

Research Article

Allocating Multiple Types of Tasks to Heterogeneous Agents Based on the Theory of Comparative Advantage

Toma Morisawa, Kotaro Hayashi , and Ikuo Mizuuchi

Department of Mechanical Systems Engineering, Tokyo University of Agriculture & Technology, 2-24-16 Nakacho, Koganei, Tokyo 184-8588, Japan

Correspondence should be addressed to Kotaro Hayashi; hayashik@cc.tuat.ac.jp

Received 10 November 2017; Revised 13 February 2018; Accepted 2 April 2018; Published 22 May 2018

Academic Editor: Yunyi Jia

Copyright © 2018 Toma Morisawa et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a method to allocate multiple tasks with uncertainty to heterogeneous robots using the theory of comparative advantage: an economic theory that maximizes the benefit of specialization. In real applications, robots often must execute various tasks with uncertainty and future multirobot system will have to work effectively with people as a team. As an example, it may be necessary to explore an unknown environment while executing a main task with people, such as carrying, rescue, military, or construction. The proposed task allocation method is expected to reduce the total makespan (total length of task-execution time) compared with conventional methods in robotic exploration missions. We expect that our method is also effective in terms of calculation time compared with the time-extended allocation method (based on the solution of job-shop scheduling problems). We simulated carrying tasks and exploratory tasks, which include uncertainty conditions such as unknown work environments (2 tasks and 2 robots, multiple tasks and 2 robots, 2 robots and multiple tasks, and multiple tasks and multiple robots). In addition, we compared our method with full searching and methods that maximize the sum of efficiency in these simulations by several conditions: first, 2 tasks (carrying and exploring) in the four uncertain conditions (later time, new objects appearing, disobedient robots, and shorter carrying time) and second, many types of tasks to many types of robots in the three uncertain conditions (unknown carrying time, new objects appearing, and some reasonable agents). The proposed method is also effective in three terms: the task-execution time with an increasing number of objects, uncertain increase in the number of tasks during task execution, and uncertainty agents who are disobedient to allocation orders compared to full searching and methods that maximize the sum of efficiency. Additionally, we performed two real-world experiments with uncertainty.

1. Introduction

Research on a multirobot task allocation system and coordinating heterogeneous robots have become a hot topic of research in the field robotics. We propose a method of allocating multiple types of tasks to heterogeneous robots, based on the theory of comparative advantage [1], in order to minimize the makespan (the total length of task-execution time). The calculation time in the proposed method is negligibly short compared with the task-execution time, and it may also be effective for dynamic allocation (e.g., in an unknown real environment).

A key driving force of multirobot systems is their force of numbers. Multirobot systems are expected to improve the effectiveness of robotic systems in terms of the temporal

efficiency [2]. Similar to humans, multiple robots work more effectively than a single robot, even those that do not have complex mechanisms or advanced intelligent control systems. Recent studies on decision-making of multirobot systems focus on two categories: allocation (centralized method for coordinating) and cooperation (decentralized method for coordinating). Yan et al. [3] valued the following two categories: the allocation that the central controller has a global view of the world, whereby the globally optimal plans can be produced, but it is not robust in relation to dynamic environments or failures in communications and other uncertainties and the cooperation can better respond to unknown or uncertain environments, but the solutions they reach are often suboptimal. In the real world, multirobot systems will be offered various tasks of the unknown and

uncertain environments. With the advancement of information and communication technology, it will require that multirobot systems for heterogeneous robots have various perception and capability by many vendors. Moreover, effective teamwork between groups of humans and robots is one of the goals in the fields of robotics and AI [4]. Multirobot systems must have a view of the environment including the humans. For example, in a rescue or industrial mission, humans and robots are required to work together. Our ultimate goal is allocating tasks to heterogeneous robots and humans, or the cooperating of heterogeneous robots with humans with minimum makespan and the real world. First of all, this study is focused on the allocation of tasks to heterogeneous robots in unknown environments by using the theory of comparative advantage: an economic theory that maximizes the benefit of specialization.

Conventional task allocation methods are categorized as static or dynamic allocation [5]. Static allocation gives the task assignments at the start until all the tasks are completed. Static allocation calculates the assignments to minimize the required time. This type of problem, in which the makespan is minimized, is called a job-shop scheduling problem [6]. The makespan refers to the total length of the schedule. However, the calculation time of static allocation increases exponentially as the number of tasks increases. In static allocation, the execution time of each task should be calculated and is perturbed by uncertainty. Dynamic allocation frequently conducts reallocation of static allocation [7] and requires more calculation time.

Dispatching methods to reduce the calculation time have been studied. In multirobot task allocation, the methods are instances of optimal assignment problems [8]. In the auction based method proposed by Gerkey and Mataric [9], the goal of the problem is to maximize the efficiency per task of allocation, where efficiency refers to the value per unit work of task-execution. However, maximizing the efficiency does not minimize the makespan in the case of allocating tasks to heterogeneous robots. For example, if a robot executes a task that another robot should do, the latter robot must execute yet another task that it is either not proficient at or is hard to execute. This type of loss is called opportunity cost [10]. We propose an allocation method based on the theory of comparative advantage, which reduces the opportunity cost and minimizes the makespan.

In this study, the allocation based on our method reduces the makespan compared with methods that maximize the sum of efficiency. We perform simulation experiments in environments with uncertainty and find that allocation based on the proposed method reduces the makespan under any uncertainty, compared with methods that maximize the sum of efficiency. In uncertain environments, the makespan with the proposed method was almost the same compared with frequent reallocation of static allocation. We also conduct a heterogeneous robots task-execution experiment and a human-robot collaborative experiment in the real environment. We propose task allocation methods for various tasks execution by heterogeneous multirobots. The proposed method is expected to be effective in real world experiments, because uncertainty can actually appear.

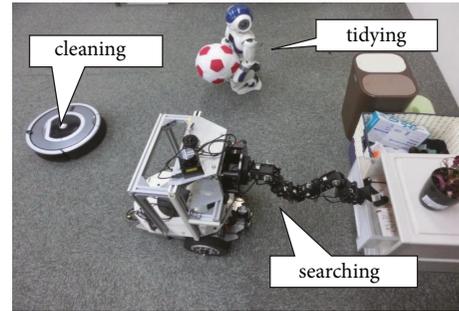


FIGURE 1: Task allocation to heterogeneous robots.

Currently, the area of the human-robot symbiosis is growing rapidly. As stated in the call of this special issue, assembly, warehouses, and home services need human-robot collaboration as soon as possible. Uncertainty of human is not only each individual difference but also effort input in human-robot collaboration. The definition of the theory of comparative advantage had to be extended to adapt to increases in the number of robots and tasks in the real world including humans.

2. Materials and Methods

2.1. Related Work. Because of the difficulties of constructing a single robot that has the required capability in real world, in recent literature, there are works on heterogeneous multirobot systems (e.g., in assembly of large-scale structures [2]). To allocate tasks to heterogeneous robots, each robot accomplishes its advantageous task, which increases the time performance [2, 11, 12]. How to optimally assign a set of robots to a set of tasks is well known as multirobot task allocation (MRTA) problem [13]. Liemhetcharat and Veloso [14] suggested that “the MRTA problem is categorized along three axes: single-task robots (ST) versus multitask robots (MT), single-robot tasks (SR) versus multirobot tasks (MR), and instantaneous assignment (IA) versus time-extended assignment (TA).” In the example shown in Figure 1, each robot accomplishes its advantageous task. Scheutz et al. [15] showed that swarms of high- and low-speed unmanned aerial vehicles exhibit higher performance than homogeneous swarms in locating and tracking chemical clouds. King et al. [16] performed experiments with a team consisting of rovers and blimps cooperatively executing two types of tasks. Zhang et al. [17] developed a fuzzy collaborative intelligence based algorithm to the collaboration of heterogeneous robots. In these studies, the types of robots are determined in advance, and each robot only executes its specific type of task. Ideally, task allocation methods should not depend on the types of robots, tasks, and environments. Many types of robots are expected to accomplish tasks flexibly as a team operating in any type of environment.

Under other circumstances, the problems of heterogeneous multirobot task allocation problems have been understood as instances of operations research, economics, scheduling, network flows, and combinatorial optimization

[5]. The allocating methods of these problems are categorized as either static allocation, in which tasks are allocated at the start until all the tasks are completed, or dynamic allocation, in which the next task is allocated to any robot as required. Gerkey and Mataric [5] have categorized task allocation problems in multirobot systems as instantaneous or time-extended assignment. Instantaneous assignment is similar to dynamic allocation, and time-extended assignment is similar to static allocation. Time-extended methods have been based on the scheduling algorithm [18]. However, it takes exponential time or longer for these methods to calculate the exact solution. In addition, static allocation is only initially efficient and will degrade over time [19]. We suppose that there are several types of uncertainty related to work in real environments. Especially in the human/multirobot collaboration, it is uncertain to predict a human's behavior such as in the following three cases:

- (i) Uncertainty in task-execution time
- (ii) Uncertainty increasing the number of tasks during task execution
- (iii) Uncertainty agent who is disobedient to allocation orders

Dynamic allocation is a method that repeatedly allocates tasks dynamically during execution. There are many studies on rescheduling in job-shop scheduling problems in factories. In these works, jobs are added during execution [20], the processing speed is uncertain [7], efficiency under an uncertain environment based on the Intuitionistic Fuzzy Sets [21], slack-based techniques at low levels of uncertainty [22], and so forth. These methods are divided into two types: (1) all task assignments are given repeatedly until all the tasks are completed and (2) no planning for future allocations is performed. The latter is called a dispatching method (e.g., first in first out (FIFO) and shortest processing time (SPT) [18]). Allocations based on rescheduling all the tasks are thought to reduce the makespan, and allocations based on dispatching are thought to reduce the calculation time. The dispatching method is also used in CPU scheduling [23], where tasks (processes) are added successively. Compared with production scheduling in a factory [24], the makespan of multirobot task allocation is thought to be smaller. Computer performance may also be restricted in some multirobot task allocation problems. Therefore, dispatching methods have often been studied in multirobot task allocation (e.g., methods that maximize the sum of efficiency [9, 25–33]). These methods entail searching for the best combination of robots and tasks that maximize the sum of efficiency. In this case, efficiency is defined per robot per task, e.g., the inverse of the task-execution time for a robot to perform a task. These methods are based on the solution to the optimal assignment problem [8], the goal of which is to allocate tasks to robots to maximize the overall expected performance. However, allocation based on this method does not necessarily minimize the makespan. In allocation to heterogeneous agents, the opportunity cost can be added to the makespan [10]. Opportunity cost is a term used in economics, representing the value of an alternative forgone (unselected combination)

to pursue a certain action. For example, consider that a robot executes a task that another robot should do, to maximize the sum of efficiency, and another robot executes a task for which it cannot minimize the makespan or that it cannot execute. If the former robot executes the task, the difference in task-execution times between the robots is added to the makespan as the opportunity cost. We focus on the makespan at the uncertainty and propose an allocating method based on the theory of comparative advantage, which is a framework to minimize makespan including the opportunity cost.

2.2. Task Allocation Based on the Theory of Comparative Advantage

2.2.1. Theory of Comparative Advantage.

To increase the efficiency of tasks executed by heterogeneous robots through task allocation, we focus on “comparative advantage.” Comparative advantage is an economic theory referring to the ability of any given economic actor to decrease the total economic cost. According to this theory, an appropriate allocation of tasks among robots is more efficient than each robot executing all tasks. We adapt this theory to the ability of robots and reduce the cost of multirobot decision.

Let e be the efficiency (the productivity per unit labor input) when the robot i executes task m . It is possible to compare the comparative advantage of robot i with robot j by

$$\frac{{}^i e_m}{{}^i e_n} > \frac{{}^j e_m}{{}^j e_n}. \quad (1)$$

If robot j has the comparative advantage in task n compared with robot i ,

$$\frac{{}^i e_n}{{}^i e_m} < \frac{{}^j e_n}{{}^j e_m}. \quad (2)$$

We define ${}^i p_m$ as the “comparative advantage between robots” and ${}^i q_m$ as the “degree of comparative advantage between tasks”:

$$\begin{aligned} {}^i p_m &= \frac{{}^i e_m}{{}^j e_m}, \\ {}^i q_m &= \frac{{}^i e_m}{{}^i e_n}. \end{aligned} \quad (3)$$

Each comparative advantage between each robot with each task becomes an index to allocate tasks. If robot i is inferior to robot j (${}^i e_m < {}^j e_m$) for task m and has the comparative advantage (see (1)), we should allocate robot i to task m . The total productivity of task m is given by

$$S_m = \sum_{i=1}^i e_m^i w_m, \quad \left(\sum_{i=1}^i w_m = 1 \right), \quad (4)$$

where ${}^i w_m$ is the working ratio (the labor input rate), which is the cost of robot i completing task m per the total cost of robot i . The Pareto front (Figure 2) gives a solution to fix the value of

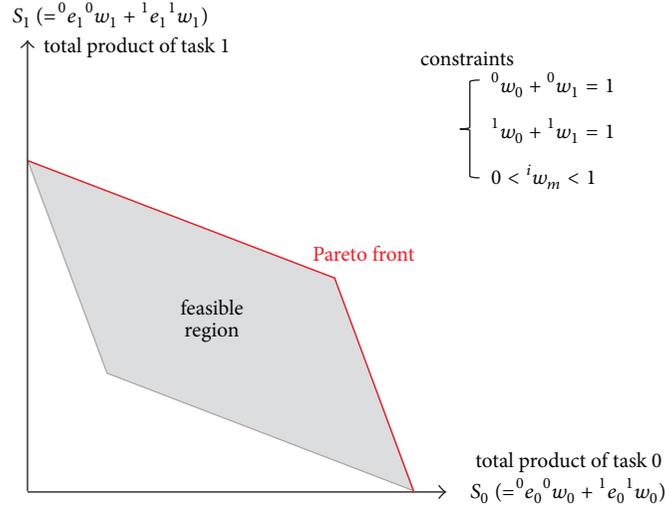


FIGURE 2: Pareto front.

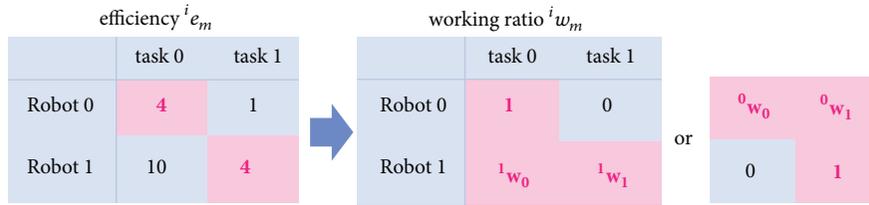


FIGURE 3: Example of an optimal allocation on the Pareto front.

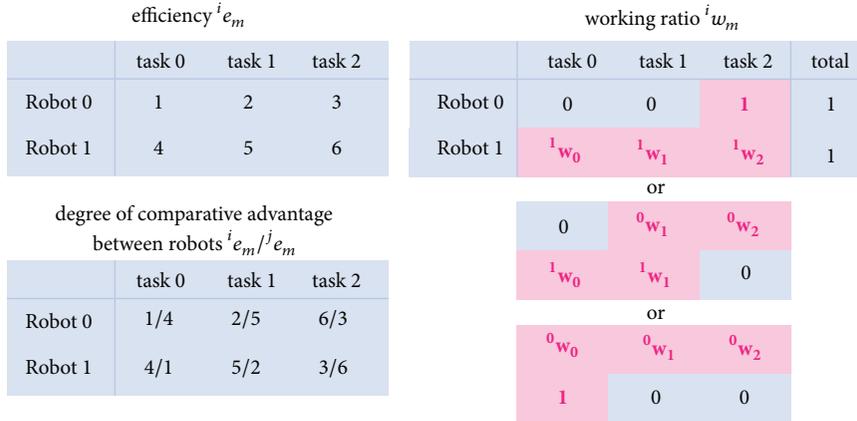


FIGURE 4: Comparative advantage with multiple tasks and two agents.

the working ratio. In this theory, to obtain the solution of the Pareto front ($iw_m = 1$), a robot with a comparative advantage (Figure 3) must specialize in a task. We define the cost of robot completing the task as the makespan.

For example, when two robots are allocated to two tasks by the Pareto front (Figure 1), robot 0 executes task 0 and robot 1 executes task 1 at first. Throughout the execution time, one robot is perfectly specialized in one task. The other robot first executes its comparatively advantageous task and subsequently executes the other task. When two robots are allocated to multiple tasks by the Pareto front, each robot

performs the tasks in order of their comparative advantages. By the Pareto front, one can argue that maximizing the sum of efficiencies is inefficient. For example, Figure 4 shows a set of robots and tasks. According to the Pareto front, robot 0 executes the tasks in the order (2, 1, 0) and robot 1 executes the tasks in the order (0, 1, 2). By maximizing the sum of efficiency, if robot 0 executes any task after finishing task 2, it would not be an optimum allocation by the Pareto front.

2.2.2. Comparative Advantage with Multiple Agents and Multiple Tasks. The theory of comparative advantage can be

efficiency ${}^i e_m$			working ratio ${}^i w_m$				
	task 0	task 1	task 0	task 1			
Robot 0	1	4	${}^0 w_0$	${}^0 w_1$	or		
Robot 1	2	5	1	0		${}^1 w_0$	${}^1 w_1$
Robot 2	3	6	1	0		1	0

	task 0	task 1		task 0	task 1
Robot 0	0	1	or	0	1
Robot 1	1	0	or	0	1
Robot 2	1	0	or	${}^2 w_0$	${}^2 w_1$

FIGURE 5: Comparative advantage with two tasks and multiple agents.

extended to allocation among multiple robots or multiple tasks [34]. Figure 4 shows an example of the allocation of two robots to multiple tasks. In Figure 4, tasks are arranged from the left in ascending order of ${}^i p_m$ of robot 0. The tasks are always arranged in descending order of ${}^i p_m$ of robot 1. Neither robot should perform tasks that have less ${}^i p_m$ than a benchmark task. We can choose any task as the benchmark task. If the benchmark is task 1, then robot 0 should first execute task 2 and robot 1 should first execute task 0 in the example in Figure 4. Figure 5 shows that each robot performs a task that has a larger ${}^i q_m$ than a benchmark agent. We can choose any agent as the benchmark agent.

In cases of multiple robots and multiple tasks, it will be complex to solve the problem. No general method for describing all the Pareto fronts has been established yet. In economics, several methods have been used to apply the theory of comparative advantage to cases of multiple agents and multiple commodities. There are methods of finding the solutions on the Pareto front or methods in which the number of agents or commodities is restricted [35–37]. In this study, we use a method proposed by Tian [37]. In the following, ${}^i r_m$ is the efficiency of robot i to execute task m (${}^i e_m$) divided by the sum of efficiency per task and robot:

$${}^i r_m = \frac{{}^i e_m}{\sum_k {}^k e_m \sum_l {}^l e_l}. \quad (5)$$

Robot i is allocated to task m for which r is maximum. Next, among the combinations of other robots and tasks, the robot and task combination that maximizes r is selected. This procedure is repeated for all the robots. A detailed allocation flowchart is given in Figure 6.

2.2.3. Optimal Assignment Problem. The optimal assignment problem [8] is a well-known problem in operations research. In this problem, M robots, N tasks, and nonnegative efficiencies that predict each robot's performance for each task are given. The goal is to assign tasks to robots to maximize the overall expected performance. In the task allocation problem, the goal is to find the appropriate ${}^m \alpha_n$ such that the sum of efficiency is maximized:

$$\operatorname{argmax}_{\alpha} \left(\sum_{m=1}^M \sum_{n=1}^N {}^m e_n \alpha_n \right), \quad (6)$$

where e represents the task performance per time and ${}^m \alpha_n$ must be either 0 or 1; if robot m executes task n , then ${}^m \alpha_n = 1$.

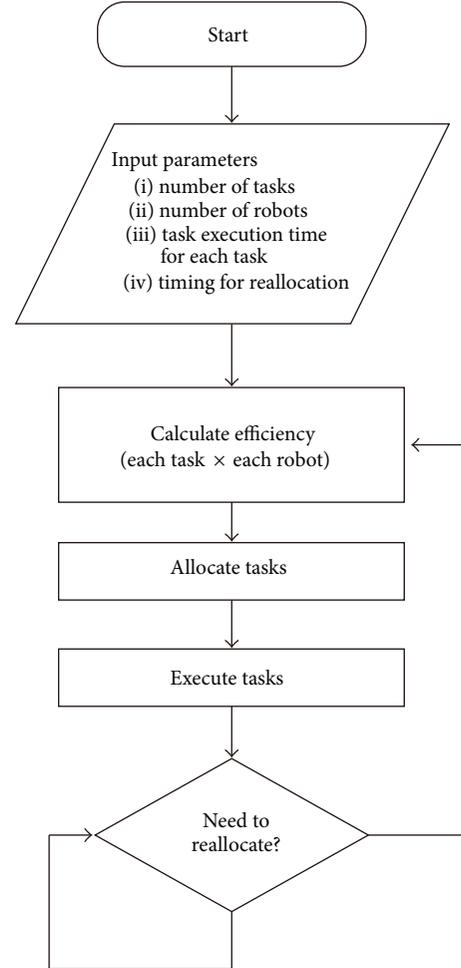


FIGURE 6: Allocation flowchart.

This is a method of instantaneous assignment which requires frequently solving the problem. It costs $O(MN^2)$ time using the Hungarian method [38]. However, maximizing the sum of efficiency does not always minimize the makespan. For example, in a case where each robot executes each different task (Table 1), robot 0 executes task 1 and robot 1 executes task 0 to maximize the sum of efficiency. Then, the total products are $S_0 = 1$ and $S_1 = 10$, where the solution is not on the Pareto front. In other instances, if robot 1 is specialized in task 1 and the working ratios of robots 0 and 1 are ${}^1 w_0 = 0.25$ and ${}^1 w_1 = 0.75$, respectively, the total products of task 0 are $S_0 = 1$ and $S_1 = 11.5$, where the solution is on the Pareto front.

TABLE 1: Example in which maximizing sum of efficiency does not minimize makespan.

	task 0	task 1
Robot 0	4	10
Robot 1	1	4

We can regard this multirobot task allocation problem as an extension of the traveling salesman problem (TSP) [39]. General TSPs with multiple agents are called multiagent traveling salesman problems [40], vehicle routing problems [41], or job-shop scheduling problems [6]. These problems are NP-hard [42]; they cannot be solved in polynomial time [43, 44]. Instead of listing all the orders (as in the full searching method), there are other methods to decrease the amount of calculation, e.g., by using dynamic programming of time computational quantity $O(2^N N^2)$ and the branch-and-cut algorithm [45]. General algorithm, such as NN (Neural Network) [46] or ACO (Ant Colony Optimization) [47, 48], based approximate solutions are also given. In the study to solve vehicle routing problem, it is well known that the method clusters the area by the number of agents and calculates each shortest path [49]. Approximation algorithms and heuristics are, therefore, developed for these problems [48, 50, 51]. However, they require exponential time or longer to calculate the exact solution.

In the uncertain real environment, our proposed method should be able to solve large-scale problems with a large number of robots, tasks, people, animals, and so on. Our method should also be able to compare with different algorithms and the trajectories of robots. There are many things to clarify if our proposed method is feasible or not. But first and foremost, we think the full searching method is plausible to compare with the proposed method.

In this paper, we compare our method with existing methods and show the effectiveness of our method on situations: 2 task and 2 robots (Section 3.1), multiple tasks and 2 robots (Section 3.2), 2 robots and multiple tasks (Section 3.3), and multiple tasks and multiple robots (Section 3.4). As the reason for this, an optimal solution can be calculated for task allocation to the 2 robots situation, to the extent the number of tasks is not large. If our method in these situations is more effective than existing methods for the number of tasks that is more than a certain value, at least, the method has possibilities to solve large task allocation to multiple robots.

3. Task Allocation Method to Specific Problems

3.1. Allocating Two Types of Tasks to Two Types of Robots. To verify the effectiveness of the proposed method, we first simulate allocating two types of tasks to two types of robots (Figure 7). In this study, our purpose is to shorten the makespan of executing tasks in an unknown environment. Tasks in an unknown environment are divided into two main tasks: exploration and other tasks. We choose the carrying

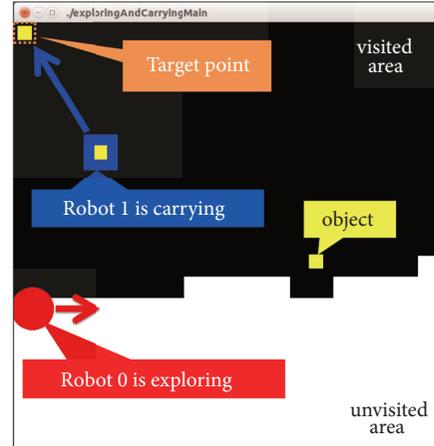


FIGURE 7: Simulation: two types of tasks by two types of robots. The field consists of 20×20 areas. The red circle and blue square represent the robots, the yellow squares represent objects to be carried, the white shading represents unvisited area, and the black shading represents visited area.

TABLE 2: Moving speeds of each robot.

	Not carrying (v_i)	Carrying ($v_{\text{obj}(i)}$)
Agent 0	0.33	0.10
Agent 1	1.00	0.033

task as the basic other task in a field that consists of 20×20 areas. We define these two types of tasks as follows:

- (i) Carrying task: carrying objects to a target point (Figure 7, dashed square). This task is finished once the robots have carried all the objects to the target point.
- (ii) Exploring task: exploration of the unknown areas for mapping and discovering objects. Robots do not know where the objects are initially. Unknown areas, indicated by white, turn into known areas, indicated by black, when robots pass through the areas. This task is finished once the robots have passed through all the areas.

The two types of robots explore unknown areas and carry discovered objects. Table 2 shows the differences between the two types of robots. Here, v_i is the moving velocity when robot i does not carry any object, and $v_{\text{obj}(i)}$ is the moving velocity when robot i carries an object. We define the efficiency (performance per unit work) of the carrying task, e_{car} . In this simulation, the efficiency is the inverse of the time taken for the robot to carry the nearest object to the target point:

$${}^i e_{\text{car}} = \left(\frac{l_0}{v_i} + \frac{l_1}{v_{\text{obj}(i)}} \right)^{-1}, \quad (7)$$

where l_0 is the distance from the robot to the object and l_1 is the distance from the object to the target point. We also define

TABLE 3: Moving velocity.

	Not carrying	Carrying
Robot 0	0.4	0.1
Robot 1	1.0	0.4

the efficiency of the exploring task, e_{exp} . In this situation, it refers to the inverse of the time taken for the robot to go to the nearest unknown area:

$${}^i e_{\text{exp}} = \left(\frac{l_2}{v_i} \right)^{-1}, \quad (8)$$

where l_2 is the distance from the robot to the nearest unknown area. In this simulation, the robots are allocated to tasks in the following situations: when one robot finishes carrying an object or exploring an area, when both robots try to carry the same object, and when the comparative advantages of the robots switch. If the allocated task is changed, while a robot is carrying an object, the robot leaves the object.

3.1.1. Comparison with Conventional Method. In this section, we compare the proposed method with conventional methods, i.e., maximizing the sum of efficiency [9] and exploring first method. In maximizing the sum of efficiency, new task is allocated to a robot which has best efficiency of the task as follows: the robot velocities are listed in Table 3. We defined the efficiency of exploring as the not-carrying velocity. We investigated the following three allocation methods:

- (i) Proposed method: the theory of comparative advantage (Pareto front).
- (ii) Maximizing the sum of efficiency: each robot executes each different task; if ${}^0 e_0 + {}^1 e_1$ is more than ${}^0 e_1 + {}^1 e_0$, then robot 0 executes task 0.
- (iii) Exploring first method: both robots execute the exploring task at first and then execute the carrying task after the exploring task is finished.

We simulate three conditions of the number of objects (4, 8, 16). Under the proposed method, robot 0 executed the exploring task and robot 1 executed the carrying task. By maximizing the sum of efficiency, each robot executed opposite tasks. Figure 8 shows the results. Allocation based on our method minimized the makespan of the three types of allocation methods under all conditions.

3.2. Allocating Multiple Types of Tasks in Unknown Environment to Two Types of Robots. In unknown environments, robots have to execute many tasks which are manifold and uncertain. We propose a task allocation method based on the theory of comparative advantage in the case where there are multiple tasks in unknown environments. This allocation method is expected to minimize the makespan by executing in order of the comparative advantage between robots. There are many studies about situations consisting of multiple tasks (e.g., restaurants [52] and disaster mitigation [53]). We feature

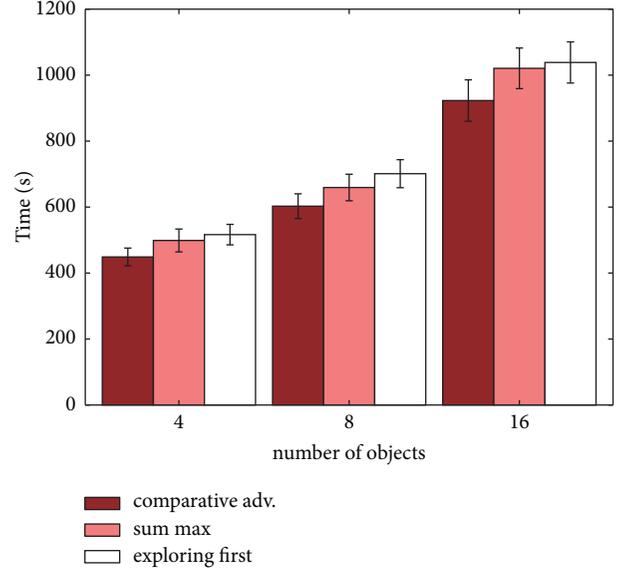


FIGURE 8: Results of proposed and conventional methods.

carrying multiple types of objects tasks as multiple types of tasks. We compare our method with the basic allocation method that gives all the task assignments of the entire period (we refer to this method as the full searching method). We also simulate allocation in unknown environments with possible accidents: later, adding, unreasonable, and speed-up.

3.2.1. Method to Allocate Tasks of Carrying Multiple Types of Objects. As noted above, we simulated the single-type carrying task allocated to heterogeneous robots by the Pareto front. We evaluate the Pareto front for allocation tasks of carrying multiple types of objects to heterogeneous robots. We define the efficiency of the carrying task ${}^i e_m$ as the time taken for the robot to carry the objects.

$${}^i e_m = \left(\frac{l_0}{v_i} + \frac{l_1}{v_{\text{obj}(i,m)}} \right)^{-1}, \quad (9)$$

where $v_{\text{obj}(i,m)}$ is the moving velocity when robot i is carrying object m . The carrying velocity $v_{\text{obj}(i,m)}$ depends on the carried object. The comparative advantage between robots is determined by ${}^i e_m / {}^j e_m$ in (3). When any robot finishes carrying an object, both robots redetermine the carrying objects.

We compare the Pareto front with an allocation method based on the full searching of carrying orders. The full searching method calculates the task-execution times of all the carrying orders in the case of M robots and N objects and calculates an order of minimal time. The number of possible combinations is

$$\frac{(N + M - 1)!}{(M - 1)!}. \quad (10)$$

3.2.2. Setting of Simulation with Possible Accidents. We did a simulation to evaluate the proposed method by comparing

TABLE 4: Moving velocity.

	Not carry	Carry (0)	Carry (1)	...
Robot 0	1.0	0.10	0.11	...
Robot 1	1.0	0.20	0.21	...

The numbers in brackets represent the object IDs.

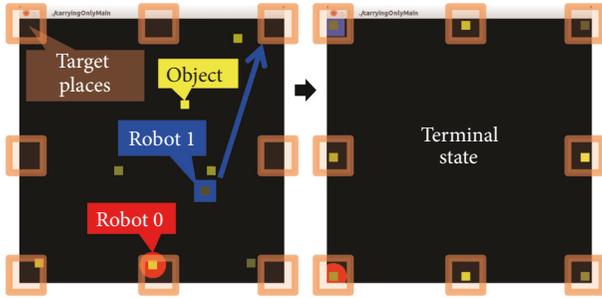


FIGURE 9: Simulation: heterogeneous robots and multiple carrying tasks.

it with the full searching method. Table 4 shows the moving velocities for types of robots and types of carrying objects. The field in the simulator consists of 20×20 areas (Figure 9). At first, the robots are in the center of the field, and the objects are in random positions. The robots know the positions of all the objects in the known area. The target place that each object should be carried to is different. Each robot carries an object that maximizes the comparative advantage between robots.

3.2.3. *Conditions.* We set simulation conditions defining unexpected events that may occur, including the following conditions:

- (i) Later condition: the time required for a robot to carry is later than predicted.
- (ii) Adding condition: new objects appear.
- (iii) Unreasonable condition: some robots are disobedient to allocation orders.
- (iv) Speed-up condition: the time to reach object positions is shorter than the carrying time.

We hypothesize that the proposed method based on the theory of comparative advantage performs better under these conditions than the full searching methods. In the proposed method, robots execute tasks that are given high priority according to the execution time difference among robots. Thus, the remaining tasks have shorter execution time difference among robots. If unexpected events occur, our method can avoid the situation in which an inappropriate robot executes a task and increases the total task-execution time.

Later Condition. This condition is the situation in which it takes more time to carry objects than predicted. For example, such situations occur when a robot carries an object that needs an unknown time to carry, accidents occur, and so

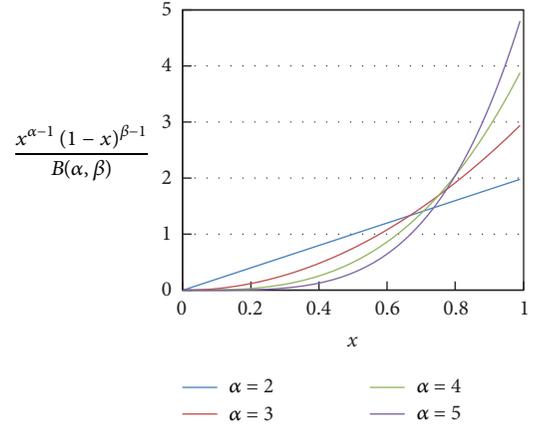


FIGURE 10: Beta distribution ($\beta = 1$).

forth. We simulated the delay by multiplying the estimated carrying velocity $v_{\text{obj}(i,m)}$ by noise x :

$$v_{\text{objreal}(i,m)} = xv_{\text{obj}(i,m)} \quad (0 < x < 1), \quad (11)$$

where $v_{\text{obj}(i,m)}$ is determined by the allocation and $v_{\text{objreal}(i,m)}$ is the carrying velocity when robots actually carry objects. We define noise x as a random number drawn on a beta distribution.

$$\frac{x^{\alpha-1} (1-x)^{\beta-1}}{B(\alpha, \beta)} \quad (0 \leq x \leq 1), \quad (12)$$

where x represents a random variable and $B(\alpha, \beta)$ is a beta function. At $x = 1$, the time is the same as predicted. At $x = 1/2$, it takes twice as much time as predicted. x is defined for each combination of robots and objects. The parameter α controls the noise variance. If the value α is smaller, the noise variance increases. Figure 10 shows a plot of the beta distribution with $\beta = 1$.

We simulated two types of full searching methods: (1) full searching (static), in which the robots carry objects in the order that is decided at first and (2) full searching (dynamic), in which the robots redetermine the order every time a robot finishes carrying an object. We ran simulations to compare the proposed method and the two types of full searching methods for certain patterns of α .

Adding Condition. This condition is the situation in which the number of objects increases. At first, there are seven objects, and some objects are added later. Table 5 lists the carrying velocity and time of added objects. The robots redetermine executing tasks in all the methods every time a robot finishes carrying an object, or an object is added.

Unreasonable Condition. This condition is the situation in which some robots are unreasonable agents (UAs). A human or agent is often disobedient to allocation orders because they are in an inaccessible place, they think of a new way, or for other reasons. In these situations, it is necessary to reallocate tasks for the shortest execution time that the other agents are not executing. The full searching method redetermines the

TABLE 5: Carrying velocity of added objects.

Appearing time [s]	1st	2nd	3rd	4th
Robot 0	0.11	0.19	0.13	0.17
Robot 1	0.21	0.29	0.23	0.27

Initial objects in Table 4.

TABLE 6: Moving velocity (object ID in brackets).

	Not carry	Carry (0)	Carry (1)	...
	v_i	$v_{\text{obj}(i,0)}$	$v_{\text{obj}(i,1)}$...
Robot 0	0.6	0.010	0.011	...
Robot 1	1.0	0.020	0.021	...

TABLE 7: Moving velocity of added objects.

Appearing time [s]	1st	2nd	3rd	4th
Robot 0	0.011	0.019	0.013	0.017
Robot 1	0.021	0.029	0.023	0.027

order for the robots every time an agent finishes a task. In this simulation, Robot 1 is a UA and the robot velocities are given in Table 4.

Speed-Up Condition. In this condition, the carrying velocities (Table 6) are increased to 1/10 of the velocities in previous experiments (Table 4). The time to move is also shorter than the carrying time. In addition, to evaluate this condition, we combined it with adding a condition. The carrying velocity and time when objects are added are shown in Table 7.

Comparing with Allocation Method of Full Searching of Carrying Orders. The calculation complexity of the proposed method is $O(MN)$. Figure 11 shows the average task-execution and calculation time for 100 experiments, where there are initially 7, 8, 9, or 10 objects in random places. The task-execution time for the proposed method is slightly longer than that of the full searching method. However, the calculation time in full searching increases exponentially with an increasing number of objects. From this result, if the executing time of the proposed method is slightly longer than the full searching method, it means that the proposed method has a shorter makespan than the full searching method. Our proposed method is effective for the task allocation to multiple robots in uncertain environment. The calculation time in the proposed method is negligibly short compared with the task-execution time: only 0.51 s for an experiment with 10 objects. Our hardware platform was a ThinkPad T530 with a Core i7-3520M processor, 16 GB memory, and HDD running Ubuntu 14.04.

Maximize the Sum of Efficiency. We also compared our method with methods that maximize the sum of efficiency. Table 4 lists the robot velocities. Figure 12 shows the experimental results of 100 time simulations. There were significant

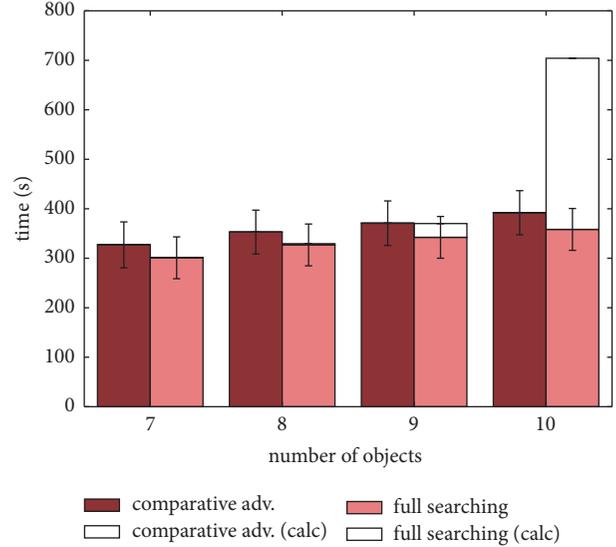


FIGURE 11: Makespan and calculation time in 100 simulations. There are 7, 8, 9, and 10 objects.

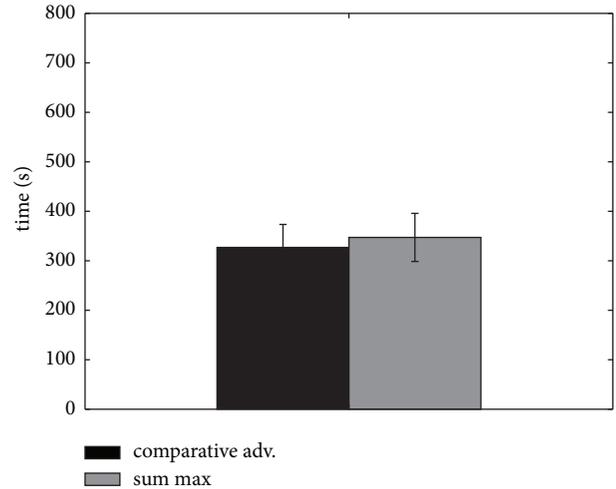


FIGURE 12: Makespan in 100 simulations.

differences: $t(99) = 9.08$; $p < .05$. This is similar to the results in Section 3.1.

3.2.4. Results of Simulations with Possible Accidents

Results: Later Condition. Figure 13 shows the results. Compared with the full searching (static) method, the allocation based on our method reduced the makespan when the noise variance is large (e.g., $\alpha = 3$). Compared with full searching (dynamic) method, the Bonferroni method shows no significant difference between the methods for noises $\alpha = 5, 4, 3$, except for a significant difference between the method that maximizes the sum of efficiency with the full searching method at $\alpha = 5$ ($p < .01$). In the full searching (dynamic) method, the calculation time should be treated as idle time whenever the robots begin working.

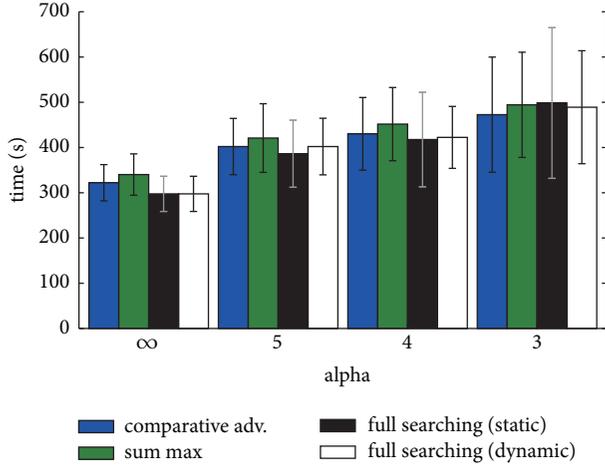


FIGURE 13: Makespan in the later condition.

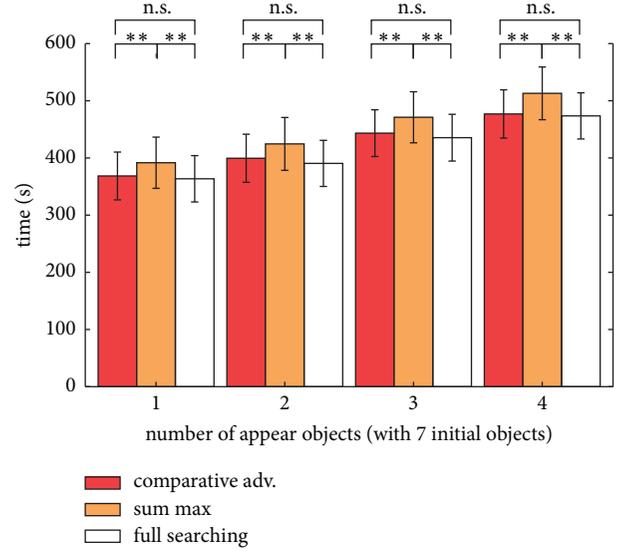
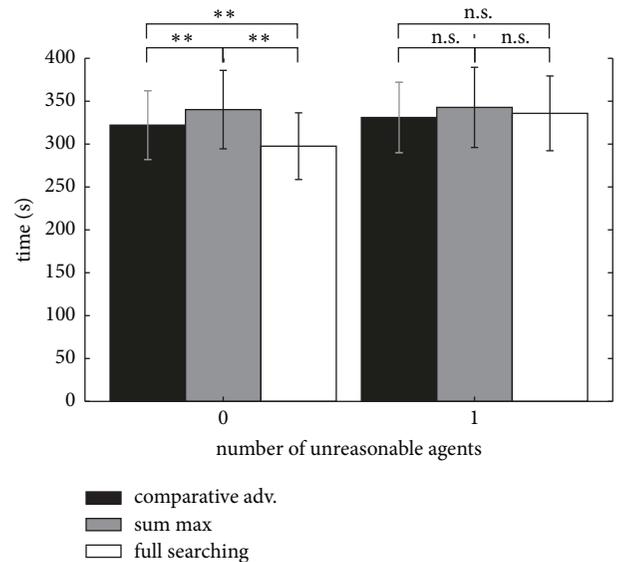
The substantial task-execution time includes the working and calculating time. Our method shortens the substantial task-execution time compared to that in the full searching (dynamic) method, depending on the number of carrying objects. Compared with methods that maximize the sum of efficiency, the allocation based on the proposed method reduces the makespan.

Results: Adding Condition. Figure 14 shows the experimental results. There are no obvious differences depending on the number of added objects. We confirmed that allocation based on the proposed method reduces the makespan compared with the methods that maximize the sum of efficiency. The Bonferroni method shows a significant difference between the proposed method and the methods that maximize the sum of efficiency under all conditions. However, the Bonferroni method shows no significant difference between the proposed method and the full searching method under all conditions.

Results: Unreasonable Condition. Figure 15 shows the results. There are significant differences. The proposed method minimizes the makespan of the three methods, in comparison with the full searching method. The UA does not execute an allocated task. In these simulations, the UAs execute the shortest-makespan task of the tasks that other agents are not executing. In the full searching method, the task order for the robots is redetermined every time an agent completes a task. Robot 1 in Table 4 is a UA. Figure 15 shows the results. There are no significant differences.

Each cell in Tables 8 and 9 shows the method that reduced the makespan. The bars (-) in the tables indicate that the t -test shows no significant difference between the methods. As a result, in cases with a UA, the proposed method is more effective than the full searching method. In certain cases without a UA, the proposed method is as effective as the full searching method.

Results: Speed-Up Condition. We think the proposed method is more effective than the full searching method in cases

FIGURE 14: Makespan in the adding condition. $**p < .01$, $*p < .05$, and $+p < .1$. A significant p value indicates that a significant difference in prescribing exists between the conditions. The lower p value indicates a strong significant difference. n.s. means no significant difference between the conditions.FIGURE 15: Makespan in the unreasonable condition. $**p < .01$, $*p < .05$, and $+p < .1$. A significant p value indicates that a significant difference in prescribing exists between the conditions. The lower p value indicates a strong significant difference. n.s. means no significant difference between the conditions.

where the differences in efficiency are smaller. Figure 16 shows the results for which the number of objects increases during the task execution. The allocation based on the proposed method reduces the makespan compared with the full searching method. The proposed method gives priority to the tasks whose carrying time differences are large. Therefore,

TABLE 8: Comparison between proposed method and full searching (with no unreasonable agents).

	$\alpha = \infty$	$\alpha = 5$	$\alpha = 4$	$\alpha = 3$
$n_{\text{add}} = 0$	full searching**	-	full searching ⁺	-
$n_{\text{add}} = 1$	full searching*	-	-	-
$n_{\text{add}} = 2$	full searching**	-	-	-
$n_{\text{add}} = 3$	full searching ⁺	-	-	-
$n_{\text{add}} = 4$	-	-	-	-

⁺ $p < .1$, * $p < .05$, and ** $p < .01$; DOF = 199.

TABLE 9: Comparison between proposed method and full searching (with an unreasonable agent).

	$\alpha = \infty$	$\alpha = 5$	$\alpha = 4$	$\alpha = 3$
$n_{\text{add}} = 0$	proposed**	-	-	-
$n_{\text{add}} = 1$	proposed**	proposed*	proposed ⁺	-
$n_{\text{add}} = 2$	proposed**	proposed**	-	proposed*
$n_{\text{add}} = 3$	proposed**	proposed*	proposed**	-
$n_{\text{add}} = 4$	proposed**	proposed*	-	-

(⁺ $p < .1$, * $p < .05$, ** $p < .01$; DOF = 199).

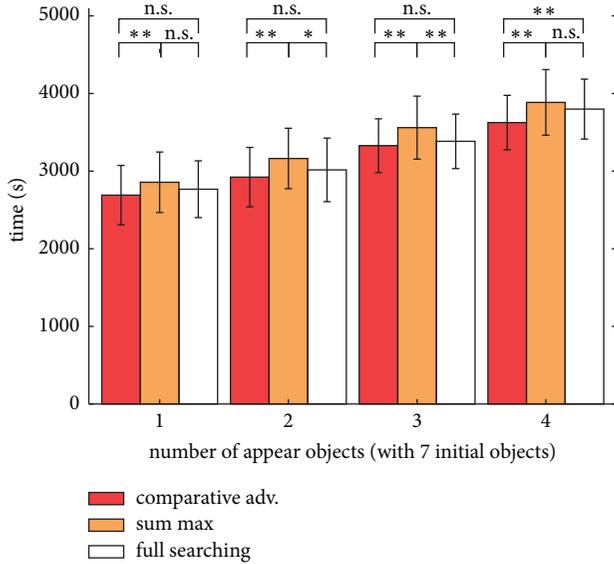


FIGURE 16: Makespan in the speed-up condition. (1/10 carrying speed condition). ** $p < .01$, * $p < .05$, and ⁺ $p < .1$. A significant p value indicates that a significant difference in prescribing exists between the conditions. The lower p value indicates a strong significant difference. n.s. means no significant difference between the conditions.

the opportunity cost, which is the difference in carrying times between robots, is unlikely to increase when compared with the full searching method. The efficiency of each task has less differences depending on the robot position, because the carrying time does not depend on it. The proposed method considers only the instantaneous robot position.

TABLE 10: Moving velocity (object ID in brackets).

	Not carrying	Carrying
Robot 0	0.2	0.1
Robot 1	0.3	0.08
Robot 2	0.4	0.06
Robot 3	0.5	0.04

3.3. Allocating Two Tasks to Many Types of Robots

3.3.1. Limitation of Allocating Method. We think the Pareto front can shorten the makespan of many robots executing two tasks (exploring and carrying). In this theory, we must nominate a benchmark robot to determine how many robots should be allocated to each task (exploring or carrying) and the benchmark robot is the only one that executes both tasks. For the adjustment to many robots, the following steps are added to the allocation method based on the theory of comparative advantage. To decide a benchmark robot, we performed a simulation.

- (1) Nominate a benchmark robot in some way.
- (2) Allocate one task to robots that have a comparatively greater degree of advantage than the benchmark robot, according to (3).
- (3) Allocate the other task to the other robots.

The whole-period Pareto front has a limitation for allocating tasks to many types of robots. When one task is finished first, every agent will be executing the other task. This means that there are robots that execute both tasks and the whole-period Pareto front is unadaptable to this situation. The whole-period Pareto front can be adapted to the situation in which each robot executes a single task. Hence, to allocate two tasks to many types of robots, it is unadaptable to shorten the overall makespan. However, we think that the particular makespan (e.g., until one task is finished) will be shortened by the whole-period Pareto front. We propose a new method to shorten the overall makespan through the simulation results.

3.3.2. Simulation: Allocate a Carrying and an Exploring Task to Many Types of Robots. We performed simulation experiments to allocate a carrying and an exploring task to four types of robots by the theory of comparative advantage (Figure 17). We set the moving velocities of four robots as the comparative advantage (Table 10), and (8) gives the efficiency of the robots for each task. We performed the following steps to simulate allocation.

- (1) Determine a benchmark robot in number order of robots (decide how many robots to allocate the exploring task).
- (2) Allocate exploring task to robots that have a comparatively greater degree of advantage than the benchmark robot.

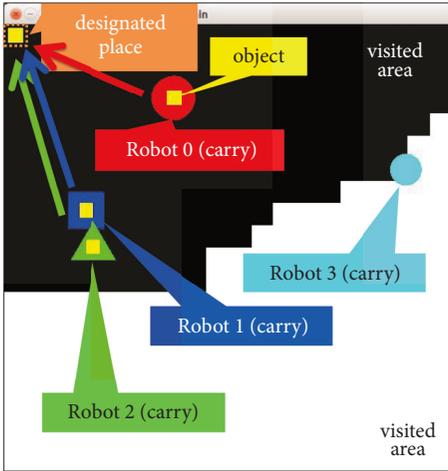


FIGURE 17: Simulation example: allocate the exploring and carrying tasks to four robots.

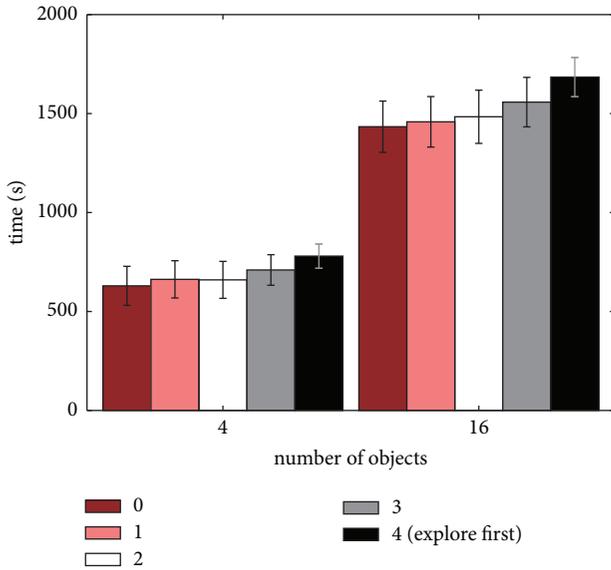


FIGURE 18: Results based on our method. The legend shows the benchmark agent id: minimal numbers of exploring agents.

- (3) Allocate the carrying task to other robots in the order of the degree of comparative advantage.
- (4) Allocate the exploring task to the remaining robots.

We simulated five robot conditions (benchmark robots are 1–4; exploring robots are 0–4) versus the object condition (number of objects are 4 or 16) on the instantaneous Pareto front. Figure 18 shows the results. In both object conditions, when the exploring robot was zero, the instantaneous Pareto front can shorten makespan the most. From these results, we can propose a method “explore until find.” Under this method, if an object is found, it should be moved instantly. Accordingly, for few tasks, robots should give preference to other tasks over the exploring task. Thus, we propose a method in the case that there are many tasks in an unknown environment.

TABLE II: Moving velocity.

	Not carry	Carry (0)	Carry (1)	...
	v_i	$v_{\text{obj}(i,0)}$	$v_{\text{obj}(i,1)}$...
Robot 0	1.0	0.10	0.11	...
Robot 1	1.0	0.20	0.21	...
Robot 2	1.0	0.30	0.31	...
Robot 3	1.0	0.40	0.41	...

3.4. Allocating Many Types of Tasks to Many Types of Robots

3.4.1. Method. We propose a method to allocate many types of tasks to many types of robots, based on the theory of comparative advantage proposed by Tian [37]. The numbers of robots and objects are represented by M and N , respectively. Here, ${}^i r_m$ is the efficiency for robot i to execute task m (${}^i e_m$) divided by the sum of the efficiencies per task and per robot by (5). Robot i executes task m , for which r is maximum. Next, among the combinations of other robots and tasks, the combination for which r is maximum is selected. This procedure is repeated M times. In case of $M = 2$, the task selected by this method is always the same as the method of Section 3.1. Figure 19 shows an example of allocation based on this method. The red-colored values indicate the selected tasks.

3.4.2. Verification. We verify the proposed method to allocate carrying tasks to many robots by a simulation, as shown in Figure 20. The simulation settings used here are the same as those in Section 3.1. We employed the following conditions for verification.

- (1) Robot condition: the robots have different comparative advantages. We set the moving velocities of four robots as the comparative advantage (Table II).
- (2) Environment condition: the system knows or does not know about the environment.
- (3) Unreasonable condition: this is the same as in Section 3.2: task-execution time, added objects, and unreasonable agent.

We compare the proposed method with the full searching method and the method that maximizes the sum of efficiency. The full searching method in this section refers to the “dynamic full searching” method: robots are reallocated to each task every time a robot finishes carrying an object.

Figure 21 shows the makespan in these three methods in a known environment. Multiple comparisons with the Bonferroni methods show significant difference between these three methods. The method that maximizes the sum of efficiency is the highest in the known environment.

We show the following results of the conditions in an unknown environment.

Task-Execution Time. We performed simulation experiments in situations where it takes an unknown time to carry objects. Figures 22 and 23 show the experimental results. Multiple comparisons with the Bonferroni method revealed that there

	efficiency $i e_m$					$\frac{i e_m}{(\sum_i i e_m)(\sum_m i e_m) \times 10^{-3}}$				
	task 0	task 1	task 2	task 3	task 4					
Robot 0	1	2	3	4	5	1.96	3.51	4.76	5.80	6.67
Robot 1	6	7	8	9	10	4.41	4.61	4.76	4.89	5.00
Robot 2	11	12	13	14	15	4.98	4.86	4.76	4.68	4.62
Robot 3	16	17	18	19	20	5.23	4.97	4.76	4.59	4.44

red: selected tasks

FIGURE 19: Example of allocation based on the proposed method.

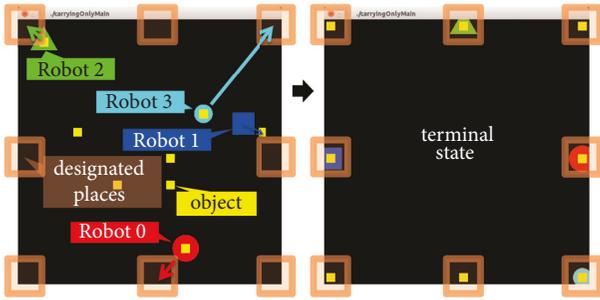


FIGURE 20: Simulation of carrying objects with four robots.

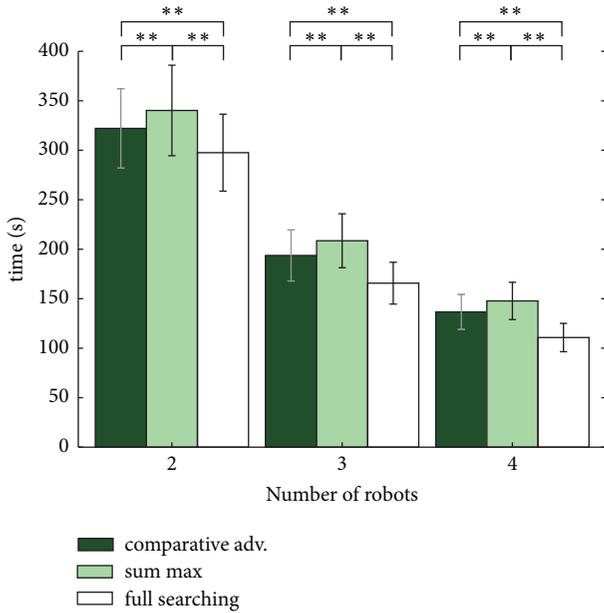


FIGURE 21: Makespan of some number of robots. $**p < .01$, $*p < .05$, and $+p < .1$. A significant p value indicates that a significant difference in prescribing exists between the conditions. The lower p value indicates a strong significant difference. n.s. means no significant difference between the conditions.

are significant differences ($p < .001$) except for $\alpha = 4$, comparative adv versus sum max (n.s.). Compared with methods that maximize the sum of efficiency, the proposed method reduces the makespan under any condition. Compared with the full searching method, the makespan ratio decreased with

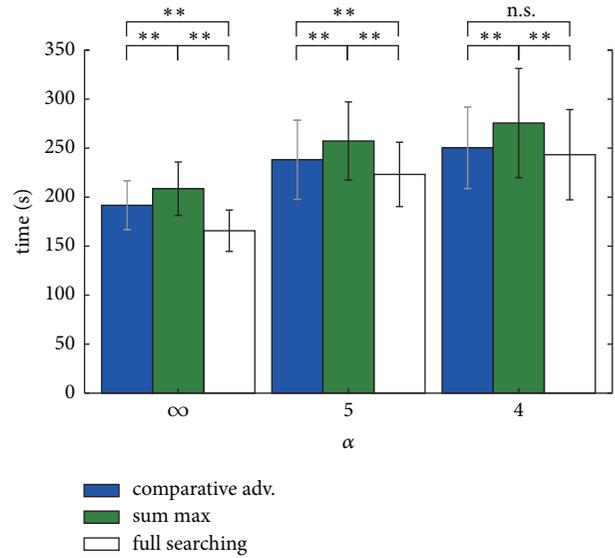


FIGURE 22: Makespan in situations where carrying objects can be late (three robots). $**p < .01$, $*p < .05$, and $+p < .1$. A significant p value indicates that a significant difference in prescribing exists between the conditions. The lower p value indicates a strong significant difference. n.s. means no significant difference between the conditions.

increasing uncertainty. The Bonferroni method shows no significant differences ($p = .05$ level for $\alpha = 4$).

Adding Objects. Figures 24 and 25 show the experimental results. Compared with the methods that maximize the sum of efficiency, the proposed method reduces the makespan under any condition. Compared with the full searching method, the makespan ratio decreased with increasing uncertainty. The Bonferroni method shows no significant differences ($p = .05$ level for $n_{\text{add}} = 2, 3$, or 4).

Unreasonable Robots. In this experiment, robots with large index numbers are unreasonable. For example, for two UAs and one reasonable agent (RA), robots 1 and 2 are unreasonable and robot 0 is reasonable. Figures 26 and 27 show the experimental results. Compared with the methods that maximize the sum of efficiency, the proposed method reduces the makespan under any condition. The Bonferroni method shows no significant differences ($p = .05$).

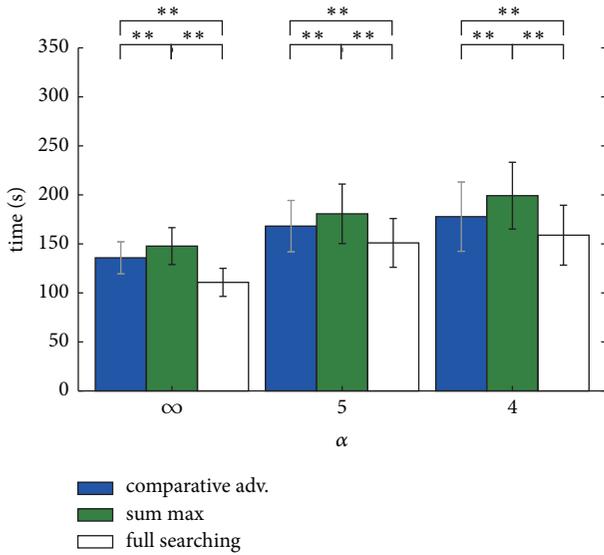


FIGURE 23: Makespan in situations where carrying objects can be late (four robots). $**p < .01$, $*p < .05$, and $^+p < .1$. A significant p value indicates that a significant difference in prescribing exists between the conditions. The lower p value indicates a strong significant difference. n.s. means no significant difference between the conditions.

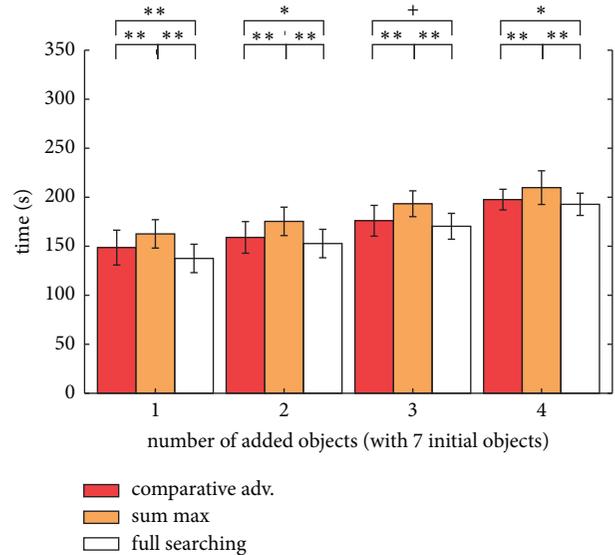


FIGURE 25: Makespan in situations where carrying objects are added later (four robots). $**p < .01$, $*p < .05$, and $^+p < .1$. A significant p value indicates that a significant difference in prescribing exists between the conditions. The lower p value indicates a strong significant difference. n.s. means no significant difference between the conditions.

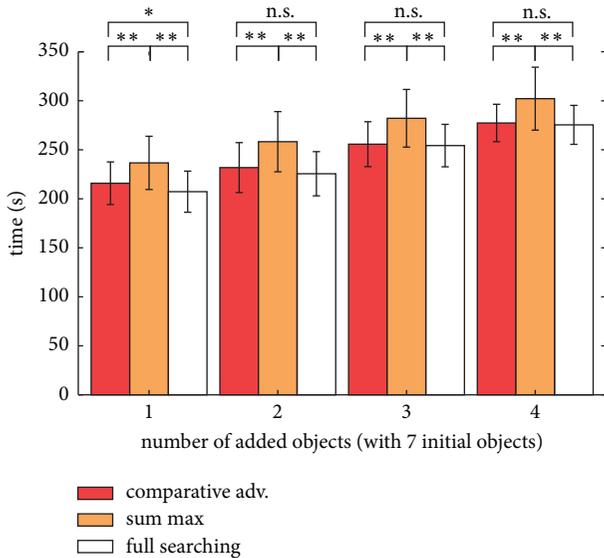


FIGURE 24: Makespan in situations where carrying objects are added later (three robots). $**p < .01$, $*p < .05$, and $^+p < .1$. A significant p value indicates that a significant difference in prescribing exists between the conditions. The lower p value indicates a strong significant difference. n.s. means no significant difference between the conditions.

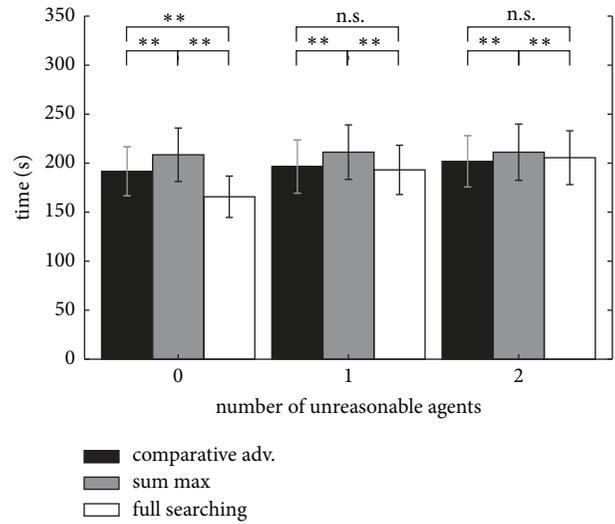


FIGURE 26: Makespan in situations where there is an unreasonable agent (three robots). $**p < .01$, $*p < .05$, and $^+p < .1$. A significant p value indicates that a significant difference in prescribing exists between the conditions. The lower p value indicates a strong significant difference. n.s. means no significant difference between the conditions.

The task-execution time for the proposed method is slightly longer than that of the full searching method from Figures 22–27. However, the calculation time in full searching increases exponentially with an increasing number of objects. We expect that our method will become shorter than full searching with an increase in the number of agents.

These results are similar to those in Figure 8, where the allocation based on our method reduces the makespan under any condition compared with methods that maximize the sum of efficiency. Compared with the full searching method, the makespan ratio decreased with increasing uncertainty.

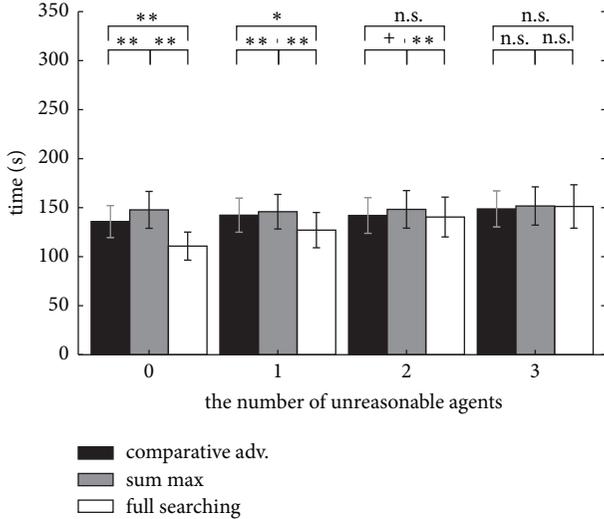


FIGURE 27: Makespan in situations where there is an unreasonable agent (four robots). $**p < .01$, $*p < .05$, and $+p < .1$. A significant p value indicates that a significant difference in prescribing exists between the conditions. The lower p value indicates a strong significant difference. n.s. means no significant difference between the conditions.

TABLE 12: Moving ability of each robot.

	Proceeding	Rotating
Roomba	100 mm/s	100°/s
NAO	60 mm/s	60°/s

3.4.3. *Experiments in the Real Environment.* Additionally, beyond the simulations, we conducted an allocation experiment in a real environment. Figure 28 shows scenes of this experiment. We used two types of robots: a wheeled robot Roomba (iRobot) and a small humanoid NAO (Aldebaran Robotics). These robots can move forward or backward and rotate in their current position. Table 12 shows the speeds of the robots. The objects that should be carried and their features are as follows:

- (i) Box: both robots can easily carry this object by pushing Roomba.
- (ii) Ball on the wall: it is difficult for Roomba to push this object. Roomba cannot move the object to the target area directly, unless Roomba pushes it off the wall. Roomba needs to go around it and move it from behind. NAO can scrape objects from the wall using its hand, which costs approximately 5 s.
- (iii) Table: it has four legs. Roomba is not wide enough to span the legs of the long side. Therefore, Roomba must alternately push the left and right legs in order to move the object. NAO has a comparative advantage to carry the table.

We calculated the carrying time using the moving path and moving speeds (Table 12). The robot controller gave a list of points as the moving path to the robots. Figure 28

shows the flow of the experiments. We regard the failure to carry as a factor of the delay of the carrying time. The proposed method is effective in allocating carrying tasks in an unknown environment as well as in the simulation.

Allocating Carrying Tasks to Wheeled Robot and Humanoid. Figure 28 shows scenes of experiments. (3) shows that Roomba failed to carry the box. If Roomba had not failed, Roomba would have carried the ball. We performed real-world experiments of allocating carrying tasks to real robots using the proposed method. We tried to confirm how our method works in the real world. As a result, the advantage of the proposed method will gradually reduce the differences between the execution times of remaining tasks.

Allocating Tasks to Robot and Human. We also conducted an experiment in which a robot and a human carry objects to confirm the effectiveness of the proposed method in allocation with an unreasonable agent. We used Roomba as the robot. Figure 29 shows the scene of this experiment, which included four types of objects (two boxes, a ball, and a table). Roomba is good at carrying the two boxes and is poor at carrying the table and ball. The human carries an object that the robot is not carrying. The results indicated that the human carried the table and ball, and the robot carried the two boxes. The robot saved carrying time for comparatively disadvantageous objects, thereby reducing the makespan. The robot executed comparatively advantageous tasks and prevented the UA from executing such tasks. The makespan would be reduced by the proposed method.

4. Discussion

4.1. Limitation

4.1.1. *Computing Power.* In this research, to allocate tasks to robots, we used the economic theory of comparative advantage as a method to reduce the total makespan. We compared the proposed method with conventional methods used in robotic exploration missions. These results have been enabled by the recent developments in computing power and the ability of robots. In the future, high computing power will reduce the total makespan, particularly the calculating time, and standardized robots that are able to execute multitasks will be developed. However, our method contributes to the efficiency of robots' labor output. Monofunctional robots can practically execute tasks other than their main task (e.g., cleaning task for the Roomba). Real robots can execute tasks that they should not or that their developers did not consider, but only at a low level (e.g., carrying task for the Roomba). In real environments, we cannot perceive everything that occurs. In such cases, our method can allocate the tasks to robots that include these low-level performances on a real-time basis according to the comparative advantage.

4.1.2. *Simulation Setting.* We conducted simulation in a field that is free (without obstacles) and small (20×20 grid), by robots without sensors and motions. Obstacles increase uncertainty and sensing capability relate closely to a robot's

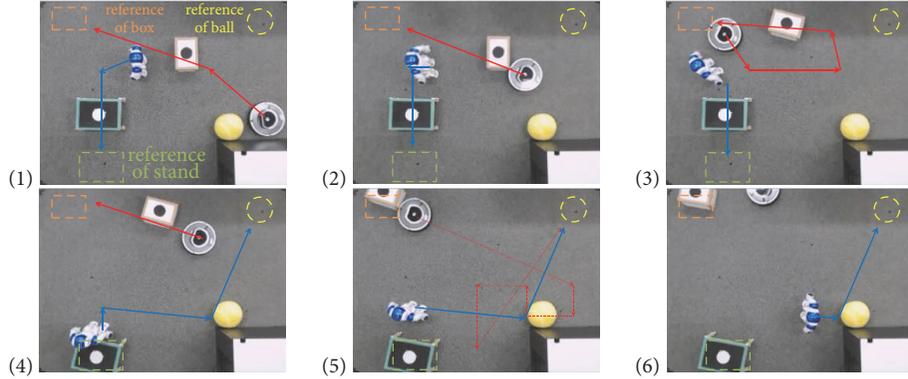


FIGURE 28: Experiment of real robots based on the proposed method. (1) Initial state. The robot decided the carrying object based on the list of carrying time. (2) Trying to move each object. (3) Roomba fails to carry the box and try to do again. (4) NAO finishes carrying the stand and begins to carry the ball. (5) Roomba finishes carrying the box. The method then determines which robot carries the ball, and NAO continues carrying the ball. (6) Roomba finished tidying, and NAO continues carrying the ball.

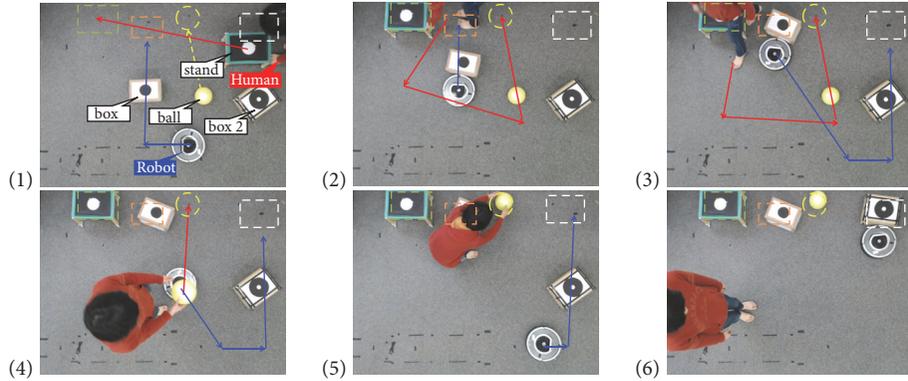


FIGURE 29: Experiment of the robot with a human.

performance. This study is a first step to investigate how to reduce the total makespan by using comparative advantage. We must unite the proposed method with existing algorithms of obstacle avoidance, exploration, and safe wandering. If it is possible to evaluate the performance of agents by real time, our method can be dynamically adaptive.

4.2. Future Work. There have been sparse improvements to the methods of allocating many tasks to many robots. We have a forward-looking approach in which the allocation of next tasks is preliminarily based on the theory of trade with the comparative advantage [54]. In the traveling salesman problem or the job-shop scheduling problem, there are many studies to reduce the calculation time. The proposed method should be applied to these studies and our method should be comparatively verified with them. In this study, one robot executes only one task (single-task robots). In the uncertain real environment, it is important to solve complex large-scale problems with a large number of robots, tasks, people, animals, and so on. Our method should also be able to be compared with different algorithms and the trajectories of robots. It is necessary to expand this to include the condition under which one robot executes certain tasks at the same

time (multitask robots). In addition, the condition under which some robots execute one task at the same time should be considered. We supposed a centralized robot system in this study; thus, we should expand the proposed method to include the distributed autonomous systems.

This study investigated the allocation methods using simulations. We must comparatively verify these results in a real environment with other conventional methods such as market-based techniques inclusive of TraderBots [55, 56] and Hoplites [57]. The uncertainty instances with humans have numerous possible causes. Dealing with true “uncertainty” requires consideration of human-robot collaborations. Performance of a human is not certain quantitatively. Our method observes and calculates the performance of human and allocates tasks to robots with competence dynamically.

5. Conclusion

We investigated a method of using the theory of comparative advantage to allocate tasks to robots with uncertainty including humans. The proposed method is a dynamic sharing algorithm to allocate uncertainty tasks in an unknown

environment, assuming timely reallocation. First, we confirmed that the proposed method reduces the total makespan (the total task-execution time) compared with conventional methods used in robotic exploration missions. We expect that our method is also effective in terms of calculation time when compared with the time-extended allocation method. We simulated carrying tasks and exploring tasks, which include uncertainty conditions of the work in an unknown environment. The proposed method is also more effective in dealing with uncertainty in task-execution time, uncertainty in the increasing number of tasks during task-execution, and uncertainty agents who are disobedient to allocation orders, compared to existing methods (the sum of efficiency and full searching methods). Finally, through experiments in a real environment, we confirmed that the proposed method can reduce the makespan.

This paper makes several contributions to human-robot interaction. First, the effectiveness of a new economic theory was shown for heterogeneous robots. In robot-robot collaboration, it is important to execute tasks even if the robot has inferior ability to accomplish a task. Second, the theory was effective in the uncertain environments including a human. Human-robot collaboration is receiving a lot of attention. The reallocation corresponding to uncertainty of people is a critical issue.

Disclosure

The full text is not published; it was accepted and presented in ROBOMECH2015 conference [58].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The authors would like to thank Editage (<https://www.editage.jp>) for English language editing.

References

- [1] D. Ricardo, *Principles of Political Economy and Taxation*, John Murray, 1817.
- [2] A. Farinelli, L. Iocchi, and D. Nardi, "Multirobot systems: A classification focused on coordination," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 5, pp. 2015–2028, 2004.
- [3] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, article no. A399, 2013.
- [4] B. Scassellati and B. Hayes, "Human-robot collaboration," *AI Matters*, vol. 1, no. 2, pp. 22–23, 2014.
- [5] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.
- [6] R. L. Graham, "Bounds for certain multiprocessing anomalies," *Bell System Technical Journal*, vol. 45, no. 9, pp. 1563–1581, 1966.
- [7] S. R. Lawrence and E. C. Sewell, "Heuristic, optimal, static, and dynamic schedules when processing times are uncertain," *Journal of Operations Management*, vol. 15, no. 1, pp. 71–82, 1997.
- [8] D. Gale, *The Theory of Linear Economic Models*, McGraw-Hill Book Co, New York, NY, USA, 1960.
- [9] B. P. Gerkey and M. J. Mataric, "Sold!: auction methods for multirobot coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, 2002.
- [10] F. von Wieser and A. Ford Hinrichs, *Social Economics*, Adelphi, New York, NY, USA, 1927.
- [11] L. E. Parker, "The effect of heterogeneity in teams of 100+ mobile robots," in *MultiRobot Systems Volume II: From Swarms to Intelligent Automata*, pp. 205–215, Kluwer, 2003.
- [12] B. P. Gerkey, *Gerkey. on multi-robot task allocation [Ph.D. thesis]*, University of Southern California, 2003.
- [13] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," *Cooperative Robots and Sensor Networks*, vol. 604, Springer International Publishing, pp. 31–51, 2015.
- [14] S. Liemhetcharat and M. Veloso, "Weighted synergy graphs for effective team formation with heterogeneous ad hoc agents," *Artificial Intelligence*, vol. 208, pp. 41–65, 2014.
- [15] M. Scheutz, P. Schermerhorn, and P. Bauer, "The utility of heterogeneous swarms of simple UAVS with limited sensory capacity in detection and tracking tasks," in *Proceedings of the 2005 IEEE Swarm Intelligence Symposium, SIS 2005*, pp. 265–272, USA, June 2005.
- [16] E. King, Y. Kuwata, M. Alighanbari, L. Bertuccelli, and J. How, "Coordination and control experiments on a multi-vehicle testbed," in *Proceedings of the 2004 American Control Conference (AAC)*, pp. 5315–5320, usa, July 2004.
- [17] L. Zhang, H. Zhong, and S. Y. Nof, "Adaptive fuzzy collaborative task assignment for heterogeneous multirobot systems," *International Journal of Intelligent Systems*, vol. 30, no. 6, pp. 731–762, 2015.
- [18] P. Brucker, *Scheduling Algorithms*, Springer, 2007.
- [19] B. Gerkey and M. Mataric, "A framework for studying multi-robot task allocation," in *Proceedings of the In Proceedings of the Second International Naval Research Laboratory Workshop on Multi-Robot Systems*, 2003.
- [20] R. T. Nelson, C. A. Holloway, and R. Mei-Lun Wong, "Centralized scheduling and priority implementation heuristics for a dynamic job shop model," *IIE Transactions*, vol. 9, no. 1, pp. 95–102, 1977.
- [21] X. Zhang, Y. Deng, F. T. S. Chan, P. Xu, S. Mahadevan, and Y. Hu, "IFSJS: a novel methodology for the Job-Shop Scheduling Problem based on intuitionistic fuzzy sets," *International Journal of Production Research*, vol. 51, no. 17, pp. 5100–5119, 2013.
- [22] A. J. Davenport, G. Christophe, and J. Christopher Beck, "Slack-based Techniques for Robust Scheduling," in *Proceedings of the 6th European Conference on Planning ECP-01*, pp. 181–192, Toledo, Spain, 2001.
- [23] J. Błażewicz, K. H. Ecker, E. Pesch, G. Schmidt, and J. Węglarz, *Scheduling Computer and Manufacturing Processes*, Springer, Berlin, Germany, 2nd edition, 2001.
- [24] A. Edward, F. Silver David, and R. Peterson, *Inventory Management and Production Planning and Scheduling*, Wiley, 1998.
- [25] B. B. Werger and M. J. Mataric, "Broadcast of local eligibility for multi-target observation," in *Distributed Autonomous Robotic Systems*, pp. 347–356, 2000.

- [26] L. Vig and J. A. Adams, "Multi-robot coalition formation," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 637–649, 2006.
- [27] L. E. Parker and F. Tang, "Building multirobot coalitions through automated task solution synthesis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1289–1304, 2006.
- [28] L. Fanelli, A. Farinelli, L. Iocchi, D. Nardi, and G. P. Settembre, "Ontology-based coalition formation in heterogeneous MRS," in *Proceedings of the International Symposium on Practical Cognitive Agents and Robots, PCAR 2006*, pp. 105–116, Australia, November 2006.
- [29] Y. Zhang and L. E. Parker, "Considering inter-task resource constraints in task allocation," *Autonomous Agents and Multi-Agent Systems*, vol. 26, no. 3, pp. 389–419, 2013.
- [30] C.-H. Fua and S. S. Ge, "COBOS: Cooperative backoff adaptive scheme for multirobot task allocation," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1168–1178, 2005.
- [31] R. Zlot, *An auction-based approach to complex task allocation for multirobot teams [PhD. thesis]*, Carnegie Mellon University, 2006.
- [32] A. Gautam, J. K. Murthy, G. Kumar, S. P. A. Ram, B. Jha, and S. Mohan, "Cluster, Allocate, Cover: An Efficient Approach for Multi-robot Coverage," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pp. 197–203, Hong Kong, October 2015.
- [33] D. Wu, G. Zeng, L. Meng, W. Zhou, and L. Li, "Gini coefficient-based task allocation for multi-robot systems with limited energy resources," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 155–168, 2018.
- [34] R. Dornbusch, S. Fischer, and P. A. Samuelson, "Comparative advantage, trade, and payments in a Ricardian model with a continuum of goods," *American Economic Review*, vol. 67, no. 5, pp. 823–839, 1977.
- [35] L. W. McKenzie, "Specialisation and efficiency in world production," *The Review of Economic Studies*, vol. 21, no. 3, pp. 165–180, 1954.
- [36] R. W. Jones, "Comparative advantage and the theory of tariffs: A multi-country, multi-commodity model," *Review of Economic Studies*, vol. 28, no. 3, pp. 161–175, 1961.
- [37] Y. Tian, "A New Idea about Ricardo's Comparative Advantage Theory on Condition of Multi-Commodity and Multi-Country," *International Journal of Business and Management*, vol. 3, no. 12, pp. 155–160, 2009.
- [38] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.
- [39] M. M. Flood, "The traveling-salesman problem," *Operations Research*, vol. 4, pp. 61–75, 1956.
- [40] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.
- [41] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.
- [42] C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley, 1994.
- [43] J. K. Lenstra and A. H. G. R. Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.
- [44] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," *Mathematics of Operations Research*, vol. 1, no. 2, pp. 117–129, 1976.
- [45] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, pp. 497–520, 1960.
- [46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [47] M. Dorigo, *Optimization, learning and natural algorithms [Ph.D. thesis]*, Politecnico di Milano, Milano, Italy, 1992.
- [48] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.
- [49] J. Bramel and D. Simchi-Levi, "A location based heuristic for general routing problems," *Operations Research*, vol. 43, no. 4, pp. 649–660, 1995.
- [50] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, pp. 568–581, 1964.
- [51] G. A. Croes, "A method for solving traveling-salesman problems," *Operations Research*, vol. 6, pp. 791–812, 1958.
- [52] T. Van Der Zant and L. Iocchi, "Robocup@home: Adaptive benchmarking of robot bodies and minds," in *Proceedings of the International Conference on Simulation Modeling and Programming for Autonomous Robots*, vol. 7072, pp. 171–182, 2010.
- [53] S. Tadokoro, H. Kitano, T. Takahashi et al., "RoboCup-Rescue project: a robotic approach to the disaster mitigation problem," in *Proceedings of the ICRA 2000: IEEE International Conference on Robotics and Automation*, pp. 4089–4094, April 2000.
- [54] Y. Shiozawa, "Final resolution of the ricardian problem on international values," in *Proceedings of the International Conference on Structural Economic Dynamics*, 2012.
- [55] R. Zlot, A. T. Stentz, M. B. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp. 3016–3023, USA, May 2002.
- [56] R. Zlot and A. Stentz, "Market-based multirobot coordination for complex tasks," *International Journal of Robotics Research*, vol. 25, no. 1, pp. 73–101, 2006.
- [57] N. Kalra, D. Ferguson, and A. Stentz, "Hoplites: A market-based framework for planned tight coordination in multirobot teams," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1170–1177, Spain, April 2005.
- [58] T. Morisawa and I. Mizuuchi, "2P1-V06 Allocating Multiple Tasks to Heterogeneous Robots Based on the Theory of Comparative Advantage," *The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec)*, vol. 2015, no. 0, pp. _2P1-V06.1–_2P1-V06.4, 2015.



Hindawi

Submit your manuscripts at
www.hindawi.com

