In Proceedings of the International Conference on Robotics and Automation (ICRA), 2021

Multi-Robot Task Allocation Games in Dynamically Changing Environments

Shinkyu Park, Yaofeng Desmond Zhong, and Naomi Ehrich Leonard

***Abstract*— We propose a game-theoretic multi-robot task allocation framework that enables a large team of robots to optimally allocate tasks in dynamically changing environments. As our main contribution, we design a decision-making algorithm that defines how the robots select tasks to perform and how they repeatedly revise their task selections in response to changes in the environment. Our convergence analysis establishes that the algorithm enables the robots to learn and asymptotically achieve the optimal stationary task allocation. Through experiments with a multi-robot trash collection application, we assess the algorithm's responsiveness to changing environments and resilience to failure of individual robots.**

I. Introduction

We envision a team of autonomous robots strategically interacting with one another to carry out large-scale missions in extensive and unstructured environments. Team members should coordinate with one another to optimally allocate and perform a set of given tasks. In this work, we propose a game-theoretic framework that models decision making in multi-robot task allocation, and we provide analysis to show that our framework enables multi-robot systems to learn and self-organize so that they achieve optimal task allocation in dynamically changing environments.

We consider that each robot in a team of mobile robots selects a task to perform, and, when it is given an opportunity, the robot can revise its current task selection. Such revision of task selection is a crucial capability for multi-robot systems as it allows the team to adapt its allocation of tasks to accommodate dynamically changing environments. We examine a class of task allocation problems for a finite number of tasks, $i = 1, \ldots, M$, where each task i requires the "consumption" of resource q_i, which *grows with time* t. Our proposed decision-making algorithm prescribes how the robots repeatedly revise their task selections to achieve the optimal stationary allocation of tasks.

This class of problems includes scenarios in which q_i is the number of services to be provided in fleet management at location i, q_i is the uncertainty to be managed in environmental surveillance of area i, and q_i is the available resource to be collected in search and retrieval in patch i. In this work, we use multi-robot trash collection, as illustrated in Fig. 1, to demonstrate how our framework can be applied in multi-robot system applications and also to design experiments to evaluate the performance of our framework.

This work was supported by Princeton University's School of Engineering and Applied Science through the generosity of Lydia and William Addy '82.

S. Park, Y. D. Zhong, and N. E. Leonard are with the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, USA. {shinkyu@princeton.edu, y.zhong@princeton.edu, naomi@princeton.edu }

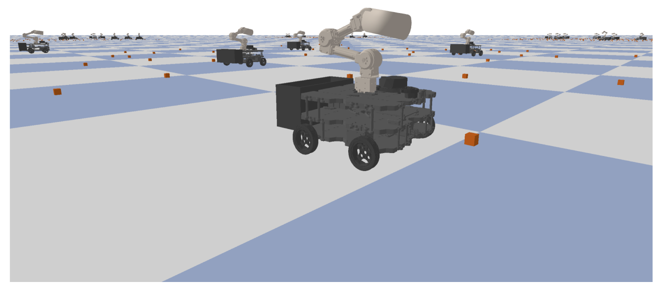


Fig. 1. Multi-robot trash collection application: Each mobile manipulator is tasked with collecting trash q_i, rendered as red objects, in a patch i.

Our primary goal is to design a multi-robot decision-making algorithm that prescribes how each robot selects a task over time, i.e., which q_i to consume over time, so that the team ultimately minimizes the unconsumed resource $(q_1, \ldots, q_M)$. We design the algorithm based on a game theory formalism that was originally conceived to model and assess long-term strategic interactions in a large population of decision-making agents. Using stability analysis methods proposed in [1]–[3], we establish a convergence guarantee of our algorithm, which we also validate through experiments designed in the multi-robot trash collection application.

In the multi-robot research community, our formulation can be categorized as a single-task robot, multi-robot task problem according to the taxonomy articulated in [4]. Existing solutions to this problem class such as market-based approaches [5]–[7], which require strong coordination among the robots to implement auction mechanisms, provide orchestrated ways to allocate tasks to multiple robots. Fully decentralized threshold-based task allocation schemes [8], [9], which are simpler to implement, allow self-interested robots to select tasks based only on information available to individual robots. Closely related to our game-theoretic and population-level formalism, hedonic game approaches [10]–[12] propose distributed task allocation for a large population of agents, where each agent interacts with its neighboring agents to select a utility-maximizing task. The work of [13] establishes a population-level model to analyze and identify emerging collective behavior in multi-robot task allocation. Other important works in the literature include a behavior-based approach [14] and a distributed algorithm for coalition formation for multi-robot task allocation [15].

The following are benefits of our approach. Our decision-making framework builds on the game theory formalism in which each robot independently makes a decision on task selection based on the information on the distribution of the robots' task selections and the environmental resource variables q_i. Hence, unlike market-based approaches [5]–[7], [12], the framework potentially reduces the complexity in orchestrating coordination among multiple robots in their decision making.

Our decision-making algorithm allows the robots to repeatedly revise their task selections in accordance with dynamics of the environment. Therefore, in contrast to some existing approaches [8]–[11], [13]–[15], which are more suitable for task allocation in static environments or lack performance guarantees in dynamically changing environments, our framework ensures multiple self-interested robots to adapt and self-organize to achieve the optimal task allocation in such dynamic environments.

We summarize the highlights of our work as follows:

- **Game-Theoretic Framework for Task Allocation in Dynamically Changing Environments:** We propose a multi-robot task allocation framework that consists of an aggregate model describing underlying dynamics of the environment and a decision-making algorithm prescribing how the robots can effectively revise their task selections.
- **Convergence Analysis and Validation through Experiments:** By leveraging analytical tools recently developed in game theory [1]–[3], we establish that the decision-making algorithm guarantees convergence to efficient task allocation at which the environmental resource variables are minimized. We implement the proposed algorithm in the multi-robot trash collection application to assess its performance in large-scale experiments.

The paper is organized as follows. In §II, we formalize the multi-robot task allocation problem. In §III, we propose a decision-making algorithm as a solution to the problem and provide mathematical analysis that establishes convergence properties of the algorithm. In §IV, we present experimental results to evaluate the effectiveness of our framework in the multi-robot trash collection application. We conclude the paper with a summary and future plans in §V.

II. Multi-Robot Task Allocation

Consider a team of N robots, each selecting and executing one task at a time among M tasks. Each task i is associated with a resource variable q_i, which grows according to an underlying dynamic model of the environment. When a robot selects task i, it contributes to consuming q_i. In this work, we investigate the design of a decision-making algorithm that prescribes how each robot can strategically switch its task selection to allow the robotic team to achieve the optimal task allocation, which minimizes the variables $(q_1, \cdots, q_M)$ associated with all the tasks.

Let $q_i(t)$ denote the resource associated with task i at time instant t and let $x_i(t)$ denote the portion of robots selecting task i at time instant t, defined as

$$x_i(t) = \frac{\#\text{ of robots selecting task } i \text{ at } t}{\text{total number of robots } N}. \tag{1}$$

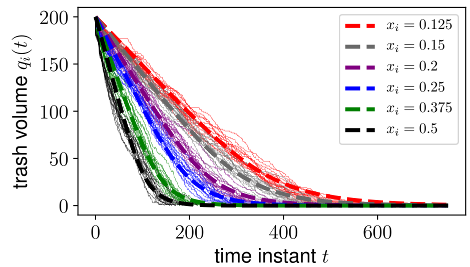


Fig. 2. A comparison of trajectories $q_i(t),\ t \geq 0$ derived from (2) with $R_i = 3.44$, $\alpha_i = 0.036$, and $\beta_i = 0.91$ with the experimental data with 40 robots. The thick dashed lines denote the trajectories of (2) and the thin solid lines represent the experiment data.

We adopt the following dynamic model to describe how $q_i(t)$ changes over time. This model will play an important role in designing the decision-making algorithm:

$$\dot{q}_i(t) = -\underbrace{\mathcal{F}_i\left(q_i(t), x_i(t)\right)}_{\text{consumption rate}} + \underbrace{w_i}_{\text{growth rate}}. \tag{2}$$

Growth rate $w_i \geq 0$, which is constant, specifies the rate of increase of $q_i(t)$ in the environment. The consumption rate $\mathcal{F}_i\left(q_i(t), x_i(t)\right)$ specifies how fast the robots performing task i reduce the value of $q_i(t)$ over time as a feedback function of current resource $q_i(t)$ and portion of robots $x_i(t)$.

The model (2) describes, for each task i, how the associated $q_i(t)$ evolves based on aggregate information $x_i(t)$ on the robots' task selections. This use of aggregate information makes it possible to study (2) as a population game [16] from game theory. By following the same naming convention as in population games, we refer to $x(t) = (x_1(t), \cdots, x_M(t))$ as the *population state* of the multi-robot system, which describes the distribution of the robots' task selections.

We view (2) as an aggregate model that approximates the dynamics of individual robots' consumption of $(q_1(t), \cdots, q_M(t))$. Such an aggregate model is beneficial in designing scalable task allocation algorithms for large multi-robot systems, which we discuss more in detail in §III. The following example is a concrete instance of (2), which we use to design a decision-making algorithm for multi-robot trash collection experiments in §IV.

Example 1: Consider a team of mobile manipulators tasked with collecting trash across multiple patches, as illustrated in Fig. 1. Task i is defined as trash collection in patch i. So $q_i(t)$ denotes the trash volume in patch i at t, and $x_i(t)$ denotes the portion of robots collecting trash in patch i at t. In this scenario, we let consumption rate $\mathcal{F}_i\left(q_i, x_i\right)$ be

$$\mathcal{F}_i\left(q_i, x_i\right) = R_i \frac{e^{\alpha_i q_i} - 1}{e^{\alpha_i q_i} + 1} x_i^{\beta_i}, \tag{3}$$

where R_i, α_i, and β_i are positive constants. The parameter R_i represents the maximum consumption rate of the robots in patch i, and α_i and β_i determine how the consumption rate varies with respect to q_i and x_i. Fig. 2 illustrates, for $N = 40$, $w_i = 0$, and for different constant values of x_i, the comparison between trajectories $q_i(t),\ t \geq 0$ derived by (2) and data obtained using the simulator described in §IV, where the parameters R_i, α_i, β_i are selected to minimize the error between the experimental data and the trajectories $q_i(t),\ t \geq 0$ computed according to (2) with the consumption rate defined by (3).

We impose the following two assumptions on $\mathcal{F}_i$.

Assumption 1: $\mathcal{F}_i$ is a bounded function.

Assumption 2: $\mathcal{F}_i$ is an increasing function of q_i and x_i.

Assumption 2 has the following interpretations in our trash collection example. When there is more trash to remove, it becomes easier for the robots to spot and collect it. Also as more robots are assigned to the same patch i, it will take less time to finish the task.

The primary goal of this work is to design a decision-making algorithm that specifies when each robot can revise its task selection and to which task it switches. We describe the decision-making algorithm as the pair $\mathbb{T}$ and μ where

$$\big(\text{task selection}\big) = \mu\left(q(t), x(t)\right),\ t \in \mathbb{T}. \tag{4}$$

$\mathbb{T}$ is the set of time instants when a robot can revise its task selection and μ is the protocol that assigns one task from the set of available tasks $\{1, \cdots, M\}$ to the robot. We formalize the multi-robot task allocation problem as follows.

*Problem 1 (***Multi-Robot Task Allocation***):* Design a decision-making algorithm (4), i.e., design $\mathbb{T}$ and μ, that solves the optimization problem defined by

$$\begin{aligned} &\text{minimize } \gamma && (5)\\ &\text{subject to } \limsup_{t\to\infty} q_i(t) \leq \gamma,\ \forall i \in \{1, \cdots, M\} \\ &\qquad\qquad \text{dynamic model (2) for } q(t) \\ &\qquad\qquad \text{decision-making algorithm (4) governing } x(t). \end{aligned}$$

A. Generalizations

Our approach can be extended to complex task allocation where there are inter-dependent tasks. For instance, if tasks i and j are inter-dependent, we can express the dynamic models associated with the tasks by

$$\begin{aligned} \dot{q}_i(t) &= -\mathcal{F}_i\left(q_i(t), q_j(t), x_i(t)\right) + w_i \\ \dot{q}_j(t) &= -\mathcal{F}_j\left(q_i(t), q_j(t), x_j(t)\right) + w_j. \end{aligned}$$

In addition, our framework can be generalized to task allocation for heterogeneous robotic teams by expanding the population state to $(x_i(t), y_i(t))$, which describes the portions of two different robotic groups selecting task i. In this case, the dynamic model expands to

$$\dot{q}_i(t) = -\mathcal{F}_i\left(q_i(t), x_i(t), y_i(t)\right) + w_i.$$

We leave in-depth investigations as future work.

III. Decision-Making Algorithm and Convergence Analysis

Our decision-making algorithm consists of a payoff (reward) mechanism, Poisson clock, and task revision protocol, which we describe in §III-A. Convergence analysis, provided in §III-B, establishes that the algorithm converges to its unique equilibrium state at which $(q_1(t), \cdots, q_M(t))$ attain the minimum of the optimization problem (5).

A. Decision-Making Algorithm

Payoff (Reward) Mechanism: In our algorithm, the robots tend to select tasks returning higher payoffs. A *payoff mechanism* specifies how the payoffs are determined from the variables relevant to the robots' decision making. We consider the following payoff mechanism, which defines the payoff p_i associated with task i:

$$p_i(t) = q_i(t) + \nu\big(-\mathcal{F}_i(\gamma^*, x_i(t)) + w_i\big), \tag{6}$$

where ν is a nonnegative constant and γ^* is the minimum of (5) satisfying $\mathcal{F}_i(\gamma^*, x_i^*) = w_i,\ \forall i \in \{1, \cdots, M\}$ at the stationary population state $x^* = (x_1^*, \cdots, x_M^*)$.

In the case $\nu = 0$, the payoff mechanism (6) allows the robots to switch their task selections based only on the instantaneous values of $(q_1(t), \cdots, q_M(t))$. Such model-free mechanism design is useful for the scenario where the robots do not have access to the dynamic model (2).

When $\nu > 0$, the payoff mechanism (6) generates rewards that prompt the robots to balance being responsive to the change in $q_i(t)$ and selecting the task allocation corresponding to the stationary population state x^*. However, this requires the robots to *know* the consumption rate $\mathcal{F}_i$ and the growth rate w_i for every task. Interestingly, such a selection of ν for (6) enables the robots' task selection to be robust against perturbation in the dynamic model (2) and their decision making. We demonstrate this using experimental results in §IV.

Poisson Clock: Each robot is allowed to switch its task selection at every tick of the *Poisson clock*. Formally, the Poisson clock is defined as a set $\mathbb{T}_\lambda = \{t_1, t_2, \cdots\}$ of time ticks, where each t_k in $\mathbb{T}_\lambda$ is defined as the k-th jump time of the Poisson process with parameter λ. Equivalently, the time difference between the two consecutive ticks t_k and t_{k+1} is drawn according to the exponential distribution: $t_{k+1} - t_k \sim \exp(\lambda)$. In our algorithm, the robots' Poisson clocks are defined by identical and independent Poisson processes, each has the same parameter λ, and each robot can assess the payoff $(p_1(t_k), \cdots, p_M(t_k))$ at every t_k in $\mathbb{T}_\lambda$.[1]

Task Revision Protocol: At every t_k of its own Poisson clock $\mathbb{T}_\lambda$, each robot switches its task selection according to the probabilistic task revision protocol defined by

$$\begin{aligned} &\mathbb{P}\Big(\text{robot switches current task } i \text{ to task } j\Big) \\ &\quad = \varrho\left[p_j(t_k) - p_i(t_k)\right]_+ && (7a)\\ &\mathbb{P}\Big(\text{robot stays with current task } i\Big) \\ &\quad = 1 - \varrho \sum_{j=1}^{M} \left[p_j(t_k) - p_i(t_k)\right]_+, && (7b) \end{aligned}$$

where $p_i(t_k)$ is the payoff assigned to task i according to (6), $[r]_+$ is the positive part of r, i.e., $[r]_+ = \max(0, r)$, and ϱ

[1] For simplicity, in this paper, we assume the robots are able to evaluate $q(t), x(t)$, each time they revise their task selections. We leave it as a future work to extend our framework to the scenario where each robot maintains its own estimates of $q(t), x(t)$ and computes the payoffs for the task revision.

is a positive constant satisfying $\varrho \sum_{j=1}^{M} [p_j(t_k) - p_i(t_k)]_+ \leq 1$, for all $t_k \in \mathbb{T}_\lambda$ and for all $i \in \{1, \cdots, M\}$.

Under the protocol (7), each robot tends to switch its task selection to another one incurring a higher reward than its current task, and the tendency to switch to another task increases as the task returns a higher reward.

B. Convergence Guarantee

Large population approximation in game theory [17] suggests that as the number N of robots tends to infinity, the variable $q(t)$ and the population state $x(t)$ can be approximated, with high accuracy, by the solution of the differential equations given by

$$\dot{q}_i(t) = -\mathcal{F}_i\left(q_i(t), x_i(t)\right) + w_i \tag{8a}$$

$$\dot{x}_i(t) = \frac{\varrho}{\lambda} \sum_{j=1}^{M} x_j(t) \left[p_i(t) - p_j(t)\right]_+ - \frac{\varrho}{\lambda} x_i(t) \sum_{j=1}^{M} \left[p_j(t) - p_i(t)\right]_+, \tag{8b}$$

where the payoff $p_i(t)$ is determined by the payoff mechanism (6). Note that the protocol (7) is embedded in (8b) and λ is the parameter of the Poisson clock.

We establish convergence of our decision-making algorithm by examining stability of the equilibrium state of (8). Our analysis hinges on analytical tools developed for higher-order learning in multi-agent games [1]–[3]. Although the same results hold for any $\nu \geq 0$, to simplify the presentation, we consider the case where $\nu = 0$ for (6) in which case the payoff is determined as $p_i(t) = q_i(t),\ \forall i \in \{1, \cdots, M\}$.

We begin by identifying that the equilibrium state (q^*, x^*) of (8) satisfies

$$\mathcal{F}_i\left(q_i^*, x_i^*\right) = w_i,\ \forall i \in \{1, \cdots, M\} \tag{9a}$$

$$\gamma^* = q_1^* = \cdots = q_M^* \tag{9b}$$

where γ^* is the minimum of the optimization (5). We adopt the following definition for the neighborhood set of (q^*, x^*).

Definition 1 (Neighborhood Set of (q^*, x^*)): Given a positive constant δ, the neighborhood set $\mathbb{S}_\delta$ of (q^*, x^*) is defined as

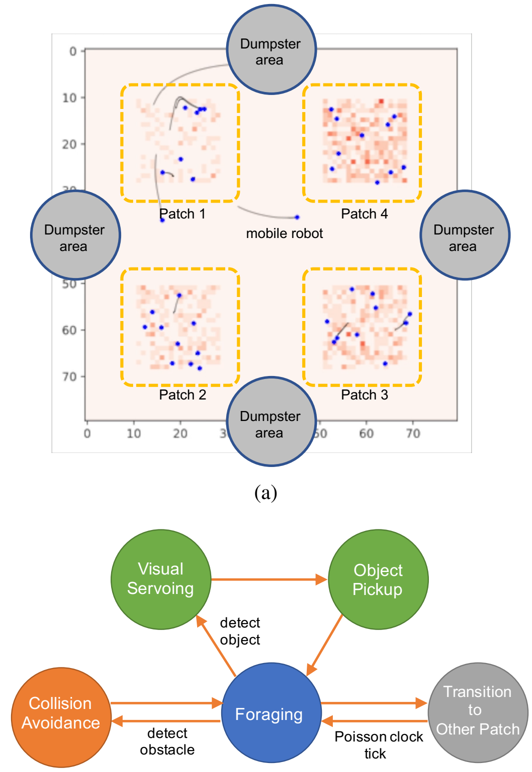
$$\mathbb{S}_\delta = \left\{(q, x) \in \mathbb{R}^M \times \mathbb{X} \,\middle|\, \|(q, x) - (q^*, x^*)\|_2 < \delta \right\}.$$

In other words, $\mathbb{S}_\delta$ is a set of points that are within δ-radius from the equilibrium state (q^*, x^*).

To examine convergence properties of our decision-making algorithm, we adopt the standard stability notions from nonlinear system theory [18].

Definition 2 (Global Attractiveness): The equilibrium state (q^*, x^*) of (8) is *globally attractive* if for any initial condition $(q(0), x(0))$ in $\mathbb{R}^M \times \mathbb{X}$, the solution $(q(t), x(t))$ of (8) converges to (q^*, x^*).

Definition 3 (Lyapunov Stability): The equilibrium state (q^*, x^*) of (8) is Lyapunov stable if for any neighborhood set $\mathbb{S}_\epsilon$ of (q^*, x^*), there is another neighborhood set $\mathbb{S}_\delta$ for which



(a)

(b)

Fig. 3. Multi-robot trash collection application. (a) The environment. (b) The finite state machine that defines how each robot operates.

the solution $(q(t), x(t))$ of (8) satisfying $(q(0), x(0)) \in \mathbb{S}_\delta$ stays in $\mathbb{S}_\epsilon$ for all $t \geq 0$.

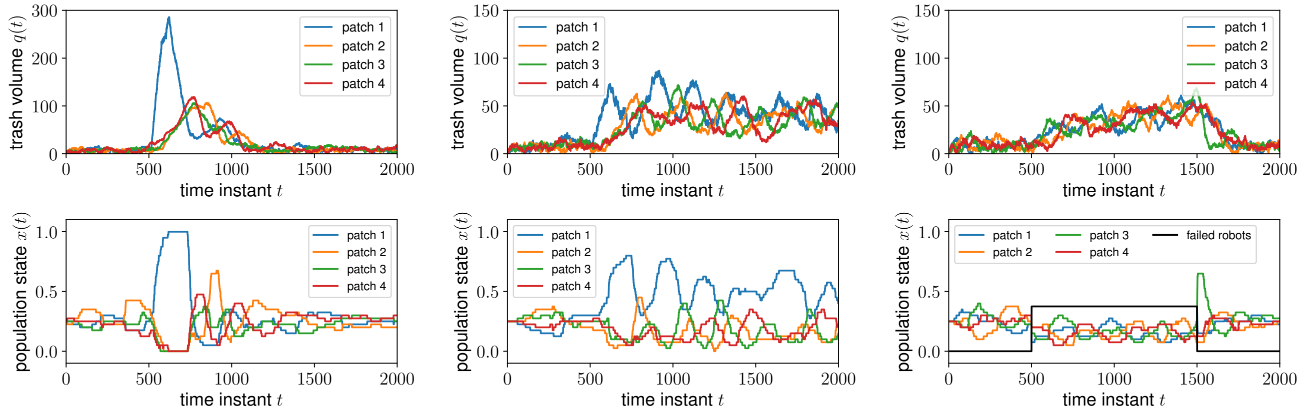
Definition 4 (Global Asymptotic Stability): The equilibrium state (q^*, x^*) is *globally asymptotically stable* if it is globally attractive and Lyapunov stable.

Both notions of global attractiveness and Lyapunov stability are important. The former guarantees the robots to asymptotically achieve the optimal task allocation. The latter ensures that if the robots have already selected nearly optimal task allocation, then our decision-making algorithm maintains the robots' task selections near the optimal allocation. Using the stability definitions, we state our main results.

Theorem 1: Under the decision-making algorithm, described in §III-A, the equilibrium state of (8) is globally asymptotically stable.

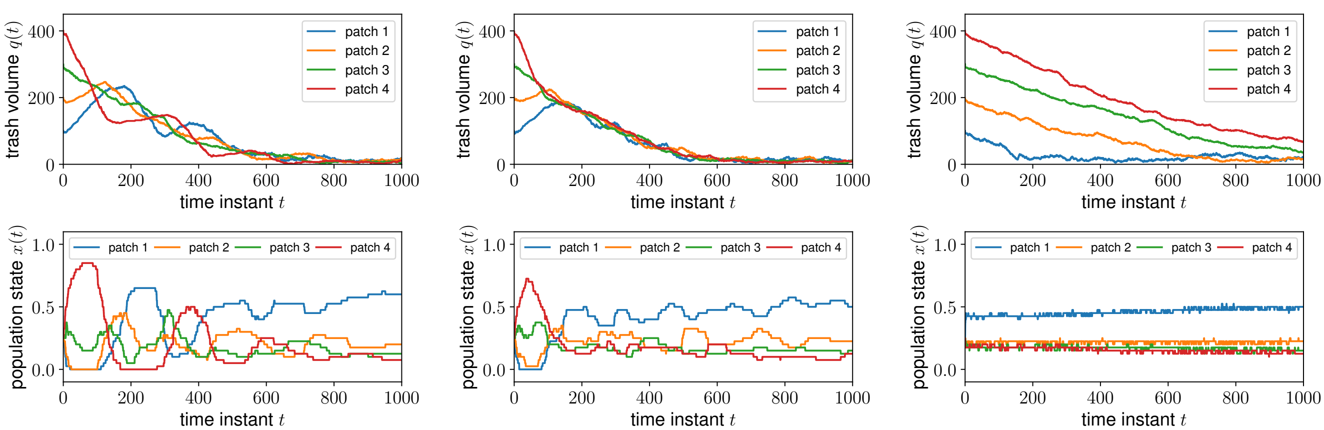
IV. Application to Multi-Robot Trash Collection

We demonstrate the effectiveness of our decision-making algorithm in a multi-robot trash collection application. As explained in Example 1 and illustrated in Fig. 1, the robots are tasked with collecting trash from designated patches, where the variable $q_i(t)$ represents the volume of trash in patch i and $x_i(t)$ denotes the portion of robots assigned to patch i at t. The decision-making algorithm is used to determine when and how each robot revises its patch selection. We examine, under the proposed algorithm, the robots' abilities to adapt in changing environments and tolerate failure of individual robots. We also investigate how the performance



(a) (b) (c)

Fig. 4. Experiment outcomes with changing growth rates and failure of individual robots. The initial conditions of the experiments are set to $q(0) = (0, 0, 0, 0)$, $x(0) = (0.25, 0.25, 0.25, 0.25)$, and $w = (0.5, 0.5, 0.5, 0.5)$.



(a) $\nu = 0$ (b) $\nu = 40$ (c) $\nu = 800$

Fig. 5. Experiment outcomes with different selections of the parameter ν in the payoff mechanism (6). The initial conditions of the experiments are set to $q(0) = (100, 200, 300, 400)$, $x(0) = (0.25, 0.25, 0.25, 0.25)$, and $w = (1.0, 0.5, 0.33, 0.25)$.

of our algorithm depends on the selection of the parameter ν in the payoff mechanism (6).

A. Experiment Settings

As depicted in Fig. 3(a), the environment has 4 patches and there are 4 dumpster areas where the robots can empty their trash collection baskets. We deploy 40 robots and use a cubic object to represent the unit volume of trash. New objects arrive at each patch i according to a Poisson process with parameter w_i, equal to the growth rate in (2), and are placed uniformly randomly within the patch.

As shown in Fig. 1, every robot is equipped with 4-wheel differential drive and a single manipulator to pick up the cubic objects from the ground. Individual robots can hold up to 10 objects at a time in their trash collection baskets. Once the basket is full, the robot should travel to the nearest dumpster area to empty the basket.

We implement a finite state machine, as depicted in Fig. 3(b), to define how each robot operates in the application. In the foraging state, inside its assigned patch, each robot moves toward a randomly selected location where trash is spotted, and when the robot detects a cubic object located within a $0.5\ m$ radius, it uses visual servoing to approach and pick up the object. The robot uses the artificial potential field scheme [19] to avoid collision with other robots. For the revision of the robots' patch selections, we implement our decision-making algorithm described in §III.

We select the parameter λ of the Poisson clock to ensure that each robot does not change its patch selection while it is transitioning between patches. To this end, we assign $\lambda = 8$ since it takes approximately $8\ s$ for each robot to transition between two patches and get ready to pick up trash. The parameter ϱ of the task revision protocol is selected to ensure the probabilistic protocol (7) is well-defined in every scenario of the experiments. The initial condition $q(0) = (100, 200, 300, 400)$ used in the third experiment scenario incurred the smallest value $\varrho = 1/600$ and we adopt this parameter selection for all experiment scenarios.

B. Experimental Results

We consider the following 3 experiment scenarios.

a) **Changing growth rate w_i:** We assess the responsiveness of the proposed algorithm in the environments where the growth rate w_i of the dynamic model (2) changes. We select $\nu = 0$ for the payoff mechanism (6) for which the robots revise their patch selections without the knowledge of the dynamic model. We run experiments for the following two cases. In the first case, we let all the patches initially have the same Poisson growth rate with the parameter $w_1 = \cdots = w_4 = 0.5$. Then, between $t = 500\ s$ and $t = 600\ s$, the growth rate at patch 1 surges to $w_1 = 5.0$ and comes back to the initial rate $w_1 = 0.5$ afterwards. This case is used to evaluate whether the algorithm reacts to the sudden increase of the trash volume in patch 1 and relocates the robots accordingly. In the second case, all the patches have the same initial growth rate as in the first case, but the growth rate at patch 1 switches to $w_1 = 1.0$ at $t = 500\ s$ and remains at the new rate. We design this experiment to assess whether the algorithm can redistribute the robots corresponding to switching growth rates.

Figs. 4(a) and 4(b), respectively, depict the outcomes of the two cases. As we can observe from the figures, under the proposed algorithm, the robots switch their patch selections in response to the changes in the trash volume. In particular, in the first case, upon detection of the increase of the trash volume at patch 1, the robots start to move to patch 1 and collect trash there. As the trash volume in patch 1 decreases, the robots return to the other patches. In the second case, interestingly, despite the fact that they are not aware of the growth rates, the robots revise their patch selections according to the new growth rates. Noticeably, both the trash volume and population state exhibit fluctuations around the equilibrium state, especially after the growth rates change. We recognize that such fluctuations result from the perturbations due to the finite size of the multi-robot system, stochasticity in the Poisson growth rate of the trash volume, and the transition time between the patches.

b) **Failure of individual robots:** To assess whether the algorithm has tolerance to failure of individual robots, we carry out an experiment in which 15 robots (out of the total 40 robots) break down at $t = 500\ s$ and discontinue their operations until $t = 1500\ s$. As in the first scenario, we select $\nu = 0$ for (6) and fix the parameters of the Poisson growth rates to $w_1 = \cdots = w_4 = 0.5$. Fig. 4(c) depicts the experiment outcomes from which we can observe that, despite the failure, the proposed algorithm allows the rest of the team members to reorganize their patch allocation and continue to carry out the trash collection. Notably, even though the multi-robot system has reduced capability in collecting trash between $t = 500\ s$ and $t = 1500\ s$, the remaining 25 robots continue to maintain the optimal uniform distribution of their patch allocation to keep the trash volumes minimized.

c) **Effect of the weight ν in payoff mechanism** (6)**:** We examine how the robots' patch selection pattern depends on the selection of ν in (6). For this purpose, we adopt (3) to specify the consumption rate $\mathcal{F}_i$ of the payoff mechanism (6), where we select the same parameters R_i, α_i, β_i as described in Example 1 and Fig. 2.

Fig. 5 depicts outcomes of the experiments with 3 different values of ν ($\nu = 0, 40, 800$). First of all, as we have discussed in Theorem 1, in all 3 cases, the trash volume and population state in every patch converge to the equilibrium state. For the model-free case ($\nu = 0$) depicted in Fig. 5(a), the trajectories of $q(t)$ and $x(t)$ oscillate before reaching the equilibrium state, whereas when $\nu = 40$, as observed in Fig. 5(b), such oscillation becomes less significant. When ν is substantially large ($\nu = 800$), as seen in Fig. 5(c), the population state tends to immediately converge to its stationary value and the trash volume trajectories have the slowest convergence among the three cases.

The experiment outcomes suggest that when the robots do not know the dynamic model and revise their patch selections based only on the instantaneous trash volume $q(t)$, they tend to go back and forth between different patches until the trash volumes converge; we previously observed a similar phenomenon in Fig. 3. However, when the robots take the dynamic model into account in their decision making, they implicitly coordinate to allow a smaller portion of the robots to react to the changes in the trash volume and the rest to move to the patch locations corresponding to the stationary population state. Hence, the parameter ν for the payoff mechanism (6) can be used to determine the trade-off between the responsiveness of our algorithm to changes in the environment and robustness against perturbation in the robots' decision making. However, when ν is too large, the robots' patch revision becomes insensitive to the changes in the trash volume which results in the slowest convergence to the equilibrium state.

V. Conclusions

In this paper, we proposed a new decision-making algorithm for multi-robot task allocation in dynamically changing environments. Our main results established the convergence guarantee that, under the proposed algorithm, the robots are able to learn and achieve the optimal task allocation and are robust against perturbation in their decision making. We evaluated the performance of the algorithm in the multi-robot trash collection application.

From the experimental results, we learned that proper design of the payoff mechanism ensures the robots to be responsive to changes in the environment and, at the same time, to be robust against perturbation inherent in the environment and their decision making. However, this requires the robots to be aware of the dynamic model of the environment. As a future plan, we will investigate online learning approaches to estimate the dynamic model (2), and further examine how we can adaptively select the weight parameter ν of the payoff mechanism to find the optimal trade-off between responsiveness to environmental changes and robustness against perturbations. We will also investigate the complex task allocation and heterogeneous robotic system scenarios which we discussed in §II-A.

References

[1] S. Park, N. C. Martins, and J. S. Shamma, "Payoff dynamics model and evolutionary dynamics model: Feedback and convergence to equilibria," arXiv:1803.07744, 2019.

[2] ——, "From population games to payoff dynamics models: A passivity-based approach," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 6584–6601.

[3] M. J. Fox and J. S. Shamma, "Population games, stable games, and passivity," *Games*, vol. 4, pp. 561–583, 2013.

[4] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[5] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[6] L. Lin and Z. Zheng, "Combinatorial bids based multi-robot task allocation method," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 1145–1150.

[7] M. B. Dias, "Traderbots: A new paradigm for robust and efficient multirobot coordination in dynamic environments," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, January 2004.

[8] M. J. Krieger and J.-B. Billeter, "The call of duty: Self-organised task allocation in a population of up to twelve mobile robots," *Robotics and Autonomous Systems*, vol. 30, no. 1, pp. 65 – 84, 2000.

[9] W. Agassounon and A. Martinoli, "Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems," ser. AAMAS '02. New York, NY, USA: Association for Computing Machinery, 2002, p. 1090–1097.

[10] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjorungnes, "Hedonic coalition formation for distributed task allocation among wireless agents," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1327–1344, 2011.

[11] I. Jang, H. Shin, and A. Tsourdos, "Anonymous hedonic game for task allocation in a large-scale multiple agent system," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1534–1548, 2018.

[12] H. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[13] K. Lerman, C. Jones, A. Galstyan, and M. J. Matarić, "Analysis of dynamic task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 25, no. 3, pp. 225–241, 2006.

[14] L. E. Parker, "Alliance: an architecture for fault tolerant multirobot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.

[15] L. E. Parker and Fang Tang, "Building multirobot coalitions through automated task solution synthesis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1289–1305, 2006.

[16] L. E. Blume, *The Economy as an Evolving Complex System II*. CRC Press, 2018, ch. Population games.

[17] W. H. Sandholm, *Population Games and Evolutionary Dynamics*. MIT Press, 2010.

[18] H. K. Khalil, *Nonlinear Systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002.

[19] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.