrate that the proposed system scales effectively in both simplified and realistic settings, and that the degree of performance improvement becomes more substantial as the task complexity and workload increase.

Table 2. Detailed task completion times by the number of transport agents and objects.

N_τ	Simplified Environment			Kindergarten-Themed Environment			
	$N_j = 9$	$N_j = 12$	$N_j = 15$	$N_j = 12$	$N_j = 16$	$N_j = 20$	$N_j = 24$
1	129	213 s	278 s	155 s	211 s	229 s	272 s
2	124 s	193 s	240 s	119 s	179 s	210 s	253 s
3	125 s	167 s	196 s	98 s	177 s	187 s	225 s
4		–		98 s	132 s	170 s	201 s
5		–		103 s	119 s	155 s	182 s

5.2. Effectiveness of Auction-Based Pairing

5.2.1. Experimental Scenarios

To evaluate the effectiveness of the auction-based pairing algorithm, we define two experimental scenarios:

- **With auction-based pairing**: Robots use the proposed dynamic pairing strategy, where transport agents and mobile manipulators are paired through an auction-based algorithm. Pairings can change dynamically based on task status and robot availability.
- **Without auction-based pairing**: Robots operate without the auction-based pairing mechanism. Instead of dynamic pairing, each mobile manipulator, assumed to be paired with a transport agent, is equipped with a container that has the same $L^{\max}$ as a transport agent and performs the task without any pairing mechanism. The trajectory planner employed in this work is based on the approach proposed in [14].

Both scenarios are tested under identical environment settings, including object placements and robot configurations. Task completion time is measured and compared to analyze the effectiveness of the auction-based pairing strategy.

5.2.2. Performance Comparison

To assess the effectiveness of the proposed auction-based pairing strategy, we compare its performance against a baseline configuration in which each mobile manipulator is equipped with a fixed container and no dynamic pairing is allowed. Both systems are evaluated in the kindergarten-themed environment.

As shown in Figure 8 and Table 3, the auction-based system achieves better performance in most settings, particularly when the number of transport agents increases.

For example, at $N_j = 12$, the completion time decreases from 156 s to 103 s as N_τ increases from 1 to 5 without an auction-based pairing; with the proposed method, the time drops from 155 s to 98 s over the same range.

At $N_j = 16$, the difference becomes even more pronounced. The baseline shows a reduction of 32 s (from 191 s to 159 s), while the auction-based method achieves a 92 s reduction (from 211 s to 119 s), demonstrating more effective resource utilization under moderate task loads.

Even in higher workloads, the auction-based method consistently maintains a more significant margin of improvement. When $N_j = 24$, the completion time drops from 274 s to 218 s without an auction-based pairing and from 272 s to 182 s with an auction-based pairing, yielding a 36 s advantage.

These results indicate that the proposed auction-based pairing enables better adaptation to varying task distributions. By allowing dynamic reallocation of roles between mobile manipulators and transport agents, the system achieves consistently shorter completion times across different conditions.

Table 3. Detailed task completion times by the number of transport agents and objects.

N_τ	Without Auction-Based Pairing				With Auction-Based Pairing			
	$N_j = 12$	$N_j = 16$	$N_j = 20$	$N_j = 24$	$N_j = 12$	$N_j = 16$	$N_j = 20$	$N_j = 24$
1	156 s	191 s	229 s	274 s	155 s	211 s	229 s	272 s
2	150 s	185 s	212 s	262 s	119 s	179 s	210 s	253 s
3	137 s	180 s	205 s	246 s	98 s	177 s	187 s	225 s
4	138 s	170 s	191 s	234 s	98 s	132 s	170 s	201 s
5	136 s	159 s	176 s	218 s	103 s	119 s	155 s	182 s

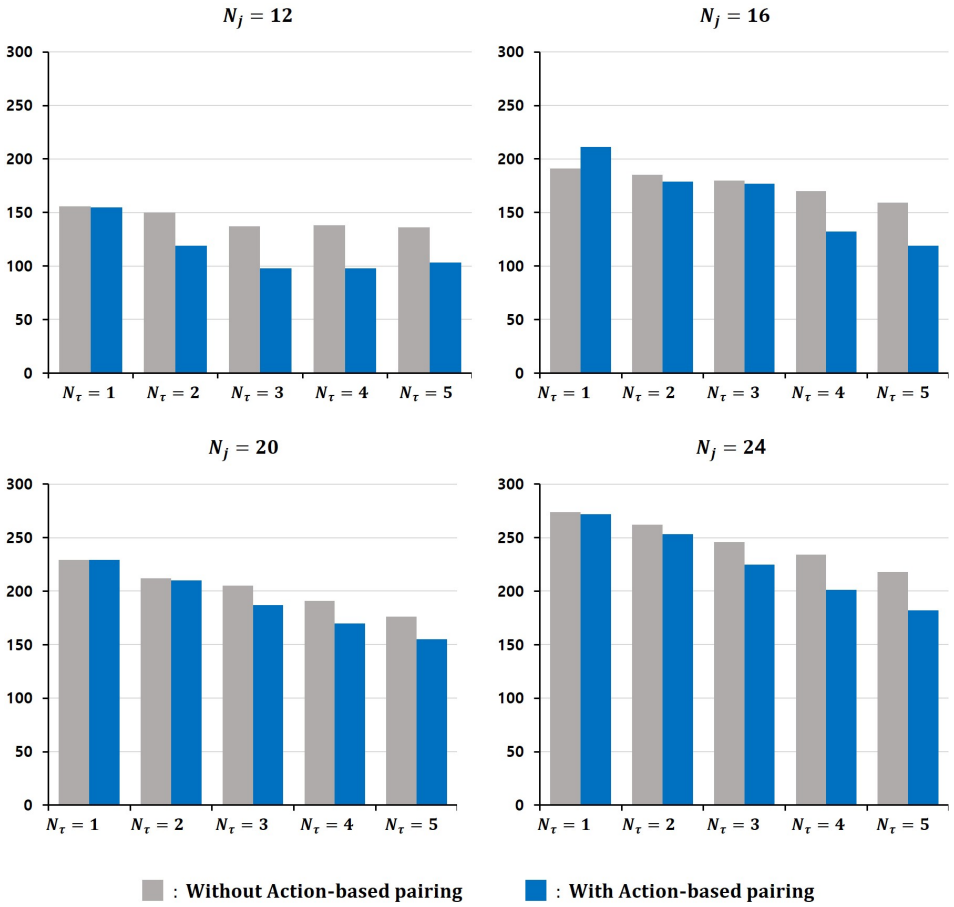


Figure 8. Visual comparison of task completion times according to the number of transport agents and objects.

6. Discussion

This study proposes an MAPD system that enables efficient collaboration between mobile manipulators and transport agents. Unlike conventional approaches that treat heterogeneous agents separately, our system integrates trajectory optimization with auction-based dynamic pairing to support flexible role adjustment and synergy.

Experimental results validate the approach, showing that the proposed method improves task completion time, especially in high-demand situations. These improvements highlight the value of strategic dynamic pairing and partial trajectory planning in optimizing heterogeneous multi-robot collaboration.

However, the current system assumes that all transport agents have identical capabilities and ignores energy limitations. These assumptions simplify the problem but may limit applicability in real-world deployments. Future improvements will focus on addressing these constraints by incorporating robot heterogeneity and energy-aware task allocation.

7. Conclusions

This work presented an MAPD framework that enables heterogeneous multi-robot collaboration through dynamic pairing and partial trajectory planning. By integrating an auction-based planner with trajectory optimization, the system achieves improved task efficiency and flexibility. Experimental validation in both simplified and realistic environments confirmed the effectiveness of the proposed method under various workloads.

For future work, we plan to extend the system to consider energy-aware scheduling [30–32], which is essential for real-world robot operation, by incorporating battery levels and standby agents in charging stations, allowing for dynamic deployment based on resource availability and demand. Furthermore, the current system assumes uniform robot capabilities. Our future research will explore heterogeneous agents with varying speeds and capacities to enhance generalizability.

Author Contributions: Conceptualization, J.-B.Y. and S.-J.Y.; methodology, J.-B.Y.; software, J.-B.Y.; validation, J.-B.Y., J.K., S.N., and D.S.; formal analysis, J.-B.Y.; investigation, J.-B.Y.; resources, S.-J.Y.; data curation, J.-B.Y.; writing—original draft preparation, J.-B.Y.; writing—review and editing, J.-B.Y., J.K, S.N., D.S., and S.-J.Y.; visualization, J.-B.Y.; supervision, S.-J.Y.; project administration, S.-J.Y.; funding acquisition, S.-J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by BK21FOUR, Creative Human Resource Education and Research Programs for ICT Convergence in the 4th Industrial Revolution and 2-Year Research Grant of Pusan National University.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MAPD	Multi-Agent Pickup and Delivery
MTZ	Miller–Tucker–Zemlin
ROS	Robot operating system
A*	A-star
ORCA	Optimal Reciprocal Collision Avoidance

References

1. Li, J.T. Design Optimization of Amazon Robotics. *Autom. Control Intell. Syst.* **2016**, *4*, 48. [CrossRef]
2. Farinelli, A.; Boscolo, N.; Zanotto, E.; Pagello, E. Advanced approaches for multi-robot coordination in logistic scenarios. *Robot. Auton. Syst.* **2017**, *90*, 34–44.
3. Fu, B.; Smith, W.; Rizzo, D.M.; Castanier, M.; Ghaffari, M.; Barton, K. Robust Task Scheduling for Heterogeneous Robot Teams Under Capability Uncertainty. *IEEE Trans. Robot.* **2023**, *39*, 1087–1105. [CrossRef]
4. Osooli, H.; Robinette, P.; Jerath, K.; Azadeh, R. A Multi-Robot Task Assignment Framework for Search and Rescue with Heterogeneous Teams. *arXiv* **2023**, arXiv:2309.12589.
5. Li, L.; Chen, Z.; Wang, H.; Kan, Z. Fast Task Allocation of Heterogeneous Robots with Temporal Logic and Inter-Task Constraints. *IEEE Robot. Autom. Lett.* **2023**, *8*, 4991–4998. [CrossRef]

6. Wu, Y.; Gao, Y.; Song, W. Multi-robot Task Assignment Algorithm for Medical Service System. In Proceedings of the 2022 IEEE International Conference on Robotics and Biomimetics (ROBIO), Jinghong, China, 5–9 December 2022; pp. 1204–1209. [CrossRef]
7. Dai, L.L.; Pan, Q.K.; Miao, Z.H.; Suganthan, P.N.; Gao, K.Z. Multi-Objective Multi-Picking-Robot Task Allocation: Mathematical Model and Discrete Artificial Bee Colony Algorithm. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 6061–6073. [CrossRef]
8. Ju, C.; Kim, J.; Seol, J.; Son, H.I. A review on multirobot systems in agriculture. *Comput. Electron. Agric.* **2022**, *202*, 107336. [CrossRef]
9. Chen, Z.; Alonso-Mora, J.; Bai, X.; Harabor, D.D.; Stuckey, P.J. Integrated Task Assignment and Path Planning for Capacitated Multi-Agent Pickup and Delivery. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5816–5823. [CrossRef]
10. Xu, G.; Wu, Y.; Tao, S.; Yang, Y.; Liu, T.; Huang, T.; Wu, H.; Liu, Y. Multi-robot Task Allocation and Path Planning with Maximum Range Constraints. *arXiv* **2024**, arXiv:2409.06531.
11. Zhang, Y.; Bastani, F.B.; Yen, I.L. Decentralized Path Planner for Multi-robot Systems. In Proceedings of the 2009 21st IEEE International Conference on Tools with Artificial Intelligence, Newark, NJ, USA, 2–4 November 2009; pp. 171–175. [CrossRef]
12. Hou, J.; Zhou, X.; Gan, Z.; Gao, F. Enhanced Decentralized Autonomous Aerial Robot Teams with Group Planning. *IEEE Robot. Autom. Lett.* **2022**, *7*, 9240–9247. [CrossRef]
13. Cecchi, M.; Paiano, M.; Mannucci, A.; Palleschi, A.; Pecora, F.; Pallottino, L. Priority-Based Distributed Coordination for Heterogeneous Multi-Robot Systems with Realistic Assumptions. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6131–6138. [CrossRef]
14. Camisa, A.; Testa, A.; Notarstefano, G. Multi-Robot Pickup and Delivery via Distributed Resource Allocation. *IEEE Trans. Robot.* **2021**, *39*, 1106–1118. [CrossRef]
15. Wang, H.; Chen, W.; Wang, J. Coupled task scheduling for heterogeneous multi-robot system of two robot types performing complex-schedule order fulfillment tasks. *Robot. Auton. Syst.* **2020**, *131*, 103560. [CrossRef]
16. Wang, H.; Chen, W. Task scheduling for transport and pick robots in logistics: A comparative study on constructive heuristics. *Auton. Intell. Syst.* **2021**, *1*, 17. [CrossRef]
17. Wang, H.; Chen, W. Task scheduling for heterogeneous agents pickup and delivery using recurrent open shop scheduling models. *Robot. Auton. Syst.* **2024**, *172*, 104604. [CrossRef]
18. Thilagavathy, R.; Sumithradevi, K. ROS-Enabled Collaborative Navigation and Manipulation with Heterogeneous Robots. *SN Comput. Sci.* **2023**, *4*. [CrossRef]
19. Aswale, A.; Pinciroli, C. Heterogeneous Coalition Formation and Scheduling with Multi-Skilled Robots. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 5402–5409. [CrossRef]
20. Camisa, A.; Notarnicola, I.; Notarstefano, G. A Primal Decomposition Method with Suboptimality Bounds for Distributed Mixed-Integer Linear Programming. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami, FL, USA, 17–19 December 2018; pp. 3391–3396. [CrossRef]
21. Camisa, A.; Notarnicola, I.; Notarstefano, G. Distributed Primal Decomposition for Large-Scale MILPs. *IEEE Trans. Autom. Control* **2022**, *67*, 413–420. [CrossRef]
22. Vujanic, R.; Esfahani, P.; Goulart, P.; Mariethoz, S.; Morari, M. A Decomposition Method for Large Scale MILPs, with Performance Guarantees and a Power System Application. *Automatica* **2014**, *67*, 144–156. [CrossRef]
23. Park, S.; Zhong, Y.D.; Leonard, N.E. Multi-Robot Task Allocation Games in Dynamically Changing Environments. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 8678–8684. [CrossRef]
24. Sawik, T. A note on the Miller-Tucker-Zemlin model for the asymmetric traveling salesman problem. *Pol. Akad. Nauk. Pol. Acad. Sci.* **2016**, *64*, 517–520. [CrossRef]
25. Michel, O. Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation. *Int. J. Adv. Robot. Syst.* **2004**, *1*, 5.
26. Macenski, S.; Foote, T.; Gerkey, B.; Lalancette, C.; Woodall, W. Robot Operating System 2: Design, Architecture, and Uses In The Wild. *arXiv* **2022**. [CrossRef] [PubMed]
27. Hart, P.; Nilsson, N.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
28. Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*; Gurobi Optimization, LLC: Beaverton, OR, USA, 2024.
29. van den Berg, J.P.; Guy, S.J.; Lin, M.C.; Manocha, D. Reciprocal n-Body Collision Avoidance. In *Robotics Research, Proceedings of the 14th International Symposium ISRR, Lucerne, Switzerland, 31 August–3 September 2009*; Springer: Berlin/Heidelberg, Germany, 2011.
30. Bagchi, M.J.; Nair, S.B.; Das, P.K. On a dynamic and decentralized energy-aware technique for multi-robot task allocation. *Robot. Auton. Syst.* **2024**, *180*, 104762. [CrossRef]

31. Mokhtari, M.; Mohamadi, P.H.A.; Aernouts, M.; Singh, R.K.; Vanderborght, B.; Weyn, M.; Famaey, J. Energy-aware multi-robot task scheduling using meta-heuristic optimization methods for ambiently-powered robot swarms. *Robot. Auton. Syst.* **2025**, *186*, 104898. [CrossRef]
32. Fouad, H.; Beltrame, G. Energy Autonomy for Resource-Constrained Multi Robot Missions. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 7006–7013. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.