Algorithms for Task Allocation in Homogeneous Swarm of Robots

Jha, Devesh K.

TR2018-151 October 23, 2018

Abstract

In this paper, we present algorithms for synthesizing controllers to distribute a swarm of homogeneous robots (agents) over heterogeneous tasks which are operated in parallel. Swarm is modeled as a homogeneous collection of irreducible Markov chains. States of the Markov chain represent the tasks performed by the swarm. The target state is a pre-defined distribution of agents over the states of the Markov chain (and thus the tasks). We make use of ergodicity property of irreducible Markov chains to ensure that as an individual agent converges to the desired behavior in time, the swarm converges to the target state. To circumvent the problems faced by a global controller and local/decentralized controllers alone, we design a controller by combining global supervision with local-feedback-based state level decisions. Some numerical experiments are shown to illustrate the performance of the proposed algorithms.

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

Algorithms for Task Allocation in Homogeneous Swarm of Robots

Devesh K. Jha†
devesh.jha@merl.com

***Abstract*— In this paper, we present algorithms for synthesizing controllers to distribute a swarm of homogeneous robots (agents) over heterogeneous tasks which are operated in parallel. Swarm is modeled as a homogeneous collection of irreducible Markov chains. States of the Markov chain represent the tasks performed by the swarm. The target state is a pre-defined distribution of agents over the states of the Markov chain (and thus the tasks). We make use of ergodicity property of irreducible Markov chains to ensure that as an individual agent converges to the desired behavior in time, the swarm converges to the target state. To circumvent the problems faced by a global controller and local/decentralized controllers alone, we design a controller by combining global supervision with local-feedback-based state level decisions. Some numerical experiments are shown to illustrate the performance of the proposed algorithms.**

I. Introduction

Autonomous robots are becoming ubiquitous and are expected to play an increasingly important role in both civilian and military applications such as intelligence, surveillance, reconnaissance (ISR), health care, and logistics, to name a few. Recent advances in sensor technology and embedded processors have allowed us to design *swarm* of robots which are capable of doing various coordinated complex tasks. Some important examples are fleets of unmanned vehicles deployed in ocean bed for the purposes of mine-hunting or surveillance or drug delivery techniques in humans using micro-scale robots at pre-specified rate and locations [1], [2]. A common element in all these applications is a desirable global behavior which can be achieved where the individual behavior is achieved by the agents themselves and thus, is not considered during global policy synthesis. In such large-scale systems, it is, in general, difficult to design controller for each agent (or robot) individually and then coordinate their behavior, for the computational requirements. It is not difficult to imagine that the corresponding state-space would grow exponentially and such a system would be highly inefficient. A critical challenge, among several others, in these highly distributed systems is to guarantee stability and performance with limited global information. In this paper, we are interested in synthesis of controllers for controlling global states of a large-scale system with the assumption that each of the individual agents are controllable.

Related Work: Traditionally, swarm control has been studied under two broad categories: centralized control with global information, where the controller broadcasts the policies to the swarm and decentralized control where, for example, some bio-inspired collective behavior laws are used to replicate social behavior in nature. A lot of work has been done in swarm control addressing issues of centralized and decentralized control. Some examples of centralized control could be found in [3]–[7].

Some distributed control approaches for swarm control could be found in [8]–[11]. However, there are no performance or convergence guarantees. This work is motivated by the idea that we can augment the centralized controller by a state-dependent feedback strategy so that the agents can use the centralized control strategy with limited perturbation from the same; the perturbation in the strategy depends on the current state of an agent and the desired steady state of the system. The distributed state-information-based feedback control leads to a time-varying stochastic linear system. Such time varying stochastic analysis has been been extensively presented in consensus literature [12], [13]. It is noted that in the following text we often call transition matrix as Markov kernel even though kernels are generally defined for continuous state-space (and our problem is discrete).

Contributions: We present a problem of swarm control and propose algorithms for centralized as well as control with distributed autonomy. In particular, this paper has the following major contributions.

1) We present a closed-form solution to the swarm control problem presented in an earlier publication [4].
2) Additionally, we present a framework where we augment the global controller with a local information-based feedback controller to reduce the movement of agents to achieve the desired state.

II. Preliminaries and Notations

In this section, we provide some preliminary concepts and notations that facilitate understanding of the concepts presented in the sequel. We denote the cardinality of any set Φ by $|\Phi|$.

Definition II.1. *A finite-state homogeneous Markov chain is a triple $G = (Q, \boldsymbol{P}, \mathbf{p}^{[0]})$, where Q is the set of states with $|Q| = M \in \mathbb{N}$, $\boldsymbol{P}$ is the $|Q| \times |Q|$ stationary probability matrix such that $\forall\, i, j \in \{1, \ldots, M\}$, $\boldsymbol{P}_{ij} \geqq 0$ with $\sum\limits_{j=1}^{M} \boldsymbol{P}_{ij} = 1$, and $\mathbf{p}^{[0]} \in [0,1]^{|Q|}$, $\sum\limits_{i=1}^{M} \mathbf{p}_i^{[0]} = 1$, is the initial distribution over the states.*

Definition II.2. *A finite-state homogeneous Markov chain $G = (Q, \boldsymbol{P}, \mathbf{p}^{[0]})$, with $|Q| = M, M \geq 2$, is called irreducible if, for any (i,j), $1 \leq i, j \leq M$, there exists a*

†The author is with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, 02139, USA.

finite positive integer $k(i,j)$ such that the ij^{th} element of the k^{th} power of $\boldsymbol{P}$ is strictly positive, i.e., $\boldsymbol{P}_{ij}^{k} > 0$. In this case, the stochastic transition matrix $\boldsymbol{P}$ is also an irreducible matrix [14].

The transition matrix of a finite Markov chain is always a stochastic matrix [14]. Any stochastic transition matrix $\mathbf{K}$ has at least one unity eigenvalue, and all the eigenvalues of $\mathbf{K}$ are located within or on the unit disc. For any $M \times M$ irreducible stochastic matrix $\mathbf{K}$ with $M > 1$, the diagonal terms are strictly less than unity, i.e., $\mathbf{K}_{ii} < 1 \, \forall i$. Upon unity sum normalization, the left eigenvector $P = [P_1, P_2, \ldots, P_M]$ corresponding to the unique unity eigenvalue of $\mathbf{K}$ is called the stationary probability vector, where $\sum_i P_i = 1$, and $P_j > 0 \quad \forall\, j \in \{1, 2, \ldots, M\}$ [14] such that $P\mathbf{K} = P$.

Next we present a brief review of language measure theory [15], [16] which is later used to synthesize the distributed architecture for the problem discussed in this paper.

A. Review of Language Measure of a Probabilistic Finite State Automata

This section summarizes the concept of signed real measure of probabilistic finite state automata (PFSA) and its role for optimal control of PFSA. While the theories of language measure and the associated optimal control are developed in [15], this section introduces pertinent definitions and summarizes the essential concepts that are used in the sequel.

Definition II.3. *(PFSA) A probabilistic finite state automaton (PFSA) over an alphabet Σ is a quintuple*

$$\mathbb{G} \triangleq (Q, \Sigma, \delta, \Pi, \boldsymbol{\chi}), \text{ where}$$

- *Q is the nonempty finite set of states, i.e., $|Q| \in \mathbb{N}$;*
- *$\delta : Q \times \Sigma^{\star} \to Q$ is the transition map that satisfies the following conditions: $\forall q \in Q \;\; \forall s \in \Sigma \;\; \forall w \in \Sigma^{\star}$ $\delta(q, \epsilon) = q$; and $\delta(q, ws) = \delta\big(\delta(q, w), s\big)$.*
- *$\Pi : Q \times \Sigma^{\star} \to [0, 1]$ is the morph function of state-specific symbol generation probabilities, which satisfies the following conditions: $\forall q \in Q \;\; \forall s \in \Sigma \;\; \forall w \in \Sigma^{\star}$ $\Pi(q, s) \geq 0$; $\sum_{s \in \Sigma} \Pi(q, s) = 1$; and $\Pi(q, \epsilon) = 1$; $\Pi(q, ws) = \Pi(q, w) \times \Pi(\delta(q, w), s)$.*
- *$\boldsymbol{\chi} : Q \to [-1, 1]^{|Q|}$ is the vector-valued characteristic function that assigns a signed (normalized) real weight to each state.*

The $(|Q| \times |Q|)$ state transition probability matrix $\boldsymbol{P}$ is defined as

$$\boldsymbol{P} = [\boldsymbol{P}_{jk}], \text{ where } \boldsymbol{P}_{jk} \triangleq \sum_{s \in \Sigma:\ \delta(q_j, s) = q_k} \Pi(q_j, s) \; \forall q_j, q_k \in Q$$

Note: $\boldsymbol{P}$ is a non-negative stochastic matrix [14]. It is also noted that a PFSA induces a Markov chain $\mathbb{M} = (Q, \boldsymbol{P}, \boldsymbol{p}^{[0]})$ where $\boldsymbol{p}^{[0]}$ is the one-hot vector representing the initial state of the PFSA.

We skip some definition and details for brevity. Interested readers are refered to [16] for details about the same. We present definition of language induced by a PFSA before presenting the definition of the real measure for the same.

Definition II.4. *(Language) Let $(Q, \Sigma, \delta, \Pi, \boldsymbol{\chi})$ be a PFSA. The language generated by all words, which terminates at a state $q_k \in Q$ after starting from $q_j \in Q$, is defined as*

$$L(q_j, q_k) \triangleq \{w \in \Sigma^{*} :\ \delta^{\star}(q_j, w) = q_k\} \tag{1}$$

The language generated by all words, which may terminate at any state $q \in Q$ after starting from $q_j \in Q$, is defined as

$$L(q_j) \triangleq \bigcup_{q \in Q} L(q_j, q) \tag{2}$$

Definition II.5. *(Language Measure) Let $L(q_j, q_k)$ and $L(q_j)$ be languages on a PFSA $(Q, \Sigma, \delta, \pi, \boldsymbol{\chi})$ and let $\theta \in (0, 1)$ be a parameter. A signed real measure $\mu_{\theta}^{jk} : 2^{L(q_j, q_k)} \to \mathbb{R}$ (that satisfies the requisite axioms of measure [17]) is defined as*

$$\mu_{\theta}^{jk}(L(q_j, q_k)) \triangleq \sum_{w \in L(q_j, q_k)} \theta (1 - \theta)^{|w|} \pi(q_j, w) \boldsymbol{\chi}(q_k) \tag{3}$$

The measure of the language $L(q_j)$ is defined as

$$\nu_{\theta}^{j}(L(q_j)) \triangleq \sum_{q_k \in Q} \mu_{\theta}^{jk}(L(q_j, q_k)) \tag{4}$$

The language measure of the PFSA $(Q, \Sigma, \delta, \pi, \boldsymbol{\chi})$ in Eq. (II.5) is expressed vectorially as

$$\boldsymbol{\nu}_{\theta} = \theta [\mathbb{I} - (1 - \theta)\mathbf{P}]^{-1} \boldsymbol{\chi} \tag{5}$$

where $\mathbf{P}$ is the state transition matrix (see Definition II.3) and the inverse on the right side exists for all $\theta \in (0, 1)$ [15]. Furthermore, as $\theta \to 0^{+}$, the matrix $\theta[\mathbb{I} - (1 - \theta)\mathbf{P}]^{-1}$ converges to the Cesaro matrix $\boldsymbol{\mathcal{P}} \triangleq \lim_{k \to \infty} \frac{1}{k} \sum_{j=0}^{k-1} \mathbf{P}^{j}$. Then, the limiting measure vector $\boldsymbol{\nu}_0$ is obtained as [15]

$$\boldsymbol{\nu}_0 \triangleq \lim_{\theta \to 0^{+}} \boldsymbol{\nu}_{\theta} = \lim_{\theta \to 0^{+}} \theta [\mathbb{I} - (1 - \theta)\mathbf{P}]^{-1} \boldsymbol{\chi} = \boldsymbol{\mathcal{P}} \ \boldsymbol{\chi} \tag{6}$$

where $\mathbb{I}$ is the $(|Q| \times |Q|)$ identity matrix.

The limiting language measure ν_0^{j} sums up to the limiting real measure of each string starting from state q_j, given the weighting function, $\boldsymbol{\chi}$.

III. Problem Formulation

Consider a set of N robots to be allocated among M heterogeneous tasks which are operated in parallel. The number of robots performing task $i \in \{1, \ldots, M\}$ at a time epoch k is denoted by $n_i^{[k]}$. The desired number of robots for task i is denoted by n_i^{d}. We assume that $n_i^{d} > 0$ for all $i \in \{1, \ldots, M\}$. Then, to make the system scalable in the number of agents, we define the population fraction at any task i as $p_i^{[k]} = n_i^{[k]}/N$. The state of the swarm is then defined as $\mathbf{p}^{[k]} = [p_1^{[k]}, \ldots, p_M^{[k]}]$. The desired state of the swarm is given by the fraction of agents at the individual tasks which is denoted by the vector $\mathbf{p}^{d} = [p_1^{d}, \ldots, p_M^{d}]$. It is noted that since $n_i^{d} > 0 \; \forall i \in \{1, 2, , \ldots, M\}$, thus we have that $p_i^{d} > 0 \, \forall \, i \in \{1, \ldots, M\}$.

Assumption III.1. $|Q| = M < \infty$ *i.e., the number of tasks operated in parallel are finite.*

Next we formalize the definition of an agent and swarm before we state the formal problem.

Definition III.1. ***(Agent in Swarm Modeling)****: An agent is a connected digraph $R = (Q, E)$, where each state $i \in Q$ represents a distinct predefined behavior (i.e., a heterogeneous task), and $E \in \{0,1\}^{|Q|\times|Q|}$ is a matrix such that $E_{ij} = 1$ implies there exist a controllable transition from state i to j (implying the connectivity of the tasks).*

The matrix E in Definition III.1 specifies state transitions of the agent R's state. An agent can only transition to the tasks it is directly connected with in a single hop (or a controllable movement). It is possible to associate probabilities with the transition of an agent between tasks based on the requirements at the tasks or the preference of the robot. The probabilities of the state transitions constitute a finite-state irreducible Markov chain, with the irreducibility property following from the connectedness of the agent graph. Thus the behavior of an agent can be represented by an irreducible Markov chain $G = (Q, \mathbf{P}, \mathbf{p}^{[0]})$, where $\mathbf{P}$ represents a stochastic matrix such that $\mathbf{P}_{ij} > 0$ if and only if $E_{ij} = 1$; $\mathbf{p}^{[0]}$ represents its initial state (which could be a one-hot vector representing the initial task of the agent).

Definition III.2. ***(Swarm)****: In the sense of Definition III.1, a homogeneous swarm $\$$ is defined to be a collection of independent identical agents $R = (Q, E)$, each of which is represented by some (finite-state) irreducible Markov chain $G = (Q, \boldsymbol{P}, \mathbf{p}^{[0]})$. Formally,*

$$\$ = \{G^\alpha : \alpha \in \mathbb{X}\}$$

such that $G^\alpha = G$, and $\mathbb{X}$ is an index set (finite, countable or uncountable). The state of the swarm is defined by the distribution of robots over the tasks i.e., $\mathbf{p}^{[k]}$). The swarm dynamics is represented by the following equation:

$$\mathbf{p}^{[k+1]} = \mathbf{p}^{[k]}\boldsymbol{P} \tag{7}$$

where $\boldsymbol{P}_{ij}$ represents the probability with which an agent decides to switch from task i to task j.

Following Definition III.2, let $\$ = \{G^\alpha : \alpha \in \mathbb{X}\}$, be a homogeneous swarm, where $G^\alpha = (Q, \mathbf{P}, \mathbf{p}^{[0]})$ is the irreducible Markov chain corresponding to the uncontrolled agent and a known target state $\mathbf{p}^d$ for the swarm, where $\sum_{i=1}^{|Q|} \mathbf{p}_i^d = 1$, and $\mathbf{p}_j^d > 0 \forall j \in \{1, \ldots, M\}$. The swarm dynamics is governed by equation (7).

With the above definitions in mind, we now define the first problem we present in this paper.

A. Central Controller Design

Imagine a swarm of robots where the initial state of the swarm is $\mathbf{p}^{[0]}$ and the desired state is represented by $\mathbf{p}^d$. The goal of the central controller is to achieve the desired distribution of agents over the tasks. This can be achieved in a probabilistic setting by using a Markov kernel for transition between tasks such that the desired state is the stationary distribution over the states of the swarm. Thus, given a desired state distribution $\mathbf{p}^d$, the problem is to synthesize a Markov kernel $\mathbf{P}^\star$ given any initial kernel $\mathbf{P}$ such that the following conditions hold.

1) $\sum_{j=1}^{M} \mathbf{P}^\star_{ij} = 1$.
2) $\mathbf{P}^\star_{ij} > 0$ if and only if $\mathbf{P}_{ij} > 0$ for any $i, j \in 1, \ldots, M$.
3) $\mathbf{P}^\star$ is an irreducible matrix.
4) $\lim_{k\to\infty} \|\mathbf{p}^{[0]}(\mathbf{P}^\star)^k - \mathbf{p}^d\|_\infty = 0.$

B. Central Control with Distributed Autonomy

In this problem, we want to have a local-information-based policy for the agents which could be calculated as a perturbation to the central control policy $\mathbf{P}^\star$. The perturbed policy could be derived as a function of $\mathbf{p}_i^{[k]}$, the desired state $\mathbf{p}_i^d$ and the neighboring states $\mathbf{p}_j^{[k]}$ where j is such that $E_{ij} = 1$. It results in time varying stochastic policies for the agents depending on their current state and local information. We denote the perturbed local policies by $\tilde{\mathbf{P}}^{[k]}$ at an instant k. Also, we want only λ fraction of the agents to switch between the tasks at steady state when compared to the number of agents switching tasks under the central policy. Then, the local-information-based perturbed policy has to satisfy the following conditions.

1) $\sum_{j=1}^{M} \tilde{\mathbf{P}}^{[k]}_{ij} = 1.$
2) $\tilde{\mathbf{P}}^{[k]}_{ij} > 0$ if and only if $\mathbf{P}^\star_{ij} > 0$ for any $i, j \in 1, \ldots, M$..
3) $\sum_{j=1}^{|Q|} \tilde{\mathbf{P}}^{[k]}_{ij} = 1.$
4) $\lim_{k\to\infty} \|\mathbf{p}^{[0]} \prod_{i=0}^{k} \tilde{\mathbf{P}}^{[i]} - \mathbf{p}^d\|_\infty = 0$
5) $\lim_{k\to\infty} \tilde{\mathbf{P}}^{[k]}_{ij} = \lambda \mathbf{P}^\star_{ij}$ for $i \neq j$ and $\lim_{k\to\infty} \tilde{\mathbf{P}}^{[k]}_{ii} = \lambda \mathbf{P}^\star_{ii} + (1 - \lambda)$

It is noted that the irreducibility of $\tilde{\mathbf{P}}^{[k]}$ follows from condition (1) and the fact that $\mathbf{P}^\star$ is an irreducible matrix. At this point, we would like to clarify that condition 5 implies that the robots stay at the same task with an increased probability of $(1-\lambda)$ and thus, the probability to switch task at any state is reduced by fraction λ.

It is also noted that both problems are related to synthesis of Markov kernels such that the stationary distribution of the underlying Markov chain achieves the desired state of the swarm in the asymptotic limit.

IV. Proposed Algorithms

In this section we present the proposed approach for estimation of the Markov kernels described in section III. As described earlier in section III, we first present a solution to III-A and then to III-B. Solutions are presented as psuedo-codes in Algorithms 1 and 2. Proofs of the underlying Theorems are being skipped for brevity.

A. Algorithm for Central Controller Synthesis

This section presents a closed-form solution to solve the control problem described in section III-A. Let $\mathbf{p}^d$ be the desired state of the swarm with initial state $\mathbf{p}^{[0]}$. Let $\mathbf{P}$ be an irreducible stochastic matrix for G and let $\tilde{\mathbf{p}}$ be its unique stationary probability distribution vector. Then, a Markov kernel which achieves the desired distribution over the swarm

Algorithm 1: `EstimatingMarkovKernel`

for $i \in \{1, \dots, M\}$ **do**
  $\hat{d}_i = \frac{p_i^0}{p_i^d}$;
for $i \in \{1, \dots, M\}$ **do**
  Normalize $d_i = \frac{\hat{d}_i}{\sum_{i=1}^{M} \hat{d}_i}$;
return $\{d_i\}_{i=1,\dots,M}$;
return $\Pi^\star = \mathtt{diag(d)}\Pi - \mathtt{diag(d)} + \mathtt{I}$
  ▷ $\mathtt{I}$ represents the identity matrix of the same size as Π

in the asymptotic limit could be obtained using the following transformation of the matrix $\mathbf{P}$.

$$\mathbf{P}_{ij} \longmapsto d_i \mathbf{P}_{ij}, i \neq j$$
$$\mathbf{P}_{ij} \longmapsto d_i \mathbf{P}_{ij} + (1 - d_i), i = j$$

where, $d_i \in (0, 1)$ $\forall i \in \{1, 2, \dots, M\}$ where the vector $\mathbf{d} = [d_1, \dots, d_M]$ is given by the following expression.

$$[d_1, \dots, d_M] = [\hat{d}_1, \dots, \hat{d}_M] / \sum_{i=1}^{M} \hat{d}_i \tag{8}$$

where $[\hat{d}_1, \dots, \hat{d}_M] = [p_1^{[0]}, \dots, p_M^{[0]}]\mathbf{X}^{-1}$ where, $\mathbf{X}^{-1} = \mathrm{diag}(p_1^d, \dots, p_M^d)^{-1}$. The vector $\mathbf{p}^d$ and $\mathbf{p}^{[0]}$ denote the final and initial distribution of the swarm, respectively. The vector $\mathbf{d}$ in equation (8) gives the closed form solution to the iterative Algorithm 1 in [4]. This helps in greatly simplifying the controller synthesis complexity which is useful for swarms with a large number of states or tasks performed in parallel. It is noted that the perturbations preserve the original topology of the graph representing the connectivity of the tasks. For convenience of presentation, the transformation is presented as a psuedo-code in Algorithm 1.

Next, we present a lemma and consequently a theorem (for brevity we skip the analysis) which show correctness of our approach.

Lemma IV.1. *We define a perturbation of the irreducible stochastic matrix Π as follows:*

$$\hat{\Pi} = D\Pi - D + I \tag{9}$$

where, $D = \mathrm{diag(d_i)}\, \mathrm{i} \in \{1, \dots, \mathrm{M}\}$, where $d_i \in (0, 1)$ and I is an identity matrix of size $M \times M$. Then $\hat{\Pi}$ is a stochastic irreducible matrix.

Theorem IV.1. *Let $\mathbf{p}^d \in \mathbb{R}^M$ be an element-wise positive probability vector and let $\Pi \in \mathbb{R}^{M \times M}$ be an irreducible stochastic matrix with a strictly positive stationary probability vector, $\mathbf{p}$. Then, $\exists$ a diagonal matrix $D \in \mathbb{R}^{M \times M}$ with $D_{ii} \in (0, 1)$ such that $\Pi^\star = D\Pi - D + I$ is an irreducible stochastic matrix with stationary probability vector $\mathbf{p}^d$.*

Theorem IV.1 guarantees the existence of a solution for the desired matrix. The diagonal elements of the matrix D is given by Equation (8). The psuedo code to estimate a feasible Markov kernel is presented in Algorithm 1. Proofs are being skipped here for brevity.

Remark IV.1. *In Lemma IV.1, we show that under the perturbations described, we retain the irreducibility of the stochastic matrix and through Theorem IV.1 guarantees the existence of a possible perturbation so that the perturbed irreducible stochastic matrix attains the desired distribution for the swarm. It is noted that the solution is not unique; however, this gives a closed-form solution for controller synthesis.*

B. Algorithm for Distributed Autonomy

In the last section, we presented a closed form solution to the controller synthesis problem described in Section III-A. It was based on the knowledge of global state of the swarm and the synthesized controller has asymptotic convergence and global stability guarantees. However, it might lead to unnecessary movement of agents at steady state. Furthermore, in the absence of any feedback, a swarm cannot react to any unforeseen changes events which might lead to changes in requirement of agents at different states. It is desirable that the robots (agents) should have some degree of autonomy to choose their action based on their current state and local state information of the swarm. This forms the motivation of the problem described in Section III-B. In this section, we present a framework to allow distributed autonomy to the robots so that they can decide to follow the global policy in a probabilistic fashion while retaining global stability. We define the degree of autonomy as the fraction of the times an agent decides to follow the global policy against the policy of staying in the same state.

The local feedback-based policy represents the probability with which an agent decides to either follow the global policy or stay in its current state. The goal is to achieve a predefined percentage of activity at steady-state. To formulate the problem of distributed autonomy in the setting of language measure theory, we augment the states of the Markov chain (that represents the swarm) with characteristic weights (see Definition II.3 in Section II). Please see Section II for more information on the characteristic weights. The characteristic weights of the states are defined as follows: $\chi_i^{[k]} = \mathbf{p}_i^d - \mathbf{p}_i^{[k]}$.

The vector $\chi^{[k]} = [\chi_1^{[k]}, \chi_2^{[k]}, \dots, \chi_M^{[k]}]^T$ thus contains the information about the deficit or excess of agents at the individual tasks (or the states of the swarm). A positive value of $\chi_i^{[k]}$ would suggest deficit of agent in state i and a negative value suggests excess of agents. As mentioned earlier, the goal is to achieve a certain fraction, λ, of original activity level at steady state. During the transient phase, some states with positive characteristic weights observe higher (than steady-state) activity rates than states with negative characteristic weights. It is presented more formally next.

To estimate a measure for *goodness* of a swarm state, we estimate value functions for each state as the expectation of characteristic weights over the states. The expected value of the characteristic weights for the Markov chain (that represents the swarm) is calculated using the language measure theory as discussed in Section II. The expectation, parameterized by a parameter $\theta \in (0, 1)$, of the characteristic weights of an irreducible Markov chain with stochastic matrix $\mathbf{P}$ is calculated by the following recursive equation

Algorithm 2: `EstimatingDistributedPolicy`

for $i \in \{1, \ldots, M\}$ *till convergence* **do**
  for $j \in$ `Nb(i)` **do**
    ▷ `Nb(i)` represents the neighborhood of state i
    $\nu_i \leftarrow \sum_{j \in \mathtt{Nb(i)}} (1-\theta)\mathbf{P}_{ij}\nu_j + \theta\chi_i$;
$\mu = \nu - \chi$;
for $i \in \{1, \ldots, M\}$ **do**
  $f_k(\mu_i) = \frac{1}{1+(\frac{1}{\lambda}-1)\exp(-\beta_k\mu_i)}$;
return $\{f_k(\mu_i)\}_{i=1,\ldots,M}$;

(we drop the superscript k for simplicity).

$$\nu_i \leftarrow (1-\theta) \sum_{j \in \mathrm{Nb}(i)} \mathbf{P}_{ij}\nu_j + \theta\chi_i \tag{10}$$

where, $\mathrm{N}b(i)$ is the neighborhood of the agent i. We use $\theta = 0.02$ for the computation of the expectation. This is guided by the fact that the optimal value function is achieved in the asymptotic limit as θ goes to zero (see Section II). Interested readers are referred to earlier publications [15] for further discussion on choice of the parameter θ. The measure ν_i is the discounted expected value of χ for agents starting in state i. Clearly, $\nu = \mathbf{0}$ if $\chi = \mathbf{0}$. Convergence of the expected weights follow from the fact that χ is a constant vector and $\mathbf{P}$ is a row stochastic matrix. All the states (i.e., tasks) synchronously calculate their own ν_i and then broadcast it to their neighbors. This is repeated recursively till the expectations converge. Based on the measure defined in equation 10, we define a quantity, $\mu = \nu - \chi$.

The quantity μ_i represents the difference between the characteristic weight for state i and the expected value of the χ for agents starting in state i. As such, a positive value of μ_i would mean that the states to which the agents can go from that state have higher expected reward than their current state and hence, such states are expected to have higher activity. An activation function is defined according to the following sigmoid function

$$f_k(\mu_i^{[k]}) = \frac{1}{1+(\frac{1}{\lambda}-1)\exp(-\beta^{[k]}\mu_i)} \tag{11}$$

where, $\lambda \in (0,1)$ and $\beta^{[k]} \in \mathbb{R}_+$ are parameters. λ denotes the steady state activity level for the agents. $\beta^{[k]}$ is a scaling factor for $\mu_i^{[k]}$. Clearly, $f_k(\mu_i^{[k]}) \in (0,1)$ for all $\mu_i^{[k]} \in \mathbb{R}$. The sigmoid functions are used to design smooth (i.e., the feedback rate is differentiable) feedback controllers.

The state-based perturbation to the central control policy $\mathbf{P}^\star$ of the swarm is then defined by the following equations.

$$\begin{aligned} \mathbf{P}^\star_{ij} &\longmapsto b_i^{[k]}\mathbf{P}^\star_{ij}, i \neq j \\ \mathbf{P}^\star_{ij} &\longmapsto b_i^{[k]}\mathbf{P}^\star_{ij} + (1-b_i), i = j \end{aligned} \tag{12}$$

where, $b_i^{[k]} = f_k(\mu_i^{[k]})$, where $\mu_i^{[k]}$ is given by equation (10). Clearly, $\lim_{k\to\infty} f_k(\mu_i^{[k]}) = \lambda$ if $\lim_{k\to\infty} \mu_i^{[k]} = 0$. Thus, the feedback matrix for at epoch k is given by,

$$\begin{aligned} \tilde{\mathbf{P}}^{[k]}_{ij} &= b_i^{[k]}\mathbf{P}^\star_{ij}, i \neq j \\ \tilde{\mathbf{P}}^{[k]}_{ij} &= b_i^{[k]}\mathbf{P}^\star_{ij} + (1-b_i^{[k]}), i = j \end{aligned} \tag{13}$$

Thus the idea is that based on the state of a task and the neighboring tasks, an agent can probabilistically decide whether to follow the global policy or stay in the same task. At steady state, $\nu = \chi = 0$ and thus, $f_k(\mu_i^{[k]}) = 0$ for all $i \in \{1, \ldots, M\}$. The policy is then given by $\tilde{\mathbf{P}} = \lambda\mathbf{P}^\star + (1-\lambda)I$ (It is easy to see that $\tilde{\mathbf{P}}$ satisfies $\mathbf{p}\tilde{\mathbf{P}} = \mathbf{p}$). At steady state (or the desired state), this should lead to λ fraction of agents staying in their task by deciding against the global policy to switch. At every iteration, the agent activity level $b_i^{[k]}$ decides the probability with which the agents in task i decide against the global policy and stay in the same task; this results in overall reduced activity. The algorithm is also presented as a psuedo-code in Algorithm 2. Rigorous analysis of the proposed algorithm are skipped and will be provided in a future extended publication.

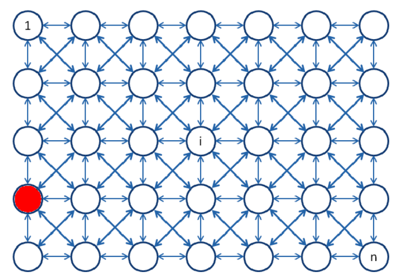


Fig. 1: The tasks modeled as a graph with its connectivity topology. Initially all the agents are in the state marked in red.

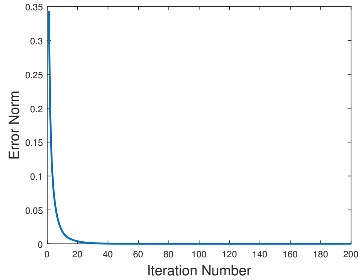
V. Numerical Results and Discussion

In this section, we present some numerical results based on the proposed algorithms described in sections IV-A and IV-B. We consider a task allocation problem which consists of the 35 tasks being operated in parallel. The connectivity of tasks is represented as a graph which is shown in Figure 1. Each node of the graph (or a task) is connected to 8 neighboring nodes (or tasks) except for the ones on the edges which are connected to either 5 or 3 nodes (see Figure 1).

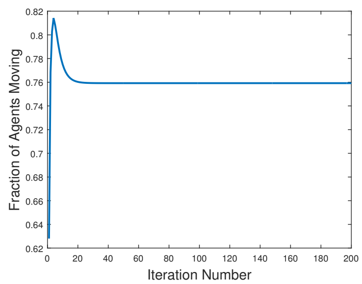
Initially all the agents are located at the same task which is marked in red in the graph. The agents need to be distributed uniformly over all the tasks at steady state. In Figure 2 we show the results obtained for the central controller. The system converges to the desired distribution monotonically (see Figure 2a) ; however, there is a lot of activity at steady state which is undesirable (see Figure 2b).

Next we consider the distributed control which is calculated as perturbation to the central control policy discussed earlier. For the distributed control, the parameter θ is selected to be 0.02 and the desired activity level is taken to be $\lambda = 0.2$ i.e., only 20% of the agents should move at steady state when compared to the movement shown in Figure 2b.

This particular case shows the use of proportional feedback given by $\beta^{[k]} = \gamma/k$ where $\gamma = 600$. In Figure 3, we show the results of the stable distributed proportional controller. As seen in Figure 3a, the error norm asymptotically converges to zero. Figure 3b shows the agent activities at steady state reaches λ fraction of the activity achieved by the central control policy. Compared to the central control,

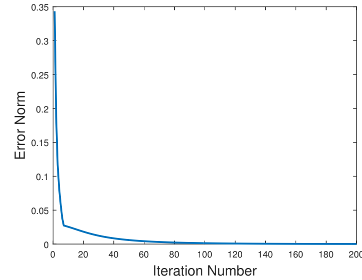


(a) Convergence of the agent distribution to the desired distribution

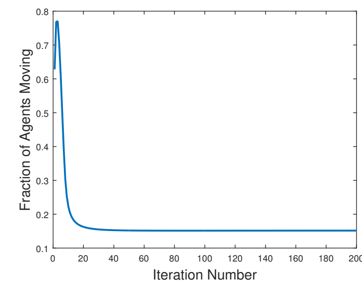


(b) Activity level at steady state

Fig. 2: Convergence of system states and agent activity obtained by the central control policy



(a) Convergence of the agent distribution to the desired distribution over the parallel tasks



(b) Activity level at steady state

Fig. 3: Convergence and agent activity level obtained by the distributed feedback controller

the distributed controller is slower (as seen by the error convergence rates in Figures 2 and 3a). However, the distributed controller is able to achieve the target state with reduced movement of agents and maintains reduced activity at steady state.

VI. Conclusions and Future Work

Swarm control problems are difficult due to the large state-space of the system. In this paper, we modeled swarm as a homogeneous collection of irreducible Markov chains. We presented a solution that uses a mix of centralized and decentralized control. In future research, we would like to investigate the use of reinforcement learning for correcting unmodeled dynamics.

References

[1] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343, no. 6172, pp. 754–758, 2014.

[2] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

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

[4] I. Chattopadhyay and A. Ray, "Supervised self-organization of homogeneous swarms using ergodic projections of Markov chains," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 6, pp. 1505–1515, 2009.

[5] N. Napp and E. Klavins, "A compositional framework for programming stochastically interacting robots," *The International Journal of Robotics Research*, vol. 30, no. 6, pp. 713–729, 2011.

[7] S.-i. Azuma, R. Yoshimura, and T. Sugie, "Broadcast control of multi-agent systems," *Automatica*, vol. 49, no. 8, pp. 2307–2316, 2013.

[6] B. Acikmese and D. S. Bayard, "A Markov chain approach to probabilistic swarm guidance," in *American Control Conference (ACC), 2012*, pp. 6300–6307, IEEE, 2012.

[8] A. Jevtić, A. Gutiérrez, D. Andina, and M. Jamshidi, "Distributed bees algorithm for task allocation in swarm of robots," *Systems Journal, IEEE*, vol. 6, no. 2, pp. 296–304, 2012.

[9] B. Fonooni, A. Jevtić, T. Hellström, and L.-E. Janlert, "Applying ant colony optimization algorithms for high-level behavior learning and reproduction from demonstrations," *Robotics and Autonomous Systems*, vol. 65, pp. 24–39, 2015.

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

[11] A. Cornejo, A. Dornhaus, N. Lynch, and R. Nagpal, "Task allocation in ant colonies," in *Distributed Computing*, pp. 46–60, Springer, 2014.

[12] B. Touri and A. Nedic, "Product of random stochastic matrices," *Automatic Control, IEEE Transactions on*, vol. 59, no. 2, pp. 437–448, 2014.

[13] D. Bajovic, J. Xavier, J. M. Moura, and B. Sinopoli, "Consensus and products of random stochastic matrices: Exact rate for convergence in probability," *Signal Processing, IEEE Transactions on*, vol. 61, no. 10, pp. 2557–2571, 2013.

[14] R. B. Bapat and T. E. Raghavan, *Nonnegative matrices and applications*, vol. 64. Cambridge University Press, 1997.

[15] I. Chattopadhyay and A. Ray, "Structural transformations of probabilistic finite state machines," *International Journal of Control*, vol. 81, no. 5, pp. 820–835, 2008.

[16] D. K. Jha, Y. Li, T. A. Wettergren, and A. Ray, "Robot path planning in uncertain environments: A language-measure-theoretic approach," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 3, p. 034501, 2015.

[17] H. L. Royden and P. Fitzpatrick, *Real analysis*, vol. 198. Macmillan New York, 1988.