On Time-Optimal Behavior Scheduling of Robotic Swarms for Achieving Multiple Goals

Sasanka Nagavalli*, *Student Member, IEEE*, Nilanjan Chakraborty†, *Member, IEEE*,
Katia Sycara*, *Fellow, IEEE*

***Abstract*— Robotic swarms are multi-robot systems whose global behaviors emerge from local interactions between individual robots. Each robot obeys a local control law that can be activated depending on an operator's choice of global swarm behavior. Real missions occur in uncontrolled environments with dynamically arising objectives and require combinations of behaviors. Given a library of swarm behaviors, a supervisory operator commanding the swarm must choose a sequence of behaviors to execute and their execution durations in order to accomplish a particular task during a mission composed of many tasks. In this paper, we address the following problem: given a library of swarm behaviors, the swarm initial state, the final goal and an unordered set of intermediate goals the operator wants to achieve, the objective is to identify a behavior schedule comprised of selected behaviors and associated time intervals of application of the behavior so that the total time to reach the final goal is minimized. Our contributions are as follows: (a) formalization of the problem of behavior scheduling to achieve multiple unordered goals with a robotic swarm, (b) an algorithm that produces a behavior schedule to achieve all intermediate goals and the final goal in minimum time such that the behavior durations are locally optimal and given which the goal sequence has bounded suboptimality and (c) application of this algorithm to configuration control of robot swarms.**

I. Introduction

Swarms are multi-robot systems that operate using simple local control laws. Global swarm behaviors emerge via interactions of swarm members. These emergent behaviors, such as flocking, deployment, rendezvous, allow the swarm to accomplish tasks even with the individual swarm members' sensor and computation limitations. Real-world applications, including area coverage, search and rescue operations, military surveillance or first responder assistance [1]–[5], are composed of complex tasks which usually cannot be achieved with a single existing behavior. One way to achieve complex tasks with swarms is to manually synthesize more sophisticated local control laws that would allow the swarm to accomplish the task while individual robots satisfy practical local constraints like collision avoidance [6,7]. Other work has used formal methods to automatically synthesize the local control logic for each robot from a set of individual robot specifications [8], which provides formal guarantees for each robot, but limited insight into overall swarm performance. This is a hard problem in general and there are few known solutions with guaranteed performance even for specific tasks like area coverage, search and rescue, and surveillance [9]–[11]. Alternatively, one can use a library of collective swarm behaviors (which may not necessarily be designed for the task at hand) and compose them using a supervisory controller [12] to accomplish the task. Often, an operator knows, not only the final task goal but also a set of intermediate goals that must be reached on the way to the final goal. In this paper, we study the problem of finding a *behavior schedule* (i.e. behavior sequence and the times of instantiation of the behaviors), so that the total time to reach the final goal is minimized and all intermediate goals are achieved.

More formally, we consider a swarm robotic system equipped with a library of collective behaviors. In each behavior, the state evolution of the swarm is modeled by a linear dynamical system. Formally, we model the state evolution of the swarm as a hybrid dynamical system, where there is one mode with linear dynamics corresponding to each behavior. We are also given a set of intermediate goals (represented in the joint state space of the robots) that are unordered and a final goal that the robots should reach. Note that each intermediate goal as well as the final goal is the equilibrium point of some behavior in our behavior library, which ensures that the intermediate and final goals are reachable. Our objective is to compute a sequence of the modes (behaviors) and the switch times such that the robot swarm reaches the final goal state in minimum time while achieving all the intermediate goal states. Our problem is related to the problem of switch time optimization studied for hybrid systems [13]. In switch time optimization problems for hybrid systems, the objective is to compute the sequence of the modes and the switch times between the modes so as to optimize a given objective function (over the path taken by the dynamical system) over a given time horizon. A key distinction of our problem from the extant literature on switch time optimization is that we have intermediate states that the dynamical system has to reach. Furthermore, in our problem, the final state is fixed while the objective is to minimize the total time to get to the final goal state.

We develop a two-step procedure to solve the swarm behavior scheduling problem. Our first step is to compute the minimum time trajectory between any two given (possibly intermediate) goal states with our dynamics as constraints. We present an algorithm to compute the different behavior switches and their timings for moving from one goal state to

*The authors are with the Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. Email: snagaval@andrew.cmu.edu, katia@cs.cmu.edu.
†The authors are with the Department of Mechanical Engineering, Stony Brook University, Stony Brook, New York, USA. Email: nilanjan.chakraborty@stonybrook.edu. This research has been sponsored in part by AFOSR Award FA9550-15-1-0442 and an NSERC PGS D scholarship.

another. Please note that it has already been established in the literature that for going to a given goal state in a time-optimal fashion, it may be beneficial to delay the choice of the behavior that results in achievement of the goal [14]. The result here is a generalization of the result in [14], since we show that one can use other behaviors (whose fixed points may not correspond to the given goals) transiently in order to move between the goals in a time-optimal fashion. Since our method is based on gradient descent, we achieve a locally optimal solution. In our second step, we formulate a Traveling Salesman Problem, where the cities are the goals and the cost of traveling between the goals is as given by the first step above. We show that the costs satisfy the triangle inequality although they are asymmetric. Therefore, we use a variant of Christofides' algorithm [15,16] to compute a goal sequence with bounded suboptimality - specifically, it is a constant factor approximation to the optimal sequence.

Our contributions are (a) formalization of the problem of behavior scheduling to achieve multiple unordered goals with a robotic swarm, (b) an algorithm that produces a behavior schedule to achieve all intermediate goals and the final goal in minimum time such that the behavior durations are locally optimal and given which the goal sequence has bounded suboptimality and (c) application of this algorithm to configuration control of robot swarms.

II. Related Work

We formulate our problem in the framework of switch time optimization for hybrid systems (see [13] for a comprehensive survey of recent results) where the mode sequence is not known a priori and also apply some results from traditional time-optimal control to characterize the length of the unknown optimal mode sequence. The authors of [17] consider the problem of finding the optimal switch times for a hybrid system where the cost functional is defined over a fixed time horizon and the final state is unconstrained. After finding an expression for the gradient of the cost functional in terms of the dynamics of the system and the costate, they propose a gradient descent algorithm that iteratively forward integrates the state equations, backward integrates the costate equations and then takes a gradient step (with Armijo step size) in the cost functional to eventually find a locally optimal solution. They also propose a procedure to use the gradient related information to iteratively alternate between adding modes to the sequence and updating the switch times. In [18], a sufficient descent version of the algorithm is presented with a procedure to satisfy dwell time constraints. The techniques presented in [18] are quite powerful and provide locally optimal solutions to the free-endpoint fixed-time horizon switch time optimization problem in hybrid systems with piecewise continuous modes.

In contrast to this locally optimal solution to the general free-endpoint fixed-time problem, in the first part of our approach we solve a particular case (for systems performing consensus) of the constrained-endpoint free-time problem using a more direct approach that does not require integration of the costate equations. Our method also relies on gradient descent techniques and provides a locally optimal solution. This solution is a behavior schedule for a robotic swarm to move from an initial state towards a goal state in minimum time using only consensus-based behaviors from a given library. In the second part of our approach, we then extend this technique with a variant [16] of Christofides' algorithm [15] for generating fixed-endpoint Hamiltonian paths with bounded suboptimality, which allows us to find a behavior schedule for a robotic swarm to move from an initial state to a desired final state while also achieving a set of unordered intermediate goals in minimum time. Our primary contribution is the application of these techniques to robotic swarms.

III. Problem Formulation

A. Consensus-based Behaviors

Consider a robotic swarm whose joint state (i.e. stacked states of individual robots) is given by $\mathbf{x}(t) \in \mathcal{X}$ at time $t \in \mathbb{R}_+$ in state space $\mathcal{X} = \mathbb{R}^n$. For this robotic swarm, the state evolution of each robot may be written as a weighted linear combination of its own state and the states of other robots in the swarm. That is, the joint state evolution of the swarm is given by the following differential equation when there is no external influence from a supervisory operator (e.g. a human).

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) \tag{1}$$

For the dynamics above, it is clear that the joint state of the swarm continuously evolves over time unless $\mathbf{x}(t)$ lies in the null space of the dynamics matrix $\mathbf{A}$. If the system is stable (i.e. all eigenvalues of $\mathbf{A}$ have negative real parts), then it is clear that the state of the system will evolve from any initial state $\mathbf{x}(0)$ to asymptotically approach a point in the null space of $\mathbf{A}$ (i.e. $\lim_{t\to\infty} \mathbf{A}\mathbf{x}(t) = \mathbf{0}$).

The state evolution of the swarm may be influenced by selecting a constant bias input $\mathbf{z} \in \mathbb{R}^n$, which is incorporated into the dynamics as follows.

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x}(t) - \mathbf{z}) \tag{2}$$

Applying the terminology we introduced in [12], Equation (2) represents the dynamics for a *swarm meta-behavior* with two unspecified parameters: the dynamics matrix $\mathbf{A}$ and the bias vector $\mathbf{z}$. This meta-behavior can be instantiated into many different *concrete swarm behaviors* via different choices for the dynamics matrix $\mathbf{A}$ and the bias vector $\mathbf{z}$. For example, if individual robot states represent their spatial positions, then the concrete swarm behavior known as rendezvous may be instantiated by specifying $\mathbf{z} = \mathbf{0}$ and $\mathbf{A} = -\mathbf{L}$, where $\mathbf{L}$ is the Laplacian matrix of the swarm's communication graph. In fact, the case when the dynamics matrix is fixed as $\mathbf{A} = -\mathbf{L}$ is an important subclass of the meta-behavior with the dynamics given in Equation (2). Specifically, fixing the dynamics matrix in this way results in the swarm performing a biased version of linear time-invariant (LTI) consensus with the following dynamics. See [19] for more information about the unbiased version of continuous-time continuous-state consensus.

$$\dot{\mathbf{x}}(t) = -\mathbf{L}(\mathbf{x}(t) - \mathbf{z}) \tag{3}$$

For this reason, we refer to this meta-behavior and any concrete behaviors instantiated from it (by specifying $\mathbf{z}$) as *consensus-based behaviors*. When the states represent spatial positions of the robots, we can instantiate concrete consensus-based behaviors like rendezvous (as described above) or various spatial configurations (e.g. line, circle) through appropriate selection of $\mathbf{z}$. In contrast to other methods of rendezvous or generating the spatial configurations, instantiating them as a consensus-based behavior would provide useful properties such as preserving the centroid of the original robot positions as they move.

It is important to recognize that the states do not necessarily have to represent spatial positions. For example, the states could instead represent more abstract quantities, such as the distribution of robots among tasks as in [20] (continuous-time macroscopic model of swarm). In that case, we can instantiate concrete consensus-based behaviors such as mission-specific robot distributions among tasks by appropriately selecting $\mathbf{z}$ and with $\mathbf{A} = -\mathbf{L} = \mathbf{K}$ ($\mathbf{K}$ as defined in [20]) representing the transition rate matrix, where $\mathbf{L}$ (not necessarily symmetric) is the weighted Laplacian of a directed graph modelling transitions between tasks. Here, the consensus-based behavior would provide the beneficial (and necessary) property that the total quantity of robots is preserved as their distribution among tasks changes.

B. Problem Statement

Assume we have a finite-sized, fixed library

$$\mathcal{B} = \{f_1, f_2, \ldots, f_m\} \tag{4}$$

of m concrete consensus-based behaviors where the i-th behavior in the library is a map $f_i : \mathbb{R}^n \to \mathbb{R}^n$ that has been instantiated from Equation (3) by specifying the bias vector $\mathbf{z}_i \in \mathbb{R}^n$.

$$f_i(\mathbf{x}(t)) = -\mathbf{L}(\mathbf{x}(t) - \mathbf{z}_i) \tag{5}$$

For notation purposes, we write $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\}$ for the set of bias vectors with which the concrete consensus-based behaviors in our library were instantiated.

A supervisory operator controls the robotic swarm by selecting a behavior $f_{b_k} \in \mathcal{B}$ where $b_k \in \{1, 2, \ldots, |\mathcal{B}|\}$ and applying it for a corresponding duration $\tau_k \in \mathbb{R}_+$. Thus, control of the robotic swarm may be represented as a sequence

$$S = ((b_1, \tau_1), (b_2, \tau_2), \ldots, (b_d, \tau_d)) \tag{6}$$

of d 2-tuples where the first element in the 2-tuple is the index b_k of the selected behavior in the library and the second element is the duration τ_k over which the behavior is applied. We refer to this sequence as a *behavior schedule* for our robotic swarm. For notation purposes, we write the behavior selection times

$$t_k = t_0 + \sum_{i=1}^{k} \tau_i \tag{7}$$

and without loss of generality assume $t_0 = 0$. The corresponding dynamics guiding the state evolution of the swarm is given as follows.

$$\dot{\mathbf{x}}(t) = \begin{cases} f_{b_1}(\mathbf{x}(t)) & \text{for } t \in [0, t_1) \\ f_{b_2}(\mathbf{x}(t)) & \text{for } t \in [t_1, t_2) \\ \vdots & \\ f_{b_d}(\mathbf{x}(t)) & \text{for } t \in [t_{d-1}, t_d) \end{cases} \tag{8}$$

Our supervisory operator has an unordered set of c intermediate goals $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_c\}$ where $\mathbf{y}_j \in \mathbb{R}^n$ that they would like to achieve with the robotic swarm in minimal time. We define a goal $\mathbf{y}_j$ to be achieved within time T if the following condition is true.

$$\exists t \leq T : \|\mathbf{P}(\mathbf{x}(t) - \mathbf{y}_j)\|_2^2 \leq \varepsilon \tag{9}$$

The parameter $\varepsilon \in \mathbb{R}_+$ is a desired tolerance chosen by the supervisory operator. The matrix $\mathbf{P}$ is an orthogonal projection matrix that rejects any component in the null space of $\mathbf{L}$. That is, if $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{\text{nullity}(\mathbf{L})}\}$ is an orthonormal basis (i.e. $\forall i : \|\mathbf{v}_i\|_2 = 1$ and $\forall i, j : \mathbf{v}_i^\top \mathbf{v}_j = 0$) for the null space of $\mathbf{L}$, then $\mathbf{P}$ can be written as follows.

$$\mathbf{P} = \prod_{i=1}^{\text{nullity}(\mathbf{L})} \left(\mathbf{I} - \mathbf{v}_i \mathbf{v}_i^\top\right) \tag{10}$$

For example, every Laplacian matrix $\mathbf{L}$ has an eigenvector $\mathbf{1}$ with corresponding eigenvalue 0. For a connected undirected graph, $\mathbf{1}$ is also the only eigenvector with eigenvalue 0, so it must be part of a basis for the null space of $\mathbf{L}$, which means for an undirected connected graph $\mathbf{P} = \mathbf{I} - \left(\mathbf{1}^\top \mathbf{1}\right)^{-1} \mathbf{1}\mathbf{1}^\top$.

In this paper, given the library $\mathcal{B}$ of consensus-based behaviors, the initial state $\mathbf{x}_{\text{initial}}$ of the swarm, the final goal $\mathbf{x}_{\text{final}}$ and the unordered set $\mathcal{Y}$ of intermediate goals the operator would like to achieve, our objective is to identify the behavior schedule S, which consists of selected behaviors and the associated time intervals of application, to minimize the total time T. Formally, our problem is written as follows.

$$\begin{aligned} \underset{S}{\arg\min} \quad & T \\ \text{subject to} \quad & \mathbf{x}(0) = \mathbf{x}_{\text{initial}} \\ & \|\mathbf{P}(\mathbf{x}(T) - \mathbf{x}_{\text{final}})\|_2^2 \leq \varepsilon \\ & \dot{\mathbf{x}}(t) = \begin{cases} f_{b_1}(\mathbf{x}(t)) & \text{for } t \in [0, t_1) \\ f_{b_2}(\mathbf{x}(t)) & \text{for } t \in [t_1, t_2) \\ \quad\vdots & \\ f_{b_d}(\mathbf{x}(t)) & \text{for } t \in [t_{d-1}, t_d) \end{cases} \\ & \forall \mathbf{y} \in \mathcal{Y}, \exists t \leq T : \|\mathbf{P}(\mathbf{x}(t) - \mathbf{y})\|_2^2 \leq \varepsilon \end{aligned} \tag{11}$$

The first constraint in this problem specifies the initial state of the swarm. The second constraint specifies a necessary and sufficient condition on the final state of the swarm for the final goal to be achieved. The third constraint restricts the trajectory describing the state evolution of the swarm based on the dynamics of the chosen behaviors from the library. The final constraint not only captures the requirement that all intermediate goals must be achieved, but also that they may be achieved in any order. The optimal solution to this problem is a behavior schedule S^* that causes the swarm to achieve all intermediate goals and then the final goal in minimal time T^*.

IV. Identifying a Behavior Schedule with No Intermediate Goals

The problem in Equation (11) is quite challenging to approach directly, so we begin with a relaxed version of the problem where there are no intermediate goals (i.e. $\mathcal{Y} = \varnothing$). In this section, we begin by analyzing the simplest case, where the behavior library only contains two behaviors ($m = 2$) and we show how to find the optimal behavior schedule to achieve the single goal $\mathbf{x}_{\text{final}}$. We then extend our analysis to the case where there are multiple behaviors in the library.

A. Behavior Library Contains Only Two Behaviors

When the behavior library only contains two behaviors, any behavior schedule must necessarily alternate between selecting the two behaviors, so the only variables are the durations. Thus, the state of the system evolves as follows.

$$\dot{\mathbf{x}}(t) = \begin{cases} -\mathbf{L}(\mathbf{x}(t) - \mathbf{z}_1) & \text{for } t \in [0, t_1) \\ -\mathbf{L}(\mathbf{x}(t) - \mathbf{z}_2) & \text{for } t \in [t_1, t_2) \\ -\mathbf{L}(\mathbf{x}(t) - \mathbf{z}_1) & \text{for } t \in [t_2, t_3) \\ -\mathbf{L}(\mathbf{x}(t) - \mathbf{z}_2) & \text{for } t \in [t_3, t_4) \\ \quad\vdots & \end{cases} \tag{12}$$

By considering the change of variables $\bar{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{z}_1$, we can always analyze the equivalent system

$$\dot{\bar{\mathbf{x}}}(t) = \begin{cases} -\mathbf{L}\bar{\mathbf{x}}(t) & \text{for } t \in [0, t_1) \\ -\mathbf{L}(\bar{\mathbf{x}}(t) + \mathbf{z}_1 - \mathbf{z}_2) & \text{for } t \in [t_1, t_2) \\ -\mathbf{L}\bar{\mathbf{x}}(t) & \text{for } t \in [t_2, t_3) \\ -\mathbf{L}(\bar{\mathbf{x}}(t) + \mathbf{z}_1 - \mathbf{z}_2) & \text{for } t \in [t_3, t_4) \\ \quad\vdots & \end{cases} \tag{13}$$

and find an optimal behavior schedule to transition from initial state $\bar{\mathbf{x}}_{\text{initial}} = \mathbf{x}_{\text{initial}} - \mathbf{z}_1$ to final state $\bar{\mathbf{x}}_{\text{final}} = \mathbf{x}_{\text{final}} - \mathbf{z}_1$. Thus, we will drop the overbar notation and analyze the following system.

$$\dot{\mathbf{x}}(t) = \begin{cases} -\mathbf{L}\mathbf{x}(t) & \text{for } t \in [0, t_1) \\ -\mathbf{L}(\mathbf{x}(t) - \mathbf{z}) & \text{for } t \in [t_1, t_2) \\ -\mathbf{L}\mathbf{x}(t) & \text{for } t \in [t_2, t_3) \\ -\mathbf{L}(\mathbf{x}(t) - \mathbf{z}) & \text{for } t \in [t_3, t_4) \\ \quad\vdots & \end{cases} \tag{14}$$

Given the system above, the following important question naturally arises: what is the maximum number of switches required to achieve a goal $\mathbf{x}_{\text{final}}$? We will now seek to answer this question by considering the following problem that occurs in traditional time-optimal control of LTI systems with bounded control signals $\mathbf{u}(t) \in [0, 1]$.

Theorem 1: The solution (if it exists) to the problem in Equation (15) is the optimal control signal $\mathbf{u}^*(t)$. This control signal corresponds to the optimal behavior schedule for the problem in Equation (11) when $\mathcal{Y} = \varnothing$ and the state evolution corresponds to Equation (14).

$$\begin{aligned} \underset{\mathbf{u}(t)}{\arg\max} \quad & \int_0^T -1 \, dt \\ \text{subject to} \quad & \mathbf{x}(0) = \mathbf{x}_{\text{initial}} \\ & \left\| \mathbf{P}(\mathbf{x}(T) - \mathbf{x}_{\text{final}}) \right\|_2^2 \le \varepsilon \\ & \dot{\mathbf{x}}(t) = -\mathbf{L}\mathbf{x}(t) + \mathbf{L}\mathbf{z}\mathbf{u}(t) \\ & \mathbf{u}(t) \in [0, 1] \end{aligned} \tag{15}$$

Proof: The payoff functional $\int_0^T -1 \; dt = -T$ is maximized when the time T at which the swarm achieves the desired goal $\mathbf{x}_{\text{final}}$ is minimized. First, we form the Hamiltonian, where $\mathbf{p}(t)$ is the costate. Due to space constraints, we will drop the dependence on time t from our notation.

$$H(\mathbf{x}, \mathbf{p}, \mathbf{u}) = \mathbf{p}^\top(-\mathbf{L}\mathbf{x} + \mathbf{L}\mathbf{z}\mathbf{u}) - 1 \tag{16}$$

This is a free-time, constrained-endpoint problem, for which Pontryagin's Maximum Principle states the following conditions under which a given input signal $\mathbf{u}(t)$ is optimal.

$$\dot{\mathbf{x}} = \nabla_p H = -\mathbf{L}\mathbf{x} + \mathbf{L}\mathbf{z}\mathbf{u} \tag{17}$$

$$\dot{\mathbf{p}} = -\nabla_x H = \mathbf{L}^\top \mathbf{p} \tag{18}$$

$$\forall t \in [0, T] : H(\mathbf{x}, \mathbf{p}, \mathbf{u}) = \max_{a \in [0,1]} H(\mathbf{x}, \mathbf{p}, a) \tag{19}$$

$$\mathbf{p}(T) = \nabla_x \left(\left\| \mathbf{P}(\mathbf{x} - \mathbf{x}_{\text{final}}) \right\|_2^2 - \varepsilon \right) \Big|_{t=T} \tag{20}$$

Here, we have two differential equations representing the state $\mathbf{x}(t)$ and costate $\mathbf{p}(t)$ evolution. In addition, we have a terminal condition on the costate trajectory — it must be perpendicular to the curve representing the set of possible final states $\mathbf{x}(T)$ for the system. By expanding the third condition, we find

$$\begin{aligned} H(\mathbf{x}, \mathbf{p}, \mathbf{u}) &= \max_{a \in [0,1]} H(\mathbf{x}, \mathbf{p}, a) \\ \mathbf{u} &= \underset{a \in [0,1]}{\arg\max} \left(\mathbf{p}^\top(-\mathbf{L}\mathbf{x} + \mathbf{L}\mathbf{z}a) - 1 \right) \\ \mathbf{u} &= \underset{a \in [0,1]}{\arg\max} \left(\mathbf{p}^\top \mathbf{L}\mathbf{z}a \right) \end{aligned} \tag{21}$$

which implies the following for optimal $\mathbf{u}^*$.

$$\forall a \in [0, 1] : \mathbf{p}^\top \mathbf{L}\mathbf{z}(\mathbf{u}^* - a) \ge 0 \tag{22}$$

$$\mathbf{u}^* = \begin{cases} 1 & \text{for } \mathbf{p}^\top \mathbf{L}\mathbf{z} \ge 0 \\ 0 & \text{for } \mathbf{p}^\top \mathbf{L}\mathbf{z} < 0 \end{cases} \tag{23}$$

Clearly, the above form of piecewise continuous $\mathbf{u}^*$ has an exactly corresponding representation as a discrete behavior schedule S^* and applying $\mathbf{u}^*$ to solve the problem in Equation (15) causes the state to evolve according to Equation (14) as required. ■

Returning to our robotic swarm, Theorem 1 implies that even when the the supervisory operator is permitted to select a convex combination of the consensus-based behaviors in the library, the optimal behavior schedule only switches discretely between individual behaviors in the library.

Theorem 2: The maximum length of the optimal behavior schedule S^* for a robotic swarm performing consensus-based behaviors with $\mathcal{Y} = \varnothing$ and with a behavior library containing $|\mathcal{B}| = 2$ behaviors is $|S^*| \leq n$.

Proof: By expanding the costate evolution we find $\mathbf{p}^\top \mathbf{L}\mathbf{z} = \mathbf{p}(0)^\top e^{\mathbf{L}t}\mathbf{L}\mathbf{z}$, which implies the well known result that for time-optimal control of a linear time-invariant system where the control signal is constrained to a convex set, each component of the control signal only takes on values at its limits (in this case, 0 and 1) — that is, the signal is bang-bang — and that the optimal control signal only switches a finite number of times. Since the dynamics matrix for our consensus-based behavior is the Laplacian of an undirected communication graph, it is symmetric positive-definite with real entries and has an eigendecomposition $\mathbf{L} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$.

$$\begin{aligned} &\mathbf{p}^\top \mathbf{L}\mathbf{z} \\ =&\mathbf{p}(0)^\top e^{\mathbf{L}t}\mathbf{L}\mathbf{z} \\ =&\mathbf{p}(0)^\top e^{\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top t}\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top\mathbf{z} \\ =&\mathbf{p}(0)^\top \mathbf{Q}e^{\mathbf{\Lambda}t}\mathbf{Q}^\top\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top\mathbf{z} \\ =&\tilde{\mathbf{p}}(0)^\top e^{\mathbf{\Lambda}t}\mathbf{\Lambda}\tilde{\mathbf{z}} \\ =&\sum_i \tilde{p}_i(0)\,\tilde{z}_i\lambda_i e^{\lambda_i t} \end{aligned} \tag{24}$$

Here $\tilde{\mathbf{z}} = \mathbf{Q}^\top\mathbf{z}$ and $\tilde{\mathbf{p}} = \mathbf{Q}^\top\mathbf{p}$. Since all the eigenvalues λ_i of $\mathbf{L}$ must be real and nonnegative, this is simply a sum of growing exponentials. Some of the coefficients may be positive and some of the coefficients may be negative, so this switches between positive and negative a maximum of n times over $t \in [0, T]$, which means that the optimal control signal must switch between 0 and 1 at most n times. This implies that the optimal behavior schedule for our swarm has a maximum length of n. ■

We note that while the proofs of Theorem 1 and Theorem 2 were presented for a symmetric dynamics matrix for ease of exposition, they can actually be extended to any diagonalizable dynamics matrix in a straightforward manner.

B. Behavior Library Contains Many Behaviors

We now turn our attention to the problem of finding the optimal behavior schedule when our behavior library can contain any number of consensus-based behaviors and there are no intermediate goals (i.e. $\mathcal{Y} = \varnothing$). Rather than a vector $\mathbf{z} \in \mathbb{R}^n$, now consider a matrix $\mathbf{Z} \in \mathbb{R}^{n\times m}$ where the i-th column is the i-th bias vector $\mathbf{z}_i$ from the set of bias vectors $\mathcal{Z}$ corresponding to behaviors in the library $\mathcal{B}$.

Theorem 3: For the problem in Equation (15), if the dynamics constraint is replaced with $\dot{\mathbf{x}}(t) = -\mathbf{L}\mathbf{x}(t) + \mathbf{L}\mathbf{Z}\mathbf{u}(t)$ and $\mathbf{u}(t)$ is constrained to a simplex in the positive orthant (i.e. $\mathbf{u}(t) \in [0,1]^m : \|\mathbf{u}(t)\|_1 \leq 1$), then the optimal signal $\mathbf{u}^*(t)$ corresponds to the optimal behavior schedule for the problem in Equation (11) when $\mathcal{Y} = \varnothing$ and the state evolution corresponds to Equation (8).

Proof: Applying Pontryagin's Maximum Principle with $\mathcal{A} = \{\mathbf{a} \mid \mathbf{a} \in [0,1]^m : \|\mathbf{a}\|_1 \leq 1\}$, we find the following condition.

$$\begin{aligned} H(\mathbf{x},\mathbf{p},\mathbf{u}) &= \max_{\mathbf{a}\in\mathcal{A}} H(\mathbf{x},\mathbf{p},\mathbf{a}) \\ \mathbf{u} &= \arg\max_{\mathbf{a}\in\mathcal{A}} \left(\mathbf{p}^\top(-\mathbf{L}\mathbf{x} + \mathbf{L}\mathbf{Z}\mathbf{a}) - 1\right) \\ \mathbf{u} &= \arg\max_{\mathbf{a}\in\mathcal{A}} \left(\mathbf{p}^\top\mathbf{L}\mathbf{Z}\mathbf{a}\right) \end{aligned} \tag{25}$$

Expanding the costate evolution with $\mathbf{L} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$, we find

$$\mathbf{p}(t)^\top \mathbf{L}\mathbf{Z}\mathbf{a} = \sum_j a_j \sum_i \tilde{p}_i(0)\,\tilde{z}_{ij}\lambda_i e^{\lambda_i t} \tag{26}$$

with $\tilde{\mathbf{Z}} = \mathbf{Q}^\top\mathbf{Z}$ and $\tilde{\mathbf{p}} = \mathbf{Q}^\top\mathbf{p}$ which implies the following for optimal $\mathbf{u}^*$.

$$j^* = \arg\max_j \left(\sum_i \tilde{p}_i(0)\,\tilde{z}_{ij}\lambda_i e^{\lambda_i t}\right) \tag{27}$$

$$\forall j \neq j^* : u_j^*(t) = 0 \tag{28}$$

$$u_{j^*}^*(t) = \begin{cases} 1 & \text{for } \sum_i \tilde{p}_i(0)\,\tilde{z}_{ij^*}\lambda_i e^{\lambda_i t} \geq 0 \\ 0 & \text{for } \sum_i \tilde{p}_i(0)\,\tilde{z}_{ij^*}\lambda_i e^{\lambda_i t} < 0 \end{cases} \tag{29}$$

Thus, the optimal signal $\mathbf{u}^*(t)$ has either all entries equal to 0 or a single entry equal to 1 at any given time t. Clearly, it is piecewise continuous and has an exactly corresponding representation as an optimal behavior schedule S^* to solve Equation (11) when $\mathcal{Y} = \varnothing$. ■

Theorem 3 implies that even when the supervisory operator is allowed to continuously switch between a convex combination of the behaviors in the library, the optimal behavior schedule marvelously only switches discretely between individual behaviors rather than selecting a weighted combination.

Theorem 4: The maximum length of the optimal behavior schedule S^* for a robotic swarm performing consensus-based behaviors with $\mathcal{Y} = \varnothing$ and with a behavior library containing $|\mathcal{B}| = m$ (where $m \geq 2$) behaviors is $|S^*| \leq \frac{m(m-1)}{2}n$.

Proof: The proof is immediate from Theorem 2 and Theorem 3 since we may switch between any pair of behaviors in the library at most n times and there are $\frac{m(m-1)}{2}$ unique pairs. ■

In practice, we expect the majority of behavior schedules to be shorter than the maximum suggested by Theorem 4.

C. Procedure for Computing Behavior Schedule with No Intermediate Goals

Now that we have characterized the maximum length of the optimal behavior schedule, we present a procedure to compute it when $\mathcal{Y} = \varnothing$. Begin by assuming the length of the behavior schedule S we are optimizing is $d = \frac{m^2(m-1)}{2}n$ with all m behaviors in the library repeated $\frac{m(m-1)}{2}n$ times (i.e. $b_k = b_{mj+k}$ for $j = 0, 1, \ldots, \frac{m(m-1)}{2}n - 1$). The reason we select this length for the behavior schedule is that we can also use it to represent all possible sequences of behaviors of length $\frac{m(m-1)}{2}n$ (i.e. the maximum length of the true optimal behavior sequence) by simply deleting $\frac{m(m-1)^2}{2}n$ entries in the schedule (i.e. setting their corresponding durations to 0). This allows us to somewhat indirectly address the sequencing problem, which is combinatorial and computationally difficult to tackle directly.

Now we simply have to compute the corresponding durations, which we can represent as a vector $\bar{\tau} = \begin{bmatrix}\tau_1 & \tau_2 & \ldots & \tau_d\end{bmatrix}^\top$. Since all of our behaviors are consensus-based behaviors, we can write the state evolution recursively.

$$\mathbf{x}(t_k) = e^{-\mathbf{L}\tau_k}\left(\mathbf{x}(t_{k-1}) - \mathbf{z}_{b_k}\right) + \mathbf{z}_{b_k} \tag{30}$$

Expanding and simplifying, we get the following closed form expression for the state evolution.

$$\begin{aligned}\mathbf{x}(t_d) = {} & e^{-\mathbf{L}\sum_{i=1}^{d}\tau_i}\mathbf{x}(0) - \sum_{j=1}^{d} e^{-\mathbf{L}\sum_{i=j}^{d}\tau_i}\mathbf{z}_{b_j} \\ & + \sum_{j=1}^{d-1} e^{-\mathbf{L}\sum_{i=j+1}^{d}\tau_i}\mathbf{z}_{b_j} + \mathbf{z}_{b_d}\end{aligned} \tag{31}$$

We can now rewrite the problem in Equation (11) as follows (assuming $\mathcal{Y} = \varnothing$) and noting that $T = t_d$.

$$\begin{aligned} \underset{\bar{\tau}\in\mathbb{R}^d}{\arg\min} \quad & \mathbf{1}^\top \bar{\tau} \\ \text{subject to} \quad & \mathbf{x}(0) = \mathbf{x}_{\text{initial}} \\ & \left\|\mathbf{P}\left(\mathbf{x}(t_d) - \mathbf{x}_{\text{final}}\right)\right\|_2^2 \le \varepsilon \\ \mathbf{x}(t_d) \quad & = e^{-\mathbf{L}\sum_{i=1}^{d}\tau_i}\mathbf{x}(0) - \sum_{j=1}^{d} e^{-\mathbf{L}\sum_{i=j}^{d}\tau_i}\mathbf{z}_{b_j} \\ & + \sum_{j=1}^{d-1} e^{-\mathbf{L}\sum_{i=j+1}^{d}\tau_i}\mathbf{z}_{b_j} + \mathbf{z}_{b_d} \end{aligned} \tag{32}$$

This is a nonlinear optimization problem with a single equality constraint and nonnegative optimization variables. Conveniently, in this form, both the objective function and the constraints are continuous and smooth in $\bar{\tau}$. The gradient of the objective function is given as follows.

$$\nabla_{\bar{\tau}} T = \mathbf{1} \tag{33}$$

The gradient of the equality constraint is given as follows.

$$\begin{aligned} \phi(\mathbf{x}) &= \left\|\mathbf{P}\left(\mathbf{x}(T) - \mathbf{x}_{\text{final}}\right)\right\|_2^2 - \varepsilon \\ \nabla_{\bar{\tau}}\phi &= 2\begin{bmatrix}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\tau_1} & \ldots & \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\tau_d}\end{bmatrix}^\top \mathbf{P}^\top\mathbf{P}\left(\mathbf{x}(T) - \mathbf{x}_{\text{final}}\right) \end{aligned} \tag{34}$$

The derivative of the state evolution with respect to each duration is given as follows.

$$\begin{aligned}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\tau_k} = {} & -\mathbf{L}e^{-\mathbf{L}\sum_{i=1}^{d}\tau_i}\mathbf{x}(0) - \sum_{j=1}^{k} -\mathbf{L}e^{-\mathbf{L}\sum_{i=j}^{d}\tau_i}\mathbf{z}_{b_j} \\ & + \sum_{j=1}^{k-1} -\mathbf{L}e^{-\mathbf{L}\sum_{i=j+1}^{d}\tau_i}\mathbf{z}_{b_j}\end{aligned} \tag{35}$$

Since we have analytic gradients for both the objective function and the constraints, we can use any number of standard gradient-based optimization techniques to find locally optimal solutions efficiently (see Section VI for an example). As noted above, we expect the vast majority of the durations in our behavior schedule to be 0 because we intentionally increased its length to enable it to represent all possible behavior sequences. In addition, if the supervisory operator for the robotic swarm knows the behavior sequence in advance, our technique can be used to identify the optimal durations for which those behaviors should be executed.

Finally, we note that our technique applies equally well to a general dynamics matrix $\mathbf{A}$ and it is obvious that computational efficiency can be improved when $\mathbf{A}$ is diagonalizable by taking the eigendecomposition $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$ and considering transformed state $\tilde{\mathbf{x}}(t) = \mathbf{Q}^{-1}\mathbf{x}(t)$, and behaviors $\tilde{f}_j(\tilde{\mathbf{x}}(t)) = \mathbf{\Lambda}(\tilde{\mathbf{x}}(t) - \tilde{\mathbf{z}}_j)$ with $\tilde{\mathbf{z}}_j = \mathbf{Q}^{-1}\mathbf{z}_j$. For these transformed behaviors, the dynamics matrix $\mathbf{\Lambda}$ is diagonal, which makes its matrix exponential $e^{\mathbf{\Lambda}t}$ much faster to compute. Clearly, this transformation is for computational purposes only and does not change either the mathematical results or the physical motion of robots in our swarm.

V. Algorithm for Sequencing of Unordered Intermediate Goals

In the previous section, we showed that when there are no intermediate goals, it is possible to identify the maximum length of the time-optimal behavior schedule, fix the sequence and then solve a nonlinear optimization problem with smooth objective function and smooth constraints for the durations. The identified behavior schedule will be locally optimal in the space of durations. Given a particular behavior library, we can do this for any choice of initial state and goal.

We now consider the problem of behavior scheduling with a set of c intermediate goals $\mathcal{Y}$ that must be achieved by the robotic swarm. Consider the weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$. The vertices $\mathcal{V} = \mathcal{Y} \cup \{\mathbf{x}_{\text{initial}}, \mathbf{x}_{\text{final}}\}$ of this graph include the initial state, the intermediate goals and the final goal. The edge set $\mathcal{E}$ includes directed edges between every pair of nodes but does not include any edges that begin at $\mathbf{x}_{\text{final}}$ or end at $\mathbf{x}_{\text{initial}}$. The entries $w_{ij} \in \mathbb{R}_+$ of the weight matrix $\mathbf{W} \in \mathbb{R}^{(c+2)\times(c+2)}$ represent the minimum time required to transition from the state represented by vertex v_i to the state represented by vertex v_j. These weights may be computed by using the procedure in the previous section to find the optimal behavior schedule to move from v_i to v_j. Thus, we note that weighted edges in our directed graph must satisfy the triangle inequality (i.e. $\forall i, j, k$: $w_{ij} + w_{jk} \ge w_{ik}$). In the rare case that particular triplets do not satisfy the triangle inequality (since our procedure is only locally optimal), we enforce the triangle inequality by replacing the computed behavior schedule for edge (v_i, v_k) with the concatenation of the computed behavior schedules for (v_i, v_j) and (v_j, v_k).

Once we have created the weight matrix $\mathbf{W}$, which is not symmetric for our problem, and computed the behavior schedules, our problem is reduced to finding the minimum

Algorithm 1 Compute the Best Behavior Schedule

1: **function** MULTIGOALBEHAVIORSCHEDULE($\mathcal{B}, \mathbf{x}_{\text{initial}}, \mathcal{Y}, \mathbf{x}_{\text{final}}$)
2: $S_{\text{all}} \leftarrow \varnothing$, $\mathcal{V} = \mathcal{Y} \cup \{\mathbf{x}_{\text{initial}}, \mathbf{x}_{\text{final}}\}$
3: **for** $v_i \in \mathcal{V} : v_i \neq \mathbf{x}_{\text{final}}$ **do**
4: **for** $v_j \in \mathcal{V} : (v_i \neq v_j) \wedge (v_j \neq \mathbf{x}_{\text{initial}})$ **do**
5: $S_{ij} \leftarrow$ ONEGOALBEHAVIORSCHEDULE($\mathcal{B}, v_i, v_j$)
6: $w_{ij} \leftarrow$ TOTALDURATION(S_{ij})
7: **end for**
8: **end for**
9: $\mathbf{W} \leftarrow$ ENFORCETRIANGLEINEQUALITY($\mathbf{W}$)
10: $P \leftarrow$ FIXEDENDPOINTSHAMILTONIANPATH($\mathbf{W}, \mathbf{x}_{\text{initial}}, \mathbf{x}_{\text{final}}$)
11: **for** $(v_i, v_j) \in P$ **do**
12: $S_{\text{all}} \leftarrow S_{\text{all}} \sqcup S_{ij}$
13: **end for**
14: **return** S_{all}
15: **end function**

cost Hamiltonian path which starts at $\mathbf{x}_{\text{initial}}$ and ends at $\mathbf{x}_{\text{final}}$. This is a variation of the Metric Asymmetric Travelling Salesman Problem (TSP), which is known to be NP-hard. Christofides' approximation algorithm solves the Metric Symmetric TSP problem in O($|\mathcal{V}|^3$) with bounded suboptimality [15]. Specifically, the cost of the tour produced by Christofides' algorithm will be at most $\frac{3}{2}$ times the cost of the optimal tour. In [21], the authors show that the Asymmetric TSP problem can be transformed into the Symmetric TSP, problem after which we can apply Christofides' algorithm as usual to find a $\frac{3}{2}$-approximation to the optimal tour. However, our problem also has fixed endpoints, so we must apply a variant of Christofides' algorithm [16] for Hamiltonian paths rather than tours. It is shown in [16] that when there are two fixed endpoints, the algorithm produces a $\frac{5}{3}$-approximation to the optimal Hamiltonian path. The behavior schedule S^* which solves the problem in Equation (11) is the concatenation of the locally optimal behavior schedules for each edge in the Hamiltonian path from initial state $\mathbf{x}_{\text{initial}}$ through intermediate goals $\mathcal{Y}$ to final goal $\mathbf{x}_{\text{final}}$. The behavior schedule S^* has bounded local suboptimality.

The method described above is summarized in Algorithm 1. The function ONEGOALBEHAVIORSCHEDULE() solves the problem in Equation (32). Notice that the locally optimal behavior schedule for every edge and every entry of the weight matrix can be computed in parallel (lines 3–8).

We conclude this section by noting that if the final goal is not fixed (i.e. no $\mathbf{x}_{\text{final}}$), then we can use the algorithm in [16] to find a single endpoint Hamiltonian path starting at $\mathbf{x}_{\text{initial}}$ and achieving all $\mathcal{Y}$ with $\frac{3}{2}$ bounded suboptimality. This fact can also be used to extend Algorithm 1 in a straightforward manner to handle ordered subsets of unordered goals (e.g. achieve all goals in $\mathcal{Y}_1$, then $\mathcal{Y}_2$, then $\mathcal{Y}_3$, etc.).

VI. APPLICATION TO CONFIGURATION CONTROL OF ROBOTIC SWARMS

In this section, we revisit the problem of optimal timing in configuration control of robotic swarms, which we presented to illustrate the phenomenon of Neglect Benevolence in [14]. There are many applications, including artistic performances [22], surveillance, cooperative manipulation or robotic printing where it is important to generate and maintain various configurations with a multi-robot team. In some of these applications, the sequence of desired configurations is known in advance and in others the desired configurations are known, but not the sequence. Given a library of consensus-based behaviors $\mathcal{B}$ that each cause the swarm to move towards a particular spatial configuration $\mathbf{z} \in \mathcal{Z}$, we now apply the results from the previous sections to identify a behavior schedule S that generates the desired configurations $\mathcal{Y}$ in minimum time. For these examples, we make the reasonable assumption that $\mathcal{Y} \subset \mathcal{Z}$. The optimal behavior schedule S will *transiently* switch between configurations $\mathcal{Z}$ in our library to ensure that the swarm achieves (to ε-convergence) all unordered goals $\mathcal{Y}$ and final goal $\mathbf{x}_{\text{final}}$ in minimum time. We instantiate our behavior library $\mathcal{B}$ with a set of configurations $\mathcal{Z}$ shown in Figure 1.

A. Known Behavior Sequence with Unknown Durations

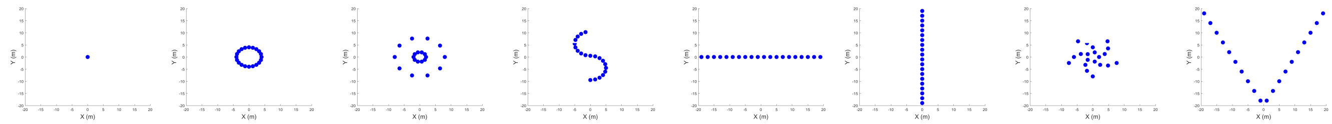
We first consider a case similar to the one we considered in [14]. The human operator is performing a mission with the robotic swarm and has applied behavior f_2, which causes the swarm to move towards the corresponding configuration $\mathbf{z}_2$. However, at time $t = t_0 = 0$ when the swarm is in state $\mathbf{x}_{\text{initial}}$ (random configuration), the operator becomes aware of a change in mission goal to $\mathbf{x}_{\text{final}}$, which can be achieved using behavior f_3 (e.g. $\|\mathbf{P}(\mathbf{z}_3 - \mathbf{x}_{\text{final}})\|_2^2 = 0$). Based on this description, the behavior schedule must have the form $S = ((2, \tau_1), (3, \tau_2))$. Note that $\mathbf{x}_{\text{initial}}$ is Neglect Benevolent simply if the optimal schedule has $\tau_1 > 0$. In our simulation (which had different $\mathbf{x}_{\text{initial}}$ than [14]) when $|S| = 2$, the schedule computed is $S =$ ((2,12.5267), (3,38.0237)) with total time $T = 50.55$. When we allow $|S| = n$, which is the maximum possible entries in the optimal schedule, the non-zero duration entries in the schedule computed are $S =$ ((3,19.2606), (2,2.9868), (3,27.9326)) with total time $T = 50.18$, a slight improvement.

B. Target Configuration with No Desired Intermediate Configurations

We now consider the case where we don't know the behavior sequence in advance, but want to find the behavior schedule to achieve $\mathbf{x}_{\text{final}} = \mathbf{z}_7$ from $\mathbf{x}_{\text{initial}}$ (random configuration) in minimum time. We implemented the procedure in Section IV-C in MATLAB and used `fmincon` (with `'interior-point'` and analytic gradients specified) to identify the optimal behavior schedule $S =$ ((5,0.1198), (6,0.8556), (8,0.1277), (5,15.9487), (8,0.4150), (5,2.1281), (7,35.4172)) for a total duration $T = 55.01$. We only show entries with non-zero duration, but note that the length of the behavior schedule is significantly less than the maximum suggested in Theorem 4. Clearly, only two of the behaviors (5 and 7) were applied for a significant portion of time.

C. Target Configuration with Desired Intermediate Configurations

Finally, we consider the case where we want to find a behavior schedule that achieves a set of intermediate configurations $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2\}$ (with $\mathbf{y}_1 = \mathbf{z}_2$ and $\mathbf{y}_2 = \mathbf{z}_3$)



(a) Rendezvous $\mathbf{z}_1$ (b) Circle $\mathbf{z}_2$ (c) Torus $\mathbf{z}_3$ (d) S $\mathbf{z}_4$ (e) Line X $\mathbf{z}_5$ (f) Line Y $\mathbf{z}_6$ (g) Spiral $\mathbf{z}_7$ (h) V $\mathbf{z}_8$

Fig. 1: Equilibrium Configurations for Behaviors in Our Library

from $\mathbf{x}_{\text{initial}}$ (random configuration) to $\mathbf{x}_{\text{final}} = \mathbf{z}_7$ in minimum time. We apply the algorithm in Section V to our problem and identify the optimal behavior schedule S = ((3,1.3048), (4,3.7551), (5,0.0914), (3,10.8722), (5,6.7721), (3,31.0313), (8,1.0880), (2,11.7012), (7,1.0925), (2,20.9164), (7,0.4393), (7,0.8738), (7,23.8005)) for a total duration $T = 113.74$. Intermediate goal $\mathbf{y}_2 = \mathbf{z}_3$ was reached at $t = 53.83$ and goal $\mathbf{y}_1 = \mathbf{z}_2$ was reached at $t = 89.06$. We only show entries with non-zero duration, but note that the length of the behavior schedule is significantly less than the maximum suggested in Theorem 4. We can also see an artifact of representing the schedule to optimize using a fixed sequence with repeated subsequences of entries in the last three terms of the identified schedule, which could more concisely and equivalently have been represented as (7,25.1136). Minor post-processing can clean up such artifacts.

VII. Conclusion

In this paper, given a library of consensus-based behaviors, we considered the problem of generating a behavior schedule that enables a robotic swarm to achieve a set of unordered intermediate goals and a final goal in minimum time. After identifying a procedure to produce a locally optimal behavior schedule between any two goals, we presented an algorithm that computes locally optimal behavior schedules between all pairs of goals and then applies a variant of Christofides' algorithm to find a $\frac{5}{3}$-approximation to the optimal Hamiltonian path through the goals. Concatenating the behavior schedules along path, we obtained an overall behavior schedule. In this work, our only cost was on time to achieve all goals. In future work, we plan to explore other cost functions and more types of behaviors in our library.

References

[1] Z. Kira and M. A. Potter, "Exerting human control over decentralized robot swarms," in *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*. IEEE, 2009, pp. 566–571.

[2] S. Bashyal and G. K. Venayagamoorthy, "Human swarm interaction for radiation source search and localization," in *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*. IEEE, 2008, pp. 1–8.

[3] M. Fields, E. Haas, S. Hill, C. Stachowiak, and L. Barnes, "Effective robot team control methodologies for battlefield applications," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 5862–5867.

[4] A. M. Naghsh, J. Gancet, A. Tanoto, and C. Roast, "Analysis and design of human-robot swarm interaction in firefighting," in *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*. IEEE, 2008, pp. 255–260.

[5] A. Kolling, S. Nunnally, and M. Lewis, "Towards human control of robot swarms," in *Proceedings of the seventh annual ACM/IEEE international conference on human-robot interaction*. ACM, 2012, pp. 89–96.

[6] S. Bhattacharya, R. Ghrist, and V. Kumar, "Multi-robot coverage and exploration on Riemannian manifolds with boundaries," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 113–137, 2014.

[7] L. Wang, A. D. Ames, and M. Egerstedt, "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 2659–2664.

[8] Y. K. Lopes, S. M. Trenkwalder, A. B. Leal, T. J. Dodd, and R. Groß, "Supervisory control theory applied to swarm robotics," *Swarm Intelligence*, vol. 10, no. 1, pp. 65–97, 2016.

[9] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[10] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 9–26, Feb 2016.

[11] G. Francesca and M. Birattari, "Automatic design of robot swarms: Achievements and challenges," *Frontiers in Robotics and AI*, vol. 3, no. 29, 2016.

[12] S. Nagavalli, N. Chakraborty, and K. Sycara, "Automated sequencing of swarm behaviors for supervisory control of robotic swarms," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2017.

[13] H. Lin and P. J. Antsaklis, "Hybrid dynamical systems: An introduction to control and verification," *Foundations and Trends® in Systems and Control*, vol. 1, no. 1, pp. 1–172, 2014.

[14] S. Nagavalli, L. Luo, N. Chakraborty, and K. Sycara, "Neglect benevolence in human control of robotic swarms," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6047–6053.

[15] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," Graduate School of Industrial Administration, Carnegie Mellon University, Technical Report 388, 1976.

[16] J. Hoogeveen, "Analysis of Christofides' heuristic: Some paths are more difficult than cycles," *Operations Research Letters*, vol. 10, no. 5, pp. 291–295, 1991.

[17] M. Egerstedt, Y. Wardi, and F. Delmotte, "Optimal control of switching times in switched dynamical systems," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3. IEEE, 2003, pp. 2138–2143.

[18] Y. Wardi, M. Egerstedt, and M. Hale, "Switched-mode systems: gradient-descent algorithms with Armijo step sizes," *Discrete Event Dynamic Systems*, vol. 25, no. 4, pp. 571–599, 2015.

[19] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 1859–1864.

[20] A. Prorok, M. A. Hsieh, and V. Kumar, "Formalizing the impact of diversity on performance in a heterogeneous swarm of robots," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5364–5371.

[21] R. Kumar and H. Li, "On asymmetric tsp: Transformation to symmetric tsp and performance bound," *Journal of Operations Research*, 1996.

[22] A. P. Schoellig, H. Siegel, F. Augugliaro, and R. DAndrea, "So you think you can dance? rhythmic flight performances with quadrocopters," in *Controls and Art*. Springer, 2014, pp. 73–105.