Formalizing the Impact of Diversity on Performance in a Heterogeneous Swarm of Robots

Amanda Prorok, M. Ani Hsieh, and Vijay Kumar

***Abstract*— We are interested in a principled study of the impact of diversity in heterogeneous large-scale distributed robotic systems. In order to evaluate the implications of heterogeneity on performance, we consider the concrete problem of distributing a large group of robots among a set of tasks that require specialized capabilities in order to be completed. We model the system of heterogeneous robots as a community of species, where each species (robot type) is defined by the traits (capabilities) that it owns. We develop a continuous model of the system at a macroscopic level, and formulate an optimization problem that produces an optimal set of transition rates for each species, so that the desired trait distribution is reached as quickly as possible. In order to evaluate the effects of heterogeneity, we propose a diversity metric that defines the notion of *eigenspecies*. We show that our metric correlates with performance: the higher the cardinality of the eigenspecies, the harder it becomes to optimize the system. Our approach is validated over multiple levels of abstraction, and real robot results confirm its validity on physical platforms.**

I. Introduction

Technological advances in embedded systems, such as component miniaturization and improved efficiency of sensors and actuators, are enabling the deployment of very large-scale robot systems, i.e., robot swarms. However, as we aspire to solve increasingly complex problems, it becomes ever more difficult to embed all necessary capabilities into one single robot type. Our premise is that, in a large-scale system of robots, one single type of robot cannot cater to all aspects of the task at hand, because at the individual level, it is governed by design rules that limit the scope of its capabilities. For example, a larger robot may be able to carry more powerful sensors, but may be less agile than its smaller counterpart. Or, we could consider the limited payload of aerial robots: if a given task requires rich sensory feedback, multiple heterogeneous aerial robots can complement each other by carrying distinct sensors. Instances of information gathering lend themselves naturally to this problem formulation, with applications to surveillance, environmental monitoring, and situational awareness [4, 6, 18].

As we allocate distinct capabilities among robot team members, we imply a certain degree of specialization among individuals [1, 2, 8]. During this process, heterogeneity becomes a design feature. The question is, then, how to best design such systems so that the resulting performance is optimized. However, since there has been very little work on quantitative measures of diversity in multi-robot systems, we still lack the analytical tools to understand the implications.

In this work, we contribute towards a general understanding of heterogeneity by proposing a measure that quantifies the diversity of a robot swarm that is tasked to complete a goal. Towards this end, we consider a concrete application with the objective of distributing a large-scale system of heterogeneous robots as efficiently as possible among tasks that require specialized competences.

A. Example of the Redistribution Problem

Figure 1 shows a system comprising five tasks that can be serviced by means of four distinct traits. The initial trait distribution is shown at t_0, and subsequent desired trait distributions are shown at t_1, t_2, and t_3. Figure 2 shows how the distribution of the traits evolves over time. This sequence is an example of how a heterogeneous robot system can be controlled to complete a global goal composed of several subtasks that require a specific set of capabilities in specific amounts. Also, the example shows how the solution to the redistribution problem can incorporate temporal constraints or precedence constraints: at a user-level, we can define arbitrary rules that govern transitions from one desired trait distribution to another as a function of the system's performance. An application of this problem considers a spatially distributed information gathering scenario: if enough data has been gathered at one site, we can reconfigure the system of robots so that it distributes to sites that have not yet been sufficiently accounted for. Which robots are deployed to which sites will depend on their capabilities and how these capabilities meet the needs that are anticipated at the sites. We note that the communication infrastructure required for such an approach is *asymmetric*, with a centralized supervisory agent broadcasting abstract state information to the robots, ensuring algorithmic invariance as we scale the system [13].

B. Background

Given a set of tasks, and knowledge about the task requirements, our problem considers which robots should be allocated to which tasks. This problem is an instance of the *MT-MR-TA: Multi-Task Robots, Multi-Robot Tasks* problem [5], and can be reformulated as a set-covering problem that stems from combinatorial optimization. This problem is strongly NP-hard [10]. A number of heuristic algorithms have been proposed. However, these algorithms are not suitable for large-scale systems. In particular, for systems that are required to adapt to changing task requirements

*Amanda Prorok and Vijay Kumar are with the University of Pennsylvania, Philadelphia, PA, USA: {prorok | kumar}@seas.upenn.edu. M. Ani Hsieh is with Drexel University, Philadelphia, PA, USA: mhsieh1@drexel.edu.

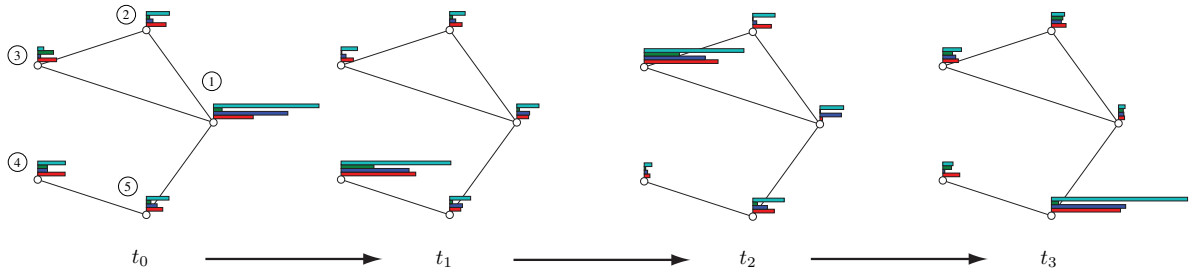


Fig. 1. Four configurations of a system with 5 tasks (nodes) and 4 traits. The trait abundance per task is represented by a bar plot. The edges of this strongly connected graph represent the possibility of switching between a pair of tasks. The system's initial distribution is shown at t_0, with subsequent desired target distributions at $t_{[1,2,3]}$.

online, we need to consider algorithms that are efficient and that run on low-cost, resource-constrained mobile platforms in real time. Hence, we consider a strategy that is scalable in the number of robots and their capabilities, and is robust to changes in the robot population [3, 7]. An important property of this strategy is its inherently decentralized architecture, with robots switching between tasks (behaviors) stochastically. The key difference between our work and previous work is that we formulate our desired state as a distribution of traits among tasks, instead of specifying the desired state as a direct measure of the robot distribution. In other words, our framework allows a user to specify how much of a given capability is needed for a given task, irrespective of which robot type satisfies that need. As a consequence, we do not employ optimization methods that utilize final robot distributions in their formulations (which is the case in previous works [3] and [12]). Instead, we explicitly optimize the distribution of traits, and implicitly solve the combinatorial problem of distributing the right number of robots of a given type to the right tasks.

II. Problem Formulation

Heterogeneity and diversity are core concepts of this work. To develop our formalism, we borrow terminology from biodiversity literature [15]. We define our robot system as a *community* of robots. Each robot belongs to a *species*, defining the unique set of *traits* that encodes the robots' capabilities. In this work, we will consider binary instantiations of traits (corresponding to the presence or absence of a given trait in a species). As an example, a trait might consider the presence/absence of a particular sensor, such as a camera or laser range finder. Another trait might consider the capability of fitting through a passageway with a fixed width. In this work, we assume that the tasks have been encoded through binary characteristics that represent the skill sets critical to task completion.

A. Notation

We consider a community of S robot species, with a total number of robots N, and $N^{(s)}$ robots per species such that $\sum_{s=1}^{S} N^{(s)} = N$. The community is defined by a set of U traits, and each robot species owns a subset of these traits. A species is defined by a binary vector

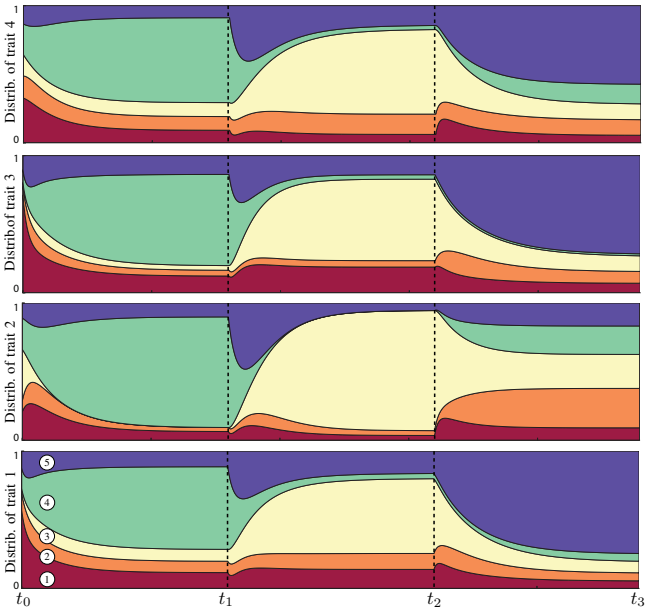


Fig. 2. Evolution over time of the trait distribution as specified by the distributions shown in Fig. 1. Each subplot represents one trait, indicating the distribution of that trait over the set of tasks (for each subplot, task 1 is shown at the bottom and task 5 at the top). The system's initial distribution is shown at t_0, with subsequent desired target distributions reached at $t_{[1,2,3]}$.

$\mathbf{q}^{(s)} = [q_1^{(s)}, q_2^{(s)}, ..., q_U^{(s)}]$. We can then define a $S \times U$ matrix $\mathbf{Q}$, with rows $\mathbf{q}^{(s)}$:

$$\mathbf{Q}_{su} = \begin{cases} 0 \text{ , if species } s \text{ does not have trait } u \\ 1 \text{ , if species } s \text{ has trait } u \end{cases}$$

We model the interconnection topology of the M tasks via a directed graph, $\mathfrak{G} = (\mathcal{E}, \mathcal{V})$ where the set of vertices, $\mathcal{V}$, represents tasks $\{1, \ldots, M\}$ and the set of edges, $\mathcal{E}$, represents the ordered pairs (i, j), such that $(i, j) \in \mathcal{V} \times \mathcal{V}$, and i and j are adjacent. Edges denote the possibility to switch between two adjacent tasks. We assume the graph $\mathfrak{G}$ is a strongly connected graph, i.e., a path exists between any pair of vertices (in contrast to a *fully* connected graph, where an edge exists between any pair of vertices), and we assume the robots have knowledge of this graph. We assign every edge in $\mathcal{E}$ a transition rate, $k_{ij}^{(s)} > 0$, where $k_{ij}^{(s)}$ defines the transition probability per unit time for one robot of species s at site i to switch to site j. Here $k_{ij}^{(s)}$ is a stochastic

transition rule. We impose a limitation on the maximum rate of each edge with $k_{ij}^{(s)} < k_{ij,\max}^{(s)}$. These values can be determined by applying system identification methods on the actual system. For example, in a system where nodes represent physically distributed sites, the transition rate represents the rate with which a specific path is chosen. This value can depend on observed factors, such as typical road congestion or the condition of the terrain. The distribution of the robots belonging to a species s at time t is described by a vector $\mathbf{x}^{(s)}(t) = [x_1^{(s)}(t), ..., x_M^{(s)}(t)]^\top$. Then, if $\mathbf{x}^{(s)}$ are the columns of $\mathbf{X}(t)$, and $\mathbf{q}^{(s)}$ are the rows of $\mathbf{Q}$, we have the $M \times U$ matrix $\mathbf{Y}$ that describes the distribution of traits on sites. For time t this relationship is given by

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

B. System

The initial state of the system is described by $\mathbf{X}(0)$, and hence, the initial distribution of traits at the sites is described by $\mathbf{Y}(0)$. The time evolution of the number of robots of species s at site i is given by a linear law

$$\frac{\mathrm{d}x_i^{(s)}}{\mathrm{d}t} = \sum_{\forall j|(i,j)\in\mathcal{E}} k_{ji}^{(s)} x_i^{(s)}(t) - \sum_{\forall j|(i,j)\in\mathcal{E}} k_{ij}^{(s)} x_i^{(s)}(t) \tag{2}$$

Then, for all species s, our base model is given by

$$\frac{\mathrm{d}\mathbf{x}^{(s)}}{\mathrm{d}t} = \mathbf{K}^{(s)}\mathbf{x}^{(s)} \quad \forall s \in 1, \ldots, S \tag{3}$$

where $\mathbf{K}^{(s)} \in \mathbb{R}^{M\times M}$ is a rate matrix with the properties

$$\mathbf{K}^{(s)\top}\mathbf{1} = \mathbf{0} \tag{4}$$

$$\mathbf{K}_{ij}^{(s)} \geq 0 \quad \forall (i,j) \in \mathcal{E} \tag{5}$$

These two properties result in the following definition:

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

Since the total number of robots and the number of robots per species is conserved, the system in Eq. 3 is subject to the constraints

$$\mathbf{X}^\top \cdot \mathbf{1} = [N^{(1)}, N^{(2)}, \ldots, N^{(S)}]^\top \tag{6}$$

$$\text{with } \mathbf{X} \succeq \mathbf{0}, \tag{7}$$

where $\succeq$ is an element-wise greater-than-or-equal-to operator.

C. Problem Statement

Our goal is to redeploy the robots of each species, distributed according to $\mathbf{X}(0)$ initially, so that a desired trait distribution $\mathbf{Y}^\bigstar$ is reached. In other words, the robots attempt to organize themselves among tasks such that the trait demand is met for each task. The problem then consists of finding an optimal rate matrix $\mathbf{K}^{(s)\bigstar}$ for each species s so that the target trait distribution is reached:

$$\mathbf{K}^{(s)\bigstar}, \tau^\bigstar = \underset{\mathbf{K}^{(s)},\tau}{\operatorname{argmin}}\, \tau \tag{8}$$

$$\text{such that} \quad \mathbf{X}(\tau^\bigstar) \cdot \mathbf{Q} = \mathbf{Y}^\bigstar \tag{9}$$

The solution leads to a robot distribution $\mathbf{X}(\tau^\bigstar)$ that satisfies Eq. 9, subject to Eq. 6 and Eq. 7. If the robots run the optimization algorithm on-board, they need knowledge of abstract state information (i.e, the initial distribution of the robot swarm among tasks, $\mathbf{X}(0)$). If the optimization is run off-board, the robots need knowledge of the transition rates of their species, $k_{ij}^{(s)}$. We note that this information is represented by a small number of values (at most M^2 values per species, or a much smaller number if the graph is sparse).

III. Diversity Metric

Since the desired configuration of our system is solely described through $\mathbf{Y}^\bigstar$, the corresponding final robot distribution $\mathbf{X}(\tau^\bigstar) = \mathbf{X}^\bigstar$ that reaches the target trait distribution $\mathbf{Y}^\bigstar$ is not known a priori. In particular, there may be several $\mathbf{X}^\bigstar$ that satisfy Eq. 9. Hence, we pose the question: *Can we infer properties of the species-trait matrix* $\mathbf{Q}$ *that quantify how easy it is to find a solution* $\mathbf{X}^\bigstar$ *that reaches* $\mathbf{Y}^\bigstar$*?* In the following, we show how $\mathbf{Q}$ embodies the *diversity* of the robot community, and how we can quantitatively evaluate the diversity to make conclusions about the system's performance.

A. Definitions

Given an unlimited number of robots per species, it may be possible to reach any given trait distribution $\mathbf{Y}^\bigstar$ with a subset of the original robot species. We call the species belonging to this subset *eigenspecies*, and we refer to the size of this subset as the *eigenspecies cardinality*. More formally, we introduce the following terminology:

Definition 1 (***Eigenspecies***):
In a robot community described by a species-trait matrix $\mathbf{Q}$, an *eigenspecies* set is a subset of $\mathbf{Q}$ with minimal cardinality, such that the system can still reach any target trait distribution $\mathbf{Y}^\bigstar$. We represent the eigenspecies by a matrix $\hat{\mathbf{Q}}$ that contains linearly independent rows of $\mathbf{Q}$.

Definition 2 (***Eigenspecies cardinality***):
The *eigenspecies cardinality* of a robot community is given by the cardinality of the eigenspecies set. It is a function $\mathcal{D} : \{0,1\}^{S\times U} \to \mathbb{N}^+$ that takes a species-trait matrix $\mathbf{Q}$ as input, and returns the number of linearly independent rows in $\mathbf{Q}$.

B. Implementation

In this section, we develop the eigenspecies cardinality. In particular, we demonstrate that the eigenspecies cardinality is a meaningful quantitative measure of the constraint in Eq. 9.

Proposition 1: The eigenspecies cardinality is

$$\mathcal{D}(\mathbf{Q}) = \operatorname{rank}(\mathbf{Q}) \tag{10}$$

Proof: The matrix $\mathbf{Q}^\top$ can be rank-factorized into the product of two matrices $\mathbf{A}$ and $\hat{\mathbf{Q}}$ such that $\mathbf{Q}^\top = \hat{\mathbf{Q}}^\top\mathbf{A}^\top$

with $\hat{\mathbf{Q}}$ containing a subset of the rows of $\mathbf{Q}$ [17]. Since $\mathbf{Y}^\star = \mathbf{X}^\star\mathbf{Q} = \mathbf{X}^\star\mathbf{A}\hat{\mathbf{Q}}$, there exists a robot distribution $\hat{\mathbf{X}} = \mathbf{X}^\star\mathbf{A}$ for which $\hat{\mathbf{X}}\hat{\mathbf{Q}} = \mathbf{Y}^\star$. Hence, as $\hat{\mathbf{Q}}$ has minimal size (due to the rank-factorization), $\hat{\mathbf{Q}}$ is an eigenspecies matrix. ■

Indeed, the rank of $\mathbf{Q}$ (which must be $\leq S$) quantifies the number of independent species in $\mathbf{Q}$ that span the solution space of the equation $\mathbf{X}^\star\mathbf{Q} = \mathbf{Y}^\star$ (with $\mathbf{X}^\star$ unknown):

- If $\mathrm{rank}(\mathbf{Q}) < S$, the system is underdetermined, and an infinite number of solutions $\mathbf{X}^\star$ will satisfy Eq. 9. In other words, at least one species in the system can be replaced by a combination of the other species. As the rank decreases, the *redundancy* of the community increases.
- If $\mathrm{rank}(\mathbf{Q}) = S$, there is only one solution $\mathbf{X}^\star$ that satisfies Eq. 9. In other words, no species in the system can be replaced by a combination of the other species, and all species are fully *complementary*.

As an example, consider matrix

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \mathbf{A} \cdot \hat{\mathbf{Q}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

The rank of $\mathbf{Q}$ is 2, hence, $\mathcal{D}(\mathbf{Q}) = 2$, which is the number of independent species.

IV. Methodology

In this section, we describe our methodology for obtaining an optimal transition matrix $\mathbf{K}^{(s)\star}$ for each species so that the desired trait distribution is reached. Two general approaches have been considered so far [3]: convex optimization and stochastic optimization. The convex optimization approach requires knowledge of the desired final robot distribution. However, our problem formulation specifies a desired trait distribution $\mathbf{Y}^\star$ without explicit definition of the final robot distribution $\mathbf{X}^\star$. Fully stochastic schemes such as Metropolis optimization have been shown to produce similar results, but they are not computationally efficient, and are ill-suited to real-time applications. In the following, we present a differentiable objective function that can be efficiently minimized through gradient descent techniques. Our method explicitly minimizes the convergence time of $\mathbf{K}^{(s)}$, unlike the convex optimization methods presented in [3], which approximate $\mathbf{K}^{(s)}$ with a symmetric equivalent (forcing bidirectionally equal transition rates between sites). Additionally, it is able to find optimal transition rates with knowledge of $\mathbf{Y}^\star$ and $\mathbf{X}(0)$ only (i.e., without knowledge of $\mathbf{X}^\star$).

A. Design of Optimal Transition Rates

We combine the solution of the linear ordinary differential equation, Eq. 3, with Eq. 1 to obtain the solution:

$$\mathbf{Y}(t) = \sum_{s=1}^{S} e^{\mathbf{K}^{(s)\star}t}\mathbf{x}_0^{(s)} \cdot \mathbf{q}^{(s)} \tag{11}$$

To find the values of $\mathbf{K}^{(s)\star}$, we consider the error

$$\mathbf{E} = \mathbf{Y}^\star - \sum_{s=1}^{S} e^{\mathbf{K}^{(s)\star}\tau}\mathbf{x}_0^{(s)} \cdot \mathbf{q}^{(s)} \tag{12}$$

where τ is the time at which the desired state is reached, and formulate our optimization problem as

$$\begin{aligned} &\text{minimize} \quad \mathcal{J}^{(1)} = \| \mathbf{E} \|_F^2 \\ &\text{such that} \quad k_{ij}^{(s)} < k_{ij,\max}^{(s)} \end{aligned} \tag{13}$$

which formulates that a minimum cost is found when the final trait distribution corresponds to the desired trait distribution, subject to maximum transition rates $k_{ij,\max}^{(s)}$. The notation $\mathbf{x}_0^{(s)}$ is shorthand for $\mathbf{x}^{(s)}(0)$. The operator $\| \cdot \|_F$ denotes the Frobenius norm of a matrix. There is no closed-form solution to the optimization problem in Eq. 13, but we can use the derivatives of $\mathcal{J}^{(1)}$ with respect to the parameters to perform gradient descent. To maximize the efficiency of our optimization function, we compute an analytical gradient. By applying the chain rule, the derivative of our objective function with respect to the transition matrix $\mathbf{K}^{(s)}$ is

$$\frac{\partial \mathcal{J}^{(1)}}{\partial \mathbf{K}^{(s)}} = \frac{\partial \mathcal{J}^{(1)}}{\partial e^{\mathbf{K}^{(s)}\tau}} \cdot \frac{\partial e^{\mathbf{K}^{(s)}\tau}}{\partial \mathbf{K}^{(s)}\tau} \cdot \frac{\partial \mathbf{K}^{(s)}\tau}{\partial \mathbf{K}^{(s)}} \tag{14}$$

We first compute the derivative of the cost with respect to the expression $e^{\mathbf{K}^{(s)}\tau}$

$$\frac{\partial \mathcal{J}^{(1)}}{\partial e^{\mathbf{K}^{(s)}\tau}} = -2\mathbf{E} \cdot \left[\mathbf{x}_0^{(s)} \cdot \mathbf{q}^{(s)}\right]^\top \tag{15}$$

The derivation of the 2nd element of Eq.14 requires the derivative of the matrix exponential. Computing the derivative of the matrix exponential is not trivial. We adapt the closed-form solution given in [9] to our problem, and write the gradient of our cost function as

$$\frac{\partial \mathcal{J}^{(1)}}{\partial \mathbf{K}^{(s)}} = \mathbf{V}^{-\mathbf{1}^\top}\left[\mathbf{V}^\top \frac{\partial \mathcal{J}^{(1)}}{\partial e^{\mathbf{K}^{(s)}\tau}} \mathbf{V}^{-\mathbf{1}^\top} \odot \mathbf{W}(\tau)\right]\mathbf{V}^\top\tau \tag{16}$$

where $\odot$ is the Hadamard product, $\mathbf{K}^{(s)} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$ is the eigendecomposition of $\mathbf{K}^{(s)}$. $\mathbf{V}$ is the $M \times M$ matrix whose jth column is a right eigenvector corresponding to eigenvalue d_i, and $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_M)$. The matrix $\mathbf{W}(t)$ is composed as follows [1]

$$\mathbf{W}(t) = \begin{cases} (e^{d_i t} - e^{d_j t})/(d_i t - d_j t) & i \neq j \\ e^{d_i t} & i = j \end{cases}$$

B. Optimization of Convergence Time

The cost function in Eq. 13 does not consider the convergence time τ as a variable. By adding a term that penalizes high convergence time values, we can compute transition rates that explicitly optimize convergence time. The modified objective function is

[1]Here, we assume that that $\mathbf{K}^{(s)}$ has M distinct eigenvalues. If this is not the case, an analogous decomposition of $\mathbf{K}^{(s)}$ to Jordan canonical form is possible, as elaborated in [9]. We note that for most models of interest, however, this is rarely the case.

$$\begin{array}{ll} \text{minimize} & \mathcal{J}^{(2)} = \mathcal{J}^{(1)} + \alpha\tau^2 \\ \text{such that} & k_{ij}^{(s)} < k_{ij,\max}^{(s)} \text{ and } \tau > 0, \end{array} \tag{17}$$

and $\alpha > 0$. By increasing α, we increase the importance of the convergence time (by penalizing high values of τ). The derivative with respect to the transition rates is

$$\frac{\partial \mathcal{J}^{(2)}}{\partial \mathbf{K}^{(s)}} = \frac{\partial \mathcal{J}^{(1)}}{\partial \mathbf{K}^{(s)}} \tag{18}$$

We also need the derivative with respect to τ. This derivative is computed analogously to the derivative with respect to $\mathbf{K}^{(s)}$ (confer Eq. 16). We have

$$\frac{\partial \mathcal{J}^{(2)}}{\partial \tau} = \frac{\partial \mathcal{J}^{(1)}}{\partial \tau} + 2\alpha\tau \tag{19}$$

with

$$\frac{\partial \mathcal{J}^{(1)}}{\partial \tau} = \sum_{s=1}^{S} \mathbf{1}^\top \mathbf{V}^{-\mathbf{1}^\top} \mathbf{A}_1 \mathbf{V}^\top \mathbf{K}^{(s)} \mathbf{1} \tag{20}$$

and

$$\mathbf{A}_1 = \mathbf{V}^\top \frac{\partial \mathcal{J}^{(1)}}{\partial e^{\mathbf{K}^{(s)}\tau}} \mathbf{V}^{-\mathbf{1}^\top} \odot \mathbf{W}(\tau) \tag{21}$$

The optimization of Eq. 17 will lead to transition rates that may lead to the desired trait distribution quickly, but there is no guarantee that this is the steady-state of $\mathbf{K}^{(s)}$. If we compute the transition rates at the outset of the experiment (without refining them online), we may wish to ensure that the state reached at the optimal time $\tau^\star$ remains near-constant. Hence, we modify our cost function in Eq. 17 as

$$\begin{array}{rcl} \min \mathcal{J}^{(3)} & = & \mathcal{J}^{(2)} \\ & + & \beta \sum_{s=1}^{S} \left\| e^{\mathbf{K}^{(s)}\tau} \mathbf{x}_0^{(s)} - e^{\mathbf{K}^{(s)}(\tau+\nu)} \mathbf{x}_0^{(s)} \right\|_2^2 \\ \text{such that} & & k_{ij}^{(s)} < k_{ij,\max}^{(s)} \text{ and } \tau > 0, \end{array} \tag{22}$$

and $\beta > 0$. The additional term in our cost function allows us to ensure that the robot distribution reached by employing $\mathbf{K}^{(s)\star}$ remains near-constant for arbitrarily long time intervals ν. This is possible because our model in Eq. 3 is stable [7], and the difference between the current robot distribution and the one at steady-state can only decrease monotonically over time. By increasing the value of β, the difference of the robot distributions at times τ and $\tau + \nu$ is decreased. In other words, the trait distribution corresponding to the steady-state of $\mathbf{K}^{(s)}$ gets arbitrarily close to the desired trait distribution $\mathbf{Y}^\star$ as β (or ν) increases.

Let us refer to this additional third term of $\mathcal{J}^{(3)}$ (and second term of Eq. 22) as $\mathcal{J}^{(3,3)}$. Then, the derivative of the new objective function with respect to the transition rates can be expressed as

$$\frac{\partial \mathcal{J}^{(3)}}{\partial \mathbf{K}^{(s)}} = \frac{\partial \mathcal{J}^{(2)}}{\partial \mathbf{K}^{(s)}} + \frac{\partial \mathcal{J}^{(3,3)}}{\partial \mathbf{K}^{(s)}} \tag{23}$$

Again, we apply the chain rule to obtain

$$\begin{array}{rcl} \dfrac{\partial \mathcal{J}^{(3,3)}}{\partial \mathbf{K}^{(s)}} & = & \dfrac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}\tau}} \dfrac{\partial e^{\mathbf{K}^{(s)}\tau}}{\partial \mathbf{K}^{(s)}\tau} \dfrac{\partial \mathbf{K}^{(s)}\tau}{\partial \mathbf{K}^{(s)}} \\ & - & \dfrac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}(\tau+\nu)}} \dfrac{\partial e^{\mathbf{K}^{(s)}(\tau+\nu)}}{\partial \mathbf{K}^{(s)}(\tau+\nu)} \dfrac{\partial \mathbf{K}^{(s)}(\tau+\nu)}{\partial \mathbf{K}^{(s)}} \end{array} \tag{24}$$

The outer derivative is

$$\begin{array}{rcl} \dfrac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}\tau}} & = & \dfrac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}(\tau+\nu)}} \\ & = & 2\beta \left[e^{\mathbf{K}^{(s)}\tau} \mathbf{x}_0^{(s)} - e^{\mathbf{K}^{(s)}(\tau+\nu)} \mathbf{x}_0^{(s)} \right] \cdot \mathbf{x}_0^{(s)\top} \end{array} \tag{25}$$

We apply the same development as in Eq. 16 to obtain the equation

$$\frac{\partial \mathcal{J}^{(3,3)}}{\partial \mathbf{K}^{(s)}} = \mathbf{V}^{-\mathbf{1}^\top} \left[\mathbf{A}_2 \tau - \mathbf{A}_3(\tau+\nu) \right] \mathbf{V}^\top \tag{26}$$

with

$$\mathbf{A}_2 = \mathbf{V}^\top \cdot \frac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}\tau}} \cdot \mathbf{V}^{-\mathbf{1}^\top} \odot \mathbf{W}(\tau) \tag{27}$$

and

$$\mathbf{A}_3 = \mathbf{V}^\top \cdot \frac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}(\tau+\nu)}} \cdot \mathbf{V}^{-\mathbf{1}^\top} \odot \mathbf{W}(\tau+\nu) \tag{28}$$

The derivative with respect to time τ is analogous:

$$\frac{\partial \mathcal{J}^{(3)}}{\partial \tau} = \frac{\partial \mathcal{J}^{(2)}}{\partial \tau} + \sum_{s=1}^{S} \mathbf{1}^\top \mathbf{V}^{-\mathbf{1}^\top} \left[\mathbf{A}_2 - \mathbf{A}_3 \right] \mathbf{V}^\top \mathbf{K}^{(s)} \mathbf{1} \tag{29}$$

For all above cost functions, $z = 1, 2, 3$, the derivative with respect to the off-diagonal elements ij of the matrix $\mathbf{K}^{(s)}$, with $(i, j) \in \mathcal{E}$, is

$$\frac{\partial \mathcal{J}^{(z)}}{\partial k_{ij}^{(s)}} = \left\{ \frac{\partial \mathcal{J}^{(z)}}{\partial \mathbf{K}^{(s)}} \right\}_{ij} - \left\{ \frac{\partial \mathcal{J}^{(z)}}{\partial \mathbf{K}^{(s)}} \right\}_{jj} \tag{30}$$

where $\{\cdot\}_{ij}$ denotes the element on row i and column j. The computational complexity of computing the gradient of our objective function is $O(S \cdot M^3 + S \cdot M^2 \cdot U)$, i.e., the computation grows linearly with the number of species and traits, and cubically with number of tasks[2].

Finally, we summarize our optimization problem as follows:

$$\mathbf{K}^{(s)\star}, \tau^\star = \underset{\mathbf{K}^{(s)}, \tau}{\operatorname{argmin}} \mathcal{J}^{(3)}, \tag{31}$$

under the constraints shown in Eq. 22. To solve the system, we implement a basin-hopping optimization algorithm [19], which attempts to find the global minimum of a smooth scalar function. Locally, our basin-hopping algorithm uses a quasi-Newton method (namely, the Broyden-Fletcher-Goldfarb-Shanno algorithm [14] with bound constraints), using the analytical gradients given by Eq. 29 and Eq. 30.

[2]The average time to compute the gradient for a system with $M = 8$, $U = 4$, and $S = 4$ is around 1.35 ms with $\nu = 0$, and 2.2 ms with $\nu > 0$ (using code implemented in Python using the NumPy and SciPy libraries, and tested on a 2 GHz Intel Core i7 using a single CPU).

V. Results

In the following, we present results that show that *(i)* our method successfully achieves the deployment of a heterogeneous system of robots so that a desired trait distribution is reached, and that *(ii)* the performance of the heterogenous system is a function of its diversity. We evaluate these claims over multiple levels of abstraction, and finally, also present real robot results validating the feasibility of our approach on physical platforms.

We consider the optimization problem posed in Eq. 31 that explicitly optimizes convergence time. Throughout this section, we use parameters $\alpha = 1$, $\beta = 5$, and $\nu = 2$, to produce optimal transition rates $\mathbf{K}^{(s)\star}$ for each species. These parameters were roughly tuned to work with the example described below, and remain the same for all experiments, including the real robot experiments. Naturally, the parameters could be tuned on a case-to-case basis to further improve performance, but this is not our focus here. Unless otherwise noted, we set $k^{(s)}_{ij,\max} = 2\,\mathrm{s}^{-1}$.

A. Example

To illustrate our method in more detail, let us consider the example portrayed earlier, in Fig. 1. The graph is generated randomly according to the Watts-Strogatz model [20] (with a neighboring node degree of $K = 3$, and a rewiring probability of $\gamma = 0.6$; the graph is guaranteed to be connected). The robot community consists of 3 species and 4 traits, and is defined as follows:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \text{ with } \mathbf{X}^\top \cdot \mathbf{1} = [24, 46, 60]^\top$$

In this example, $N = 130$ robots transition among $M = 5$ tasks. We sample a random initial robot distribution $\mathbf{X}(t_0)$ with a majority of traits in use at task 1. We specify a randomly generated desired trait distribution, which is visualized in Fig. 1 at t_1. The final robot distribution $\mathbf{X}(t_1)$ then serves as the initial distribution for a subsequent reconfiguration targeting the trait distribution visualized at t_2. As this process is repeated, it demonstrates how our method can be used to redistribute a swarm of robots through time.

B. Performance Metric

Previous work has shown the benefit of validating methods over multiple levels of abstraction (sub-microscopic, microscopic, and macroscopic) [16]. In this section, we propose an evaluation of our methods on two levels: macroscopic, and microscopic. Indeed, the most efficient way of simulating a large-scale system of robots is by considering a continuous macroscopic model, derived directly from the ordinary differential equation, Eq. 3, as is done in the example above. In order to validate our methods at a lower level of abstraction, we also implement a discrete microscopic model that emulates the behavior of individual robot controllers. This agent-level control is based on the transition rates $k^{(s)}_{ij}$ encoded by the transition matrix $\mathbf{K}^{(s)}$: A robot of species s at site i transitions to site j according to probability $p^{(s)}_{ij}$ that is an element of the matrix $\mathbf{P}^{(s)} = e^{\mathbf{K}^{(s)}\Delta T}$, where ΔT is the duration of one time-step. Running multiple iterations of the microscopic model enables us to capture the stochasticity resulting from our control system. In the remainder of this paper, we use $\Delta T = 0.04\,\mathrm{s}$, unless stated otherwise.

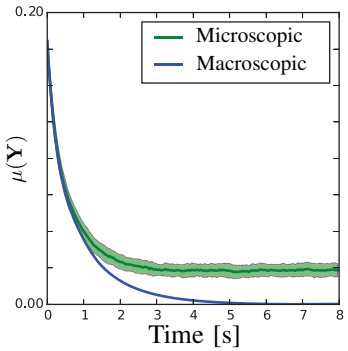


Fig. 3. Ratio of misplaced traits over time for a graph with $M = 10$ nodes. The plot shows the macroscopic model as well as the average over 60 iterations of the microscopic model. The shaded area shows the standard deviation.

The degree of convergence to $\mathbf{Y}^\star$ is expressed by the fraction of misplaced traits. We have:

$$\mu(\mathbf{Y}) = \frac{\|\mathbf{Y}^\star - \mathbf{Y}\|_1}{2\|\mathbf{Y}^\star\|_1} \tag{32}$$

Fig. 3 shows the ratio of misplaced traits $\mu(\mathbf{Y})$ over time for a graph with $M = 10$ nodes, a random initial robot distribution, and a random desired trait distribution. We run 60 iterations of the discrete microscopic model. We observe that the trait error decreases exponentially. Initially, the microscopic and macroscopic models show good agreement, up to about $t = 1$ second. Afterwards, the stochasticity of the microscopic model forces the error ratio (which counts absolute differences) to be larger than 0. Note that the latter result depends on the noise intensity, and hence, the dynamics of the system. Systems with slower dynamics achieve lower average errors at steady-state.

C. Impact of Diversity

Our aim is to observe the impact of diversity on system performance. We accomplish this by evaluating the time of convergence to the desired trait distribution as a function of our proposed diversity measure, the eigenspecies cardinality. We consider a system of $M = 10$ tasks and $S = 6$ species, and generate random species-trait matrices $\mathbf{Q}$ with eigenspecies cardinality values ranging from 1 to 6. The system is evaluated on 60 graphs, for each eigenspecies cardinality value, with a random initial robot distribution and a random desired trait distribution per graph. We measure the time $t_{\mu,\mathrm{thresh}}$ at which the system converges to a value $\mu_{\mathrm{thresh}} = 2.5\%$ of misplaced traits, and say that one system converges faster than another if it takes less time for $\mu(\mathbf{Y})$ to decrease to μ_{thresh}. Similar performance metrics have been proposed in [3, 7]. Fig. 4 shows the results. Our optimization method is shown in green. We see that as the eigenspecies cardinality of the system increases, the time to convergence

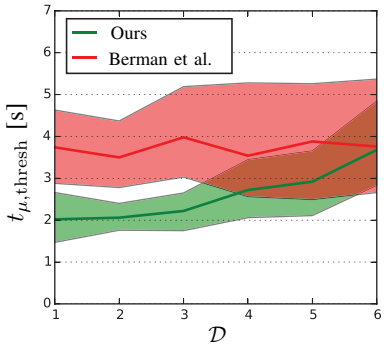
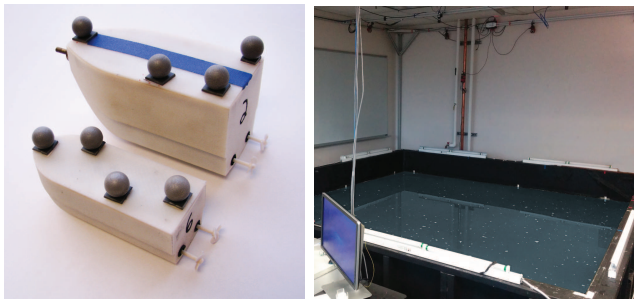


Fig. 4. The plot shows the median convergence time evaluated on the microscopic model, with $t_{\mu_{\text{thresh}}}$ for $\mu_{thresh} = 2.5\%$, as a function of the eigenspecies cardinality, for 60 random graphs per cardinality value. The system has $M = 10$ tasks and $S = 6$ species. The shaded area shows the 25th and 75th percentiles.

also increases. Indeed, the size of the solution space of Eq. 9 decreases as the eigenspecies cardinality increases. In other words, the more the species are complementary, the harder the system is to optimize. Also, we compare our method to a benchmark convex optimization approach that stems from [3], denoted in the latter work as **[P1]**[3]. We choose this method because it is to-date one of the most efficient methods that optimizes the convergence time of homogenous swarm systems, and because it has roughly the same computational complexity as our method. The results of this method are shown in red. We see that the performance does not correlate with the eigenspecies cardinality. Since the method does not optimize the reconfiguration for desired trait distributions, it is input with a potentially sub-optimal final robot distribution (which is acerbated for low eigenspecies cardinality). Our method improves upon this state-of-the-art benchmark method by 25% for $\mathcal{D} = 5$ and by 46% for $\mathcal{D} = 1$.



(a) (b)

Fig. 5. (a) The micro-autonomous surface vehicles used in experiments. (b) Indoor testbed tank that is $3 \times 3 \times 1\,\text{m}^3$ in size.

[3]This method implicitly optimizes the convergence time by optimizing the asymptotic convergence rate (of a system of homogenous robots). We adapt the method to our problem: we minimize the second eigenvalue λ_2 of a symmetric matrix $\mathbf{S}^{(s)}$, such that $\lambda_2(\mathbf{S}^{(s)}) \geq \text{Re}(\lambda_2(\mathbf{K}^{(s)}))$. Since this method requires the knowledge of the desired species distribution $\mathbf{X}^\star$, we artificially bootstrap the method by computing a random instantiation of $\mathbf{X}^\star$ that satisfies the desired trait distribution defined by Eq. 1. We note that in practical applications, computing a good instantiation of $\mathbf{X}^\star$ is not trivial.

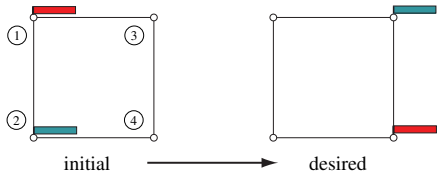


Fig. 6. Initial and desired trait distribution for a system of robots with two traits and two species.

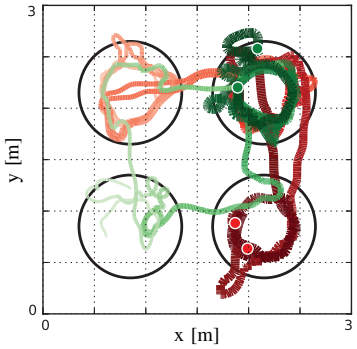


Fig. 7. Trail of four mASVs (with two mASVs per species) for an experiment of 6 minutes duration (the earlier the trail, the more transparent the color).

D. Validation on Robotic Platform

In this section, we show experimental results that *(i)* demonstrate how our approach is able to synthesize controllers for actual robotic platforms, and *(ii)* that show how real dynamics can be accounted for when choosing an appropriate rate $k_{ij,\max}$. We deploy a fleet of micro-autonomous surface vehicles (mASVs) in an indoor testbed tank that is $3 \times 3 \times 1\,\text{m}^3$ in size, see Fig. 5. The mASVs are differential drive surface vehicles equipped with a microcontroller board, an XBee radio module, and an inertial measurement unit [11]. We use two vehicle types: smaller platforms that are approximately 12 cm long and have a mass of about 45 g, and larger platforms that are approximately 15 cm long and have a mass of about 110 g. Localization for the mASVs is provided by an external motion capture system. The vehicles are tasked to monitor a site by circling within its perimeter. We use a fleet of four mASVs, emulating two species with two mASVs each. The vehicles are tasked to monitor a site by circling within its perimeter, and we imagine that the boats' traits emulate specific sensing capabilities, for instance. We compute matrices $\mathbf{K}^{(s)\star}$ with transition rates for both species, using the same parameters as above, a maximum switching rate of $k_{ij,\max} = 0.01\,\text{s}^{-1}$, and $\Delta T = 0.2\,\text{s}$. We run experiments of 6 minutes duration, and calculate the ratio of misplaced traits over time. In a first set of runs, each species owns 1 distinct trait (i.e., the eigenspecies cardinality is $\mathcal{D} = 2$), and the robots must reconfigure according to Fig. 6. In a second set of runs, both species own the same traits (i.e., the eigenspecies cardinality is $\mathcal{D} = 1$). An example from our first set of runs is shown in Fig. 7, which visualizes the trail left by the four mASVs. We see how the mASVs start out at the initial distribution

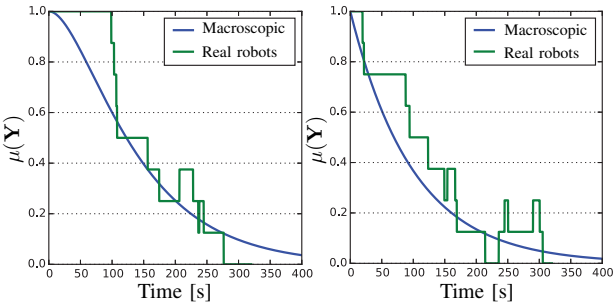


Fig. 8. Ratio of misplaced traits over time for real robot experiments with four robots, two species, and two robots per species. We show the average over 2 experimental runs per plot. The macroscopic model is also shown. (a) $\mathcal{D} = 2$ (b) $\mathcal{D} = 1$.

and, over time, converge to the desired distribution. Fig. 8 shows the average ratio of misplaced traits for two real robot runs per experiment. We see that the ratio decreases over time and eventually reaches zero as the mASVs converge to the desired distribution, thus confirming the feasibility of our method on real robotic platforms. Despite the limited number of experimental platforms and trials, we see a good agreement between real robot results and the macroscopic model.

VI. Conclusion

In this work, we contribute to the understanding of the impact of diversity in heterogeneous robot systems. We consider the specific problem of distributing a heterogeneous robot system among a set of tasks with the goal of satisfying a desired distribution of robot capabilities among those tasks. We propose a formulation for heterogeneous robot systems through *species* and *traits*, and show how this formulation is used to achieve an optimal distribution of robots by specifying the desired final trait distribution. Using this formulation, we propose a diversity metric based on *eigenspecies* that indicates how performance is affected by diversity. We show that the more the robot community is diverse, the harder it is to optimize: by adding redundant (non-complementary) species, we increase the size of the solution space and facilitate the optimization. Future work will consider methods that automatically generate desired trait distributions as a function of underlying real-world problems.

VII. Acknowledgment

We gratefully acknowledge the support of ONR grants N00014-15-1-2115, N00014-14-1-0510, and N00014-13-1-0731, ARL grant W911NF-08-2-0004, NSF grant IIS-1426840, and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. The authors would also like to thank Dhanushka Kularatne for his support in conducting the real robot experiments.

References

[1] W. Abbas and M. Egerstedt. Characterizing heterogeneity in cooperative networks from a resource distribution view-point. *Communications in Information and Systems*, 14:1–22, 2014.

[2] T. Balch. *Hierarchic Social Entropy: An Information Theoretic Measure of Robot Group Diversity*. *Autonomous Robots*, 8:209–237, 2000.

[3] S. Berman, Á. Halasz, M. A. Hsieh, and V. Kumar. Optimized Stochastic Policies for Task Allocation in Swarms of Robots. *IEEE Transactions on Robotics*, 25:927–937, 2009.

[4] B. Charrow. *Information-theoretic active perception for multi-robot teams*. PhD thesis, University of Pennsylvania, 2015.

[5] B. P. Gerkey and M. J. Mataric. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *Interntional Journal of Robotics Research*, 23(9):939–954, 2004.

[6] M. A. Hsieh, A. Cowley, F. J. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C. J. Taylor, Y. Endo, R. C. Arkin, B. Jung, F. D. Wolf, G. S. Sukhatme, , and D. C. MacKenzie. Adaptive teams of autonomous aerial and ground robots for situational awareness. *Journal of Field Robotics*, 24:991–1014, 2007.

[7] M. A. Hsieh, Á. Halasz, S. Berman, and V. Kumar. Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, 2(2-4):121–141, 2008.

[8] E. G. Jones, B. Browning, M. B. Dias, B. Argall, and M. Veloso. Dynamically Formed Heterogeneous Robot Teams Performing Tightly Coordinated Tasks. *International Conference on Robotics and Automation*, pages 570–575, 2006.

[9] J. D. Kalbfleisch and J. F. Lawless. The Analysis of Panel Data Under a Markov Assumption. *Journal of American Statistical Association*, 80:863–871, 1985.

[10] B. Korte and J. Vygen. Combinatorial Optimization: Theory and Algorithms. Springer-Verlag, Berlin., 2000.

[11] D. Larkin, M. Michini, A. Abad, S. Teleski, and M. A. Hsieh. Design of the multi-robot Coherent Structure Testbed (mCoSTe) for Distributed Tracking of Geophysical Fluid Dynamics. *Proc. of the ASME International Design Engineering Technical Conferences (IDETC)*, 2014.

[12] L. Matthey, S. Berman, and V. Kumar. Stochastic strategies for a swarm robotic assembly system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1953–1958. IEEE, 2009.

[13] N. Michael, J. Fink, S. Loizou, and V. Kumar. Architecture, Abstractions, and Algorithms for Controlling Large Teams of Robots: Experimental Testbed and Results. In *Robotics Research; Springer Tracts in Advanced Robotics*, pages 409–419. Springer Berlin Heidelberg, 2011.

[14] A. Mordecai. Nonlinear programming: analysis and methods. Courier Corporation, 2003.

[15] O. L. Petchey and K. J. Gaston. Functional diversity (FD), species richness and community composition. *Ecology Letters*, pages 402–411, 2002.

[16] A. Prorok, N. Correll, and A. Martinoli. Multi-level spatial modeling for stochastic distributed robotic systems. *The International Journal of Robotics Research (IJRR)*, 30:574–589, Apr. 2011.

[17] G. W. Stewart. *Matrix Algorithms I. Basic Decompositions*. SIAM, 1998.

[18] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler. Sensor Planning for a Symbiotic UAV and UGV system for Precision Agriculture. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5321–5326, 2013.

[19] D. J. Wales and J. P. K. Doye. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101:5111–5116, 1997.

[20] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.