

A Unified Modeling Framework for Task Assignment in Large Heterogeneous Teams

Harish Ravichandar, Kenneth Shaw, Sonia Chernova

Georgia Institute of Technology

Abstract

Large teams of agents have the potential to solve complex multi-task problems that are intractable for a single agent working independently. However, solving complex multi-task problems require leveraging the relative strengths of different agents in the team. We consider the problem of assigning tasks to each member of a heterogeneous team. Specifically, given how different *traits* (capabilities) are required across multiple tasks, we compute the optimal distribution of agents across tasks such that the desired trait requirements are achieved. Inspired by prior work in robot swarms and biodiversity, we categorize agents into different *species* (groups) based on their traits. We model each trait as a continuous variable and differentiate between traits that can and cannot be aggregated from different agents. Our framework is capable of reasoning about both species-level and agent-level differences in traits. Further, we define measures of *diversity* for any given team based on the team's continuous-space trait model. We illustrate the necessity and effectiveness of the proposed framework using detailed simulations and a capture the flag game environment.

1. Introduction

Prior work in multi-agent systems has produced significant insights into the process of engineering collaborative behavior in group of robots [1, 2]. These insights have resulted in large teams of agents capable of accomplishing complex tasks that are intractable for a single agent. The far-reaching applications of robot swarms include environmental monitoring [3], agriculture [4], warehouse automation [5], construction [6], defense [7], and targeted drug delivery [8].

Complex multi-task scenarios, such as search-and-rescue missions and disaster management operations, require a collection of complementary abilities for each task involved. The requirements often dynamically change owing to the harsh operating conditions. Accomplishing such complex tasks with a team of homogeneous agents is often difficult, if not impossible, since any one type of agent is unlikely to possess all the capabilities necessary to meet the requirements. A heterogeneous team of agents, on the other hand, can leverage the relative advantages of the different types of agents within the team.

While a heterogeneous team of agents provides a unique set of advantages, such a team does pose a significant challenge. The challenge is that of optimally assigning specific tasks to the various agents in the team. Indeed, in order to efficiently meet the requirements of complex missions, highly coordinated efforts from all the agents is necessary. Such coordination, however, relies on the team's ability to optimally divide the various tasks among its agents based on their individual and relative strengths.

Assuming that a set of tasks and their corresponding requirements are available, we focus on an instance of the multi-Robot task assignment (MRTA) problem [9, 10, 11]. The MRTA problem involves assigning tasks to each agent of the team to fulfill certain task requirements. In other words, a solution to the MRTA problem specifies which tasks must be assigned to which agents based on task requirements.

In our work, we focus on an instance of the MRTA problem with a focus on large heterogeneous teams. Indeed, assigning tasks to the different agents in the team requires reasoning about their complementary traits and the limited resources of the team. To this end, we introduce a framework to model the traits of the team. Our framework enables the computation of the optimal number of agents from each species necessary to meet the requirements of all the tasks. Inspired by prior work in robot swarms [12] and biodiversity [13],

we assume that each agent in the team belongs to a particular *species* (group). Further, each species is defined based on the *traits* (capabilities) possessed by its members.

One of our contributions is that we reason about the traits of each species in the *continuous* space in the context of task assignment. Prior work [12] has considered binary instantiations of capabilities. However, binary models fail to capture the nuances of the scales and natural variations in capabilities. For instance, consider an unmanned aerial vehicle and a bipedal robot. While both agents share the mobility trait (the ability to move), their speeds are likely to be considerably different. By using continuous-space models, our framework explicitly reasons about the variations among the different species in the team.

In addition to species-level differences in traits, our framework explicitly reasons about agent-level differences within each species. To take these variations into account, we use *stochastic* summaries (independent probability densities) of each species’ traits. Using a stochastic model allows us to automatically find clusters in the trait space and thus aids in identifying the different species and their expected traits along with the observed variations.

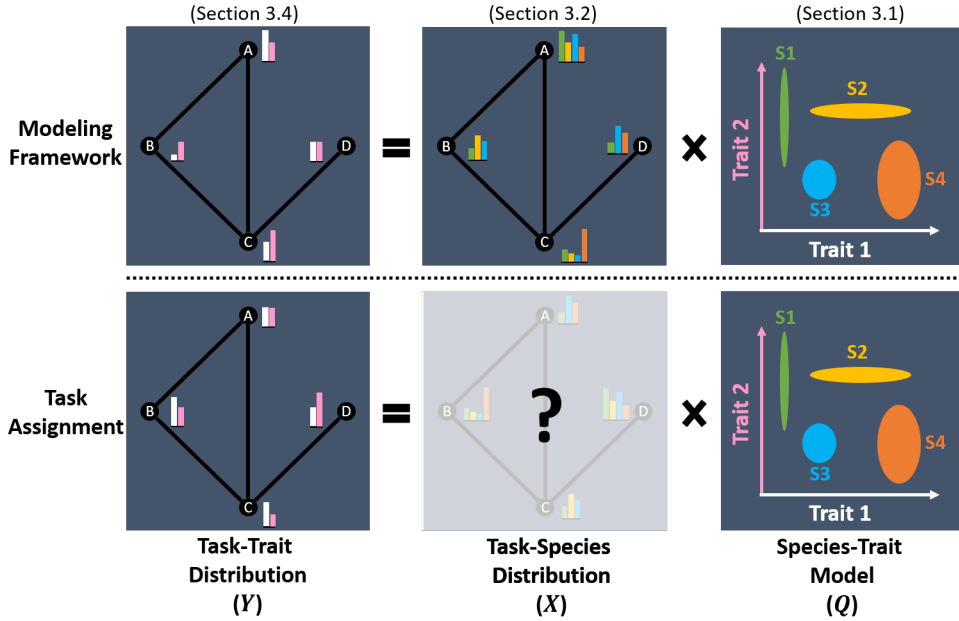


Figure 1: *Top row:* Our modeling framework illustrating how the task-trait distribution is defined based on the task-species distribution and the species-trait model. *Bottom row:* Given the desired task-trait distribution and the species-trait model, we aim to perform the optimal task assignment (as defined by the task-species distribution) by finding such that the desired task-trait distribution is achieved.

In Fig. 1, we illustrate our modeling framework and the task assignment problem. As seen in the top row, the modeling framework captures the effects of task assignments (task-species distribution X) and the species’ traits (species-trait model Q) on how the traits are aggregated for each task (task-trait distribution Y). Our task assignment method, as seen in the bottom row, implicitly identifies the optimal task assignments based on the desired task-trait distribution and the species-trait model. We infer the optimal task assignments implicitly since our algorithm directly optimizes for the transition rates on the task graph which in turn dictate how task assignments vary as a function of time.

Thanks to our stochastic trait model, our modeling framework can be used to model the capabilities of all kinds of agents (both robots and humans). Indeed, humans and robots can possess vastly different traits that are complementary [14]. For instance, when compared to humans, robots can carry heavier payloads, move faster, and be immune to fatigue. On the other hand, humans’ abilities to assimilate and maintain situational awareness, process noisy information, and adapt to highly unstructured environments are unmatched by their robotic counterparts. Further, individual differences are considerable in teams involving humans (see [14]

and references therein). These variations can be readily captured in our framework by categorizing the human agents into a separate set of one or more species.

Based on the trait models, we formulate and solve a constrained optimization problem to distribute the agents across the different tasks to satisfy certain higher-level task requirements (the MT-MR-TA problem). The topology of the tasks is modeled as a densely connected graph with each node representing a task and or a physical location. For any distribution of agents on the graph, based on our trait models that relate agents and traits, we can compute the corresponding trait distribution across the graph.

We compute the optimal transition rates to redistribute the agents on the graph such that the desired trait distribution is achieved and maintained as quickly as possible. We assume that the task requirements are expressed in terms of the desired trait distribution (the ideal distributions of traits across all the tasks required to complete the task). Further, our optimization explicitly reasons about the expected variance of the trait distribution. To enable this reasoning, we study the effect of the variance of the agents' traits on the achieved trait distribution and derive a closed-form expression to compute the variance.

When reasoning about trait requirements, attention must be paid to the fact that not all capabilities are improved in quantity by aggregation of individual agents' abilities. For instance, any number of slow agents do not add up to a fast agent. Taking this observation into account, we consider two types of traits: *cumulative* and *non-cumulative*. We consider a trait to be (non) cumulative if it can (not) be aggregated from different agents in order to achieve certain task requirements. While examples of cumulative traits include ammunition, equipment, and coverage area, those of non-cumulative traits include speed, special skills, and training.

Our framework allows for the optimization of two separate goals: *exact matching* and *minimum matching*. In exact matching, the algorithm aims to distribute the agents such that the achieved trait distribution is as close to the desired as possible. In minimum matching, the algorithm aims to distribute the agents such that the achieved trait distribution is higher than or equal to the desired as possible, i.e., over-provisioning is not penalized.

Finally, we extend the *diversity measures* introduced in [12] to the continuous space. We derive two separate diversity measures, one for each goal function. The diversity measures provide insights about the trait-based heterogeneity of the team. Specifically, the diversity measures help define a minimum subset of the species that can collectively compensate for the rest of the team.

Our key contributions include:

1. A unified modeling framework that can be used to model both human and robot agents in a large heterogeneous team.
2. A stochastic trait model that captures both between-species and within-species variations.
3. A task assignment and distribution algorithm based on our modeling framework. Our algorithm is capable of task assignment with respect to two separate goals: exact matching and minimum matching.
4. Derivation of diversity measures, corresponding to each goal, for teams with continuous trait models.
5. Evaluation of the proposed framework using detailed simulations and a capture-the-flag game environment.

2. Related Work

Significant efforts have been focused on problems in multi-robot systems task assignment (MRTA). As a result, numerous categories of problems and the associated solutions have been proposed for MRTA [9, 10, 11]. Broadly, the problems are categorized based on three binary characteristics: (1) Task type (single-robot vs multi-robot), (2) Robots type (single-task vs multi-task, and (3) Assignment type (instantaneous vs time-extended) [9]. While task type indicates the number of robots required to complete the tasks, robot type indicates whether the robots are capable of performing a single task or multiple tasks. The assignment type is used to differentiate between tasks that involve scheduling constraints and those that do not. Indeed, numerous approaches for task assignment are available in the literature. However, we limit our focus to approaches involving multi-agent systems. We refer readers to [9, 10] for comprehensive categorizations and examples of all approaches pertaining to task assignment.

Approaches involving single-task robots, solve the assignment problem by assuming that each agent is specialized and can only perform one task. The method proposed in [15] addresses a transportation task involving multiple single-task robots. Some of the items to be transported can be transported by a single agent, while others need coordinated efforts from several agents. The authors of [15] use a greedy set-partitioning algorithm to form coalitions of robots required to perform the tasks. Potential coalitions are iteratively computed for each task involved. The coalition formation algorithm introduced in [15] was later modified to be more suitable for multi-agent systems [16]. The modified algorithm in [16] reduced the communication effort, balanced the coalitions, and constrained the requirements to specify if and when all the required traits must be possessed by a single robot. These approaches, however, require the listing of all potential coalitions and thus are not suitable for problems involving large number of possible coalitions. Indeed, the number of possible coalitions is a factor of both the number of agents in the team and the inherent diversity of the team. Specifically, as the number of agents in the team and their similarities increase, so does the number of possible coalitions. Our work, on the other hand, is scalable with the number of agents as it does not list all possible coalitions.

Scalability issues are addressed by the methods introduced in [17, 18, 19]. In [17], multiple tasks are assigned to a team of homogeneous robots. The authors develop of a continuous abstraction of the team by modeling population fractions and defining the task allocation problem as the selection of rates of robot switching from and to each task. We adopt a similar methodology to make sure that our approach is scalable. In [18], the authors extend the method in [17] with a wireless communication-free quorum sensing mechanism in order to reduce task assignment time. In [19], a decentralized approach for heterogeneous robot swarms is introduced. The approach computes optimal rates at which the agents must switch between the different tasks. These rates, in turn, are used to compute probabilities that determine stochastic control policies of each agent. policies for individual robots. However, A common shortcoming of the approaches in [17, 18, 19] is that they assume that the desired distribution of agents across the tasks is known.

Auction or market-based methods also provide solutions to the MRTA problem involving single-task robots [20, 21, 22, 23]. In [20], the agent responsible for any given task is the agent who discovers the task. Once discovered, the agent holds an auction to recruit other agents into a coalition. The authors of [21] introduce combinatorial bidding to form coalitions and provides explicit cooperation mechanism for agents to form coalitions and bid for tasks. A homogeneous task assignment algorithm for robot soccer is presented in [22]. Sensed information from the agents are shared to compute a shared potential function that would help the agents move in a coordinated manner. We refer readers to [23] for a survey of market-based approaches applied to multi-robot coordination. A common trait of auction or market-based methods is that they require extensive communication for bidding and scale poorly with the number of agents in the team. Further, the methods discussed thus far are limited to single-task robots. In contrast, our approach considers robots capable of performing multiple tasks.

Our work falls under the category of *Multi-Task Robots Multi-Robot Tasks Instantaneous Assignment (MT-MR-IA)* problem [9]. In other words, we are interested in task assignments for a team of multiple agents, each capable of performing several tasks. The assignment type is instantaneous since our task assignment does not reason about future task assignments or scheduling constraints. The MT-MR-IA is an instance of the set-covering problem in combinatorial optimization and is known to be strongly NP-Hard [9]. Albeit not developed for MRTA, a few heuristics-driven solutions have been proposed for the set covering problem [24, 25].

Based on prior work in set covering problems, the authors of [26] extended their prior work on task assignment for single-task robots (introduced in [15]) to a distributed set-covering algorithm to solve a MT-MR-IA problem. The problem results in overlapping coalitions since the agents are allowed to contribute their capabilities to more than one task at a time. An auction-based solution to the MT-MR-IA problem is proved by the CoMutaR (Coalition formation based on Multitasking Robots) framework [27]. The CoMutaR framework focuses on coalition formation as well as coordination withing each coalition.

Our framework shares the most similarities with the task assignment framework introduced in [12]. In contrast to all prior work discussed thus far, the framework in [12] is capable of receiving the task requirements provided in the form a desired trait distribution cross tasks. We take a position similar to [12], and do not assume that the desired distribution of agents is known. Another similarity between our

approach and [12] is being suitable for a decentralized implementation. Thus, both approaches are scalable in the number of agents and their capabilities, and are robust to changes in the agent population.

While our work shares several similarities by [12], there are a number of notable relative advantages. First, our species-trait model is continuous, while [12] deals with binary model. Second, we differentiate between cumulative and non-cumulative traits. Third, we consider the inherent randomness in the agents' traits. The framework in [12] utilizes deterministic and binary models of traits, thereby failing to capture the variations at both species and agent levels. Finally, the diversity measures introduced in [12] are limited to binary trait models. In our work, we introduce similar definitions for continuous-space models.

3. Modeling framework

In this section, we introduce our modeling framework that enables task assignment in large heterogeneous teams.

Example: We illustrate the modeling framework using an example task assignment problem. We progressively build the example as we introduce the different parts of the framework. In this example, the team is made up of 4 species, each consisting of 25 agents. Each species is categorized based on the following 4 traits: viewing distance (in miles), speed (in miles per hour), number of health packs, and ammunition.

3.1. Trait Model

Base model: Consider a heterogeneous team of agents. We assume that each agent is a member of a particular species. The number of species $S \in \mathbb{N}$ is finite, and the number of agents in the i th species is denoted by N_i . Thus, the total number of agents in the team is given by $N = \sum_{i=1}^S N_i$. We define each species by its abilities (*traits*). Specifically, the traits of each species are defined as follows

$$q^{(s)} \triangleq [q_1^{(s)}, q_2^{(s)}, \dots, q_U^{(s)}], \quad \forall s = 1, 2, \dots, S \quad (1)$$

where $q_i^{(s)}$ is the i^{th} trait of the s^{th} species, and U is the number of traits. Thus, the traits of the team is defined by a $S \times U$ *species-trait matrix* $Q = [q^{(1)T}, \dots, q^{(S)T}]^T$, with each row corresponding to one species.

Stochastic traits: To capture the natural variation found in each species, we maintain a stochastic summary of each species' traits. Specifically, each element of Q is assumed to be an independent Gaussian random variable: $q_i^{(s)} \sim \mathcal{N}(\mu_{si}, \sigma_{si}^2)$. Thus, expected value of the species-trait matrix $\mu_Q = [\mu_{q^{(1)}}, \dots, \mu_{q^{(S)}}]^T$ and the corresponding matrix summarizing the variance of each of its elements are given by

$$\mathbb{E}\{Q\} = \mu_Q = \begin{bmatrix} \mu_{11} & \dots & \mu_{1U} \\ \vdots & \ddots & \vdots \\ \mu_{S1} & \dots & \mu_{SU} \end{bmatrix} \quad \text{Var}\{Q\} = \Sigma_Q = \begin{bmatrix} \sigma_{11}^2 & \dots & \sigma_{1U}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{S1}^2 & \dots & \sigma_{SU}^2 \end{bmatrix} \quad (2)$$

The proposed framework learns both the means and variances directly from data. Given the trait values of each agent in the team and the number of species S , a Gaussian mixture model (GMM) is used to approximate the joint distribution over the traits of the team. The parameters of the GMM with S Gaussian kernels and diagonal covariance matrices are learned using the Expectation Maximization algorithm. Each Gaussian cluster in the mixture is assumed to represent a single species using the mean and variance values for each trait.

Example: Let the expected value of the species-trait matrix and the corresponding matrix of variances for our example team be given by

$$\mu_Q = \begin{bmatrix} 0.1 & 15 & 20 & 140 \\ 0.3 & 10 & 10 & 0 \\ 0.5 & 0 & 25 & 60 \\ 0.4 & 45 & 30 & 140 \end{bmatrix} \quad \Sigma_Q = \begin{bmatrix} 0.03 & 1 & 1.5 & 5.6 \\ 0.02 & 1.5 & 0.5 & 0 \\ 0.01 & 0 & 2.4 & 8.7 \\ 0.06 & 2.3 & 3.9 & 9.2 \end{bmatrix} \quad (3)$$

Note that our framework allows for modeling traits of different orders of magnitude. Further, for the same trait, the variation observed in each species is different. For instance, consider the ammunition trait (4th columns of μ_Q and Σ_Q). The distribution of this trait is considerably different in each species. Specifically, while Species 1 has the largest average units of ammunition (140), it also has the smallest variance (5.6). On the other hand, Species 3 has the considerably lower units of ammunition (60) while its variance (8.7) is considerably higher than that of Species 1. Encoding these aspects of each species enables the task assignment to reason about the various trade-offs when recruiting agents to meet the task requirements.

Cumulative traits: We model the cumulative traits as continuous variables (i.e., $q_i^{(s)} \in \mathbb{R}_+, \forall i \in \mathcal{C}$), and the non-cumulative traits as binary variables (i.e., $q_j^{(s)} \in \{0, 1\}, \forall j \notin \mathcal{C}$), where $\mathcal{C} \subseteq \{1, 2, \dots, U\}$ is the set of indices corresponding to cumulative traits. In the case of non-cumulative traits, the binary values are assigned based on the following rule

$$q_j^{(s)} = \begin{cases} 1, & \text{if } \mu_{sj} \geq q_i^{\min} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where q_i^{\min} is a user-defined minimum acceptable value for the i th trait. The binary representation of non-cumulative traits captures information about whether the agents of each species possess the minimum required capabilities. Further, when aggregated (Section 3.4), the binary representation provides the total number of agents meeting the minimum requirements, as opposed to aggregating the trait values.

Example: In our example, the first two traits (viewing distance and speed) are non-cumulative since they can't be aggregated. The minimum acceptable values for the traits are $q_1^{\min} = 0.2$ miles, and $q_2^{\min} = 15$ mph. Thus, the modified expected value of the species-trait matrix is given by

$$\mu_Q = \begin{bmatrix} 0 & 1 & 20 & 140 \\ 1 & 0 & 10 & 0 \\ 1 & 0 & 25 & 60 \\ 1 & 1 & 30 & 140 \end{bmatrix} \quad (5)$$

Note that the average viewing distance of Species 1 (0.1 miles) is lower than the minimum requirement of 0.2 miles. Thus, Species 1 is considered as not meeting the requirements for viewing distance and is assigned a zero for the same trait. Similarly, Species 2 and 3 are assigned zeros for speed.

3.2. Task Model

Given the trait model from the previous section, we require the team to accomplish M tasks. We model the topology of the tasks using a densely connected graph $\mathbb{G} = (\mathcal{E}, \mathcal{V})$. The vertices \mathcal{V} represent the M tasks, and the edges \mathcal{E} connect adjacent tasks and each edge represents the possibility of an agent to switch between the corresponding two adjacent tasks. For each species, we aim to identify the optimal transition rate $k_{ij}^{(s)}$ for every edge in \mathcal{E} , such that $0 < k_{ij}^{(s)} < k_{ij,max}^{(s)}$. The transition rate $k_{ij}^{(s)}$ defines the rate which

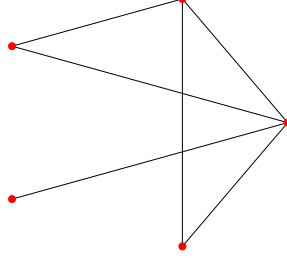


Figure 2: The task graph of our example task assignment problem.

which an agent from species s currently performing task i switches to task j . The transition rates dictate how all the agents are distributed around the graph as time evolves.

Example: In our example, there are 5 tasks and the task graph is shown in Fig. 2. Note that the graph is not fully connected. This reflects the restrictions on how the agents can switch between tasks. For instance, let each task be carried out in a different physical location. The presence (absence) of an edge between any two tasks implies that it is (not) possible for the agents to move between the two tasks. Our framework explicitly takes these restrictions into consideration.

3.3. Agent Model

With the capabilities and the tasks of the team defined, the modeling of individual agents and their assignments remains. The distribution of agents from species s across the M tasks at time t is defined by $\mathbf{x}^{(s)}(t) = [x_1^{(s)}(t), x_2^{(s)}(t), \dots, x_M^{(s)}(t)]^T \in \mathbb{N}^M$. Thus the distribution of the whole team across the tasks at time t can be described using an *abstract state information matrix* $\mathbf{X}(t) \in \mathbb{N}^{M \times S}$.

Example: Let us assume that the initial distribution of agents, perhaps a result of earlier task requirements, be given by

$$\mathbf{X}(0) = \begin{bmatrix} 25 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6)$$

Thus, initially, all the agents from Species 1 are assigned to Task 1, all agents from Species 2 are to Task 2, and so on. Further, no agents are assigned to Task 5. Note that each column adds up to the number of agents in the corresponding species.

The dynamics of the number of agents from Species s at Task i is give by

$$\dot{x}_i^{(s)}(t) = \sum_{\forall j | (i,j) \in \mathcal{E}} k_{ji}^{(s)} x_j^{(s)}(t) - k_{ij}^{(s)} x_i^{(s)}(t) \quad (7)$$

and thus the dynamics of each species' abstract state information can be computed as

$$\dot{\mathbf{x}}^{(s)}(t) = \mathbf{K}^{(s)} \mathbf{x}^{(s)}(t), \quad \forall s = 1, 2, \dots, S \quad (8)$$

where $\mathbf{K}^{(s)} \in \mathbb{R}_+^{M \times M}$ is the rate matrix of species s , defined as follows

$$K_{ij}^{(s)} = \begin{cases} k_{ji}^{(s)}, & \text{if } i \neq j, (i,j) \in \mathcal{E} \\ 0, & \text{if } i \neq j, (i,j) \notin \mathcal{E} \\ -\sum_{i=1, (i,j) \in \mathcal{E}}^M k_{ij}^{(s)}, & \text{if } i = j \end{cases} \quad (9)$$

The solution of the dynamics in (8) for each species is given by

$$\mathbf{x}^{(s)}(\tau) = e^{K^{(s)}\tau} \mathbf{x}^{(s)}(0) \quad (10)$$

Thus, the solution to the dynamics of the abstract state information is given by

$$X(\tau) = \sum_{s=1}^S e^{K^{(s)}\tau} z^{(s)}(0) \quad (11)$$

where $z^{(s)}(0) = X(0) \odot (\mathbf{1} \cdot e_s) \in \mathbb{N}^{M \times S}$, $\mathbf{1}$ denotes a M -dimensional vector of ones, and e_s is the S -dimensional unit vector with its s th element equal to one.

3.4. Trait Aggregation and Distribution

Finally, we represent the trait distribution across the tasks by the *trait distribution matrix* $Y(t) \in \mathbb{R}_+^{M \times U}$ and is computed as

$$Y(t) = X(t)Q \quad (12)$$

Thus, $Y(t)$ is computed by aggregating the traits of all the agents assigned to a particular task. For cumulative traits, each column of $Y(t)$ represents the aggregated amounts of the corresponding trait available at each task, and for non-cumulative traits, each column of $Y(t)$ represents the total number of agents (who meet the minimum requirements for the corresponding trait) assigned to each task.

Note that since the stochastic nature of Q results in the elements of $Y(t)$ being random variables. The expected value of $Y(t)$ can be computed as follows

$$\mathbb{E}\{Y(t)\} = \mu_Y(t) = X(t)\mu_Q \quad (13)$$

and since the elements of Q are independent random variables, the variance of each element of Y is given by

$$\text{Var}\{Y(t)\} = \Sigma_Y(t) = (X(t) \odot X(t)) \Sigma_Q \quad (14)$$

where \odot denotes the Hadamard (entry-wise) product. Furthermore, the covariance between any two elements of Y is given by

$$\text{Cov}\{Y_{ij}, Y_{kl}\} = \begin{cases} \sum_{s=1}^S (x_i^{(s)} x_k^{(s)} \sigma_{sj}^2), & \text{if } j = l \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

Example: The expected value of the species-trait matrix and the initial abstract state information of our example problem are defined in (5) and (6), respectively. Thus, the corresponding initial trait distribution is given by

$$Y(0) = X(0)\mu_Q = \begin{bmatrix} 0 & 25 & 500 & 3500 \\ 25 & 0 & 250 & 0 \\ 25 & 0 & 625 & 1500 \\ 25 & 25 & 750 & 3500 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (16)$$

Given that the agents are distributed as given by (6), the above matrix explains how the team's aggregate capabilities are distributed across the different tasks.

4. Problem Formulation

Based on the modeling framework described in Section 3, we consider the problem of task assignment that achieves a desired trait distribution across tasks. Specifically, we wish to find the transition rates $K^{(s)}$ for each species such that the trait distribution over tasks $Y(t)$, defined in (12), reaches the desired trait distribution Y^* as quickly as possible.

We express the problem as the following optimization problem

$$\tau^*, K^{(s)*} = \arg \min_{\tau, K^{(s)}} \tau \quad (17)$$

$$s.t. \quad X(\tau^*)Q \in \mathcal{G}(Y^*) \quad (18)$$

where $\mathcal{G}(Y^*) : \mathbb{R}_+^{M \times U} \rightarrow \Omega$, named the goal function, is a function that defined the set of admissible trait distribution matrices Ω . As in [12], we consider two goal functions:

1. *Exact matching*: $\mathcal{G}_1(Y^*) = \{Y | Y^* = Y\}$
2. *Minimum matching*: $\mathcal{G}_2(Y^*) = \{Y | Y^* \preceq Y\}$

where \preceq denotes the element-wise less-than-or-equal-to operator. While goal function \mathcal{G}_1 requires achieving the exact desired trait distribution, goal function \mathcal{G}_2 requires the trait distribution be greater than or equal to the desired trait distribution. In other words, \mathcal{G}_1 does not allow any deviation from the desired trait distribution, and \mathcal{G}_2 allows for over-provisioning.

Example: Let the desired trait distribution for our example be given by

$$Y^* = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 25 & 500 & 3500 \\ 25 & 0 & 250 & 0 \\ 25 & 0 & 625 & 1500 \\ 25 & 25 & 750 & 3500 \end{bmatrix}. \quad (19)$$

Note that the initial trait distribution $Y(0)$ is defined in (16). Thus, the task assignment algorithm is required to find the optimal transition rates $K^{(s)*}$ such that the team's trait distribution changes from $Y(0)$ to Y^* as quickly as possible and continuous to remain at Y^* .

5. Diversity Measures

Large heterogeneous teams with multiple species might result in capabilities that are complementary and or redundant. Inspired by [12], we study the properties of the average species-trait matrix μ_Q to understand the similarities and variations among the species of a given team. Measures of team diversity were defined in [12] for species defined by binary traits. In this section, we extend and define diversity measures for species defined by continuous traits. We define two measures of trait diversity for a given team, one for each of the two goal functions defined in Section 4. To this end, we utilize the following definitions.

Definition 5.1. *Minspecies*: In a human-robot team described by an average species-trait matrix μ_Q , a *mins-species* set is a subset of rows of μ_Q with minimal cardinality, such that the system can still achieve the goal $\mathcal{G}(Y^*)$.

Definition 5.2. *Minspecies cardinality*: The cardinality of the mins-species set is defined as the Minspecies cardinality and is represented by the function $\mathcal{D}_G : \mathbb{R}_+^{S \times U} \rightarrow \mathbb{N}_+$ that takes the average species-trait matrix μ_Q as the input and returns the minimum number of rows to achieve the goal $\mathcal{G}(Y^*)$.

5.1. Eigenspecies

First, we define a diversity measure related to the exact matching goal, \mathcal{G}_1 .

Proposition 5.1. *The cardinality of eigenspecies (the minspecies corresponding to goal function \mathcal{G}_1) is computed as follows*

$$\mathcal{D}_{\mathcal{G}_1} = \min |\mathcal{M}_1| \quad (20)$$

$$\text{s.t. } \sum_{s \in \mathcal{M}_1} \alpha_{s\tilde{s}} \mu_{q(s)} = \mu_{q(\tilde{s})}, \forall \tilde{s} \notin \mathcal{M}_1, \forall \alpha_{s\tilde{s}} \in \mathbb{N} \quad (21)$$

$$(22)$$

where \mathcal{M}_1 is a subset of all the species in the team, $|\cdot|$ denotes the cardinality, and \mathbb{N} is the set of all non-negative integers.

Proof. The species-trait matrix Q can be factorized as $\mu_Q = A\hat{\mu}_Q$ where $A \in \mathbb{N}^{S \times |\mathcal{M}_1|}$ and $\hat{\mu}_Q \in \mathbb{R}^{|\mathcal{M}_1| \times U}$. Now, $Y^* = X^* \mu_Q = X^* A \hat{\mu}_Q = \hat{X} \hat{\mu}_Q$ where $\hat{X} = X^* A$. Thus, there exists an agent distribution \hat{X} that can achieve the goal \mathcal{G}_1 with only a subset of the species, defined using the minimal species-trait matrix $\hat{\mu}_Q$. \square

Thus, \mathcal{M}_1 contains the minimal set of species that can exactly match any desired trait distribution without recruiting agents from species not in \mathcal{M}_1 , and $\mathcal{D}_{\mathcal{G}_1}$ denotes the number of species that form \mathcal{M}_1 . Note that weighting factors of the sum are restricted to be natural numbers. The motivation behind this restriction is that, when aggregating traits, the weighting factor corresponds to the number of agents we are considering when aggregating traits.

Example: For the team in our example, the average species-trait matrix is defined in (5). Note that the sum of the first row rows is equal to the last row. Further, no other rows is equal to the weighted (by natural numbers) sum of the remaining rows. Specifically, $\mu_{q(1)} + \mu_{q(2)} = \mu_{q(4)}$. The average species-trait matrix can thus be factorized as $\mu_Q = A\hat{\mu}_Q$, where

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \hat{\mu}_Q = \begin{bmatrix} 0 & 1 & 20 & 140 \\ 1 & 0 & 10 & 0 \\ 1 & 0 & 25 & 60 \end{bmatrix} \quad (23)$$

Thus, $\mathcal{M}_1 = \{1, 2, 3\}$ and consequently $\mathcal{D}_{\mathcal{G}_1} = 3$. In other words, the traits of only one species (Species 4) can be exactly matched by aggregating the traits of other species (Species 1 and 2).

5.2. Coverspecies

Next, we define a diversity measure related to the minimum matching goal, \mathcal{G}_2 .

Proposition 5.2. *The cardinality of coverspecies (the minspecies corresponding to goal function \mathcal{G}_2) is computed as follows*

$$\mathcal{D}_{\mathcal{G}_2} = \min |\mathcal{M}_2| \quad (24)$$

$$\text{s.t. } \sum_{s \in \mathcal{M}_2} \alpha_{s\tilde{s}} \mu_{q(s)} \succeq \mu_{q(\tilde{s})}, \forall \tilde{s} \notin \mathcal{M}_2, \forall \alpha_{s\tilde{s}} \in \mathbb{N} \quad (25)$$

Proof. The species-trait matrix Q can be factorized into two matrices A and $\hat{\mu}_Q$ such that $A \in \mathbb{N}^{S \times |\mathcal{M}_2|}$, $\hat{\mu}_Q \in \mathbb{R}^{|\mathcal{M}_2| \times U}$, and $\mu_Q \preceq A\hat{\mu}_Q$. Now, $Y^* \preceq X^* \mu_Q \preceq X^* A\hat{\mu}_Q$. Thus, there exists an agent distribution $\hat{X} = X^* A$ that can achieve the goal \mathcal{G}_2 with only a subset of the species, defined using the species-trait matrix $\hat{\mu}_Q$. \square

Thus, \mathcal{M}_2 contains the minimal set of species that can satisfy (with potential over-provision) any desired trait distribution without recruiting agents from species not in \mathcal{M}_2 , and $\mathcal{D}_{\mathcal{G}_2}$ denotes the number of species that form such a minimal set.

Example: For the team in our example, the average species-trait matrix is defined in (5). Note that the sum of the first two rows is equal to the the last row. Further, the sum of the first two rows is grater than the third row. Specifically, $\mu_{q(1)} + \mu_{q(2)} = \mu_{q(4)}$ and $\mu_{q(1)} + \mu_{q(2)} \succeq \mu_{q(3)}$. The average species-trait matrix can thus be factorized as $\mu_Q \preceq A\hat{\mu}_Q$, where

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \hat{\mu}_Q = \begin{bmatrix} 0 & 1 & 20 & 140 \\ 1 & 0 & 10 & 0 \end{bmatrix} \quad (26)$$

Thus, $\mathcal{M}_2 = \{1, 2\}$ and consequently $\mathcal{D}_{\mathcal{G}_2} = 2$. In other words, the traits of two species (Species 3 and 4) can be minimum matched by aggregating the traits of the rest of the species (Species 1 and 2).

6. Solution Approach

This section details the proposed solution to the optimization problem defined in (17)-(18). Our solution finds the optimal transition rates $K^{(s)*}$ without assuming the knowledge of the optimal agent distribution X^* .

6.1. Error Measures

We begin by considering the time evolution of average trait distribution over the tasks. To this end, we combine (13) and (10), yielding

$$\mu_Y(\tau, K^{(1,\dots,S)}, x^{(s)}(0)) = \sum_{s=1}^S e^{K^{(s)}\tau} x^{(s)}(0) \mu_{q(s)} \quad (27)$$

In order to satisfy the goal function constraint, as defined in (18), we impose constraints on two error functions [12]. The first error function measures the trait distribution error and is defined separately for each goal function as follows:

$$E_1^{\mathcal{G}_1}(\tau, K^{(1,\dots,S)}, X(0)) = \|Y^* - \mu_Y(\tau)\|_F^2 \quad (28)$$

$$E_1^{\mathcal{G}_2}(\tau, K^{(1,\dots,S)}, X(0)) = \|\max[(Y^* - \mu_Y(\tau)), 0]\|_F^2 \quad (29)$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. Note that we have omitted the dependence of $Y(\cdot)$ on the transition rates and initial conditions for brevity. The second error function measures the deviation from the steady state agent trait distribution and is defined as follows for both goal functions:

$$E_2(\tau, K^{(1,2,\dots,S)}, X(0)) = \sum_{s=1}^S \|e^{K^{(s)}\tau} x^{(s)}(0) - e^{K^{(s)}(\tau+\nu)} x^{(s)}(0)\|_2^2 \quad (30)$$

The first error function E_1 (for both goal functions) penalizes the system when the trait distribution at time τ does not satisfy the appropriate goal, and the error function E_2 penalizes the system if its trait distribution does not reach steady state at time τ .

6.2. Optimization Problem

Based on the definitions in Sections 6.1, we reformulate the optimization problem in (17)-(18) for goal \mathcal{G}_1 as follows

$$\tau^*, K^{(s)*} = \arg \min_{\tau, K^{(s)}} \tau \quad (31)$$

$$s.t. E_1^{\mathcal{G}_1}(\tau, K^{(1,...,S)}, X(0)) \leq \epsilon_1 \quad (32)$$

$$E_2(\tau, K^{(1,...,S)}, X(0)) \leq \epsilon_2 \quad (33)$$

$$\|\Sigma_Y(\tau, K^{(1,...,S)}, X(0))\|_F^2 \leq \epsilon_3 \quad (34)$$

$$k_{ij}^{(s)} \leq k_{ij,\max}^{(s)}, \quad \forall i, j = \{1, \dots, M\}, \quad \forall s = \{1, \dots, S\} \quad (35)$$

$$\tau > 0 \quad (36)$$

where ϵ_1 , ϵ_2 , and ϵ_3 are user-defined positive scalars. Note that the optimization problem for goal \mathcal{G}_2 is identical except that we replace the constraint in (32) with $E_1^{\mathcal{G}_2}(\tau, K^{(1,...,S)}, X(0)) \leq \epsilon_1$.

Note that the solution to the optimization problem in (31)-(36) guarantees minimum levels of performance, both in terms of achieving and maintaining the appropriate goal, as defined by the arbitrary constants ϵ_1 and ϵ_2 , respectively. The constraint in (34) helps ensure that the expected variance of the achieved trait distribution is below a desired threshold. Thus, for each task, the constraint in (34) encourages the system to recruit agents who possess traits (required for the task) with relatively low variance, there by minimizing the variation in the actual trait distribution.

6.3. Analytical Gradients

To efficiently solve the the optimization problem in (31)-(36), we derive and utilize the analytical gradients of all the constraints with respect to the decision variables. In this section we define the analytical gradients of constraints defined in (32)-(36) with respect to the unknowns τ and $K^{(s)}$. We refer the readers to [12] for closed-form expressions of the derivatives of $E_1^{\mathcal{G}_1}$, $E_1^{\mathcal{G}_2}$ and E_2 with respect to both $K^{(s)}$ and τ .

We adapt the closed-form expressions for derivatives of the matrix exponential provided in [28], and use the chain rule to derive the derivatives of $\|\Sigma_Y\|_F^2$ with respect $K^{(s)}$ and τ as follows ¹

$$\frac{\partial \|\Sigma_Y\|_F^2}{\partial K^{(s)}} = \frac{\partial \|\Sigma_Y\|_F^2}{\partial e^{K^{(s)}\tau}} \frac{\partial e^{K^{(s)}\tau}}{\partial K^{(s)}\tau} \frac{\partial K^{(s)}\tau}{\partial K^{(s)}} \quad (37)$$

$$\frac{\partial \|\Sigma_Y\|_F^2}{\partial \tau} = \frac{\partial \|\Sigma_Y\|_F^2}{\partial e^{K^{(s)}\tau}} \frac{\partial e^{K^{(s)}\tau}}{\partial K^{(s)}\tau} \frac{\partial K^{(s)}\tau}{\partial \tau} \quad (38)$$

where

$$\frac{\partial \|\Sigma_Y\|_F^2}{\partial e^{K^{(s)}\tau}} = 4[(\Sigma_Y \Sigma_Q^T) \odot X(t)]z^{(s)}(0) \quad (39)$$

Thus, the closed form expressions of the derivatives are given by

$$\frac{\partial \|\Sigma_Y\|_F^2}{\partial K^{(s)}} = (V^{(s)})^{-T} B^{(s)} (V^{(s)})^T \tau \quad (40)$$

$$\frac{\partial \|\Sigma_Y\|_F^2}{\partial \tau} = \sum_{s=1}^S 1^T [(V^{(s)})^{-T} B^{(s)} (V^{(s)})^T K^{(s)}] 1 \quad (41)$$

where $K^{(s)} = V^{(s)} D^{(s)} (V^{(s)})^{-1}$ is the eigenvalue decomposition of $K^{(s)}$, $D^{(s)} = \text{diag}(d_1, \dots, d_M)$ is the diagonal matrix with the eigenvalues of $K^{(s)}$, $B^{(s)}$ is a $M \times M$ matrix defined as

¹we drop the arguments of Σ_Y for brevity

$$B^{(s)} = \left[(V^{(s)})^T \frac{\partial \|\Sigma_Y\|_F^2}{\partial e^{K^{(s)}\tau}} (V^{(s)})^{-T} \odot W^{(s)} \right] \quad (42)$$

and $W(\tau)$ is a $M \times M$ matrix with its kl th element is given by

$$W_{kl}^{(s)} = \begin{cases} \frac{(e^{d_k\tau} - e^{d_l\tau})}{d_k\tau - d_l\tau}, & k \neq l \\ e^{d_k\tau}, & k = l \end{cases} \quad (43)$$

7. Experimental Evaluation

We evaluate the proposed framework using two sets of experiments. In the following experiments we compare the proposed algorithm's performance with that of a baseline. Our baseline method is a bootstrapped version of the binary-trait-based method introduced in [12]. Since the baseline is not compatible with the continuous-space trait distribution Y^* , we make the following modifications to the baseline. We define binary species-trait matrix to be $\bar{Q} = \text{sign}(\mu_Q)$, where $\text{sign}(\cdot)$ is the signum operator applied to each element of its matrix argument. We also define a modified desired trait distribution for the baseline: $\bar{Y} = \lfloor Y^* \oslash \mu_Y \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function applied to each element of a matrix, \oslash refers to Hadamard (element-wise) division, $\mu_Y = [\mu_Y^1 \oslash 1_M, \dots, \mu_Y^U \oslash 1_M]$, μ_Y^i is the mean value of the i th trait computed across all species, and 1_M is a M -dimensional vector of ones.

The task assignment performance of each method is evaluated in terms of the following error measures:

$$\delta_{G_1}(\mu_Q) = \frac{\|Y^* - \hat{Y}(\mu_Q)\|_1}{2\|Y^*\|_1}, \quad \delta_{G_2}(\mu_Q) = \frac{\|\max(Y^* - \hat{Y}(\mu_Q), 0)\|_1}{\|Y^*\|_1} \quad (44)$$

$$\delta_{G_1}(Q) = \frac{\|Y^* - \hat{Y}(Q)\|_1}{2\|Y^*\|_1}, \quad \delta_{G_2}(Q) = \frac{\|\max(Y^* - \hat{Y}(Q), 0)\|_1}{\|Y^*\|_1} \quad (45)$$

where $\hat{Y}(\mu_Q)$ is the deterministic trait distribution achieved by the algorithm when the species-trait matrix is assumed to be $Q = \mu_Q$ (ignores variance), and $\hat{Y}(Q)$ is the stochastic trait distribution achieved when considering the stochastic species-trait matrix.

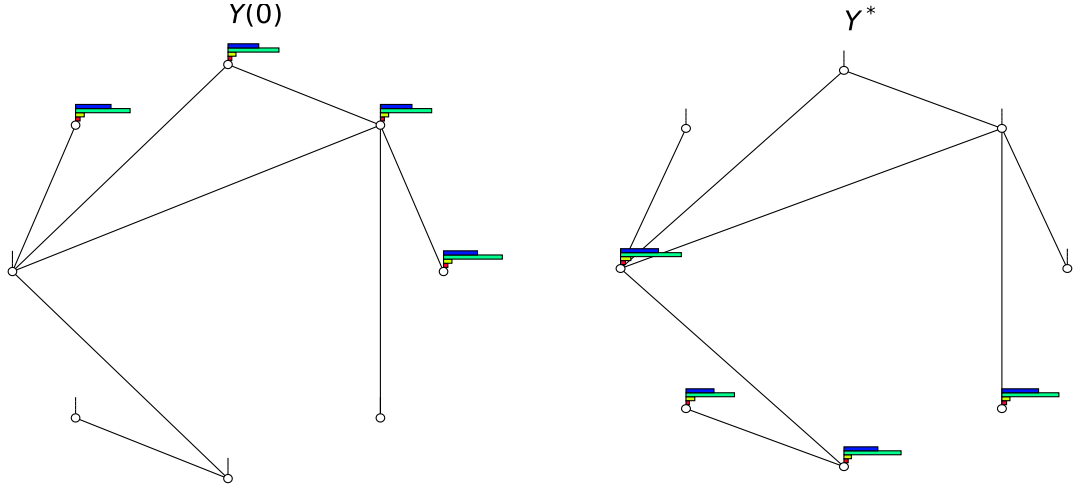


Figure 3: The initial (left) and desired (right) team configurations from an exemplary run with eight tasks (nodes) and four traits. The bar plots denote the trait distribution at each task and the edges represent the possibility of switching between the corresponding tasks.

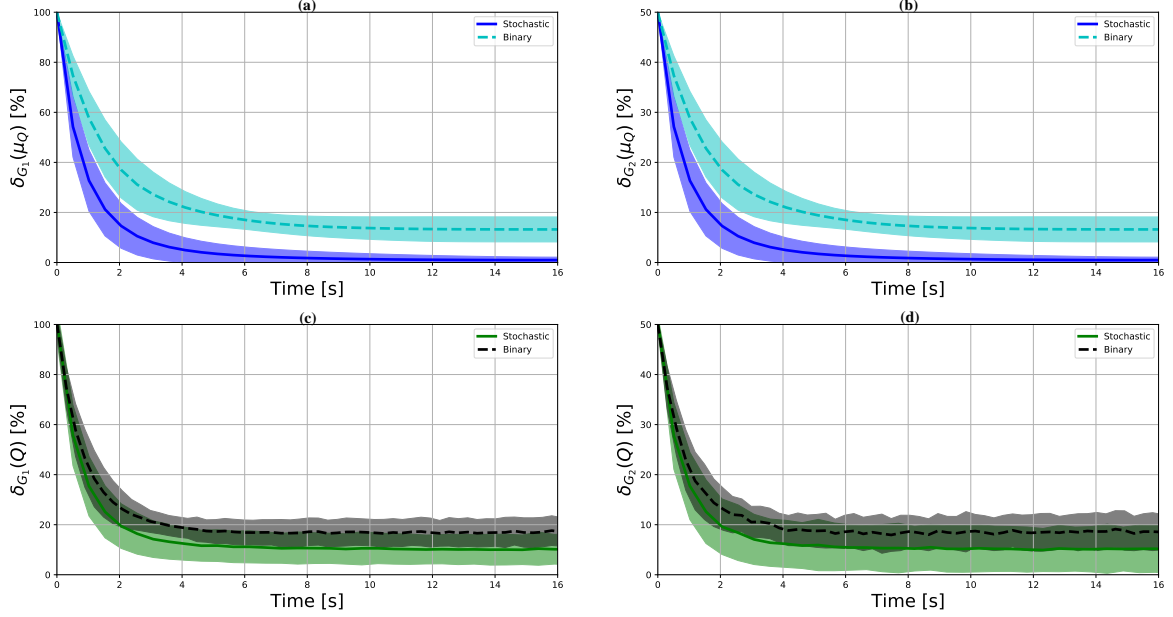


Figure 4: Error measures and their standard deviations of the proposed (stochastic) and baseline (binary) frameworks when optimizing for exact trait matching (\mathcal{G}_1).

7.1. Simulation

In the first set of experiments, we study the performances of the proposed framework and the baseline in terms of matching the desired trait requirements for a large heterogeneous team. To this end, we simulate a task assignment problem with $M = 8$ nodes (tasks), $U = 5$ traits (3 cumulative and 2 non cumulative traits), and $S = 5$ species (each with 200 agents). We present the results computed from 100 independent simulation runs.

In each run, we make the following design choices. The task graph along with its connections is randomly generated. The initial and final agent distributions, $X(0)$ and X^* , are uniformly randomly generated. Based on the obtained X^* , a desired trait distribution Y^* is computed for each run. The expected value of the species-trait matrix is chosen to be $\mu_Q = [a, a, a, b, b]^T$, where each element of $a \in \mathbb{R}_+^U$ is sampled from a uniform distribution: $a_i \sim \mathcal{U}(0, 10)$, and each element of U -dimensional b is sampled from a discrete uniform distribution: $b_i \sim \mathcal{U}\{0, 1\}$. Each element of Σ_Q is sampled from a uniform distribution: $\mathcal{U}(0, 2)$. An example initial and desired trait distribution is illustrated in Fig. 3

To ensure a fair comparison, we limit both frameworks to a maximum of 20 meta iterations of the basin hopping algorithm during each run. In order to measure $\delta_{G_1}(Q)$ and $\delta_{G_2}(Q)$ for each run, 10 samples of the trait-species matrix Q are generated and used to compute $\hat{Y}(Q)$.

Exact Trait Matching: First, we compute the optimal transition rates according to both the proposed framework and the binary trait framework [12] with respect to the function $\text{Goal } \mathcal{G}_1$. The proposed framework is found to converge during 79 of the 100 simulation runs and the binary trait framework during 10 runs. In Fig 4, we present the performances of both frameworks by plotting the errors defined in (44)-(45) as functions of time. Note that the error plots for each method reflect the error measures computed only across the converged runs.

As shown in Fig. 4(a) and (b), the proposed method consistently performs better than the baseline in terms of deterministic performance, as measured by both $\delta_{G_1}(\mu_Q)$ and $\delta_{G_2}(\mu_Q)$. As shown in Fig. 4(c) and (d), when the agents' traits are randomly sampled, the proposed method performs better than the baseline on average. The stochastic nature of the species-trait matrix, force the errors to be larger than 0.

Minimum Trait Matching: Next, we compute the optimal transition rates according to both the

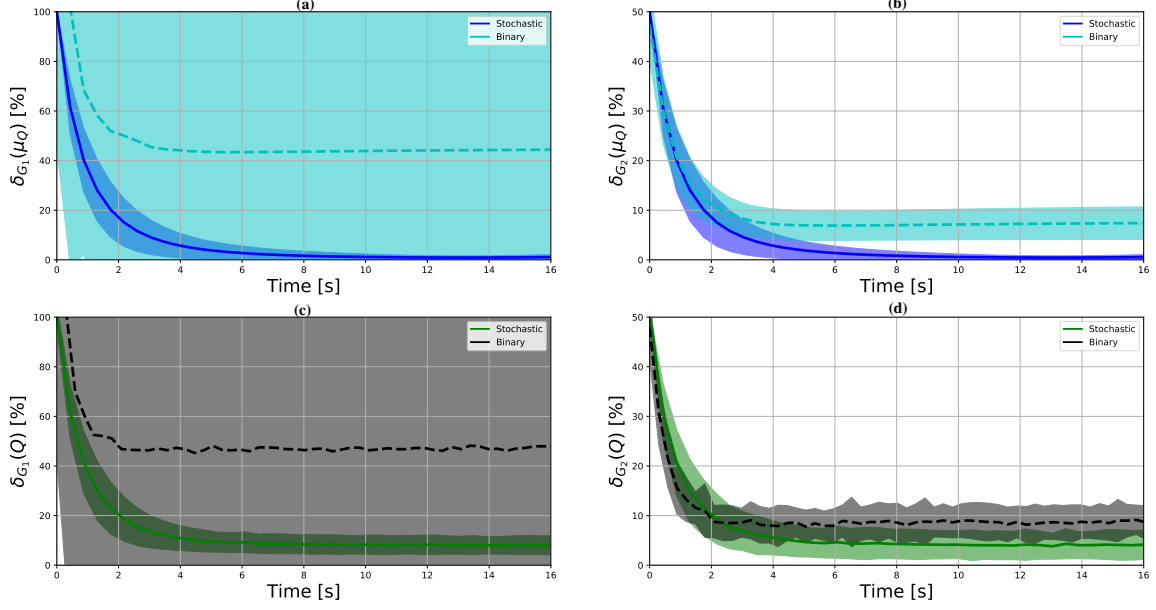


Figure 5: Error measures and their standard deviations of the proposed (stochastic) and baseline (binary) frameworks when optimizing for minimum trait matching (\mathcal{G}_1).

proposed framework and the binary trait model [12] with respect to the function Goal \mathcal{G}_2 . The proposed framework is found to converge during 85 of the 100 simulation runs and the binary trait framework during 16 runs. In Fig 5, we present the performances of both frameworks by plotting the errors defined in (44)-(45) as functions of time.

The proposed framework consistently performance better than the baseline when optimizing to satisfy minimum trait distribution, as measured by $\delta_{G_2}(\mu_Q)$. On average, the proposed method performs better than the baseline when considering stochastic species-trait matrix, as measured by $\delta_{G_2}(Q)$. These assertions are supported by the plots in Fig. 5(b) and (d). In Fig. 5(a) and (c), the baseline exhibits high error and variance in terms of both $\delta_{G_1}(\mu_Q)$ and $\delta_{G_1}(Q)$. This implies that when optimizes for \mathcal{G}_2 , the binary trait model, unlike the proposed method, results in a high level of over-provisioning.

Summary: As demonstrated by the results in this section, reasoning about stochastic traits is essential to consistently satisfy complex task requirements. Firstly, continuous trait distributions might not be achievable when reasoning over binary traits due to incompatibility with the continuous trait space. Further, to construct the modified trait distribution \bar{Y} , the binary trait method is required to consider, at minimum, the average value of each trait in the team. In the process, however, the binary trait model ignores all variations both at the species and individual levels. The advantages of considering these variations are reflected in the results above. Particularly, as seen in Table 1, the proposed assignment framework successfully converged in significantly more runs than the binary trait framework for both exact trait matching (\mathcal{G}_1) and minimum trait matching (\mathcal{G}_2).

	Converged Runs (\mathcal{G}_1)	Converged Runs (\mathcal{G}_2)
Stochastic Trait Model	79/100	85/100
Binary Trait Model [12]	10/100	16/100

Table 1: Number of converged simulation runs.

7.2. Capture the Flag

In this section, we study the question: Does the proposed task assignment framework improve higher-level team performance? To this end, we quantify the effect of the proposed framework on team performance in a capture the flag (CTF) game. We built the environment using the Unity 3D game engine.

We compare the performances of three teams, named A, B, and C. The details pertaining to each team are listed in Table 2. The three tasks (roles) in the game are *defend*, *attack*, and *heal*, in that order. The four traits are *speed*, *viewing distance*, *health*, and *ammunition*, in that order. Indeed, speed and viewing distance are non-cumulative traits while the other traits are cumulative.

	# Species	# Agents per Species	# Tasks	# Traits	Role Assignment
Team A	4	3	3	4	Proposed
Team B	4	3	3	4	Baseline (Binary)
Team C	4	3	3	4	Random

Table 2: Specifications of the teams implemented in the capture the flag environment.

The baseline task assignment strategy, similar to the experiment in Section 7.1, is a bootstrapped version of the binary-trait-based method introduced in [12]. For the random task assignment strategy, each agent is assigned uniformly randomly to one of the three roles. Note that all the algorithms are provided with identical teams, consisting of 12 agents. Thus, any variation in performance is limited to the task assignment strategy used by each team and the inherent randomness of the game.

The traits of the agents are sampled from the following stochastic species-trait matrix

$$\mu_Q = \begin{bmatrix} 1.5 & 15 & 90 & 40 \\ 1.5 & 30 & 60 & 40 \\ 3 & 15 & 80 & 30 \\ 3 & 30 & 350 & 30 \end{bmatrix} \quad \Sigma_Q = \begin{bmatrix} 0.35 & 5 & 10 & 3 \\ 0.35 & 5 & 10 & 3 \\ 0.35 & 5 & 10 & 3 \\ 0.35 & 5 & 10 & 3 \end{bmatrix} \quad (46)$$

The minimum acceptable value for the non-cumulative traits are chosen to be as follows: $q_{min}^{(1)} = 0m/s$ for speed and $q_{min}^{(2)} = 10m$ for viewing distance. The desired trait distribution is designed to be

$$Y^* = \begin{bmatrix} 2 & 2 & 120 & 80 \\ 6 & 6 & 380 & 200 \\ 4 & 4 & 340 & 140 \end{bmatrix} \quad (47)$$

All games are played with two teams at a time, one versus the other. We compare the performances of both the proposed and the baseline frameworks against random task assignment. First, we simulate a total of 100 games between Teams A (proposed) and C (random). Then, we simulate 100 games between Teams B (baseline) and C (random). We consider a team to have won a game it captures the opponent’s flag and brings it back to the starting position. If neither team is able to capture and bring back the opponent’s flag in 120 seconds, then the team with the highest number of live agents is considered the winner. Lastly, if both teams retain the same number of live agents after 120 seconds, the game is considered to have ended in a draw. The performances of the proposed and the baseline frameworks in terms of number of matches won is given in Fig. 6.

Summary: As shown in Fig. 6, the baseline framework (Team B) performs slightly better than random task assignment (241/500 vs 203/500). This observation is perhaps due to the fact that the baseline framework reasons about the average trait values and a corresponding task requirements in the binary trait space. This type of reasoning while limited is still more beneficial than not reasoning about any of the factors that influence team performance. In contrast, the proposed algorithm (Team A) is significantly more likely to win (337/500) than lose (122/500) against a team with random task assignment. Further, based on the z-test, we find that the proposed algorithm’s proportion of wins (337/500) is significantly ($p < 0.001$) higher than that of the baseline (241/500). Thus, the proposed algorithm’s ability to reason about the traits and task requirements translates to high-level team performance.

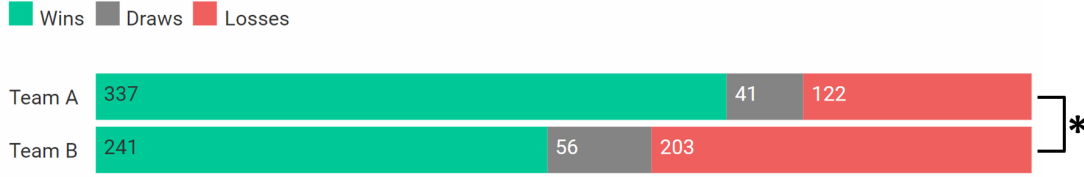


Figure 6: Performance of the proposed framework (Team A) and the baseline framework (Team B) against random task assignment (Team C) in the capture-the-flag game. The asterisk indicates the statistically significant ($p < 0.001$) difference between the proportion of wins of Team A and that of Team B.

8. Conclusion

We presented an modeling framework and a task assignment algorithm for large heterogeneous teams of agents. The members of the team are assumed to belong to different species, each defined by a set of its capabilities. Our framework models capabilities in the continuous space and explicitly takes into account both species-level and agent-level variations. Further, we quantified the diversity of a given team by introducing two separate notions of *minspecies*, each specifying the minimal subset of species necessary to achieve the corresponding goal. Finally, we illustrated the necessity and effectiveness of the proposed framework using two sets of experiments. The experimental results suggest that the proposed framework (1) successfully distributes a large heterogeneous team to meet complex task requirements, (2) consistently performs better than the baseline framework that only considers binary traits, and (3) results in improved higher-level team performance in a simulated capture the flag game.

References

- [1] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [2] Rafael H Bordini, Mehdi Dastani, Jürgen Dix, and A El Fallah Seghrouchni. *Programming Multi-Agent-Systems: 4th International Workshop, ProMAS 2006, Hakodate, Japan, May 9, 2006, Revised and Invited Papers*, volume 4411. Springer, 2007.
- [3] Florian Shkurti, Anqi Xu, Malika Meghjani, Juan Camilo Gamboa Higuera, Yogesh Girdhar, Philippe Giguere, Bir Bikram Dey, Jimmy Li, Arnold Kalmbach, Chris Prahacs, et al. Multi-domain monitoring of marine environments using a heterogeneous robot team. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1747–1753. IEEE, 2012.
- [4] Pratap Tokekar, Joshua Vander Hook, David Mulla, and Volkan Isler. Sensor planning for a symbiotic uav and ugv system for precision agriculture. *IEEE Transactions on Robotics*, 32(6):1498–1511, 2016.
- [5] Peter R Wurman, Raffaello D’Andrea, and Mick Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine*, 29(1):9, 2008.
- [6] Justin Werfel, Kirstin Petersen, and Radhika Nagpal. Designing collective behavior in a termite-inspired robot construction team. *Science*, 343(6172):754–758, 2014.
- [7] Gerardo Beni. From swarm intelligence to swarm robotics. In *International Workshop on Swarm Robotics*, pages 1–9. Springer, 2004.
- [8] Jinxing Li, B Esteban-Fernández de Ávila, Wei Gao, Liangfang Zhang, and Joseph Wang. Micro/nanorobots for biomedicine: Delivery, surgery, sensing, and detoxification. *Sci. Robot.*, 2(4), 2017.
- [9] Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, 2004.
- [10] G Ayorkor Korsah, Anthony Stentz, and M Bernardine Dias. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*, 32(12):1495–1512, 2013.
- [11] Alaa Khamis, Ahmed Hussein, and Ahmed Elmogy. Multi-robot task allocation: A review of the state-of-the-art. In *Cooperative Robots and Sensor Networks 2015*, pages 31–51. Springer, 2015.
- [12] Amanda Prorok, M Ani Hsieh, and Vijay Kumar. The impact of diversity on optimal control policies for heterogeneous robot swarms. *IEEE Trans. Robotics*, 33(2):346–358, 2017.
- [13] Owen L Petchey and Kevin J Gaston. Functional diversity (fd), species richness and community composition. *Ecology letters*, 5(3):402–411, 2002.
- [14] Arwen H DeCostanza, Amar R Marathe, Addison Bohannon, A William Evans, Edward T Palazzolo, Jason S Metcalfe, and Kaleb McDowell. Enhancing human-agent teaming with individualized, adaptive technologies: A discussion of critical scientific questions. Technical report, US Army Research Laboratory Aberdeen Proving Ground United States, 2018.

- [15] Onn Shehory and Sarit Kraus. A kernel-oriented model for autonomous-agent coalition-formation in general environments. In *Australian Workshop on Distributed Artificial Intelligence*, pages 31–45. Springer, 1995.
- [16] Lovekesh Vig and Julie A Adams. Multi-robot coalition formation. *IEEE transactions on robotics*, 22(4):637–649, 2006.
- [17] Spring Berman, Ádám Halász, M Ani Hsieh, and Vijay Kumar. Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics*, 25(4):927–937, 2009.
- [18] M Ani Hsieh, Ádám Halász, Spring Berman, and Vijay Kumar. Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, 2(2-4):121–141, 2008.
- [19] Loïc Matthey, Spring Berman, and Vijay Kumar. Stochastic strategies for a swarm robotic assembly system. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1953–1958. IEEE, 2009.
- [20] José Guerrero and Gabriel Oliver. Multi-robot task allocation strategies using auction-like mechanisms. *Artificial Research and Development in Frontiers in Artificial Intelligence and Applications*, 100:111–122, 2003.
- [21] Liu Lin and Zhiqiang Zheng. Combinatorial bids based multi-robot task allocation method. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1145–1150. IEEE, 2005.
- [22] Douglas Vail and Manuela Veloso. Multi-robot dynamic role assignment and coordination through shared potential fields. *Multi-robot systems*, pages 87–98, 2003.
- [23] M Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.
- [24] Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.
- [25] Reuven Bar-Yehuda and Shimon Even. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2(2):198–203, 1981.
- [26] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial intelligence*, 101(1):165–200, 1998.
- [27] Pedro M Shiroma and Mario FM Campos. Comutar: A framework for multi-robot coordination and task allocation. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4817–4824. IEEE, 2009.
- [28] JD Kalbfleisch and Jerald Franklin Lawless. The analysis of panel data under a markov assumption. *Journal of the American Statistical Association*, 80(392):863–871, 1985.