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

JAAMAS Track

Harish Ravichandar
Georgia Institute of Technology
harish.ravichandar@gatech.edu

Kenneth Shaw
Carnegie Mellon University
kshaw2@andrew.cmu.edu

Sonia Chernova
Georgia Institute of Technology
Chernova@gatech.edu

ABSTRACT

Large teams of heterogeneous 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 requires leveraging the relative strengths of the different kinds of agents in the team. We present *Stochastic TRAit-based Task Assignment (STRATA)*, a unified framework that models large teams of heterogeneous agents and performs effective task assignments. Specifically, given information on which *traits* (capabilities) are required for various tasks, STRATA computes the assignments of agents to tasks such that the 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. STRATA is capable of reasoning about both species-level and agent-level variability in traits. We illustrate the necessity and effectiveness of STRATA using detailed numerical simulations and in a capture-the-flag game environment.

ACM Reference Format:
Harish Ravichandar, Kenneth Shaw, and Sonia Chernova. 2021. STRATA: Unified Framework for Task Assignments in Large Teams of Heterogeneous Agents: JAAMAS Track. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

The study of multi-agent systems has produced significant insights into the process of engineering collaborative behavior in groups of agents [2, 6]. These insights have resulted in large teams of agents capable of accomplishing complex tasks that are intractable for a single agent. Teams of heterogeneous agents are particularly well suited for performing complex tasks that require a variety of skills, since they can leverage the relative advantages of the different agents and their capabilities. In this work, we are motivated by robotics applications, and the multi-robot task assignment (MRTA) problem in particular [3–5] which formally defines the challenges involved in optimally assigning agents to tasks.

We present *Stochastic TRAit-based Task Assignment (STRATA)*, a unified modeling and task assignment framework, to solve an instance of the MRTA problem with an emphasis on large heterogeneous teams. We assume that the optimal agent-to-task associations are unknown and that the task requirements are specified in terms of the various *traits* (capabilities) required for each task. Thus, in order to effectively perform the tasks, the agents must reason about their combined capabilities and the limited resources of the team. To enable this reasoning, we take inspiration from prior work in robot swarms [8] and biodiversity [7], and propose a group modeling approach [1] to model the capabilities of the team. Specifically, we assume that each agent in the team belongs to a particular *species*. Further, each species is defined based on the traits possessed by its members. Assuming that the agents are initially sub-optimally assigned to tasks, STRATA computes assignments such that the agents can reorganize themselves to collectively aggregate the traits necessary to meet the task requirements as quickly as possible.

Our representation is inspired by [8], which considered binary instantiations of traits. However, binary models fail to capture the nuances in the scales and natural variations of the agents' traits.In STRATA, we have extended the representation to model traits in the *continuous* space. Additionally, STRATA also captures agent-level differences within each species by using a *stochastic* trait model. Please see [9] for a full version of this short paper.

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

2 MODELING FRAMEWORK

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

Trait Model. Consider a heterogeneous team made up of $S \in \mathbb{N}$ *species* (i.e., agent types), each with N_s agents. We define each species by its capabilities. To capture the natural variability in *traits* (i.e., capabilities) of each species, we maintain a stochastic summary of each species' traits. Specifically, each element of the *species-trait matrix* $Q \in \mathbb{R}_+^{S\times U}$ is assumed to be an independent Gaussian random variable, $q_u^{(s)} \sim \mathcal{N}(\mu_{su}, \sigma_{su}^2)$. Thus, a vector random variable with all the traits of all the species can be written as $q = [q^{(1)}, q^{(2)}, \cdots, q^{(S)}] \sim \mathcal{N}(\mu_q, \Sigma_q)$, with its mean given by $\mu_q = [\mu_{q^{(1)}}, \cdots, \mu_{q^{(S)}}] \in \mathbb{R}_+^{SU}$, where $\mu_{q^{(s)}} = [\mu_{s1}, \cdots, \mu_{sU}] \in \mathbb{R}^U$ contains the expected trait values of the s^{th} species, and its covariance given by the following block-diagonal matrix $\Sigma_q = \text{diag}([\Sigma_{q^{(1)}}, \cdots, \Sigma_{q^{(S)}}]) \in \mathbb{R}_+^{SU\times SU}$, where $\Sigma_{q^{(s)}} = \text{diag}([\sigma_{s1}^2, \cdots, \sigma_{sU}^2]) \in \mathbb{R}_+^{U\times U}$ is the diagonal covariance matrix associated with the s^{th} species. The expected trait values can be rewritten in the form of the *expected species-trait matrix* $\mu_Q = [\mu_{q^{(1)}}^T, \cdots, \mu_{q^{(S)}}^T]^T \in \mathbb{R}_+^{S\times U}$. Similarly, the non-zero diagonal elements of the covariance matrix can be rewritten in matrix form as Var_Q.

Task Graph. We model the topology of the M tasks using a strongly connected graph $\mathbb{G} = (\mathcal{E}, \mathcal{V})$. The vertices $\mathcal{V}$ represent the M tasks, and the edges $\mathcal{E}$ connect tasks such that the existence of an edge between two tasks represents the agents' ability to switch between them. For each species, we aim to optimize the transition rate $k_{ij}^{(s)}$ for every edge in $\mathcal{E}$, such that $0 < k_{ij}^{(s)} < k_{ij,max}^{(s)}$. The transition rates implicitly dictate how the agents distribution evolves in time.

Agent Distribution. The distribution of agents from species s across the M tasks at time t is defined by $\mathrm{x}^{(s)}(t) = [x_1^{(s)}(t), \cdots x_M^{(s)}(t)]^T \in \mathbb{N}^M$. Thus the distribution of the whole team across the tasks at time t can be described using a *abstract state information matrix* $\boldsymbol{X}(t) = [\mathrm{x}^{(1)}(t), \mathrm{x}^{(2)}(t), \cdots, \mathrm{x}^{(S)}(t)] \in \mathbb{N}^{M\times S}$. As in [8], the time evolution of the number of agents from Species s at Task i is explained by the dynamical system, $\dot{x}_i^{(s)}(t) = \sum_{\forall j|(i,j)\in\mathcal{E}} k_{ji}^{(s)} x_j^{(s)}(t) - k_{ij}^{(s)} x_i^{(s)}(t)$. Thus the dynamics of each species' abstract state information can be computed as $\dot{\mathrm{x}}^{(s)}(t) = K^{(s)}\mathrm{x}^{(s)}(t), \quad \forall s = 1, 2, \cdots, S$, where $K^{(s)} \in \mathbb{R}_+^{M\times M}$ is the rate matrix of species s. Thus, the time evolution of the abstract state information is given by $\boldsymbol{X}(\tau) = \sum_{s=1}^{S} e^{K^{(s)}\tau} z^{(s)}(0)$, where $z^{(s)}(0) = \boldsymbol{X}(0) \odot (\mathbf{1} \cdot e_s) \in \mathbb{N}^{M\times S}$, $\mathbf{1}$ is an M-dimensional vector of ones, and e_s is the S-dimensional unit vector with its s^{th} element equal to one.

Trait Aggregation and Distribution. We represent the trait distribution using the *trait distribution matrix* $\boldsymbol{Y}(t) \in \mathbb{R}_+^{M\times U}$, given by

$$\boldsymbol{Y}(t) = \boldsymbol{X}(t)Q \tag{1}$$

Thus, each column of $\boldsymbol{Y}(t)$ represents the aggregated amounts of the corresponding trait available at each task at time t. Put another way, $\boldsymbol{Y}(t)$ represents the aggregation of various traits assigned to each task at time t. The expected value of $\boldsymbol{Y}(t)$ can be computed as $\boldsymbol{\mu}_Y(t) = \boldsymbol{X}(t)\boldsymbol{\mu}_Q$, and the variance of each element of $\boldsymbol{Y}$ is $\mathrm{Var}_Y(t) = (\boldsymbol{X}(t) \odot \boldsymbol{X}(t))\,\mathrm{Var}_Q$, where $\odot$ denotes the Hadamard product.

3 OPTIMIZED TASK ALLOCATION

We solve the following optimization problem to optimize allocation

$$\tau^*, K^{(s)^*} = \arg\min_{\tau, K^{(s)}} \tau \tag{2}$$

$$s.t. \quad \boldsymbol{X}(\tau)\mu_Q \in \mathcal{G}(\boldsymbol{Y}^*) \tag{3}$$

$$\|\mathrm{Var}_Y(\tau)\|_F \le \epsilon_{\text{var}} \tag{4}$$

where ϵ_{var} is the threshold used to limit the variance in $\boldsymbol{Y}(\tau)$, $\mathcal{G}(\boldsymbol{Y}^*) : \mathbb{R}_+^{M\times U} \to \Omega$, named the goal function, is a function that defines the set of admissible expected trait distribution matrices Ω. Note that the constraint in (4) helps minimize the expected variance, and thereby, maximize the chances that the actual trait distribution $\boldsymbol{Y}(\tau^*)$ meets the goal function. We consider two goal functions: $\mathcal{G}_1(\boldsymbol{Y}^*) = \{\mu_Y | \boldsymbol{Y}^* = \mu_Y\}$ (*exact matching*), and $\mathcal{G}_2(\boldsymbol{Y}^*) = \{\mu_Y | \boldsymbol{Y}^* \preceq \mu_Y\}$ (*minimum matching*), where $\preceq$ denotes the element-wise less-than-or-equal-to operator. Note that $\mathcal{G}_1$ does not allow any deviation from the desired trait distribution, and $\mathcal{G}_2$ allows for over-provisioning.

In order to satisfy the goal function constraint, as defined in (3), we impose constraints on two error functions. The first error function computes the trait distribution error and is defined separately for each goal function as follows:

$$E_1^{\mathcal{G}_1}(\tau, K^{(1,..,S)}, \boldsymbol{X}(0)) = \|Y^* - \boldsymbol{\mu}_Y(\tau)\|_F^2 \tag{5}$$

$$E_1^{\mathcal{G}_2}(\tau, K^{(1,..,S)}, \boldsymbol{X}(0)) = \|\max[(Y^* - \boldsymbol{\mu}_Y(\tau)),\ 0]\|_F^2 \tag{6}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The second error function measures the deviation from the steady state trait distribution, given by $E_2(\tau, K^{(1,..,S)}, \boldsymbol{X}(0)) = \sum_{s=1}^{S} \|e^{K^{(s)}\tau}\mathrm{x}^{(s)}(0) - e^{K^{(s)}(\tau+\nu)}\mathrm{x}^{(s)}(0)\|_2^2$.

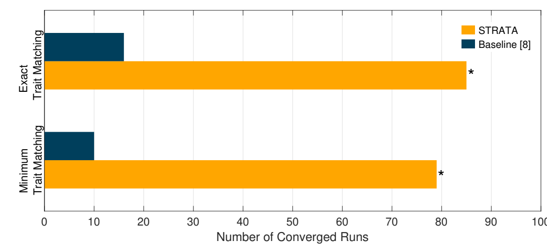

Figure 1: Number of converged runs out of 100 simulations.

4 EXPERIMENTAL EVALUATION

We evaluate STRATA using two sets of experiments (source code: https://github.com/harishravichandar/STRATA). In both experiments, we compare STRATA's performance against a bootstrapped version of a binary-trait-based method [8].

Simulation. In the first set of experiments, we simulate 100 independent task assignment problems with $M = 8$ nodes (tasks), $U = 5$ traits (3 cumulative and 2 non cumulative traits), and $S = 5$ species (each with 200 agents). We randomly sample the hyper parameters of the stochastic traits. We limit both STRATA and the baseline framework to a maximum of 20 meta iterations of the basin hopping algorithm during each run. We measure the performance of each algorithm using trait mismatch error (see [9] for details).

We find that STRATA consistently outperforms the baselines in terms of trait satisfaction. This is because binary trait models are incapable of reasoning about requirements in the continuous trait space, and ignore all variations both at the species and individual levels. Further, as seen in Fig. 1, STRATA successfully converged to a solution in significantly ($p < 0.001$) more runs than the binary trait framework for both exact trait matching ($\mathcal{G}_1$) and minimum trait matching ($\mathcal{G}_2$). This observation demonstrates that considering stochastic and continuous trait models over binary models is considerably more likely to satisfy complex trait requirements.

Capture the Flag. Next, we quantified the effect of STRATA on team performance in a capture the flag (CTF) game with 3 tasks, 4 traits and 12 agents (see [9]). We compare the performances of three teams (representing STRATA, the baseline, and random assignment).

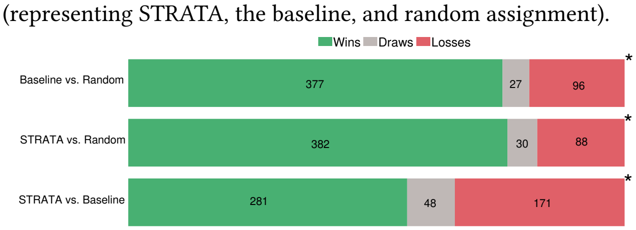

Figure 2: Performance on the capture-the-flag game.

As shown in Fig. 2, both the baseline framework and STRATA are more likely to win against random task assignment. However, STRATA is more likely to win against the baseline framework. Further, based on the z-test, we find that the proportions of wins are statistically significantly ($p < 0.001$) higher than those of losses in all three conditions. Thus, STRATA's ability to satisfy task requirements translates to improved team performance.

REFERENCES

[1] Stefano V Albrecht and Peter Stone. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence* 258 (2018), 66–95.

[2] Rafael H Bordini, Mehdi Dastani, Jürgen Dix, and A El Fallah Seghrouchni. 2007. *Programming Multi-Agent-Systems: 4th International Workshop, Revised and Invited Papers*. Vol. 4411. Springer.

[3] Brian P Gerkey and Maja J Matarić. 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research* 23, 9 (2004), 939–954.

[4] Alaa Khamis, Ahmed Hussein, and Ahmed Elmogy. 2015. Multi-robot task allocation: A review of the state-of-the-art. In *Cooperative Robots and Sensor Networks 2015*. Springer, 31–51.

[5] G Ayorkor Korsah, Anthony Stentz, and M Bernardine Dias. 2013. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research* 32, 12 (2013), 1495–1512.

[6] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. 2007. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* 95, 1 (2007), 215–233.

[7] Owen L Petchey and Kevin J Gaston. 2002. Functional diversity (FD), species richness and community composition. *Ecology letters* 5, 3 (2002), 402–411.

[8] Amanda Prorok, M Ani Hsieh, and Vijay Kumar. 2017. The Impact of Diversity on Optimal Control Policies for Heterogeneous Robot Swarms. *IEEE Trans. Robotics* 33, 2 (2017), 346–358.

[9] Harish Ravichandar, Kenneth Shaw, and Sonia Chernova. 2020. STRATA: unified framework for task assignments in large teams of heterogeneous agents. *Autonomous Agents and Multi-Agent Systems* 34 (2020), 1–25.