Fast Redistribution of a Swarm of Heterogeneous Robots

Amanda Prorok
GRASP Laboratory
University of Pennsylvania
Philadelphia, USA
prorok@seas.upenn.edu

M. Ani Hsieh
SAS Laboratory
Drexel University
Philadelphia, USA
mhsieh1@drexel.edu

Vijay Kumar
GRASP Laboratory
University of Pennsylvania
Philadelphia, USA
kumar@seas.upenn.edu

ABSTRACT

We present a method that distributes a swarm of heterogeneous 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. Our method is based on a continuous abstraction of the swarm at a macroscopic level, as we model robots switching between tasks. We formulate an optimization problem that produces an optimal set of transition rates for each species, so that the desired trait distribution among the tasks is reached as quickly as possible. Our solution is based on an analytical gradient, and is computationally efficient, even for large choices of traits and species. Finally, we show that our method is capable of producing fast convergence times when compared to state-of-the-art methods.

Keywords

swarm robotics, heterogeneous multi-robot systems, stochastic systems, task allocation

1. 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., swarms of robots. However, the smaller we design our platforms, the more stringent the tradeoffs we need to make with respect to endowed capabilities. As a consequence, investigators are composing their robot systems with multiple, heterogeneous types of robots in order to tackle increasingly challenging tasks [1, 9]. Our premise is that, in a swarm 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.

In this work, our objective is to distribute a swarm of heterogeneous robots as quickly and efficiently as possible among a set of tasks that require specialized competences. This objective is part of a larger vision to develop control and coordination strategies for teams of heterogeneous robots with specific 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, altogether more than a single one could carry on its own. Initially, we will consider tasks that are to be performed in parallel, continuously, and independently (without precedence constraints). Instances of information gathering lend themselves naturally to this problem formulation, with applications to surveillance, environmental monitoring, and situational awareness [3, 7, 20].

Given a set of tasks, and knowledge about what the task requirements are, 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 concept considers subsets of robots in a multi-robot system, and pairs them optimally (given a cost function) to tasks. This problem is strongly NP-hard [11]. A number of heuristic algorithms have been proposed. However, the running times of these algorithms are functions of the sizes of the feasible subsets (of robots paired to a task), and hence, become very expensive for large robot teams and swarms. Furthermore, the algorithms are penalized when multiple robot-subset-to-task combinations are feasible (this is the case, for example, when robots have overlapping capabilities). These algorithms are not suitable for large-scale systems, such as robot swarms. In particular, for systems that are required to adapt to changing task requirements online, we need to consider algorithms that are efficient and that run on low-cost, resource-constrained mobile platforms. 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 [2, 6, 8]. An important property of this strategy is its inherently decentralized architecture, with robots switching between tasks (behaviors) stochastically. This model is inspired by previous work in the swarm-robotic domain that explores self-organized behavior of natural systems [12, 13].

The present work focuses on the optimization of transition rates that enables a heterogenous robot swarm to converge

quickly to a configuration that satisfies a desired trait distribution. The key difference between our work and previous work [2, 8] 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 [2, 15]). 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.

2. PROBLEM FORMULATION

Heterogeneity and diversity are core concepts of this work. To develop our formalism, we will borrow terminology from biodiversity literature [18, 19]. 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, one 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.

2.1 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 $\mathbf{q}^{(s)} = [q_1^{(s)}, q_2^{(s)}, ..., q_U^{(s)}]$. 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, $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ 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 $\mathcal{G}$ 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 that 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 task i to switch to task 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 setup. 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 $\mathbf{x}^{(s)}(t) = [\mathbf{x}_1^{(s)}(t), ..., \mathbf{x}_M^{(s)}(t)]^\top$. 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 among tasks. For time t this relationship is given by

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

2.2 Problem Statement

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

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

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:

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

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. Given a target distribution $\bar{\mathbf{Y}}$, the goal is to find an optimal rate matrix $\mathbf{K}^{(s)\bigstar}$ for each species s so that we have

$$\bar{\mathbf{Y}} = \bar{\mathbf{X}} \cdot \mathbf{Q} \tag{8}$$

In other words, the task is to redeploy the robots of each species configured according to $\mathbf{X}(0)$ initially, so that a desired trait configuration $\bar{\mathbf{Y}}$ is reached. In doing this, we reach a robot configuration $\bar{\mathbf{X}}$ that satisfies Eq. 1, subject to Eq. 6. We note that there may be several such $\bar{\mathbf{X}}$.

3. METHODOLOGY

In this section, we describe our methodology for obtaining an optimal transition matrix $\mathbf{K}^{(s)\bigstar}$ for each species so that the desired trait distribution is reached. Berman et al. [2] present an exposé of optimization methods that can be used to obtain optimal transition rates for a homogenous robot swarm that is required to converge to a desired distribution. Two general approaches are considered: convex optimization and stochastic optimization. The convex optimization approach requires knowledge of the desired final robot distribution. Indeed, our problem formulation specifies a desired trait distribution $\bar{\mathbf{Y}}$ without explicit definition of the final

robot distribution $\bar{\mathbf{X}}$. Hence, convex optimization strategies as in [2] are not applicable to our problem. Given this rationale, we choose an optimization approach that is able to find optimal transition rates with knowledge of $\bar{\mathbf{Y}}$ and $\mathbf{X}(0)$, without knowledge of $\bar{\mathbf{X}}$. Although fully stochastic schemes such as Metropolis optimization have been shown to produce similar results [2], 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. Additionally, our method explicitly minimizes the convergence time of $\mathbf{K}^{(s)}$, unlike the convex optimization methods presented in [2] which approximate $\mathbf{K}^{(s)}$ with a symmetric equivalent (forcing bidirectionally equal transition rates between tasks).

3.1 Design of Optimal Transition Rates

We combine the solution of the linear ordinary differential equation, Eq. 3, and Eq. 8 to obtain the solution:

$$\mathbf{Y}(t) = \sum_{s=1}^{S} e^{\mathbf{K}^{(s)\bigstar} t} \mathbf{x}_0^{(s)} \cdot \mathbf{q}^{(s)} \tag{9}$$

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

$$\mathbf{E} = \bar{\mathbf{Y}} - \sum_{s=1}^{S} e^{\mathbf{K}^{(s)\bigstar} \tau} \mathbf{x}_0^{(s)} \cdot \mathbf{q}^{(s)} \tag{10}$$

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

$$\begin{aligned} &\text{minimize} \quad \mathcal{J}^{(1)} = \| \mathbf{E} \|_F^2 \\ &\text{such that} \quad k_{ij}^{(s)} < k_{ij,\max}^{(s)} \end{aligned} \tag{11}$$

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. 11, but we can use the derivatives of $\mathcal{J}^{(1)}$ with respect to the parameters to perform gradient descent. So that the implementation of the optimization function is efficient, it is important that the function is differentiable and that an analytical gradient can be computed. By applying the chain rule, the derivative of our objective function with respect to the transition matrix $\mathbf{K}^{(s)}$ is

$$\frac{\partial \mathcal{J}^{(1)}}{\partial \mathbf{K}^{(s)}} = \frac{\partial \mathcal{J}^{(1)}}{\partial e^{\mathbf{K}^{(s)}\tau}} \cdot \frac{\partial e^{\mathbf{K}^{(s)}\tau}}{\partial \mathbf{K}^{(s)}\tau} \cdot \frac{\partial \mathbf{K}^{(s)}\tau}{\partial \mathbf{K}^{(s)}} \tag{12}$$

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

$$\frac{\partial \mathcal{J}^{(1)}}{\partial e^{\mathbf{K}^{(s)}\tau}} = -2\mathbf{E} \cdot \left[\mathbf{x}_0^{(s)} \cdot \mathbf{q}^{(s)} \right]^\top \tag{13}$$

The derivation of the 2nd element of Eq. 12 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 [10] to our problem, and write the gradient of our cost function as

$$\frac{\partial \mathcal{J}^{(1)}}{\partial \mathbf{K}^{(s)}} = \mathbf{V}^{-\mathbf{1}^\top} \left[\mathbf{V}^\top \frac{\partial \mathcal{J}^{(1)}}{\partial e^{\mathbf{K}^{(s)}\tau}} \mathbf{V}^{-\mathbf{1}^\top} \odot \mathbf{W}(\tau) \right] \mathbf{V}^\top \tau \tag{14}$$

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 $\mathbf{D} = \text{diag}(d_1, \ldots, d_M)$. The matrix $\mathbf{W}(t)$ is composed as follows [1]

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

3.2 Optimization of Convergence Time

The cost function in Eq. 11 does not consider 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

$$\begin{aligned} &\text{minimize} \quad \mathcal{J}^{(2)} = \mathcal{J}^{(1)} + \alpha\tau^2 \\ &\text{such that} \quad k_{ij}^{(s)} < k_{ij,\max}^{(s)} \text{ and } \tau > 0, \end{aligned} \tag{15}$$

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

$$\frac{\partial \mathcal{J}^{(2)}}{\partial \mathbf{K}^{(s)}} = \frac{\partial \mathcal{J}^{(1)}}{\partial \mathbf{K}^{(s)}} \tag{16}$$

In order to optimize the convergence time, we need the derivative with respect to τ. This derivative is computed analogously to the derivative with respect to $\mathbf{K}^{(s)}$ (confer Eq. 14). We have

$$\frac{\partial \mathcal{J}^{(2)}}{\partial \tau} = \frac{\partial \mathcal{J}^{(1)}}{\partial \tau} + 2\alpha\tau \tag{17}$$

with

$$\frac{\partial \mathcal{J}^{(1)}}{\partial \tau} = \sum_{s=1}^{S} \mathbf{1}^\top \mathbf{V}^{-\mathbf{1}^\top} \mathbf{A}_1 \mathbf{V}^\top \mathbf{K}^{(s)} \mathbf{1} \tag{18}$$

and

$$\mathbf{A}_1 = \mathbf{V}^\top \frac{\partial \mathcal{J}^{(1)}}{\partial e^{\mathbf{K}^{(s)}\tau}} \mathbf{V}^{-\mathbf{1}^\top} \odot \mathbf{W}(\tau) \tag{19}$$

The optimization of Eq. 15 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 $t^\bigstar$ remains near-constant. Hence, we modify our cost function in Eq. 15 as follows.

$$\begin{aligned} \min \mathcal{J}^{(3)} \quad &= \quad \mathcal{J}^{(2)} \\ &+ \quad \beta \sum_{s=1}^{S} \left\| e^{\mathbf{K}^{(s)}\tau} \mathbf{x}_0^{(s)} - e^{\mathbf{K}^{(s)}(\tau+\nu)} \mathbf{x}_0^{(s)} \right\|_2^2 \\ \text{such that} \quad &k_{ij}^{(s)} < k_{ij,\max}^{(s)} \text{ and } \tau > 0, \end{aligned} \tag{20}$$

and $\beta > 0$. The additional term in our cost function allows us to ensure that the state reached by employing $\mathbf{K}^{(s)\bigstar}$ remains near-constant for arbitrarily long time intervals ν. By

[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 [10]. We note that for most models of interest, however, this is rarely the case.

increasing the value of β, the difference of the robot distributions at times τ and $\tau+\nu$ is decreased. In other words, as we will see in Section 4, the trait distribution corresponding to the steady state of $\mathbf{K}^{(s)}$ gets arbitrarily close to the desired trait distribution $\bar{\mathbf{Y}}$ as β increases (the same is true when we increase ν). Note that when $\alpha = 0$, β should not be infinitely large, as in this case, $\mathbf{K}^{(s)\bigstar} = \mathbf{0}$. However, for all practical purposes β is bounded and $\alpha > 0$.

Let us refer to this additional third term of $\mathcal{J}^{(3)}$ (and second term of Eq. 20) as $\mathcal{J}^{(3,3)}$. Then, the derivative of the new objective function with respect to the transition rates can be expressed as

$$\frac{\partial \mathcal{J}^{(3)}}{\partial \mathbf{K}^{(s)}} = \frac{\partial \mathcal{J}^{(2)}}{\partial \mathbf{K}^{(s)}} + \frac{\partial \mathcal{J}^{(3,3)}}{\partial \mathbf{K}^{(s)}} \tag{21}$$

Again, we apply the chain rule to obtain

$$\begin{aligned} \frac{\partial \mathcal{J}^{(3,3)}}{\partial \mathbf{K}^{(s)}} &= \frac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}\tau}} \frac{\partial e^{\mathbf{K}^{(s)}\tau}}{\partial \mathbf{K}^{(s)}\tau} \frac{\partial \mathbf{K}^{(s)}\tau}{\partial \mathbf{K}^{(s)}} \\ &- \frac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}(\tau+\nu)}} \frac{\partial e^{\mathbf{K}^{(s)}(\tau+\nu)}}{\partial \mathbf{K}^{(s)}(\tau+\nu)} \frac{\partial \mathbf{K}^{(s)}(\tau+\nu)}{\partial \mathbf{K}^{(s)}} \end{aligned} \tag{22}$$

The outer derivative is

$$\begin{aligned} \frac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}\tau}} &= \frac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}(\tau+\nu)}} \\ &= 2\beta \left[e^{\mathbf{K}^{(s)}\tau} \mathbf{x}_0^{(s)} - e^{\mathbf{K}^{(s)}(\tau+\nu)} \mathbf{x}_0^{(s)} \right] \cdot {\mathbf{x}_0^{(s)}}^{\top} \end{aligned} \tag{23}$$

We apply the same development as in Eq. 14 to obtain the equation

$$\frac{\partial \mathcal{J}^{(3,3)}}{\partial \mathbf{K}^{(s)}} = {\mathbf{V}^{-\mathbf{1}}}^{\top} \left[\mathbf{A}_2 \tau - \mathbf{A}_3(\tau+\nu) \right] \mathbf{V}^{\top} \tag{24}$$

with

$$\mathbf{A}_2 = \mathbf{V}^{\top} \cdot \frac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}\tau}} \cdot {\mathbf{V}^{-\mathbf{1}}}^{\top} \odot \mathbf{W}(\tau) \tag{25}$$

and

$$\mathbf{A}_3 = \mathbf{V}^{\top} \cdot \frac{\partial \mathcal{J}^{(3,3)}}{\partial e^{\mathbf{K}^{(s)}(\tau+\nu)}} \cdot {\mathbf{V}^{-\mathbf{1}}}^{\top} \odot \mathbf{W}(\tau+\nu) \tag{26}$$

The derivative with respect to time τ is analogous:

$$\frac{\partial \mathcal{J}^{(3)}}{\partial \tau} = \frac{\partial \mathcal{J}^{(2)}}{\partial \tau} + \sum_{s=1}^{S} \mathbf{1}^{\top} {\mathbf{V}^{-\mathbf{1}}}^{\top} \left[\mathbf{A}_2 - \mathbf{A}_3 \right] \mathbf{V}^{\top} \mathbf{K}^{(s)} \mathbf{1} \tag{27}$$

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

$$\frac{\partial \mathcal{J}^{(z)}}{\partial k_{ij}^{(s)}} = \left\{ \frac{\partial \mathcal{J}^{(z)}}{\partial \mathbf{K}^{(s)}} \right\}_{ij} - \left\{ \frac{\partial \mathcal{J}^{(z)}}{\partial \mathbf{K}^{(s)}} \right\}_{jj} \tag{28}$$

where $\{\cdot\}_{ij}$ denotes the element on row i and column j.

Finally, we summarize our optimization problem as follows:

$$\mathbf{K}^{(s)\bigstar}, t^{\bigstar} = \underset{\mathbf{K}^{(s)}, \tau}{\operatorname{argmin}} \mathcal{J}^{(3)}, \tag{29}$$

under the constraints shown in Eq. 20. To solve the system, we implement a basin-hopping optimization algorithm [21], 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 [16] with bound constraints), using the analytical gradients given by Eq. 27 and Eq. 28.

3.3 Computational Complexity

The computational complexity of computing the gradient of our objective function is $O(S \cdot M^3 + S \cdot M^2 \cdot U)$. The first part of this complexity is dictated by the eigenvalue decomposition, which is known to be $O(M^3)$ for non-sparse matrices [4][2]. We compute this decomposition only once per optimization (see Eq. 14, where $\mathbf{K}^{(s)} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$), for each optimization of $\mathbf{K}^{(s)}$. The second part is dictated by the multiplication of the matrices in Eq. 14, for which the cost is $O(M^2 \cdot U)$. Globally speaking, the computation grows linearly with the number of species and traits, and it grows slightly slower than the cube of the number of tasks. When studying heterogenous system, it is indeed a valuable result that the gradient scales at most linearly with the number of traits and species in order to allow for the exploration of a wider range of robot capabilities. Overall, for the results shown in Section 4, 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 $\nu > 0$ (the number of parameters to optimize can be as large as 225 in this case, depending on the graph's adjacency matrix). The code was implemented in Python using the NumPy and SciPy libraries, and tested on a 2 GHz Intel Core i7 using a single CPU.

3.4 Robot Controller

The optimization described above returns optimal transition rates $\mathbf{K}^{(s)\bigstar}$. 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), and needs to be transmitted to the robots only at the start of each new redistribution.

The agent-level control is based on the transition rates $k_{ij}^{(s)}$ encoded by the transition matrix $\mathbf{K}^{(s)}$: A robot of species s at task i transitions to task j according to probability $p_{ij}^{(s)}$ 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. Hence, in order to determine which task the robot must transition to next, it samples a new task with a probability according to $\mathbf{P}^{(s)}$. This is equivalent to sampling from the discrete probability distribution $\mathcal{P}(p_{i1}^{(s)}, \ldots, p_{iM}^{(s)})$, where i represents the current task. This procedure is shown in Algorithm 1. We note that as the robot is transitioning to a new task, it continues the control loop (i.e., sampling new tasks). Although we do not explicitly model transitioning time, the resulting behavior is very close to what is predicted by the macroscopic model.

4. RESULTS

Previous work has shown the benefit of validating methods over multiple, complementary levels of abstraction (submicroscopic, microscopic, and macroscopic) [14]. In the present work, we propose an evaluation of our methods on two levels: microscopic and macroscopic. Indeed, the most

[2]In the special case where all eigenvalues are distinct, the eigenvalue decomposition can be reduced to $O(M^{2.376}\log(M))$ [17].

Algorithm 1 `Robot_Controller`($\mathbf{K}^{(s)}$, ΔT)

```
P^(s) = e^(K^(s) ΔT)
while 1 do
    m ~ P(p_i1^(s), ..., p_iM^(s))
    if m ≠ i then
        Switch to task m
        i ← m
    end if
    Wait ΔT
end while
```

efficient way of simulating the swarm of robots is by considering a continuous macroscopic model, derived directly from the ordinary differential equation, Eq. 3. In order to validate the methods at a lower level of abstraction, we also implement a discrete microscopic model that emulates the behavior of individual robot controllers. Running multiple iterations of the microscopic model enables us to capture the stochasticity resulting from our control system.

Our performance metric considers the degree of convergence to $\bar{\mathbf{Y}}$, expressed by the fraction of misplaced traits

$$\mu(\mathbf{Y}) = \frac{\|(\mathbf{Y} - \bar{\mathbf{Y}})\|_1}{2\|\mathbf{Y}\|_1} \tag{30}$$

We say that one system converges faster than another if it takes less time for $\mu(\mathbf{Y})$ to decrease to some relative error μ_{thresh}, such as $\mu_{\text{thresh}} = 2.5\%$. Similar performance metrics have been proposed in [2, 6, 8].

We will consider two optimization methods, one that stems from this paper, and one that stems from [2]:

Explicit We consider the optimization problem posed in Eq. 29 that explicitly optimizes convergence time, with $\alpha = 1$, $\beta = 5$, and $\nu = 2$, producing a fixed $\mathbf{K}^{(s)\bigstar}$ for each species.

Implicit We adapt the convex optimization method presented in [2], denoted in the latter work as **[P1]**. This adapted method implicitly optimizes the convergence time by optimizing the asymptotic convergence rate (of a system of homogenous robots). In order to do this, 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 $\bar{\mathbf{X}}$, we artificially bootstrap the method by computing a random instantiation of $\bar{\mathbf{X}}$ that satisfies the desired trait distribution defined by Eq. 8. We note that in practical applications, computing a good instantiation of $\bar{\mathbf{X}}$ is not trivial. We choose this method because it is comparable to ours, and is to-date one of the most efficient methods that optimizes the convergence time of such systems.

4.1 Example

To illustrate our method, we consider an example of $N = 800$ robots switching between $M = 8$ tasks. We sample a random initial robot distribution $\mathbf{X}(0)$ with robots distributed among three tasks, and generate a random, feasible

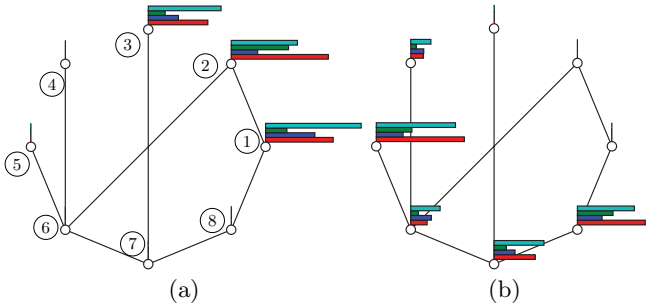

Figure 1: A strongly connected instance of a graph with 8 tasks (nodes), and possibilities of switching between tasks (edges). The system includes 4 traits. The trait abundance is represented by a bar plot. (a) Initial distribution (b) Desired distribution.

desired trait distribution $\bar{\mathbf{Y}}$ with robots distributed among the remaining five tasks. The initial trait distribution is visualized in Fig. 1(a), and the desired trait distribution is visualized in Fig. 1(b). The graph is generated randomly according to the Watts-Strogatz model [22] (with a neighboring node degree of $K = 3$, and a rewiring probability of $\gamma = 0.6$; the graph is guaranteed to be connected). We set $k^{(s)}_{ij,\max} = 1\,\text{s}^{-1}$ for all edges. The robot community consists of 3 species and 4 traits, and is defined as follows:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

with

$$\mathbf{X}^\top \cdot \mathbf{1} = [231, 312, 257]^\top \tag{31}$$

We solve the system for $\bar{\mathbf{Y}}$ as shown in Fig. 1(b). We show the evolution of the trait distribution in Fig. 2. The plots qualitatively show how the desired distribution is reached for each trait. Fig. 3 shows the ratio of misplaced traits $\mu(\mathbf{Y})$ over time for the initial and desired trait distributions depicted in Fig. 1. We run 100 iterations of the discrete microscopic model with method **Explicit**. The plot shows that our solution reaches the desired trait distribution, and that the trait error decreases exponentially. Initially, the microscopic and macroscopic models show good agreement, up to about $t = 5$ seconds. 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.

4.2 Comparison of Methods

We compare the two optimization methods, **Explicit**, and **Implicit**, and evaluate their performance with respect to the metric in Eq. 30. We instantiate 40 random graphs with $M = 6$ nodes, and random matrices $\mathbf{Q}$ with $S = 4$ species and $U = 4$ traits, and generate random desired trait distributions $\bar{\mathbf{Y}}$ for each graph. We set $k^{(s)}_{ij,\max} = 2\,\text{s}^{-1}$ for all edges. The microscopic model is iterated 4 times on each graph instantiation. For the method **Implicit**, we compute a random robot distribution $\bar{\mathbf{X}}$ that satisfies the desired trait distribution. We measure the time $t_{\mu,\text{thresh}}$ at which the

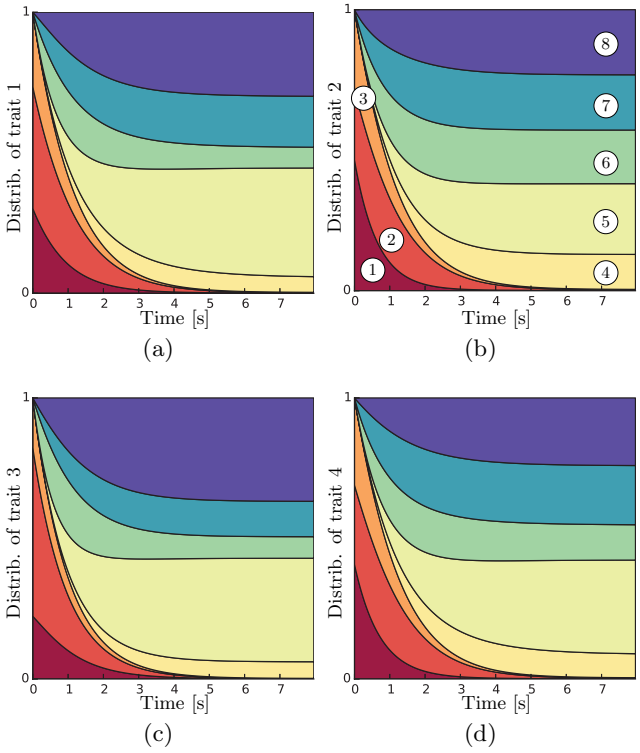

Figure 2: Evolution over time of the trait distribution as specified by the distributions shown in Fig. 1. Each subplot represents one trait, indicating the distribution of that trait over the set of tasks (task 1 is shown at the bottom and task 8 at the top).

system converges to a value $\mu_{\text{thresh}} = 2.5\%$ of misplaced traits.

Fig. 4 shows the results. The median of **Explicit** is able to improve upon the median of **Implicit** by 21%. This result is expected, as our method explicitly minimizes the convergence time of the actual system (rather than maximizing the asymptotic convergence rate of an approximated system). Also, we note that the spread of values between the 25th and 75th percentiles is 43% smaller for **Explicit**, showing that our method is more robust to different initial conditions. Finally, we compute the error obtained through method **Explicit** by comparing the analytical steady-state distribution of traits (obtained by taking the eigenvectors that correspond to the zero-eigenvalues of each rate matrix $\mathbf{K}^{(s)}$ and multiplying them by $\mathbf{Q}$) with the desired trait distribution $\bar{\mathbf{Y}}$. The median, 90th percentile and maximum error from the steady-state to the desired trait distribution are 0.108%, 0.572% and 0.812%, respectively. These results demonstrate that, despite the fact that our method is not explicitly optimizing for the steady-state, it reaches a steady-state error smaller than system noise (at steady-state).

5. CONCLUSION

We present a method that distributes a swarm of heterogeneous robots 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

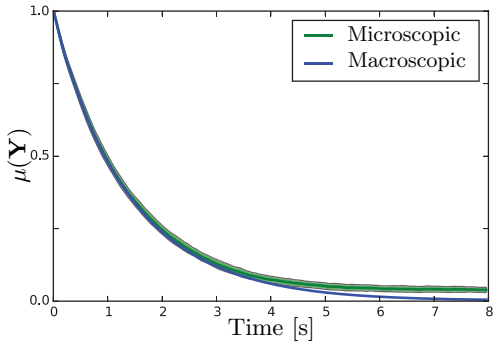

Figure 3: Ratio of misplaced traits over time for the initial and desired trait distributions depicted in Fig. 1. The simulation was run with 800 robots. The plot shows the macroscopic model as well as the average over 100 iterations of the microscopic model. The shaded area shows the standard deviation.

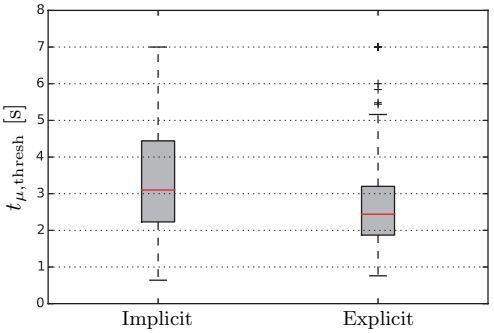

Figure 4: The plot shows the convergence time for the optimization methods, evaluated on the microscopic model, with $t_{\mu_{\text{thresh}}}$ for $\mu_{thresh} = 2.5\%$, for 40 random graphs with $M = 6$ and random matrices Q with 4 species and 4 traits. The boxplots show the median and the 25th and 75th percentiles.

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 configuration. To find the optimal transition rates, we pose an optimization problem, and develop a solution based on an analytical gradient that is computationally efficient and capable of producing fast convergence times, even for large choices of traits and species. Indeed, the gradient computation is fully scalable with respect to the number of robots, number of species and number of traits. We validate our approach on random graph instantiations, and show that our baseline method outperforms a classical alternative approach. We believe that this method is well-suited to applications that control large-scale teams of robots that need to converge quickly to desired configurations as a function of their capabilities.

6. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of ONR grants N00014-15-1-2115 and N00014-14-1-0510, 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.

References

[1] T. Balch and L. E. Parker. Special issue on Heterogeneous Multi-Robot Systems. *Autonomous Robots*, 8:207–383, 2000.

[2] 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.

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

[4] J. Demmel, I. Dumitriu, and O. Holtz. Fast Linear Algebra is Stable. *arXiv.org*, pages 1–26, 2007.

[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] Á. Halasz, M. A. Hsieh, S. Berman, and V. Kumar. Dynamic Redistribution of a Swarm of Robots Among Multiple Sites. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[7] 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.

[8] 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.

[9] 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 (ICRA)*, pages 570–575, 2006.

[10] 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.

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

[12] M. J. B. Krieger, J. B. Billeter, and L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406:992–995, 2000.

[13] T. H. Labella, M. Dorigo, and J.-L. Deneubourg. Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Transactions on Autonomous Adaptive Systems*, 1:4–25, 2006.

[14] A. Martinoli. Swarm Intelligence in Autonomous Collective Robotics: From Tools to the Analysis and Synthesis of Distributed Collective Strategies. PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1999.

[15] 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.

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

[17] V. Y. Pan and Z. Q. Chen. The Complexity of the Matrix Eigenproblem. *ACM Symposium on Theory of Computing*, pages 507–516, 1999.

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

[19] D. Tilman. Functional Diversity. *Encyclopedia of Biodiversity*, 3:109–120, 2001.

[20] 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.

[21] 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.

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