STRATA: Unified Framework for Task Assignments in Large Teams of Heterogeneous Robots

Harish Ravichandar, Kenneth Shaw, Sonia Chernova

***Abstract*—Large teams of robots have the potential to solve complex multi-task problems that are intractable for a single robot working independently. However, solving complex multi-task problems requires leveraging the relative strengths of different robots in the team. We present *Stochastic TRAit-based Task Assignment (STRATA)*, a unified framework that models large teams of heterogeneous robots and performs optimal task assignments. Specifically, given information on which *traits* (capabilities) are required for various tasks, STRATA computes the optimal assignments of robots to tasks such that the task-trait requirements are achieved. Inspired by prior work in robot swarms and biodiversity, we categorize robots 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 robots. STRATA is capable of reasoning about both species-level and robot-level differences in traits. Further, we extend measures of *diversity* for any given team based on the team's continuous-space trait model. We illustrate the necessity and effectiveness of STRATA using detailed experiments.**

I. Introduction

The multi-robot task assignment (MRTA) problem [5], [11], [10] provides a formulation to represent a wide variety of challenges that require optimally assigning robots to tasks. In this work, we focus on an instance of the MRTA problem with an emphasis on large heterogeneous teams. Teams of heterogeneous robots are particularly well suited for performing complex tasks that require a variety of skills, since they can leverage the relative advantages of the different robots.

We present *Stochastic TRAit-based Task Assignment (STRATA)*, a task assignment algorithm that enables a heterogeneous team of robots to optimally divide the various tasks among its members. STRATA models the topology of tasks as a densely connected graph, with each node representing a task and or a physical location and the edges indicating the possibility of switching between any two tasks. We assume that the optimal robot-to-task associations are unknown and that the task requirements are specified in terms of the various *traits* (capabilities) required for each task. To enable reasoning about a team's traits, we take inspiration from prior work in robot swarms [16] and biodiversity [15], and propose a group modeling approach [1] to model the capabilities of the team. Specifically, we assume that each robot in the team belongs to a particular *species* (group).

In Fig. 1, we illustrate the basic building blocks of STRATA and the task assignment problem. As seen in the top row, STRATA models 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). We also derive a closed-form expression to quantify the effect of the variance of the robots' traits on the achieved task-trait distribution. The task assignment problem, as seen in the bottom row, involves computing the optimal task assignments, given a desired task-trait distribution.

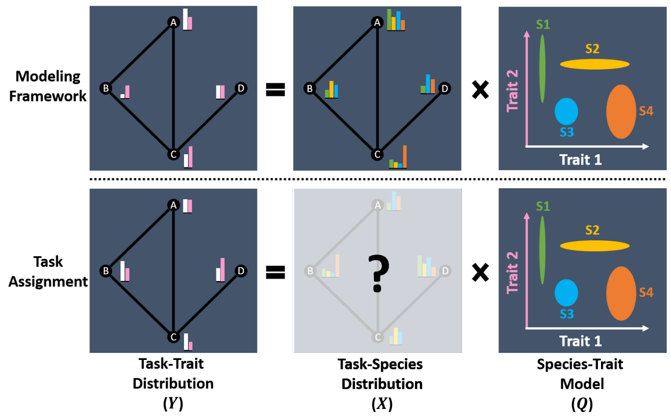


Fig. 1. *Top row:* The modeling framework underlying STRATA. *Bottom row*: The optimal task assignment problem solved by STRATA.

The STRATA representation of task, species, and traits is inspired by [16], which considered binary instantiations of traits. However, binary models fail to capture the nuances in the scales and natural variations of the robots' traits. For instance, consider an unmanned aerial vehicle and a bipedal robot. While both robots share the mobility trait (the ability to move), their speeds are likely to be considerably different. To address these challenges, in STRATA we have extended the representation to model traits in the *continuous* space. Additionally, STRATA also captures robot-level differences within each species by using a *stochastic* trait model.

When reasoning about the collective strengths of the team, attention must be paid to the fact that not all capabilities are improved in quantity by aggregation of individual robots' abilities. For instance, a coalition of any number of slow robots does not compensate for a faster robot. 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 robots in order to achieve certain task requirements.

Finally, we extend the *diversity measures* introduced in [16] to the continuous space. The diversity measures provide insights about the trait-based heterogeneity of the team. Specifically, the diversity measures help define a a minimum subset of the species that can collectively compensate for the rest of the team.

Georgia Institute of Technology, Atlanta, GA. Email: {harish.ravichandar, kshaw, chernova} @gatech.edu

In summary, the key contributions of our work include a unified framework for optimal task assignment of large heterogeneous teams that:

1) incorporates a stochastic trait model that captures both between-species and within-species variations,
2) optimally assigns tasks to robots with respect to two goals: exact matching and minimum matching, and
3) computes measures of diversity in teams with continuous trait models.

II. Related Work

A plethora of approaches have been proposed to address the multi-robot systems task assignment (MRTA) problem [5], [11], [10]. Broadly, the problems are categorized based on three binary characteristics: (1) Task type (single-robot [SR] vs multi-robot [MR]), (2) robot type (single-task [ST] vs multi-task [MT], and (3) assignment type (instantaneous [IA] vs time-extended [TA]) [5].

Coalition formations methods (e.g. [17], [20]) address transportation tasks involving multiple single-task robots. These approaches, however, require the listing of all potential coalitions and thus are not suitable for problems involving large number of robots. To ensure scalability, decentralized approaches for task assignment are introduced in [9], [3], [8], [13]. However, a common shortcoming of these decentralized approaches is that they assume that the desired behavior is specified as a function of the distribution of robots across the tasks.

Auction or market-based methods also provide solutions to the MRTA problem involving single-task robots [6], [12], [19]. We refer readers to [4] for a survey of market-based approaches applied to multi-robot coordination. A common of trait of auction or market-based methods is that they require extensive communication for biding and scale poorly with the number of robots in the team.

Our work falls under the category of *Single-Task Robots Multi-Robot Tasks Instantaneous Assignment (ST-MR-IA)* problem, also known as the *coalition formation* problem [5]. The ST-MR-IA is an instance of the set-partitioning problem in combinatorial optimization and is known to be strongly NP-Hard [5]. Albeit not developed for MRTA, a few efficient approximate solutions have been proposed for the set partitioning problem [2], [7]. Based on prior work in set partitioning problems, centralized and distributed algorithms to solve a ST-MR-IA problem have been proposed in [17], [18]. The method in [18] has also been adapted to be more efficient by reducing the required communication and discouraging imbalanced coalitions [20]. A method for coalition formation is introduced in [14] by building a solution by dynamically connecting a network of behaviors from individual robots.

A limitation of most of the existing approaches is that the desired behavior is assumed to be specified in terms of optimal robot distribution. A notable exception to this generalization is the framework introduced in [16], which is capable of receiving the task requirements provided in the form a desired trait distribution cross tasks. While STRATA shares several similarities with [16], there are a number of notable relative advantages. First, our species-trait model is continuous, while [16] uses a binary model. Second, we differentiate between cumulative and non-cumulative traits. Third, the framework in [16] utilizes a deterministic model of traits. In contrast, we consider the inherent randomness in the robots' traits, thereby capturing the variations at both species and robot levels. Finally, while the diversity measures introduced in [16] are limited to binary trait models, our measures are compatible with continuous-space models.

III. Modeling framework

In this section, we introduce the elements of STRATA that enables task assignments in large heterogeneous teams.

A. Trait Model

Base model: Consider a heterogeneous team of robots. We assume that each robot is a member of a particular species. The number of species $S \in \mathbb{N}$ is finite, and the number of robots in the ith species is denoted by N_i. The traits of each species are denoted by

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

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* $Q = [{q^{(1)}}^T, \cdots, {q^{(S)}}^T]^T$, 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 $\mu_Q = [\mu_{q^{(1)}}^T, \cdots, \mu_{q^{(S)}}^T]^T$ and the corresponding matrix summarizing the variance of each of its elements are given by

$$\boldsymbol{\mu}_Q = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1U} \\ \vdots & \ddots & \vdots \\ \mu_{S1} & \cdots & \mu_{SU} \end{bmatrix} \quad \boldsymbol{\Sigma}_Q = \begin{bmatrix} \sigma_{11}^2 & \cdots & \sigma_{1U}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{S1}^2 & \cdots & \sigma_{SU}^2 \end{bmatrix} \tag{2}$$

Using a stochastic model allows us to find clusters in the trait space and thus aids in automatically identifying the different species and their expected traits along with their variance.

Cumulative traits: STRATA explicitly differentiates between cumulative and non cumulative traits. While examples of cumulative traits include ammunition, equipment, and coverage area, those of non-cumulative traits include speed, special skills, and training. We model the cumulative traits as continuous variables (i.e., $q_i^{(s)} \in \mathbb{R}_+, \forall i \in \mathcal{C}$), and the non-commutative traits as binary variables (i.e., $q_j^{(s)} \in \{0, 1\}, \forall j \notin \mathcal{C}$), where $\mathcal{C} \subseteq \{1, 2, \cdots, U\}$ 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

$$q_j^{(s)} = \begin{cases} 1, & \text{if } \mu_{sj} \geq q_i^{\min} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $q_i^{\min}$ is a user-defined minimum acceptable value for the ith trait. The binary representation of non-cumulative traits captures information about whether the robots of each species posses the minimum required capabilities. Further, when aggregated (Section III-D), the binary representation provides the number of robots meeting the minimum requirements.

B. 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 a robot to switch between the corresponding two adjacent tasks. For each species s, the rate with which a robot currently performing task i switches to task j is given by $k_{ij}^{(s)}$, such that $0 < k_{ij}^{(s)} < k_{ij,max}^{(s)}$.

C. Robot Model

The distribution of robots from species s across the M tasks at time t is defined by $\mathrm{x}^{(s)}(t) = [x_1^{(s)}(t), x_2^{(s)}(t), \cdots x_M^{(s)}(t)]^T \in \mathbb{N}^M$. Thus, the distribution of the whole team across the tasks at time t can be described using a *abstract state information matrix* $\boldsymbol{X}(t) \in \mathbb{N}^{M \times S}$.

The dynamics of the number of robots of Species s at Task i is given by

$$\dot{x}_i^{(s)}(t) = \sum_{\forall j | (i,j) \in \mathcal{E}} k_{ji}^{(s)} x_j^{(s)}(t) - k_{ij}^{(s)} x_i^{(s)}(t) \tag{4}$$

and thus the dynamics of $\mathrm{x}^{(s)}(t)$ can be computed as

$$\dot{\mathrm{x}}^{(s)}(t) = K^{(s)} \mathrm{x}^{(s)}(t), \quad \forall s = 1, 2, \cdots, S \tag{5}$$

where $K^{(s)} \in \mathbb{R}_+^{M \times M}$ is the rate matrix of species s. Thus, the solution to the dynamics of the abstract state information is given by

$$X(\tau) = \sum_{s=1}^{S} e^{K^{(s)}\tau} \ z^{(s)}(0) \tag{6}$$

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 sth element equal to one.

D. Trait Aggregation and Distribution

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

$$Y(t) = X(t)Q \tag{7}$$

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 robots (who meet the minimum requirements for the corresponding trait) assigned to each task.

Note that 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

$$\mathbb{E}\{Y(t)\} = \boldsymbol{\mu}_Y(t) = X(t)\mu_Q \tag{8}$$

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

$$\mathrm{Var}\{Y(t)\} = \boldsymbol{\Sigma}_Y(t) = (X(t) \odot X(t)) \ \boldsymbol{\Sigma}_Q \tag{9}$$

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

$$\mathrm{Cov}\{Y_{ij}, Y_{kl}\} = \begin{cases} \sum_{s=1}^{S} (x_i^{(s)} \ x_k^{(s)} \ \sigma_{sj}^2), & \text{if } j = l \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

IV. Problem Formulation

In this section, we detail the problem of finding the transition rates $K^{(s)}$ for each species such that the trait distribution over tasks $Y(t)$ reaches the desired trait distribution Y^* as quickly as possible. We express the problem as the following optimization problem

$$\tau^*, K^{(s)^*} = \arg \min_{\tau, K^{(s)}} \tau \tag{11}$$

$$s.t. \quad X(\tau^*)Q \in \mathcal{G}(Y^*) \tag{12}$$

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 [16], we consider two goal functions:

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

swhere $\preceq$ 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, $\mathcal{G}_2$ allows for over-provisioning, while $\mathcal{G}_1$ does not.

V. Diversity Measures

Large heterogeneous teams with multiple species might result in capabilities that are complementary and or redundant. Inspired by [16], 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 [16] 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 IV. To this end, we utilize the following definitions.

Definition V.1. *Minspecies*: In a team described by an average species-trait matrix μ_Q, a *minspecies* 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 V.2. *Minspecies cardinality*: The cardinality of the minspecies set is defined as the Minspecies cardinality and is represented by the function $\mathcal{D}_\mathcal{G} : \mathbb{R}_+^{S \times U} \rightarrow \mathbb{N}_+$ that takes the average species-trait matrix μ_Q as the input and returns the minimum number of rows to achieve the goal $\mathcal{G}(Y^*)$.

A. Eigenspecies

First, we define a diversity measure related to the exact matching goal, $\mathcal{G}_1$ by identifying the minimal set of species $\mathcal{M}_1$ that can *exactly* match any desired trait distribution without recruiting robots from species not in $\mathcal{M}_1$.

Proposition V.1. *The cardinality of eigenspecies (the minspecies corresponding to goal function $\mathcal{G}_1$) is given by*

$$\mathcal{D}_{\mathcal{G}_1} = \min |\mathcal{M}_1| \tag{13}$$

$$\text{s.t.} \sum_{s \in \mathcal{M}_1} \alpha_{s\tilde{s}} \mu_{q^{(s)}} = \mu_{q^{(\tilde{s})}}, \ \forall \tilde{s} \notin \mathcal{M}_1, \forall \alpha_{s\tilde{s}} \in \mathbb{N} \tag{14}$$

$$\tag{15}$$

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 a robot 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 $\hat{\mu}_Q$. □

B. Coverspecies

Next, we define a diversity measure related to the minimum matching goal, $\mathcal{G}_2$ by identifying the minimal set of species $\mathcal{M}_2$ that can *exactly* match any desired trait distribution without recruiting robots from species not in $\mathcal{M}_2$.

Proposition V.2. *The cardinality of coverspecies (the min-species corresponding to goal function $\mathcal{G}_2$) is given by*

$$\mathcal{D}_{\mathcal{G}_2} = \min|\mathcal{M}_2| \tag{16}$$

$$\text{s.t.} \sum_{s\in\mathcal{M}_2} \alpha_{s\tilde{s}}\mu_{q^{(s)}} \succeq \mu_{q^{(\tilde{s})}}, \ \forall\tilde{s} \notin \mathcal{M}_2, \forall\alpha_{s\tilde{s}} \in \mathbb{N} \tag{17}$$

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 a robot 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 $\hat{\mu}_Q$. □

VI. Solution Approach

This section details the proposed solution to the optimization problem defined in (11)-(12).

A. Goal Constraints

We begin by considering the time evolution of average trait distribution over the tasks, obtained by combining (8) and (6):

$$\boldsymbol{\mu}_Y(\tau, K^{(1,..,S)}, \mathrm{x}^{(s)}(0)) = \sum_{s=1}^{S} e^{K^{(s)}\tau}\ \mathrm{x}^{(s)}(0)\ \mu_{q^{(s)}} \tag{18}$$

In order to satisfy the goal function constraint in (12), we impose constraints on two error functions [16]. The first error function computes the trait distribution error and is defined separately for each goal function as follows:

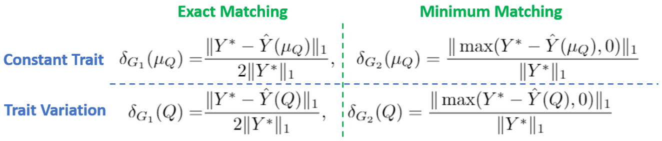
$$E_1^{\mathcal{G}_1}(\tau, K^{(1,..,S)}, X(0)) = \|Y^* - \boldsymbol{\mu}_Y(\tau)\|_F^2 \tag{19}$$

$$E_1^{\mathcal{G}_2}(\tau, K^{(1,..,S)}, X(0)) = \|\max[(Y^* - \boldsymbol{\mu}_Y(\tau)),\ 0]\|_F^2 \tag{20}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. The second error function measures the deviation from the steady state robot trait distribution and is defined as follows for both goal functions:

$$E_2(\tau, K^{(1,2,..,S)}, X(0)) = \sum_{s=1}^{S} \|e^{K^{(s)}\tau}\ \mathrm{x}^{(s)}(0) - e^{K^{(s)}(\tau+\nu)}\ \mathrm{x}^{(s)}(0)\|_2^2 \tag{21}$$

The 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 function E_2 penalizes the system if its trait distribution does not reach steady state at time τ.



	Exact Matching	Minimum Matching
Constant Trait	$\delta_{G_1}(\mu_Q) = \frac{\|Y^* - \hat{Y}(\mu_Q)\|_1}{2\|Y^*\|_1}$,	$\delta_{G_2}(\mu_Q) = \frac{\|\max(Y^* - \hat{Y}(\mu_Q), 0)\|_1}{\|Y^*\|_1}$
Trait Variation	$\delta_{G_1}(Q) = \frac{\|Y^* - \hat{Y}(Q)\|_1}{2\|Y^*\|_1}$,	$\delta_{G_2}(Q) = \frac{\|\max(Y^* - \hat{Y}(Q), 0)\|_1}{\|Y^*\|_1}$

Fig. 2. The four error measures used to quantify trait mismatch.

B. Optimization Problem

Based on the definitions in Sections VI-A, we reformulate the optimization problem in (11)-(12) for goal $\mathcal{G}_1$ as follows

$$\tau^*, K^{(s)^*} = \arg\min_{\tau, K^{(s)}} \tau \tag{22}$$

$$s.t.\ E_1^{\mathcal{G}_1}(\tau, K^{(1,..,S)}, X(0)) \leq \epsilon_1 \tag{23}$$

$$E_2(\tau, K^{(1,..,S)}, X(0)) \leq \epsilon_2 \tag{24}$$

$$\|\boldsymbol{\Sigma}_Y(\tau, K^{(1,..,S)}, X(0))\|_F^2 \leq \epsilon_3 \tag{25}$$

$$k_{ij}^{(s)} \leq k_{ij,\max}^{(s)},\ \forall i,j = \{1,..,M\}, \tag{26}$$

$$\tau > 0 \tag{27}$$

where ϵ_1, ϵ_2, and ϵ_3 are user-defined positive scalars. Further, for each task, the constraint in (25) encourages the system to recruit robots who possess traits (required for the task) with relatively low variance. Note that the optimization problem for goal $\mathcal{G}_2$ is identical except that we replace the constraint in (23) with $E_1^{\mathcal{G}_2}(\tau, K^{(1,..,S)}, X(0)) \leq \epsilon_1$.

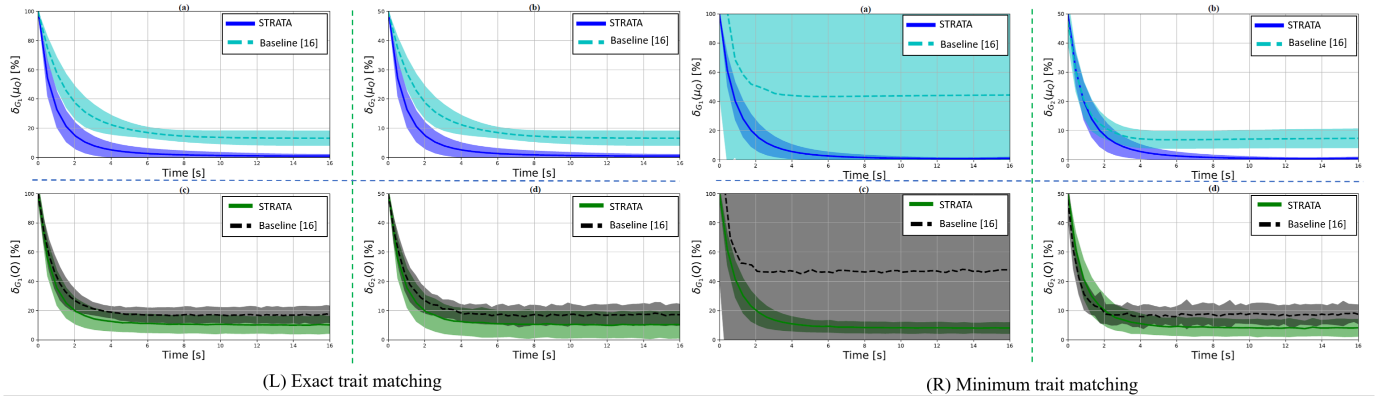
VII. Experimental Evaluation

In the following experiments, we compare STRATA's performance with that of a baseline. We use the binary-trait-based method introduced in [16] as our baseline method [1]. Since the baseline requires the desired trait distribution to be specified in the binary trait space, we make the following modifications to the baseline. We define a 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, $\mu_Y = [\mu_Y^1 \odot 1_M, \cdots, \mu_Y^U \odot 1_M]$, μ_Y^i is the mean value of the ith trait computed across all species, and 1_M is a M-dimensional vector of ones.

We study the performances of STRATA 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 robots). 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 robot distributions, $X(0)$ and X^*, are uniformly randomly generated. 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

[1] Code available at https://github.com/amandaprorok/diversity



(L) Exact trait matching

(R) Minimum trait matching

Fig. 3. Comparison of the performances of STRATA and the baseline [16] (binary) framework when optimizing for *Left*: exact matching ($\mathcal{G}_1$) and *Right*: minimum matching ($\mathcal{G}_2$). The performance of each framework is quantified in terms of four measures of percentage trait mismatch.

sampled from a discrete uniform distribution: $b_i \sim \mathcal{U}\{0, 1\}$. Each element of Σ_Q is sampled from a uniform distribution: $\mathcal{U}(0, 2)$. Based on the obtained X^* and μ_Q, a desired trait distribution Y^* is computed for each run.

The task assignment performance of each method is evaluated in terms of four measures of trait mismatch, as defined in Fig. 2. In each scenario, the performance is measured in terms of both exact and minimum trait matching, irrespective of the optimization goal. To ensure a fair comparison, we limit both STRATA and the baseline framework 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.

Exact Trait Matching: First, we compute the optimal transition rates according to both STRATA and the binary trait framework [16] with respect to the function Goal $\mathcal{G}_1$. STRATA is found to converge during 79 of the 100 simulation runs and the binary trait framework during 10 runs. In Fig 3L, we present the performances of both frameworks by plotting the errors (defined in Fig. 2) 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. 3L(a) and 3L(b), STRATA consistently performs better than the baseline in terms of deterministic performance, as measured by both $\delta_{G_1}(\mu_Q)$ and $\delta_{G_2}(\mu_Q)$. Further, as shown in Fig. 3L(c) and 3L(d), when the robots' traits are randomly sampled, STRATA performs better than the baseline on average. The stochastic nature of the species-trait matrix forces the errors to be larger than 0.

Minimum Trait Matching: Next, we compute the optimal transition rates according to both STRATA and the binary trait model [16] with respect to the function Goal $\mathcal{G}_2$. STRATA is found to converge during 85 of the 100 simulation runs and the binary trait framework during 16 runs. In Fig 3R, we present the performances of both frameworks by plotting the errors (defined in Fig. 2) as functions of time.

STRATA consistently performs better than the baseline when optimizing to satisfy minimum trait distribution, as measured by $\delta_{G_2}(\mu_Q)$. On average, STRATA 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. 3R(b) and 3R(d). In Fig. 3R(a) and 3R(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 STRATA, results in a high level of over-provisioning.

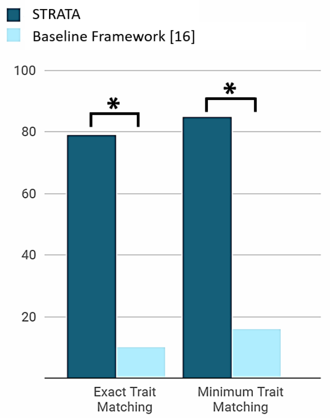


Fig. 4. Number of converged simulation runs.

Summary: The advantages of considering the variations both at the species and individual levels are reflected in the results above. Further, note that desired continuous trait distributions might not be achievable when reasoning over binary traits due to incompatibility with the continuous trait space. In fact, as seen in Fig. 4, STRATA successfully converged in significantly ($p < 0.001$) more runs than the binary trait framework for both exact trait matching ($\mathcal{G}_1$) and minimum trait matching ($\mathcal{G}_2$).

VIII. Conclusion

We presented STRATA, a unified framework capable of optimal task assignments in large teams of heterogeneous robots. The members of the team are modeled as belonging to different species, each defined by a set of its capabilities. STRATA models capabilities in the continuous space and explicitly takes into account both species-level and robot-level variations. Further, to quantify the diversity of a given team, we extended two diversity measures to the continuous trait space. The experimental results demonstrate that STRATA (1) successfully distributes a large heterogeneous team to meet complex task requirements, and (2) consistently performs better than the baseline that only considers binary traits.

Acknowledgement

This work was supported by the Army Research Lab under Grant W911NF-17-2-0181 (DCIST CRA).

References

[1] S. V. Albrecht and P. Stone, “Autonomous agents modelling other agents: A comprehensive survey and open problems,” *Artificial Intelligence*, vol. 258, pp. 66–95, 2018.

[2] A. Atamtürk, G. L. Nemhauser, and M. W. Savelsbergh, “A combined lagrangian, linear programming, and implication heuristic for large-scale set partitioning problems,” *Journal of heuristics*, vol. 1, no. 2, pp. 247–259, 1996.

[3] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar, “Optimized stochastic policies for task allocation in swarms of robots,” *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 927–937, 2009.

[4] 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.

[5] 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.

[6] J. Guerrero and G. Oliver, “Multi-robot task allocation strategies using auction-like mechanisms,” *Artificial Research and Development in Frontiers in Artificial Intelligence and Applications*, vol. 100, pp. 111–122, 2003.

[7] K. L. Hoffman and M. Padberg, “Solving airline crew scheduling problems by branch-and-cut,” *Management science*, vol. 39, no. 6, pp. 657–682, 1993.

[8] M. A. Hsieh, Á. Halász, S. Berman, and V. Kumar, “Biologically inspired redistribution of a swarm of robots among multiple sites,” *Swarm Intelligence*, vol. 2, no. 2-4, pp. 121–141, 2008.

[9] I. Jang, H.-S. 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.

[10] A. Khamis, A. Hussein, and A. Elmogy, “Multi-robot task allocation: A review of the state-of-the-art,” in *Cooperative Robots and Sensor Networks 2015*. Springer, 2015, pp. 31–51.

[11] G. A. Korsah, A. Stentz, and M. B. Dias, “A comprehensive taxonomy for multi-robot task allocation,” *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

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

[13] L. Matthey, S. Berman, and V. Kumar, “Stochastic strategies for a swarm robotic assembly system,” in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, 2009, pp. 1953–1958.

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

[15] O. L. Petchey and K. J. Gaston, “Functional diversity (fd), species richness and community composition,” *Ecology letters*, vol. 5, no. 3, pp. 402–411, 2002.

[16] A. Prorok, M. A. Hsieh, and V. Kumar, “The impact of diversity on optimal control policies for heterogeneous robot swarms.” *IEEE Trans. Robotics*, vol. 33, no. 2, pp. 346–358, 2017.

[17] O. Shehory and S. Kraus, “A kernel-oriented model for autonomous-agent coalition-formation in general environments,” in *Australian Workshop on Distributed Artificial Intelligence*. Springer, 1995, pp. 31–45.

[18] ——, “Methods for task allocation via agent coalition formation,” *Artificial intelligence*, vol. 101, no. 1, pp. 165–200, 1998.

[19] D. Vail and M. Veloso, “Multi-robot dynamic role assignment and coordination through shared potential fields,” *Multi-robot systems*, pp. 87–98, 2003.

[20] L. Vig and J. A. Adams, “Multi-robot coalition formation,” *IEEE transactions on robotics*, vol. 22, no. 4, pp. 637–649, 2006.